MATHMENU REFERENCE MANUAL

MATHMENU

Version 1.0

February 1983

INTER+ACTION

113 WARD STREET NEW HAVEN, CT. 06519

MATHMENU REFERENCE MANUAL

CONTENTS

MATHMENU REFERENCE MANUAL

Introduction

MATHMENU is a collection of 15 math/engineering programs covering a wide variety of topics, written in BASIC for the Color Computer. In the disk version, the programs are linked by a menu program for single key selection and automatic return.

All programs are self documented with an internal informational text contained within the programs. The programs are additionally supported by this documentation package, which treats each program in greater detail.

To run the disk version, simply insert the disk into Drive 0, type RUN "MATHMENU", and press ENTER. The drive will clunk and whir for a moment, and then you will be presented with a two-page menu of the available programs. Select the desired program by pressing the appropriate number key. When you are done, you will be returned to this menu automatically.

To run the cassette version, place the tape in your cassette deck. The simplest way to run a program is to rewind the tape to the beginning and type CLOAD followed by the name of the program enclosed in quotes. For example, to run 3D PLOT, type CLOAD "3D PLOT" and press the ENTER key. When the computer has found the program, and loaded it, it will display OK followed by the blinking cursor. Now type RUN and press ENTER.

You can speed up this process by fast-forwarding the tape to just before the beginning of the program you want to load; however, make sure you don't position the tape within a program or you'll get an I/O error and have to try again. Use the AUDIO ON, MOTOR ON, and MOTOR OFF commands to facilitate this. See your Radio Shack manual, Getting Started with COLOR BASIC, for more information if you are unfamiliar with loading programs from cassette.

16K cassette owners must do a PCLEAR1 command before loading MATRIXOP and VECTOROP to avoid an OM ERROR. If memory problems are encountered on other programs, a PCLEAR1 or CLEAR 100 command should fix the problem. Remember that the plotting programs require Extended Color Basic.

Cassette owners may upgrade to the disk system at a later date for an upgrade fee of $8.00.

Every attempt has been made to ensure that the programs contained in the MATHMENU collection are free from errors. In the event of problems, please consult this documentation. If the problem appears to be software related, please notify us. We will try our best to supply you with a speedy fix or replacement copy if appropriate. As always, our liability will be limited to the replacement of errant software.

Program Summary

3D PLOT - Plots 3D surfaces on the High-Res (PMODE4) graphics screen. Any equation of the form Y=f(X,Z) can be plotted within minutes. User definable features include: axes limits, axes scales, 'striping' width, and plotting step. Surfaces may be saved to disk or tape for future recall.

2D PLOT - Plots all single variable functions of the form Y=f(X). Plots in an X,Y coordinate system on the High-Res graphics screen. User definable axes limits, and axes scales. Function plots may be saved to disk or tape.

MATRIXOP - Performs operations on matrices as large as 8 rows by 8 columns. The following operations are available:

- Determinant
- Inverse
- Transpose
- Matrix Addition
- Matrix Subtraction
- Scalar Multiplication
- Matrix Multiplication
- Matrix Exponentiation

VECTOROP - Vector operations. Vectors may have as many as 20 elements each. The following operations are available:

- Dot Product
- Cross Product
- Normal Length
- Unit Vector
- Addition
- Subtraction
- Scalar Multiplication
- Plane Containing Vectors

NUMDIFN - Numerical differentiation. Evaluates numerically the derivative of any single variable function at one or more selected points.

NUMINTEG - Numerical integration. Applies Simpson's Rule to approximate the integral of single variable functions between definite limits.

LSTSQRS - Least squares curve fitting. Computes the best fit equation to a given set of data points for linear, quadratic, or cubic data distributions.

BASECONV - Converts numbers entered in any base to equivalent expressions in another selected base system. Accepts entries and reports results using alphabetic characters if desired.

RPL - Turns the Color Computer into a Reverse Polisch Logic calculator with decimal and hexidecimal computing capabilities. Visible stacks and memory registers.

BINOMEXP - Binomial expansion. Computes the terms resulting from the expansion of (A+B) raised to the N for integer values of N up to 32.

PRIME CK - Verifies primality or reports the first divisor for numbers up to one million.

LG#ADD, LG#MULT - Performs operations on large numbers without losing figures to scientific notation. Addition to 100 digits each number, multiplication to 30 digits.

RECT-POL - Converts both ways between the Cartesian (X,Y) coordinate system and the polar (radius, angle) coordinate system.

QUAD EQN - Computes real, double, or complex roots for a user defined quadratic equation A*X*X + B*X + C =0.

RPL CALCULATOR

RPL is a program which turns the Color Computer into a reverse Polish logic calculator with decimal (base 10) and hexidecimal (base 16) computing capabilities. After an initial text screen is displayed, the screen will become the calculator keyboard, displaying the valid key operations.

