TEXAS INSTRUMENTS

SOFTWARE

TI-74 MATHEMATICS LIBRARY

GUIDEBOOK

TEXAS INSTRUMENTS

TI-74 MATHEMATICS LIBRARY GUIDEBOOK

Manuals developed by: The staff of Texas Instruments Instructional Communications

Jacquelyn F. Quiram
Kenneth E. Heichelheim
Brenda Cornitius

With contributions by: Bruno Didier
Robert A. Pollan
S. S. Millwright
J. H. Puckett
Robert E. Whitsitt, II

Important

Texas Instruments makes no warranty, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding these programs or book materials and makes such materials available solely on an "as-is" basis. In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability to Texas Instruments, regardless of the form of action, shall not exceed the purchase price of this cartridge. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the user of these programs or book materials by any other party.

Table of Contents

How to Use this Guidebook

This guidebook describes the contents and use of the TI-74 Mathematics Library cartridge. The guidebook is organized to help you use the programs and subprograms in the cartridge.

Organization of the Guidebook

The guidebook includes five chapters, an appendix describing the accessible subprograms in the cartridge, and an appendix explaining service and warranty information.

Chapter 1 describes the programs in the cartridge, the general use and care of the cartridge, how to use an optional printer, and how to use the Directory program.

Chapters 2 through 5 provide detailed information about each program in the cartridge. The discussion of each program includes:

- A brief presentation of general information about the program and the inputs required by the program.
- Step-by-step instructions for using the program.
- An application example demonstrating the use of the program.

Appendix A describes the subprograms called by the programs of this library. These subprograms may be useful to you as you develop your own programs. Instead of writing the statements to perform a task, you can call the subprogram that does the task.

Service and warranty information is provided in Appendix B, which is at the end of the guidebook for easy reference.

Chapter 1: Getting Started

This chapter introduces you to the TI-74 Mathematics Library cartridge and helps you get started using the cartridge.

Table of Contents

Contents of the Mathematics Cartridge

The TI-74 Mathematics cartridge enables you to perform calculations quickly and easily. The title, filename, and a brief description of each cartridge program are provided below for your reference.

The Programs

The Mathematics cartridge contains the following programs.

Program Title	Filename	Description
Directory	DIR or CONTENTS	Displays the title and filename of each program in the cartridge
Complex Functions	COMPF	Performs operations on complex numbers and their algebraic, trigonometric, exponential, and logarithmic functions
Gamma Function	GAM	Computes the value of gamma(x) and ln(abs(gamma(x)))
Polynomial Multiplication	POL	Performs multiplication on real polynomials
Prime Factors	PRI	Determines the prime factors of any positive integer
Cubic Splines	SPL	Interpolates between input points
Relative Minimum	MIN	Finds a relative minimum for a given function
Root Finder-Bisection	BIS	Uses the bisection method to find the roots of a function
Root Finder-Newton	NEW	Uses the Newton-Raphson method to find the roots of a function

The Programs (Continued)

Program Title	Filename	Description
Convolution	CON	Finds a linear system's output for a specified input waveform
Differential Equations (Runge-Kutta)	DIF	Uses a fifth-order Runge-Kutta method to solve differential equations
Gauss Quadrature	GAU	Uses Gaussian integration to approximate integrals
Complex System	COMPS	Solves a system of n×n complex equations
Matrices	MAT	Performs matrix operations, solves linear simultaneous equations, and evaluates determinants
Tridiagonal System	TRI	Uses Gaussian elimination to solve a tridiagonal system of equations

Using Cartridges with Your TI-74

Although the cartridges for your TI-74 are durable devices, you should follow these guidelines for installing and caring for cartridges.

Caring for the Cartridge

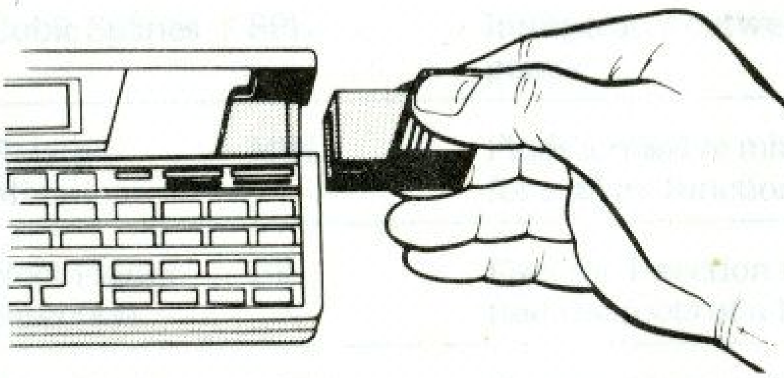
Handle the cartridge with the same care you would give any other piece of electronic equipment. You should:

- Avoid static electricity. Prior to handling the cartridge, touch a metal object to discharge any static electricity.
- Store the cartridge in its original container or in the cartridge port, on the upper right side of the TI-74.

Installing a Cartridge

The TI-74 is shipped with a port protector in the cartridge port. The port protector resembles a cartridge and is installed and removed in the same way.

1. Make sure the TI-74 is turned off. Installing a cartridge while the TI-74 is on may result in memory loss.
2. If the port protector or a cartridge is currently in the port, remove it by placing your thumb on the ridged area on top of the cartridge and sliding the cartridge to the right. Store the removed cartridge in its container.
3. Turn the Mathematics cartridge so that the ridges are facing upward.
4. Insert the cartridge into the port, small end first.



5. Slide the cartridge to the left until it snaps into place.

You should keep a cartridge or the port protector in the port at all times to prevent the accumulation of dust.

Running a Program

To run a cartridge program, follow these general steps:

1. Place the TI-74 in BASIC mode.

2. Type **RUN *"filename"*** and press **[ENTER]**.

   For example, the first program in this library is Complex Functions, and its filename is "COMPF". To execute this program, you type **RUN "COMPF"** and press **[ENTER]**.

   The program title is displayed for several seconds.

3. Press **[ENTER]** to begin the program immediately, or wait until the program begins automatically.

When running a program, enter all commands and responses exactly as they appear in the step-by-step instructions.

Pausing During A Program

To pause during the execution of a program, press the **[BREAK]** key.

To resume program execution from the breakpoint, first press **[CLR]** or **[ENTER]** to clear the display. Then type **CON** and press **[ENTER]**.

If you press **[BREAK]** when a prompt is in the display, the prompt may not reappear when you resume the program. However, the prompt still exists and you must respond to the prompt before the TI-74 will proceed to the next step.

Note: If you press the **[BREAK]** key and then leave the BASIC mode or turn the TI-74 off, you cannot resume the program with CON.

Cancelling A Program

If you are running a program and wish to cancel it, press the **[BREAK]** key and then press either the **[ENTER]** or **[CLR]** key.

You can then restart the same program, execute a new program, or perform any other TI-74 operation.

Guidelines for Using the TI-74 Programs

The TI-74 programs contained in the cartridge provide helpful display prompts during the programs. An explanation of these prompts and a few guidelines for using the programs may be helpful as you get started.

Display Prompts

Display prompts are brief sentences that appear in the display to help you enter information into the program. There are three types of display prompts:

- Entry prompts—these prompts end with a colon and instruct you to enter a value. When applicable, default values appear immediately after the colon. To respond, press **[ENTER]** to accept the current value or type a new value for the variable and press **[ENTER]**.
- Question prompts—these prompts end with a "?" and require a yes or no response. To respond, press **[Y]** for yes or **[N]** for no.
- Message prompts—these prompts explain an operation of a program. Because no information needs to be entered, you can press **[ENTER]** or **[CLR]** to immediately proceed to the next prompt.

Editing Data in a Program

Many of the programs enable you to edit the elements (data values) that you enter into a program. You have the option of editing all the data values or of selecting specific elements to edit.

Note: If you replace a value with a value that has fewer digits, be sure to delete any "leftover" digits that remain in the display. For example, suppose the current value is 389 and you want to change it to 23. If you just type **23**, the value in the display becomes 239. You must delete the remaining digit (the 9) before pressing **[ENTER]**.

To enter a value that has fewer digits than the original value, you can:

- Press **[CLR]**, type the value, and press **[ENTER]**.
- Type the value and press **[CTL] [↓] [ENTER]**.
- Type the value and use the space bar to delete the rest of the entry; then press **[ENTER]**.

Using the Optional Printer

When you run the Mathematics programs, you may want to use a compatible printer, such as the PC-324 Printer. A printer provides a convenient, permanent record of your programs and data files. Refer to the printer manual for information on connecting the printer to your TI-74.

The Printout

The printout contains all of the information needed to execute a particular program (with the exception of subprograms). The printout includes:

- Program title and filename
- Selected options
- Pertinent input data
- Results

Accessing the Printer

To use an optional printer:

1. Respond to the `Use Printer?` prompt by pressing **[Y]**. The `Enter Device Code:` prompt appears in the display.

2. Respond to the `Enter Device Code:` prompt by typing the device code (a number that identifies the device) and pressing **[ENTER]**. (Refer to the printer manual for the device code.)

Note: When a printed record is produced, the program does not pause while displaying individual results. Instead, the TI-74 prints a continuous list of results until the output is complete.

Using the Directory Program

The Directory program serves as a table of contents for the programs available with the cartridge. This program also provides a handy reference for program filenames. If you have an optional printer, you may want to print out a copy of the Directory program to use for future reference.

The Directory Program

The Directory program displays two names for each program in the cartridge. These include:

- The title of the program
- The filename that is used to execute the program

For example, the title for the first program in this library is Complex Functions and its filename is "COMPF".

Displaying the Names in the Directory

In the Directory program, the four arrow keys display the names in the directory.

- **[↑]**—Displays the previous title in the directory. If you press this key at the first title in the list, the Directory program prompts: `Exit Program?` Press **[Y]** to exit the Directory program, or press **[N]** to return to the first program title.
- **[↓]**—Displays the next title in the directory. If you press this key after the last title in the list, the Directory program prompts: `Exit Program?` Press **[Y]** to exit the Directory program, or press **[N]** to return to the last program title.
- **[→]**—Displays the filename of a program. The TI-74 ignores this key if a filename is already in the display.
- **[←]**—Displays the title of a program. The TI-74 ignores this key if a title is already in the display.

Running a Program from the Directory

When the Directory program displays a program title or filename, you can execute that program by pressing the **[RUN]** key. This procedure exits the Directory program and starts the selected program.

Running the Directory Program

To execute the Directory program, type

RUN "DIR" (or **RUN "CONTENTS"**)

and press **[ENTER]**. To exit the Directory program, press **[↓]** when the last title or filename is displayed or press **[↑]** when the first title or filename is displayed. Then press **[Y]** when the prompt `Exit Program?` appears in the display.

Step	Display	Procedure/Comment
1.	`MATHEMATICS LIBRARY`	Wait while cartridge name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to print the directory, press **[Y]**. b. If you do not want to print the directory, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**. When directory is printed, program stops.
4.	*program title*	Displays title. a. To display filename of same program, press **[→]**. Go to next step. b. To display title of previous program, press **[↑]**. Return to step 4. c. To display title of next program, press **[↓]**. Return to step 4. d. To execute the program, press **[RUN]**.
5.	*filename*	Displays filename. a. To display title of same program, press **[←]**. Return to step 4. b. To display title of previous program, press **[↑]**. Return to step 4. c. To display title of next program, press **[↓]**. Return to step 4. d. To execute the program, press **[RUN]**.

Setting the Language of the Prompts

The instructions in this manual show the titles, names, and prompts in English. However, the Mathematics cartridge contains a subprogram that lets you also select German or French. You can use this subprogram only when a cartridge program is not running.

The Languages

Each of the three languages available in the cartridge is identified by a number in the SETLANG subprogram.

Language Number	Language
0	English (Default)
1	German
2	French

If you run a cartridge program and the prompts are not in the language you prefer, press **[BREAK] [CLR]**. Then use the SETLANG subprogram to change the language.

Setting the Language

The CALL SETLANG(*language-number*) command specifies the language in which the prompts and messages in the cartridge programs are displayed. For example, the following command sets the language to French.

CALL SETLANG(2)

Note: All system messages are displayed in English, regardless of the language setting.

The Constant Memory feature of the TI-74 retains the language setting when you turn the calculator off and on. The language setting remains in effect until you change it again or until the TI-74 is initialized.

Yes/No Response

Changing the language setting from English to German or French also changes the key that indicates a "yes" response. For German, the "yes" response is the **[J]** key. For French, the "yes" response is the **[O]** key. The "no" response is always the **[N]** key, regardless of the language setting.

Chapter 2: Numeric Calculation Programs

This chapter describes four programs that perform numeric calculations.

Table of Contents

The Complex Functions (COMPF) Program

The Complex Functions program allows you to perform 20 mathematical operations and functions on complex numbers.

Menu Options

The Complex Functions program displays the following options in the menu of the program.