The four operating stacks are shown in the upper left area of the screen, and the three memory registers are shown in the upper right section. On the lower half of the screen are shown the keys recognized and the operations they represent. A return to information screen can be effected by pressing '?' from the keyboard screen. This return will not affect impending calculations.

The five operations available are addition, subtraction, multiplication, division and exponentiation. These operations can be performed on decimal or hexidecimal numbers. Mixed base calculations are possible by selecting 'X' for change mode in mid-operation. The numbers in the stacks and in memory will be converted automatically. Hexidecimal entries must be made using the alphabetic characters for numbers 10 through 15.

The reverse Polish logic (RPL) operating system differs from the algebraic operating system (AOS) found on most hand held calculators. In the RPL system, a number is ENTERed into the stacks and then operated on. In the AOS system, the equal (=) sign is used to solve calculations whereas the RPL system solves by the ENTER key. An example of equivalent calculations in each of the systems is shown below:

	Key Sequence	Result
AOS	15 * 12 =	180
RPL	15 ENTER 12 *	180

While the AOS system is more prevalent in calculator design, the RPL system is preferred by scientists and engineers for its efficiency in terms of number of required keystrokes.

2D PLOT

Below is an in depth analysis of the features of 2D PLOT, arranged in the order of appearance on the Main Menu.

(0) GENERAL INFORMATION

Selecting (0) guides the user through the following text screens:

This program will plot a user defined function on an X-Y coordinate system. The equation must be solved explicitly for Y and entered at line 10. An upper limit must be selected for the X axis, and boundaries for plotting along the X axis must be entered. The equation can then be plotted.

The function will continue to plot and will remain on the graphics screen until 'R' is pressed to return to the menu. Returning to the menu does not affect the function on the graphics screen. The function plotted will remain intact until it is cleared from the menu. This allows for inspection of two or more functions on the screen simultaneouslty.

(1) ENTER NEW EQUATION

Presing (1) prompts the user to enter the equation of the function which is to be plotted. This equation should be solved explicitly for Y; that is, Y should appear to the left of the equal sign, and the right side should be an expression which varies only in X. The equation is entered into the program at line 10.

By entering an equation at line 10, the program has been edited, and all variables are reset to zero. On the disk version, any variables which may have been entered are saved to disk before entry of the equation. These include the upper limit of the X axis, the plotting boundaries, and any changes made via the Special Features Menu.

On the cassette version, these variables are lost and must be re-entered after changing the equation at line 10. However, if only one equation is being plotted, and operations are performed in the order presented on the menu, there will be no need to re-enter these variables since the equation will have been entered before the variables are entered.

(2) ENTER NEW LIMITS

Since many 2D equations represent curves which stretch on endlessly along the X axis, it is necessary to indicate how large a section of the function is to be plotted. It is only possible to plot regions situated about the origin. (X=0,Y=0)

First, a value for the upper limit of the X axis must be selected. This value is the value of the point where the X axis meets the right-hand border of the graphics screen. Selection of this value automatically fixes the value of the upper limit of the Y axis. Unless altered from the Special Features Menu, the ratio of the upper limits of the axes X:Y is 1:1.

Having selected an upper limit for the X axis, it is now necessary to select the plotting boundaries for the X axis. This selection determines how large a section of the function will be plotted. While it is desirable to fill the graphics screen with the function as much as possible, plotting boundaries which cause the function to exit the upper and lower borders of the screen must be avoided. If necessary, a larger upper limit for the X axis, or an alteration to the scale factor of the Y axis can be entered to bring the entire function back onto the screen.

(3) BEGIN FUNCTION PLOTTING

After having entered the equation at line 10, selected an upper limit for the X axis, and entered plotting boundaries on the X axis, the function can be plotted. Pressing (3) begins this process. A return to the Menu during plotting can be achieved by pressing 'R'. Similarly, when the function is complete, pressing 'R' will affect a return to the Menu.

(4) DISPLAY GRAPHIC SCREEN

Pressing (4) causes the display to return to the graphics screen.

This selection may be made at any time from the Menu to verify the contents of the graphics screen.

(5) CLEAR GRAPHICS SCREEN

Selecting (5) causes the current function plotted on the graphics screen to be erased. To protect against accidental clearing, an additional response to "SURE? (Y/N)" is required to clear the screen. Only a 'Y' response to this query will clear the graphics screen.

(6) REMOVE COORDINATE AXES

This selection causes the coordinate axes to be erased from the graphic screen. The axes will not reappear unless selection (7) is made. If a function plot without axes is desired, the axes should be removed before plotting. Removing the axes after plotting is complete may also erase some of the function.

(7) REPLACE COOOORDINATE AXES

This selection restores the coordinate axes to the graphics screen. The axes will appear on all future plots unless (6) is selected.

(8) SPECIAL FEATURES MENU

The Special Features Menu gives added control of function plotting to the advanced function plotter.