Option	Action/Function	Option	Action/Function
0	Return to menu	11	ATN(X)
1	X^2	12	Polar to Rectangular
2	SQR(X)	13	Rectangular to Polar
3	1/X	14	X + Y
4	LN(X)	15	X − Y
5	EXP(X)	16	X × Y
6	SIN(X)	17	X ÷ Y
7	COS(X)	18	Y^X
8	TAN(X)	19	$\sqrt[X]{Y}$
9	ASIN(X)	20	$LOG_Y X$ gelöscht !
10	ACOS(X)	21	Exit the program

The program displays three menu options at a time, pausing for three seconds and then continuing with the next three options until all have been displayed. The program then prompts you to enter the option of your choice.

To perform these operations, both X and Y must be in rectangular form. If necessary, you can use option 12 to convert polar data to rectangular form. You can then use option 13 to convert results back to polar coordinates. These two options also allow you to designate whether the values are in degrees, radians, or grads.

Menu Options (Continued)

Options 1 through 11 are calculated for the complex number X. The result becomes the new X.

Options 14 through 20 are calculated for the complex numbers X and Y. The answer to any multi-value function is expressed only as the principal value of the function; that is, the result becomes the new X.

Chained Operations

Operations may be chained in the following manner.

1. Select the desired function.
2. Enter the real and imaginary parts of X.
3. Enter the real and imaginary parts of Y. After the function has been performed, the result becomes the new X.
4. Select the function again, press **[ENTER]** to accept the calculated X value, and enter a new Y value.

Method Used in the Program

For rectangular input, the following derivations are used in the program.

$$X = a + bi$$

$$Y = c + di$$

$$X + Y = (a + c) + (b + d)i$$

$$X - Y = (a - c) + (b - d)i$$

$$X \times Y = (ac - bd) + (ad + bc)i$$

$$X \div Y = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i$$

For polar data, X has magnitude R and an angle of θ (theta).

Using the Complex Functions Program

To execute the Complex Functions program, type RUN "COMPF" and press [ENTER].

Step	Display	Procedure/Comment
1.	`COMPLEX FUNCTIONS`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**.

4.	`0-Menu 1-Sqr 2-SqrRt...`
	`3-Recip 4-Ln 5-Exp...`
	`6-Sin 7-Cos 8-Tan...`
	`9-ArcSin 10-ArcCos 11-ArcTan...`
	`12-PtoR 13-RtoP 14-Add...`
	`15-Sub 16-Mult 17-Div...`
	`18-YtoX 19-XrtY 20-LOGyX...`
	`21-EXIT.`
		Displays the menu, pausing for 3 seconds while each line is displayed.

5.	`Enter Option:`	Type the number of the option you want to perform and press **[ENTER]**. a. If you select option 0, go to step 4. b. If you select option 1 through 11 or 14 through 20, go to step 6. c. If you select option 12 or 13, go to step 10. d. If you select option 21, you exit the program.
6.	`Enter Real(X):`	Type the real part of X and press **[ENTER]**.
7.	`Enter Imag(X):`	Type the imaginary part of X and press **[ENTER]**. a. If using option 1 through 11, go to step 13. b. If using option 13, go to step 15. c. If using option 14 through 20, go to step 8.

Step	Display	Procedure/Comment
8.	`Enter Real(Y):`	Type the real part of Y and press **[ENTER]**.
9.	`Enter Imag(Y):`	Type the imaginary part of Y and press **[ENTER]**. Go to step 13.
10.	`Enter 1-Deg 2-Rad 3-Grad:`	Type the number of the desired angle units (options 12 and 13 only) and press **[ENTER]**. a. If using option 12, go to step 11. b. If using option 13, return to step 6.
11.	`Enter R:`	Type the magnitude of X and press **[ENTER]**.
12.	`Enter Theta:`	Type the angle of X and press **[ENTER]**.
13.	`Real(Z)=`	Displays the real part of the answer. Press **[ENTER]** to continue.
14.	`Imag(Z)=`	Displays the imaginary part of the answer. a. If using option 1 through 12 or 14 through 20, press **[ENTER]** to return to step 5. b. If using option 13, press **[ENTER]** to proceed to the next step.
15.	`R=`	Displays the magnitude of the answer (option 13 only). Press **[ENTER]** to proceed.
16.	`Theta=`	Displays the argument of the answer (option 13 only). Press **[ENTER]** to return to step 5.

Application: Performing a Chained Calculation

The following example adds 3 + −4i to −5 + 6i and then multiplies the result by 1.9 + i. To execute the Complex Functions program, type RUN "COMPF" and press [ENTER].

Step	Display	Procedure/Comment
1.	`COMPLEX FUNCTIONS`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	Press **[N]**.

3.	`0-Menu 1-Sqr 2-SqrRt`
	`3-Recip 4-Ln 5-Exp`
	`6-Sin 7-Cos 8-Tan`
	`9-ArcSin 10-ArcCos 11-ArcTan`
	`12-PtoR 13-RtoP 14-Add`
	`15-Sub 16-Mult 17-Div`
	`18-YtoX 19-XrtY 20-LOGyX`
	`21-EXIT`
		Displays the menu.

4.	`Enter Option: 0`	Type **14** and press **[ENTER]**.
5.	`Enter Real(X): 0`	Type **3** and press **[ENTER]**.
6.	`Enter Imag(X): 0`	Type **−4** and press **[ENTER]**.
7.	`Enter Real(Y): 0`	Type **−5** and press **[ENTER]**.
8.	`Enter Imag(Y): 0`	Type **6** and press **[ENTER]**.
9.	`Real(Z)=-2`	Displays real part of answer. Press **[ENTER]** to proceed.
10.	`Imag(Z)= 2`	Displays imaginary part of answer. Press **[ENTER]** to proceed.

Step	Display	Procedure/Comment
11.	`Enter Option: 14`	Type **16** and press **[ENTER]**.
12.	`Enter Real(X): -2`	Press **[ENTER]** to accept the current value.
13.	`Enter Imag(X): 2`	Press **[ENTER]** to accept the current value.
14.	`Enter Real(Y): -5`	Type **1.9** and press **[ENTER]**.
15.	`Enter Imag(Y): 6`	Type **1** and press **[ENTER]**.
16.	`Real(Z)=-5.8`	Displays real part of answer. Press **[ENTER]** to proceed.
17.	`Imag(Z)= 1.8`	Displays imaginary part of answer. Press **[ENTER]** to proceed.
18.	`Enter Option: 16`	Type **21** and press **[ENTER]** to exit the program.

The Gamma Function (GAM) Program

The Gamma Function program computes the value of the gamma function for positive integers and positive and negative non-integers. The program also computes ln(gamma (x)).

Numeric Ranges

Although the program computes the value of the gamma function for positive integers and positive and negative non-integers, the output is limited to a display of $\pm 9.999999 \times 10^{127}$ for positive exponents and $\pm 1 \times 10^{-128}$ for negative exponents. Arguments are restricted to the range $-84.9 < x \leqslant 85.767$. Gamma(0) is not defined.

In computing ln(gamma (x)), the program is restricted to the range $-9.999999 \times 10^{11} < x \leqslant 7.59 \times 10^{13}$.

Note: If the argument is out of the gamma range but in the ln(gamma) range, the ln(gamma) value is computed, and the gamma computation is omitted.

The gamma function is defined by the improper integral

$$(p) = \Gamma(x) = \int_0^\infty t^{x-1}\, e^{-t}\, dt$$

which converges for all $p > 0$. The gamma function is $(N-1)!$ for all numbers greater than zero or non-integers less than zero.

Method Used in the Program

Gamma(X), ln(gamma(X)), X!, and ln(X!) are all calculated from the asymptotic series

$$\ln(\Gamma(X)) = \ln(X-1)! \approx (X-1/2)\ln X - X + 1/2\ln(2\pi) + \sum_{m=1}^{\infty} \frac{B_{2m}}{2m\,(2m-1)\,X^{2m-1}}$$

where B_{2m} is the 2mth Bernoulli number. The error caused by truncation is

$$|\varepsilon_{\ln}| \leqslant \frac{691}{360360\,X^{11}} \text{ for } \ln\Gamma(X) \text{ and } \ln X!$$

and

$$|\varepsilon_{\Gamma(X)}| \leqslant e^{\ln\Gamma(X)}\,(e^{|\varepsilon_{\ln}|}-1) \text{ for } \Gamma(X) \text{ and } X!$$

where e is the base of natural logarithms and ln(X) is the calculated value of the series. For those values of X, such that $-5 < X < 6$, the identity

$$\Gamma(X+1) = X\,\Gamma(X)$$

is used to compute gamma(X) from gamma(X+n) where $X+n > 6$ is used to enhance the accuracy of the calculation.

For values of X such that $X < -5$, the identity

$$\Gamma(X)\Gamma(1-X) = \frac{\pi}{\sin X\pi}$$

is used to enhance the accuracy of gamma.

For example, $(.5)! = \sqrt{\pi} \times 0.5 = .8862269255$. Alternatively,

$$(.5)! = \text{gamma}(1.5) = \frac{4.5!}{(4.5)(3.5)(2.5)(1.5)} = .8862269255$$

Note that gamma(X) is not defined when X is a negative integer. Also, gamma(0) is not defined.

Using the Gamma Function Program

To execute the Gamma Function program, type RUN "GAM" and press [ENTER].

Step	Display	Procedure/Comment
1.	`GAMMA FUNCTION`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**.
4.	`Enter X:`	Type the X value and press **[ENTER]**.
5.	`Ln(Gamma(#))=`	Displays the natural log of gamma for the given X. Press **[ENTER]** to continue.
6.	`Gamma(#)=`	Displays gamma for the given X. Press **[ENTER]** to continue.
7.	`Exit Program?`	a. To exit the program, press **[Y]**. b. To continue with the program, press **[N]**. Go to step 4.

Application: Calculating a Gamma Function

The following example calculates gamma (X) for x = 6. To execute the Gamma Function program, type RUN "GAM" and press [ENTER].

Step	Display	Procedure/Comment
1.	GAMMA FUNCTION	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	Press **[N]**.
3.	Enter X: 0	Type **6** and press **[ENTER]**.
4.	Ln(Gamma(6))= 4.787491743	Displays the natural log of gamma(6). Press **[ENTER]** to continue.
5.	Gamma(6)= 120.	Displays the value of gamma(6). Press **[ENTER]** to continue.
6.	Exit Program?	Press **[Y]**.

The Polynomial Multiplication (POL) Program

The Polynomial Multiplication program performs the multiplication of two polynomials.

Terminology of the Program

This program refers to the polynomials being multiplied as A and B, and refers to the resulting polynomial as C.

$$P_A \times P_B = P_C$$

The polynomials are of the form

$$A_0 + A_1X + A_2X^2 + \ldots + A_nX^n$$
$$B_0 + B_1X + B_2X^2 + \ldots + B_nX^n$$
$$C_0 + C_1X + C_2X^2 + \ldots + C_nX^n$$

where the coefficients are real numbers.

With 8K of memory, the TI-74 can handle two polynomials such that the order of polynomial A plus two times the order of the polynomial B does not exceed 400.

$$\text{order}\ (P_A) + (2)\ \text{order}\ (P_B) \leqslant 400.$$

Using the Polynomial Multiplication Program

To execute the Polynomial Multiplication program, type RUN "POL" and press [ENTER].

Step	Display	Procedure/Comment
1.	POLYNOMIAL MULTIPLICATION	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	Enter Device Code:	Type the output device code and press **[ENTER]**.
4.	Enter Order Of A:	Press **[ENTER]** to accept current value, or type the order of polynomial A and press **[ENTER]**.
5.	Enter Order Of B:	Press **[ENTER]** to accept current value, or type the order of polynomial B and press **[ENTER]**.
6.	Enter A(#):	Press **[ENTER]** to accept current value, or type appropriate coefficient of polynomial A and press **[ENTER]**. Repeat this step until all coefficients are entered.
7.	Edit?	a. To edit polynomial A, press **[Y]**. b. To accept data, press **[N]**. Go to step 12.

Step	Display	Procedure/Comment
8.	`Edit All Input?`	a. To edit entire polynomial A, press **[Y]**. Go to step 6. b. To edit individual elements, press **[N]**.
9.	`Enter Element To Be Edited:`	Type the number of the element to be edited and press **[ENTER]**.
10.	`Enter A(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**.
11.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 9. b. To accept data, press **[N]**.
12.	`Enter B(#):`	Press **[ENTER]** to accept current value, or type the correct coefficient of polynomial B and press **[ENTER]**. Repeat this step until all coefficients are entered.
13.	`Edit?`	a. To edit polynomial B, press **[Y]**. b. To accept data, press **[N]**. Go to step 18.
14.	`Edit All Input?`	a. To edit entire polynomial B, press **[Y]**. Go to step 12. b. To edit individual elements, press **[N]**.
15.	`Enter Element To Be Edited:`	Type the number of the element to be edited and press **[ENTER]**.