The following features are offered:

(1) Alter length of step along X axis

The length of step refers to the nominal number of points between steps along an axis. This value affects the length of the line segments drawn, and hence the smoothness of the curve. For some functions, larger step values can speed plotting without loss of accuracy. For very curvy functions, a step value of 2 will increase accuracy at a slight sacrifice in speed. A step value of 1 is meaningless and cannot be entered. The default value is 3.

(2) Alter scale factor of Y axis

This feature allows for altering the upper limit of the Y axis. Normally, that limit is the same as the upper limit of the X axis. By changing the scale factor of the Y axis, function plots may either be extended or reduced in their Y extent. This feature is useful when plotting a relationship in which the units of Y differ from the units of X, or if a function is very large or very small in its Y extent.

(3) Save function to disk (or tape)

Depending upon the version of the program (disk or cassette), this feature allows for function plots to be stored permanently on diskette or cassette. By assigning a meaningful name to the function plotted, recall can be facilitated.

(4) Read function from disk (or tape)

This feature allows for the reading in of stored function plots.

(From the Main Menu...)

(9) PRINTOUT INFORMATION

Selecting (9) guides the user through the following text screens:

Printouts of the functions plotted by this program can be made using the Radio Shack screen print routine or other comercially available software. After having plotted the desired function, exit this program using the (BREAK) key. Load the screen print routine and follow the normal procedures for printing the graphics screen.

Printouts to a Line VII Printer with the Radio Shack screen print routine will appear distorted along the X axis since that routine translates the square graphics screen into a rectangle on the printer. By setting the scale factor of the Y axis to .71, a 'printer corrected' hardcopy can be obtained.

3D PLOT

Below is an in depth analysis of the features of 3D PLOT, arranged in the order of appearance on the Main Menu.

(0) GENERAL INFORMATION

Selecting (0) guides the user through the following text screens:

This program will plot a three dimensional surface entered by the user in an X,Y,Z coordinate system. The equation must be solved explicitly for Y and entered at line 10. An upper limit for the X axis must be selected, and plotting boundaries for the X and Z axes must also be entered. The surface can then be plotted.

The surface will continue to plot and will remain on the graphics screen until 'R' is pressed to return to the menu. Returning to the menu does not destroy the surface on the graphics screen. The surface will remain intact until it is cleared from the menu.

This allows for the plotting of several surfaces on a single graphics screen.

Visualization of the 3D surfaces plotted by this program can be facilitated by noting:

- The first set of lines are plotted front to back and are all parallel to the X-Y plane.

- The second set of lines are plotted right to left and are all parallel to the Y-Z plane.

The Z axis has been adjusted for the perceptual distortion which is inherent to a 45 degree oblique projection. The scaling factor used is 2.828, which equates to a 50% shortening along the Z axis. This value can be altered from the Special Features Menu.

A tone will sound indicating that the plot is complete.

(1) ENTER NEW EQUATION

Presing (1) prompts the user to enter the equation of the surface which is to be plotted. This equation should be solved explicitly for Y; that is, Y should appear to the left of the equal sign, and the right side should be an expression which varies only in X and Z. The equation is entered into the program at line 10.

By entering an equation at line 10, the program has been edited, and all variables are reset to zero. On the disk version, any variables which may have been entered are saved to disk before entry of the equation. These include the upper limit of the X axis, the plotting boundaries, and any changes made via the Special Features Menu.

On the cassette version, these variables are lost and must be re-entered after changing the equation at line 10. However, if only one equation is being plotted, and operations are performed in the order presented on the menu, there will be no need to re-enter these variables since the equation will have been entered before the variables are entered.

A section at the end of this manual discusses in further detail the types of equations which might be entered, and the resulting surfaces.

(2) ENTER NEW LIMITS

Since many 3D equations represent surfaces which stretch endlessly through space, it is necessary to indicate how large a portion of the surface is to be plotted.

First, a value for the upper limit of the X axis must be selected. This value is the value of the point where the X axis meets the right-hand border of the graphics screen. Selection of this value automatically fixes values for the upper limits of the Y and Z axes. Unless altered from the Special Features Menu, the ratio of the upper limits of the axes X:Y:Z is 1:1:2.828. The greater value of the Z axis serves to correct for the perceptual distortion associated with oblique projections.

Having selected an upper limit for the X axis, it is now necessary to select the plotting boundaries for the X and Z axes. Their selection determines how large a portion of the surface will be plotted. While it is desirable to fill the graphics screen with the surface as much as possible, plotting boundaries which cause the surface to exit the borders of the screen must be avoided. A rule of thumb is to select plotting boundaries which are about 75% of the upper limits of the axes.

(3) BEGIN SURFACE PLOTTING

After having entered the equation at line 10, selected an upper limit for the X axis, and entered plotting boundaries on the X and Z axes, the surface can be plotted. Pressing (3) begins this process. The time required to plot a surface depends on many factors; complexity of the equation, plotting boundaries, axes limits, and length of step, all play a role. Most surfaces will plot in less than 5 minutes.