Step	Display	Procedure/Comment
16.	`Enter B(#):`	Press **[ENTER]** to accept current value, or type correct coefficient of polynomial B and press **[ENTER]**.
17.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 15. b. To accept data, press **[N]**.
18.	`C(#)=`	Displays each coefficient of polynomial C. Press **[ENTER]** to view the next coefficient until all coefficients have been displayed.
19.	`Exit Program?`	a. To exit the program, press **[Y]**. b. To continue with the program, press **[N]**. Go to step 4.

Application: Multiplying Two Polynomials

Multiply the polynomials ($9X^5 - 3X^3 + 13X^2 - 2X - 3$) and ($4X^4 + 3X^3 - 2X^2 + X - 4$). To execute the Polynomial Multiplication program, type RUN "POL" and press [ENTER].

Step	Display	Procedure/Comment
1.	POLYNOMIAL MULTIPLICATION	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	Press **[N]**.
3.	Enter Order Of A: 0	Type **5** and press **[ENTER]**.
4.	Enter Order Of B: 0	Type **4** and press **[ENTER]**.
5.	Enter A(0): 0	Type **−3** and press **[ENTER]**.
6.	Enter A(1): 0	Type **−2** and press **[ENTER]**.
7.	Enter A(2): 0	Type **13** and press **[ENTER]**.
8.	Enter A(3): 0	Type **−3** and press **[ENTER]**.
9.	Enter A(4): 0	Press **[ENTER]** to accept the current value.
10.	Enter A(5): 0	Type **9** and press **[ENTER]**.
11.	Edit?	Press **[N]**.
12.	Enter B(0): 0	Type **−4** and press **[ENTER]**.
13.	Enter B(1): 0	Type **1** and press **[ENTER]**.
14.	Enter B(2): 0	Type **−2** and press **[ENTER]**.

Step	Display	Procedure/Comment
15.	`Enter B(3): 0`	Type **3** and press **[ENTER]**.
16.	`Enter B(4): 0`	Type **4** and press **[ENTER]**.
17.	`Edit?`	Press **[N]**.

18.	`C(0)= 12`
	`C(1)= 5`
	`C(2)= -48`
	`C(3)= 20`
	`C(4)= -47`
	`C(5)= 1`
	`C(6)= 52`
	`C(7)= -30`
	`C(8)= 27`
	`C(9)= 36`
		Displays each coefficient of resulting polynomial. Press **[ENTER]** after each coefficient to continue.

19.	`Exit Program?`	Press **[Y]**.

The Prime Factors (PRI) Program

The Prime Factors program factors an integer into prime numbers. The search for a large prime factor may require several minutes.

Input Range

This program can find the prime factors of:

- Any positive integer (within the range of the machine) that has twelve or fewer mantissa digits.
- Any integer from 0 to 2,000,000,000,000. A 13-digit integer greater than this is rounded, which makes factoring imprecise.

To factor a negative integer, enter the number's absolute value and remember to include -1 when listing the factors.

Using the Prime Factors Program

To execute the Prime Factors program, type RUN "PRI" and press [ENTER].

Step	Display	Procedure/Comment
1.	`PRIME FACTORS`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type the output device code and press **[ENTER]**.
4.	`Enter # To Be Factored:`	Type the number to be factored and press **[ENTER]**.
5.	`F#=`	Displays each factor. Press **[ENTER]** to view the next factor until all factors have been displayed.
6.	`Exit Program?`	a. To exit the program, press **[Y]**. b. To continue with the program, press **[N]**. Go to step 4.

Application: Factoring a Number

Find the prime factors of 22278. To execute the Prime Factors program, type RUN "PRI" and press [ENTER].

Step	Display	Procedure/Comment
1.	`PRIME FACTORS`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	Press **[N]**.
3.	`Enter # To Be Factored: 0`	Type **22278**.

4.	`F1=2`
	`F2=3`
	`F3=47`
	`F4=79`
		Displays each factor. Press **[ENTER]** after each factor to continue.

5.	`Exit Program?`	Press **[Y]**.

Chapter 3: Curve Analysis Programs

This chapter describes four programs that can help you determine an unknown point on a curve.

Table of Contents

The Cubic Splines (SPL) Program

The Cubic Splines program fits a sequence of cubic polynomials to m input data points (a maximum of 100 data points with the standard 8K-byte memory capacity) using cubic spline interpolation. This method consists of a piecewise polynomial approximation using cubic polynomials between successive pairs of known points.

Method Used in the Program

To construct g(x) it is important to note that $g''(x)$ is continuous over the entire region $x_0 \leqslant x \leqslant x_m$. Due to this continuity, the cubic is generally superior to polynomial interpolation because regions of inflection are easily fitted.

In general, the derivatives at the endpoints of this region $f''(x_i)$ and $f''(x_m)$ are taken to be zero.

$f''(x_i) = f''(x_m) = 0$

A more accurate fit is obtained if the second derivatives at the end points are known.

The second derivative varies linearly over each interval. At any point x, the second derivative is

$$g''(x) = g''(x_i) + \frac{x - x_i(g''(x_{i+1}) - g''(x_i))}{x_{i+1} - x_i}$$

integrating twice and applying the conditions

$g(x_i) = f(x_i)$ and $g(x_{i+1}) = f(x_{i+1})$

Thus, when $x_i \leqslant x \leqslant x_{i+1}$

$$\begin{aligned} g(x) = f_i(x) = {} & \frac{g''(x_i)}{6}\left(\frac{(x_{i+1}) - x)^3}{\Delta x_i} - \Delta x_i(x_{i+1} - x)\right) \\ & + \frac{g''(x_{i+1})}{6}\left(\frac{(x - x_i)^3}{\Delta x_i} - \Delta x_i(x - x_i)\right) \\ & + f(x_i)\left(\frac{x_{i+1} - x}{\Delta x_i}\right) \\ & + f(x_{i+1})\left(\frac{x - x_i}{\Delta x_i}\right) \end{aligned}$$

where $\Delta x_i = x_{i+1} - x_i$.

Reference

Numerical Methods, Robert W. Hornbeck, Quantam Publishers. Inc., 1975, pp. 47–50.

Using the Cubic Splines Program

To execute the Cubic Splines program, type RUN "SPL" and press [ENTER].

Step	Display	Procedure/Comment
1.	`CUBIC SPLINES`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**.
4.	`Enter # Of Data Points:`	Press **[ENTER]** to accept current value, or type the number of data points and press **[ENTER]**.
5.	`Enter X(#):`	Press **[ENTER]** to accept current value, or type the correct X value and press **[ENTER]**.
6.	`Enter Y(#):`	Press **[ENTER]** to accept current value, or type the correct Y value and press **[ENTER]**. Repeat steps 5 and 6 until all values are entered.
7.	`Edit?`	a. If you want to edit the input values, press **[Y]**. Go to step 8. b. If you do not want to edit, press **[N]**. Go to step 13.
8.	`Edit All Input?`	a. If you want to edit all input, press **[Y]**. Go to step 5. b. If you want to edit single elments, press **[N]**.