A return to the Menu during plotting can be achieved by pressing 'R'. Similarly, when the surface is complete, pressing 'R' will affect a return to the Menu.

(4) DISPLAY GRAPHICS SCREEN

Pressing (4) causes the display to return to the graphics screen.

This selection may be made at any time from the Menu to verify the contents of the graphics screen.

(5) CLEAR GRAPHICS SCREEN

Selecting (5) causes the current surface on the graphics screen to be erased. To protect against accidental clearing, an additional response to "SURE? (Y/N)" is required to clear the screen. Only a 'Y' response to this query will clear the graphics screen.

(6) REMOVE COORDINATE AXES

This selection causes the coordinate axes to be erased from the graphics screen. The axes will not reappear unless selection (7) is made. If a surface without axes is desired, the axes should be removed before plotting. Removing the axes after plotting is complete may also erase some of the surface.

(7) REPLACE COOORDINATE AXES

This selection restores the coordinate axes to the graphics screen. The axes will appear on all future plots unless (6) is selected.

(8) SPECIAL FEATURES MENU

The Special Features Menu gives added control of surface plotting to the advanced surface plotter. The following features are offered:

(1) Alter slicing along X axis

This refers to the width (nominal number of points) between 'stripes' in the X direction. Some surfaces may look better the default value of 14 is made larger or smaller.

(2) Alter slicing along Z axis

As above, some surfaces may appear better if this value is changed.

(3) Alter length of step along X axis

The length of step refers to the nominal number of points between steps along an axis. This value affects the length of the line segments drawn, and hence the smoothness of the curves. For some surfaces (planes for example), larger step values can speed plotting without loss of accuracy. For curved surfaces, smaller steps increase accuracy at the sacrifice of speed. The default value is 5.

(4) Alter length of step along Z axis

As above, some surfaces may benefit by altering this value.

(5) Alter scale factor of Y axis

This feature allows for altering the upper limit of the Y axis. Normally, that limit is the same as the upper limit of the X axis. By changing the scale factor of the Y axis, surfaces may either be extended or reduced in their Y extent. This feature is useful when plotting a relationship in which the units of Y differ from the units of X.

(6) Alter scale factor of Z axis

Normally, the upper limit of the Z axis is 2.828 times the upper limit of the X axis. By changing the scale factor of the Z axis, surfaces may be extended or reduced in their Z extent. Again, this feature is useful in applications where different scales are needed along each of the axes.

(7) Save surface to disk (or tape)

Depending upon the version of the program (disk or cassette), this feature allows for surfaces to be stored permanently on diskette or cassette. By assigning a meaningful name to each surface, recall will be facilitated.

(8) Read surface from disk (or tape)

This feature allows for the reading in of stored surfaces. Sample surfaces are provided with the MATHMENU collection.

(From the Main Menu...)

(9) PRINTOUT INFORMATION (Appears on Disk version only)

Selecting (9) guides the user through the following text screens:

Printouts of the surfaces generated by this program can be made using the Radio Shack screen print routine or other commercially available software. After having generated the desired surface, exit this program using the (BREAK) key. Load the screen print routine and follow the normal procedures for printing the graphics screen.

Printouts to a Line VII Printer with the Radio Shack screen print routine will appear distorted along the X axis since that routine translates the square graphics screen into a rectangle on the printer. By setting the scale factors of the Y and Z axes to .71 and 2.51 respectively, a perceptually adjusted, 'printer corrected' hardcopy can be obtained.

THE HIDDEN LINE PROBLEM

This program does not attempt to remove lines which would be invisible to the viewer. Instead, surfaces are plotted as though they were transparent sheets of plastic. It is this plotting method which enables surface plotting to be a fairly speedy process.

There are at least two routines for the Color Computer which will plot Surfaces of Revolution with hidden lines removed. Surfaces which plot in 5 minutes with 3D PLOT may take over an hour to plot with a program which removes hidden lines. More importantly, these programs do not have the flexibility to plot all equations of the form Y = F(X,Z). Instead, they plot revolutions of a 2D function, Y = F(X), about the Y axis.

This author would be most interested to hear from anyone who believes they have a routine which can be adapted to 3D PLOT which will remove hidden lines for all 3D surfaces.

SURFACES OF REVOLUTION

Plots of rotations of 2D functions can be achieved as follows:

Given a 2D function of the form Y=F(X) for which a surface of revolution is desired, enter at Line 10 the expression for the radius in terms of X and Z. That expression is: R=SQR(X*X+Z*Z)

Next, enter the equation Y=F(X) at Line 11 with R as the variable instead of X. (Lines 10-14 may be used to enter equations, a RETURN statement is located at Line 15.) An example follows:

To plot a revolution of Y=SIN(X), enter:

```
10  R=SQR(X*X+Z*Z)
11  Y=SIN(R)
```

Upon return to the Main Menu, select an upper limit of the X axis, and plotting boundaries for the X and Z axes. A good selection for this surface is to set the upper X axis to 5, the lower plotting boundaries to -3, and upper to 3 on both the X and Z axes. The surface should appear similar to the one which appeared in the ad for MATHMENU.