Step	Display	Procedure/Comment
9.	`Enter Element To Be Edited:`	Press **[ENTER]** to accept current value, or type the number of the element to be edited and press **[ENTER]**.
10.	`Enter X(#):`	Press **[ENTER]** to accept current value, or type the correct X value and press **[ENTER]**.
11.	`Enter Y(#):`	Press **[ENTER]** to accept displayed value, or type the correct Y value and press **[ENTER]**.
12.	`Edit Other Elements?`	a. If you want to edit other elements, press **[Y]**. Go to step 9. b. If you do not want to edit other elements, press **[N]**.
13.	`Enter Y"0:`	Press **[ENTER]** to accept current value, or type the second derivative at X_0 and press **[ENTER]**.
14.	`Enter Y"#:`	Press **[ENTER]** to accept current value, or type the second derivative at Xf and press **[ENTER]**.
15.	`Edit?`	a. If you want to edit the input values, press **[Y]**. Go to step 13. b. If you do not want to edit, press **[N]**.
16.	`Print Second Derivatives?`	a. To display second derivatives, press **[Y]**. b. To continue the program, press **[N]**. Go to step 19.

Step	Display	Procedure/Comment
17.	Y"#=	Displays second derivatives. Press **[ENTER]** until all the values have been displayed.
18.	Interpolate?	a. To interpolate, press **[Y]**. b. To continue with the program, press **[N]**. Go to step 22.
19.	Enter X:	Type the X value to be used for interpolation and press **[ENTER]**.
20.	F#=	Displays result. Press **[ENTER]** to continue.
21.	Repeat For Different X?	a. To interpolate for different X, press **[Y]**. Go to step 19. b. To continue with the program, press **[N]**.
22.	Exit Program?	a. To exit the program, press **[Y]**. b. To continue the program, press **[N]**. Go to step 4.

Application: Interpolating a Normal Distribution

This example interpolates the area under the standard unit normal distribution for X = .8, using five tabulated values.

x =	**.6**	**.7**	**.9**	**1.0**	**1.1**
f(x) =	**.7257**	**.758**	**.8159**	**.8413**	**.8643**

To execute the program, type RUN "SPL" and press [ENTER].

Step	Display	Procedure/Comment
1.	`CUBIC SPLINES`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	Press **[N]**.
3.	`Enter # Of Data Points:`	Type **5** and press **[ENTER]**.
4.	`Enter X(0): 0`	Type **.6** and press **[ENTER]**.
5.	`Enter Y(0): 0`	Type **.7257** and press **[ENTER]**.
6.	`Enter X(1): 0`	Type **.7** and press **[ENTER]**.
7.	`Enter Y(1): 0`	Type **.758** and press **[ENTER]**.
8.	`Enter X(2): 0`	Type **.9** and press **[ENTER]**.
9.	`Enter Y(2): 0`	Type **.8159** and press **[ENTER]**.
10.	`Enter X(3): 0`	Type **1** and press **[ENTER]**.
11.	`Enter Y(3): 0`	Type **.8413** and press **[ENTER]**.
12.	`Enter X(4): 0`	Type **1.1** and press **[ENTER]**.
13.	`Enter Y(4): 0`	Type **.8643** and press **[ENTER]**.
14.	`Edit?`	Press **[N]**.

Step	Display	Procedure/Comment
15.	`Enter Y″0: 0`	Press **[ENTER]** to accept the current value.
16.	`Enter Y″4: 0`	Press **[ENTER]** to accept the current value.
17.	`Edit?`	Press **[N]**.
18.	`Print Second Derivatives?`	Press **[Y]**.

19.	`Y″0=0`
	`Y″1=-.2628688525`
	`Y″2=-.2163934426`
	`Y″3=-.3059016393`
	`Y″4=0`
		Displays the values, one at a time. Press **[ENTER]** after each output to continue.

20.	`Interpolate?`	Press **[Y]**.
21.	`Enter X: 0`	Type **.8** and press **[ENTER]**.
22.	`F(.8)=.7881481557`	Displays result. Press **[ENTER]** to continue.
23.	`Repeat For Different X?`	Press **[N]**.
24.	`Exit Program?`	Press **[Y]**.

The Relative Minimums (MIN) Program

The Relative Minimums program finds a value at which a function is minimum within an interval. The minimum of the interval is not necessarily the absolute minimum.

Method Used in the Program

This program utilizes the "golden section search." The interval is sectioned using the golden ratio, and only one functional evaluation is performed per iteration. The program is completed when the difference between the functional values of a section becomes less than a specified limiting value.

You decide the interval over which the program searches. Since this program assures only a relative minimum, it is preferable to use a function with a single minimum or to choose an interval intelligently so that your minimum will be absolute over the chosen region.

Entering a Subprogram

Given a function, f(x), this program finds a value of x for which f(x) has a relative minimum. Before running this program, you must enter the function f(x) as a subprogram.

The function must be entered in the following form. (The function shown in this example is Y=M*X+B.)

```
NNN SUB FX(X,Y)
NNN Y=M*X+B
NNN SUBEND
```

where

NNN = a line number larger than any line number in any main program that you have entered.
X = independent input variable.
Y = dependent output variable.

Method Used in the Program (Continued)

The golden section method is a numerical approximation to Fibonacci search.

$$F_k = \frac{1}{\sqrt{5}}\tau^{k+1} = 0.4472\,(1.6180)^{k+1}$$

$$\tau = 1.6180$$

gives a constant ratio of the approximate Fibonacci numbers

$$\frac{F_k}{F_{k+2}} = \frac{1}{\tau^2} \approx 0.382$$

$$\frac{F_k}{F_{k+1}} = \frac{1}{\tau} \approx 0.618$$

This method reduces the initial interval $[a^1, b^1]$ by a factor

$$F_k\tau^{1-k} \approx 1.17 \qquad \tau = 1.6180$$

After k function evaluations, the interval becomes

$$\tau^{1-k}$$

In the first iteration two functional values are computed; however, only one value need be determined in subsequent iterations.

Reference

Iterative Methods for Nonlinear Optimization Problems, Jacoby, Kowalik, and Pizzo, Prentice-Hall, 1972.