OTHER SURFACES

Any equation of the form Y=F(X,Z) describes a 3D surface. Not all surfaces are smooth, some may have wild oscillations which make plotting very difficult. In general, equations which involve lower powers of X and Z will be well behaved. A knowledge of the behavior of 3D equations and the surfaces they represent will greatly assist the novice plotter. An advanced Calculus text is a good source of such information.

MATRIXOP

The following Information pages can be accessed by pressing (Q):

This program will perform a variety of operations on a single matrix (A), or a pair of matrices (A) and (B), each of which may have as many as 8 rows and 8 columns. Addition, subtraction, and scalar multiplication of a VECTOR can be achieved by entering one dimensional matrices. Performing operations on (A) and (B) does not change these matrices. (Calculating the determinant of (A) does, however, cause (C) to take on the value of (A) Inverse.)

'Clerical' functions are contained on menu one. These include entry of matrices (A) and (B), display and printing of matrices (A), (B) and (C), rename functions, and an edit function, which allows individual matrix elements to be changed. Use of the 'RENAME: (C) to (A)' function causes matrix (A) to take on the value of (C) so operations can be performed on it without having to re-enter its elements. New matrices may be entered at any time by selecting options (1) or (2).

Menu two displays the various matrix operations available. These include calculation of the determinant, inverse, and transpose of (A), multiplication of (A) by a scalar, and the raising of (A) to a power. In accordance with matrix definition, (A) raised to the zero power yields the Identity matrix, (I). The sum, difference, or product of (A) and (B) can also be calculated. Calculation of (B)-(A) and (B)*(A) can be achieved by using the rename function on menu one.

The solution to a system of simultaneous linear equations can be obtained by calculating the inverse of the coefficient matrix and premultiplying the constant vector by it. Ex.

Given: (A) * (X) = (B)

for which the solution vector (X) is desired, proceed as follows:

- Enter the elements of (A)
- Enter the elements of (B)
- Calculate (C) = (A) INVERSE
- Rename: (C) to (A)
- Calculate (C) = (A) * (B)

Matrix (C) will contain the solution vector (X).

A short discussion of each of the features on menus one and two follows.

Selections from menu one:

(1) ENTER NEW MATRIX (A)

This selection enables the user to enter the individual elements of the matrix which will be operated on. After entering the number of rows (horizontal) and columns (vertical), the user will be prompted to enter the elements of matrix (A) from left to right, beginning with Row 1. In accordance with matrix subscripting convention, the elements are labelled as shown below:

11	12	13	14
21	22	23	24
31	32	33	34
41	42	42	44

Return to the menu is automatic upon completion of matrix entry.

(2) ENTER NEW MATRIX (B)

As above, the elements of a second matrix (B) are entered by selecting (2).

(3), (4), (5) DISPLAY MATRICES (A), (B), (C)

These selections allow the user to at any time verify the contents of either (A), (B), or (C).

(6) PRINT MATRICES (A), (B), (C)

Selecting (6) prompts the user to select the matrix which is to be printed to the printer. This is useful if a hardcopy of matrix calculations is required.

(7) RENAME: (A) to (B), (B) to (A)

This feature causes matrices (A) and (B) to be renamed to each other. This selection is used for calculating (B)-(A) and (B)*(A), as these operations are not offered directly on menu two.

(8) RENAME: (C) to (A)

Selection (8) causes matrix (A) to take on the value of result matrix (C). In this way, operations can be performed on (C) without having to re-enter its elements manually.

(9) EDIT MATRICES

This feature allows the user to make changes to matrix (A) or (B). Individual elements can be changed in either matrix by entering a new element for the element which is in error.

Operations from menu two:

(0) INFORMATION

Selecting (0) from menu two guides the user through the same information screens as are available from menu one.

(1) K = (A) DETERMINANT

Selecting (1) calculates the determinant of a square matrix (A) and displays the result. This operation causes matrix (C) to take on the value of (A) Inverse.

(2) C = (A) TRANSPOSE

The transpose of a matrix is that matrix which results from transposing the rows and columns of the original matrix. Operation (2) causes (C) to become the transpose of (A).

(3) C = (A) INVERSE

The inverse of a matrix is that matrix which, when multiplied by the original matrix, yields the Identity matrix (I). The inverse of a matrix is useful for solving a system of linear equations.

(4) (C) = K * (A)

Operation (4) causes (C) to take on the value of matrix (A) multiplied by a selected scalar, K.