Using the Relative Minimums Program

To execute the Relative Minimums program, type RUN "MIN" and press [ENTER].

Step	Display	Procedure/Comment
1.	`RELATIVE MINIMUMS`	Wait while the program name is displayed or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**.
4.	`Have You Entered Sub FX?`	a. If you have entered Sub FX, press **[Y]**. Go to step 6. b. If you have not entered Sub FX, press **[N]**.
5.	`Enter Sub FX...` `Refer To Manual For Aid.`	a. To repeat prompt, press **[SHIFT] [PB]**. b. To exit the program, press **[ENTER]**.
6.	`Enter A&B So That...` `A&B Bracket The Minimum.`	a. To repeat prompt, press **[SHIFT] [PB]**. b. To continue the program, press **[ENTER]**.
7.	`Enter A:`	Press **[ENTER]** to accept current value, or type the lower bound and press **[ENTER]**.
8.	`Enter B:`	Press **[ENTER]** to accept current value, or type the upper bound and press **[ENTER]**.
9.	`Enter Epsilon:`	Press **[ENTER]** to accept current value, or type value for epsilon and press **[ENTER]**.
10.	`Xmin=`	Displays result. Press **[ENTER]** to continue.
11.	`Exit Program?`	a. To exit the program, press **[Y]**. b. To continue the program, press **[N]**. Go to step 7.

Application: Finding a Function's Minimum

Find the relative minimum of $X^3 - X^2 - X + 2$ over the interval [− 5, 5], with an epsilon of .001.

Entering the Subprogram

Enter the function by typing the following lines into program memory, adjusting the line numbers as needed.

```
1 SUB FX(X,Y)
2 Y=X^3-X^2-X+2
3 SUBEND
```

Then execute the Relative Minimums program. Type **RUN "MIN"** and press **[ENTER]**.

Step	Display	Procedure/Comment
1.	RELATIVE MINIMUMS	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	Press **[N]**.
3.	Have You Entered Sub FX?	Press **[Y]**.
4.	Enter A&B So That... A&B Bracket The Minimum.	Press **[ENTER]** to continue.
5.	Enter A: 0	Type **−5** and press **[ENTER]**.
6.	Enter B: 0	Type **5** and press **[ENTER]**.
7.	Enter Epsilon: 0	Type **.001** and press **[ENTER]**.
8.	Xmin=-4.999747247	Displays result. Press **[ENTER]** to continue.
9.	Exit Program?	Press **[Y]**.

The Root Finder—Bisection (BIS) Program

The Root Finder-Bisection (BIS) program uses the bisection method to find the roots of a function. For an alternate method, see the Root Finder-Newton program.

Method Used in the Program

This program uses the bisection method. For each root which is computed, you must give two points bounding the root. The function must evaluate to two values of opposite sign at the boundaries of the interval if one root is present in the interval.

Entering a Subprogram

A function that is to be used by the bisection program must be entered as a subprogram in the following form before you start the program. (The function shown in this example is Y = M*X + B.)

```
NNN SUB FX(X,Y)
NNN Y=M*X+B
NNN SUBEND
```

where

NNN = a line number larger than any line number in any main program that you have entered.
X = independent input variable.
Y = dependent output variable.

Using the Root Finder—Bisection Program

To execute the Root Finder-Bisection program, type RUN "BIS" and press [ENTER].

Step	Display	Procedure/Comment
1.	ROOT FINDER-BISECTION	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	Enter Device Code:	Type output device code and press **[ENTER]**.
4.	Have You Entered Sub FX?	a. If you have entered subprogram FX, press **[Y]**. Go to step 6. b. If you have not entered subprogram F1, press **[N]**.
5.	Enter Sub FX... Refer To Manual For Aid.	a. To repeat prompt, press **[SHIFT] [PB]**. b. To exit the program, press **[ENTER]**.
6.	Enter A&B So That... F(A)&F(B) Have Opposite Signs.	a. To repeat prompt, press **[SHIFT] [PB]**. b. To continue the program, press **[ENTER]**.
7.	Enter A:	Press **[ENTER]** to accept current value, or type the lower bound and press **[ENTER]**.
8.	Enter B:	Press **[ENTER]** to accept current value, or type the upper bound and press **[ENTER]**.
9.	Enter Epsilon:	Press **[ENTER]** to accept current value, or type the allowable error and press **[ENTER]**.
10.	Root=	Displays a root of the function. Press **[ENTER]** to continue.
11.	Exit Program?	a. To exit the program, press **[Y]**. b. To continue with the program, press **[N]**. Go to step 7.

Application: Finding a Root through Bisection

Find the root of the function $Y = 3X^2 + 2X - 7$, which lies between 0 and 20.

Entering the Subprogram

Enter the function by typing the following lines into program memory, adjusting line numbers as needed.

```
1 SUB  FX(X,Y)
2 Y=3*X^2+2*X-7
3 SUBEND
```

Then execute the Root Finder-Bisection program. Type **RUN "BIS"** and press **[ENTER]**.

Step	Display	Procedure/Comment
1.	ROOT FINDER-BISECTION	Wait while the program name is displayed.
2.	Use Printer?	Press **[N]**.
3.	Have You Entered Sub FX?	Press **[Y]**.
4.	Enter A&B So That... F(A)&F(B) Have Opposite Signs	Press **[ENTER]** to continue.
5.	Enter A: 0	Press **[ENTER]** to accept the current value.
6.	Enter B: 0	Type **20** and press **[ENTER]**.
7.	Enter Epsilon: 0	Type **.001** for error limit and press **[ENTER]**.
8.	Root= 1.229858398	Displays the root of the function. Press **[ENTER]** to continue.
9.	Exit Program?	Press **[Y]**.

The Root Finder—Newton (NEW) Program

The Root Finder-Newton program uses the Newton-Raphson technique to approximate the roots of a function. For an alternate method, see the Root Finder—Bisection program.

Entering Subprograms

The function is entered as a subprogram in the following manner. (The function shown in this example is Y=M*X^2+B*X.)

```
NNN SUB FX(X,Y)
NNN Y=M*X^2+B*X
NNN SUBEND
```

where

NNN = a line number larger than any line number in any main program you have entered.
X = independent input variable.
Y = dependent output variable.

You must enter the derivative of the function as a subprogram in the following manner.

```
MMM SUB FD(X,Y)
MMM Y=2*M*X+B
MMM SUBEND
```

For each root, you must enter an initial guess for the root. You may find it helpful to draw a rough plot of f(x) in order to make a good approximation.