(5) (C) = (A) RAISED TO THE Nth POWER

In accordance with a little known definition of matrix exponentiation, this operation will compute the matrix resulting from (A) being raised to a scalar power. Matrix exponentiation is defined by:

(C) = N * ((A) - (I)) + (I)

where N is the scalar exponent, and (I) is the Identity matrix.

(6) (C) = (A) + (B)

Operation (6) sets (C) equal to the sum of matrices (A) and (B).

(7) (C) = (A) - (B)

Operation (7) sets (C) equal to the difference of matrices (A) and (B).

(8) (C) = (A) * (B)

Operation (8) sets (C) to the matrtix resulting from (B) being premultiplied by (A). (This is not necessarily equivalent to (B) * (A).)

VECTOROP

The following Information pages can be accessed by pressing (0):

This program will perform a variety of operations on a single vector (A), or a pair of vectors (A) and (B), each of which may have as many as 20 elements each. Performing operations on (A) and (B) does not change these vectors.

'Clerical' functions are contained on menu one. These include entry of vectors (A) and (B), display and printing of vectors (A), (B) and (C), rename functions, and an edit function, which allows individual vector elements to be changed. Use of the 'RENAME: (C) to (A)' function causes vector (A) to take on the value of vector (C) so operations can be performed on it without having to re-enter its elements. New vectors may be entered at any time by selecting options (1) or (2).

Menu two displays the various vector operations available. These include calculation of the cross and dot products, normal length of (A), multiplication of (A) by a scalar, and the unit vector (A). The sum, difference, or angle between (A) and (B) can also be calbulated. Calculation of (B)-(A) and (B) CROSS (A), although not offered directly, can be achieved by using the rename function on menu one.

Additionally, the plane containing vectors (A) and (B) can be found. The plane is given in the form of a general equation into which the coordinates (X0,Y0,Z0) of a point through which the plane is known to pass, must be inserted.

A short discussion of each of the features on menus one and two follows.

Selections from menu one:

(1) ENTER NEW VECTOR (A)

This selection enables the user to enter the individual elements of the vector which will be operated on. After entering the number of elements in vector (A), the user will be prompted to enter the elements of (A) from left to right. Return to the menu is automatic upon completion of vector entry.

(2) ENTER NEW MATRIX (B)

As above, the elements of a second vector (B) are entered by selecting (2).

(3), (4), (5) DISPLAY VECTORS (A), (B), (C)

These selections allow the user to at any time verify the elements of either (A), (B), or (C).

(6) PRINT VECTORS (A), (B), (C)

Selection (6) prompts the user to select the vector which is to be printed to the printer. This is useful if a hardcopy of vector calculations is required.

(7) RENAME: (A) to (B), (B) to (A)

This feature causes vectors (A) and (B) to be renamed to each other. This selection is used for calculating (B)-(A) and (B) CROSS (A), as these operations are not offered directly on menu two.

(8) RENAME: (C) to (A)

Selection (8) causes vector (A) to take on the value of result vector (C). In this way, operations can be performed on (C) without having to re-eneter its elements manually.

(9) EDIT VECTORS

This feature allows the user to make changes to vector (A) or (B). Individual elements can be changed in either vector by entering a new value for the element which is in error.

Operations from menu two:

(0) INFORMATION

Selecting (0) from menu two guides the user through the same information screens as are available from menu one.

(1) (C) = (A) + (B)

Operation (1) sets (C) equal to the sum of vectors (A) and (B).

(2) (C) = (A) - (B)

Operation (2) sets (C) equal to the difference of vectors (A) and (B).

(3) (C) = K * (A)

Operation (3) causes (C) to take on the value of vector (A) multiplied by a selected scalar, K.

(4) (C) = (A) CROSS (B)

Selection (4) computes the vector resulting from vector (A) being crossed with vector (B). This operation is meaningful only if vectors (A) and (B) have 3 elements each.

(5) (C) = UNIT VECTOR OF (A)

The unit vector of (A) is that vector which points in the same direction as (A) and has a unit length of 1. This operation is valid for all vectors having from 2 to 20 elements.

(6) NORMAL LENGTH OF (A)

Operation (6) computes the length of vector (A). This operation is valid for all vectors having from 2 to 20 elements.

(7) ANGLE BETWEEN (A) and (B)

This operation will compute the angle between vectors (A) and (B). Such a calculation is meaningful only for vectors having 2 or 3 elements each. The angle will be displayed in degrees and radians.

(8) DOT PRODUCT OF (A) AND (B)

Operation (8) will calculate the dot (or scalar) product of vectors (A) and (B). Unlike the cross product, the dot product results in a scalar result. This operation is valid for vectors of any number of elements.

(9) PLANE CONTAINING (A) AND (B)

The plane containing vectors (A) and (B) can be found if these vectors have 3 elements each. (If the vectors have only two elements each then they reside in the XY plane.) The resulting plane will be given by an equation into which the coordinates of a point which is to lie in the plane must be entered. The resulting expression can then be simplified to give an equation which defines the plane containing vectors (A), (B), and the point entered.

NUMDIFN

NUMDIFN will approximate the derivative of an equation Y = f(X) at one or more points of interest. The function to be evaluated is entered into the program at line 10 by the user. The 'define function' (DEF FN) command of BASIC is utilized for this purpose. As shown in the example in the program, the equation:

Y = 2*X*X + SIN(X)

would be entered at line 10 as:

```
10 DEF FN F(X)=2*X*X+SIN(X)    (ENTER).
```

to re-enter the program, type:

```
GOTO 100    (ENTER)
```

Upon return to the program, the point or points at which a derivative is desired can be entered. If a hardcopy of the results is required, the derivatives can be printed to the printer.

The solution is obtained numerically rather then by direct differentiation. Thus any equation can be evaluated, even if solution by direct methods would be difficult or impossible. The program obtains the approximation by evaluating the definition of the derivative of f(X) for a small value of h:

$$\frac{df}{dX} = \frac{f(X + h) - f(X)}{h}$$

The value of h selected is proportional to X. The relationship h = .00001 * X was found to afford good accuracy for all values of X. The approximation of the derivative offered by this program should be sufficiently accurate for most applications.

NUMINTEG

NUMINTEG will evaluate a single variable integral between definite limits by performing Simpsons Approximation over a specified number of panels (iterations). The expression which is to be integrated is entered into the program at line 10 by the user. The 'define function' command is used for this purpose.

Ex. To evaluate the integral of:

```
3*X*X        2*X
-------  +  -------
SIN(X)       LOG(X)
```

between the limits of .5 and 1, the equation would first be defined at line 10 as:

```
10 DEF FN F(X)=(3*X*X/SIN(X))+(2*X/LOG(X))   (ENTER)
```

To re-enter the program type:

```
GOTO 100   (ENTER)
```

When prompted for the lower limit type '.5'

When prompted for the upper limit type '1'

The number of iterations to be performed affects the accuracy of the result. For most functions which do not involve high powers of X or integration over a large interval, a selection of 20 will provide sufficient accuracy. Selecting higher values will slow the calculations and provide extra accuracy only in limited instances.

The results of the integration and equation being integrated can be printed to the printer if desired.

NUMINTEG performs integration numerically rather than by direct methods. This enables integrals of all equations to be evaluated, not just those for which a direct solution is possible. Simpsons Approximation is a method by which the interval of integration is divided into small regions for which a close approximation of the area can be calculated. By summing these small areas, an approximation for the integral over the entire region is obtained.

LG#ADD AND LG#MULT

LG#ADD and LG#MULT are two programs which perform operations on large numbers (numbers with many digits) without losing figures to scientific notation as would be the case if the operation had been performed in BASIC or by a hand held calculator.

LG#ADD will compute the sum of two numbers which may have as many as 100 digits each.

LG#MULT will compute the product of two numbers which may have as many as 30 digits each.

In both programs the numbers are entered digit by digit, from left to right, commas omitted. The left arrow key has been programmed to erase entries to the immediate left, just as it would behave if number entry were performed by using the BASIC INPUT statement. Number entry is terminated by the (ENTER) key.

The period of time required for calculation is dependent on the number of digits in the numbers entered. The most lengthy calculation is the multiplication of two 30-digit numbers, which takes less than a minute.

QUAD.EQN

QUAD EQN will compute the roots of (values of X which satisfy) a quadratic equation. The general form of a quadratic equation is:

A*X*X + B*X + C =0

where A, B, and C are real valued constants. The program employs the well known quadratic formula:

```
        -B +/- SQR ( B*B - 4*A*C )
X = ------------------------------
                2*A
```

This formula gives rise to two values of X which both satisfy the quadratic equation. Depending upon the values of the coefficients A, B, and C, the roots will fall into one of the following categories:

1) REAL, UNEQUAL — the two roots are real numbers (no imaginary parts) and are not equal to one another.

Ex. The equation X*X - 3*X + 2 = 0 has roots:

R1 = 1, R2 = 2

2) REAL, EQUAL - the two roots are real numbers and are equal to each other.

Ex. The equation X*X - 8*X + 16 = 0 has roots:

R1 = 4, R2 =4

3) COMPLEX - the roots are made up of a real part and an imaginary part. Imaginary numbers arise when square roots of negative numbers are taken. 'I' is commonly used to denote the square root of (-1).

Ex. The equation 5*X*X + 2*X + 2 = 0 has roots:

R1 = -.2 + .6*I, R2= -.2 - .6*I

QUAD EQN will compute any of the 3 types of roots which arise from solution of a quadratic equation. The results will be displayed on the screen, and may be printed to a printer if desired.

LSTSQRS

LSTSQRS is a program which will apply least squares curve fitting techniques to a collection of up to 100 data points. Depending upon the method selected, a cubic, quadratic, or linear equation will be calculated for the data entered.

The first step is to determine which type of curve fit is required. A knowledge of the relationship being plotted is necessary in order to make this decision. For data which portray a linear (straight line) relationship, the linear selection is appropriate. If a parabolic curve (A*X*X + B*X + C) is anticipated, then the quadratic fit should be selected. For data which behaves according to a cubic relationship (A*X*X*X + B*X*X + C*X + D) the cubic fit is appropriate.

After the method of fit is selected, the data points are entered. The data is referred to as being in X,Y pairs, but of course any letters may have been used for the data which you will be plotting. The data point pairs are entered as prompted until data entry is terminated by entering a letter as a data point.

After a short period of calculation the equation of the curve which best fits the data points is displayed. This curve needn't pass exactly through any of the points entered. Least squares curve fitting finds a curve which runs amidst the data points in such a way as to minimize the sum of the squared deviations from that line. This method of curve fitting is especially useful if there is a good chance that one or more of the data points are in error. Least squares tends to minimize the effect of errant data points on the solution curve.

PRIME CK

PRIME CK will determine whether or not any number up to one million is a prime number. A prime number is a number which is not divisible without remainder by any number but itself and 1. The first 10 primes are:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

This program uses trial divisions by prime numbers to determine a number's primality. It is only necessary to carry out divisions up to the square root of the number in question. By this method a number can be determined prime (or not prime) in just a few seconds. If the number tested is not prime, its first divisor will be displayed.

The search for prime numbers has interested mathematicians since antiquity. Lists of prime numbers have been found in the Egyptian pyramids. The computer age has seen renewed interest in the study since larger numbers can now be examined. The largest known prime is some 13,395 digits long, a number which would require over 13K of memory just to store.

BINOMEXP

BINOMEXP will compute the terms resulting from the expansion of (A+B) raised to an integer power less than 33. While it is rare that powers exceeding 10 are needed, lower power expansions are common and can be quite time consuming if attempted by hand. The resulting terms will be displayed to the screen, and can be routed to the printer if necessary.

The program employs a mnemonic device called Pascals Triangle which is helpful for recalling the coefficients resulting from a binomial expansion. The first six rows of this triangle are shown below:

```
               1                    N=0
            1     1                 N=1
         1     2     1              N=2
      1     3     3     1           N=3
   1     4     6     4     1        N=4
1     5    10    10     5     1     N=5
```

The powers to which each row corresponds are shown at the right. The pattern is thus: excluding the end entries which are always 1, any number in the triangle will be the sum of the numbers directly above it to the left and right.

By recalling this triangle for the coefficients, and knowing that the sum of the exponents in any term in the result is equal to the power to which the binomial is being raised, binomial expansion can be achieved by hand at a significant time savings over that which would be required to expand the binomial by repetitive multiplication.

By using the above triangle and applying the 'sum of exponents' theory, the result of (A+B) raised to the 4th power is found to be:

```
A#4 + 4*A#3*B + 6*A#2*B#2 + 4*A*B#3 + B#4
```

where the number sign (#) has been used to designate exponentiation.

For those who find these mnemonics confusing, BINOMEXP will supply the result at an even greater time savings.

BASECONV

BASECONV will convert an integer number from any base to its equivalent number in another selected base. While this program will probably prove most useful for calculations between bases 2, 10, and 16, it can be used to convert between any two bases, no matter how obscure they might be.

The following information will be requested in this order:

1st - Enter the base that the number is to be converted FROM.

2nd - Enter the number which is to be converted, leaving a space between each place.

3rd - Enter the base that the number is to be converted TO.

For base systems larger than 10, the letters A to Z may be used during number input for the numbers 10 to 35. The use of this coding during result output is optional. If letter output is selected, a key showing the letters and their equivalent numbers will appear with the result. Results can be printed to the printer if required.

RECT-POL

RECT-POL will convert a pair of Cartesian coordinates into their equivalent polar (radius, angle) coordinates. The reverse operation, polar to rectangular conversions, is also available.

The angle THETA may be entered in either radian measure or in degrees. The resulting angle, when appropriate, will be reported in both radians and degrees. The result of the conversions can be printed to the printer if necessary.

The polar coordinate system uses an angle THETA and distance from origin R to define a point in 2D space. The more familiar Cartesian coordinate system defines a point in space as having an X and Y value associated with it. Certain problems can be solved more easily in a polar coordinate system than they could be in a Cartesian reference system.

The equations governing conversions between the two coordinate systems are:

Rectangular to Polar:	R = SQR (X*X + Y*Y) THETA = ARCTAN (Y/X)
Polar to Rectangular:	X = R * COS(THETA) Y = R * SIN (THETA)