Note: Although the Newton-Raphson is one of the most widely used root-finding techniques, there are many functions for which the Newton-Raphson method will not converge to a solution. Another method of root finding, the bisection method, is also available in this cartridge and may be used as a complement to this program.

Method Used in the Program

An initial estimate, x_0, of a root of the equation $f(x) = 0$ is made. An improved estimate of the root can then be made as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

In general, the expression

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

may be used to approximate a root of $f(x) = 0$, so that $x_{n+1} - x_n < \varepsilon$.

The following approximations are used by the program to compute $f'(x)$.

$$f'(x) \approx \frac{f[x(1+10^{-9})] - f[x(1-10^{-9})]}{2\times 10^{-9}} \qquad x \neq 0$$

$$f'(x) \approx \frac{f[(10^{-9})] - f[(-10^{-9})]}{2\times 10^{-9}} \qquad x = 0$$

Reference

Handbook of Engineering Fundamentals, Ovid W. Eshbach, John Wiley & Sons, Inc., New York, 1954, pp.2–16.

Using the Root Finder—Newton Program

To execute the Root Finder-Newton program, type RUN "NEW" and press [ENTER].

Step	Display	Procedure/Comment
1.	ROOT FINDER-NEWTON	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	Enter Device Code:	Type output device code and press **[ENTER]**.
4.	Have You Entered Subs FX&FD?	a. If you have entered Subs FX and FD, press **[Y]**. Go to step 6. b. If you have not entered Subs FX and FD, press **[N]**.
5.	Enter Subs FX&FD... Refer To Manual For Aid.	a. To repeat prompt, press **[SHIFT] [PB]**. b. To exit the program, press **[ENTER]**.
6.	Enter Xo:	Press **[ENTER]** to accept current value, or type the initial value for X and press **[ENTER]**.
7.	Enter Epsilon:	Press **[ENTER]** to accept current value, or type the value for epsilon and press **[ENTER]**.
8.	Root=	Displays result. Press **[ENTER]** to continue.
9.	Exit Program?	a. To exit the program, press **[Y]**. b. To continue the program, press **[N]**. Go to step 6.

Application: Finding a Root by Newton's Method

Find the roots of the function:

$f(x) = X^3 - 4X^2 - LN(X) + 4$

Given: $f'(x) = 3X^2 - 8X - 1/X$ **and epsilon = .00001**

Entering the Subprograms

Enter the function and its derivative by typing the following lines into program memory, adjusting line numbers as needed.

```
1 SUB  FX(X,Y)
2 Y = X^3 - 4*X^2 - LN(X) + 4
3 SUBEND
4 SUB  FD(X,Y)
5 Y = 3*X^2 - 8*X - 1/X
6 SUBEND
```

To execute the Root Finder-Newton program, type **RUN "NEW"** and press **[ENTER]**.

Step	Display	Procedure/Comment
1.	`ROOT FINDER-NEWTON`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	Press **[N]**.
3.	`Have You Entered Subs FX&FD?`	Press **[Y]**.
4.	`Enter Xo:  0`	Type **5** for initial guess and press **[ENTER]**.
5.	`Enter Epsilon:  0`	Type **.00001** and press **[ENTER]**.
6.	`Root= 3.817439938`	Displays result. Press **[ENTER]** to continue.
7.	`Exit Program?`	Press **[Y]**.

Chapter 4: Calculus Programs

This chapter describes three programs that perform calculus operations.

Table of Contents

The Convolution (CON) Program

Given the impulse response for a linear system, the Convolution program uses the convolution integral to find the output for a specified input waveform. The program uses the trapezoidal rule to generate outputs at intervals of Δt.

Entering Subprograms

Before running the Convolution program, you must enter two subprograms to evaluate the functions x(t) and h(t). The subprograms must be entered in the following form. (The function in this example is Y = MX + B.)

```
NNN SUB FX(X,Y)
NNN Y = MX + B
NNN SUBEND
NNN SUB FH(X,Y)
NNN Y = MX + B
NNN SUBEND
```

NNN = a line number larger than any line number in any main program that you have entered.
X = an independent input variable.
Y = a dependent output variable.

Method Used in the Program

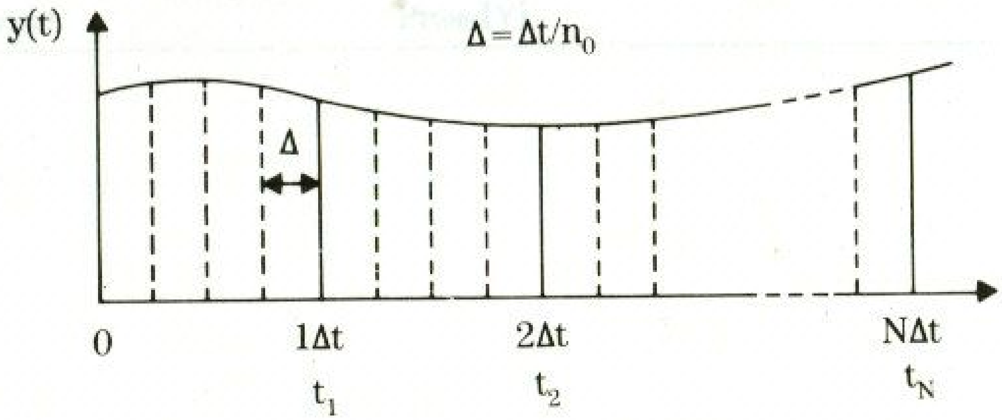
To calculate the output y(t), the input x(t) is convoluted with the impulse response h(t) in the form

$$y(t) = \int_0^t x(\lambda)h(t-\lambda)\,d\lambda$$

Applying the trapezoidal rule to this integral yields

$$y(t) \approx \frac{\Delta}{2}\left(x(0)h(t) + 2 \sum_{k=1}^{n-1} x(k\Delta)h(t-k\Delta) + x(t)h(0) \right)$$

The program prompts you to enter the number of panels per Δt, n_0, the number of output values (N) for which the integral is to be evaluated, and the step size for the integration. Thus, $\Delta = \Delta t/n_0$. Note that Δ is held constant for each successive evaluation of y(t).



Using the Convolution Program

To execute the Convolution program, type RUN "CON" and press [ENTER].

Step	Display	Procedure/Comment
1.	`CONVOLUTION`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**.
4.	`Have You Entered Subs FX&FH?`	a. If you have entered subprograms FX and FH, press **[Y]**. Go to step 6. b. If you have not entered subprograms FX and FH, press **[N]**.
5.	`Enter Subs FX&FH...` `Refer To Manual For Aid.`	a. To repeat prompt, press **[SHIFT] [PB]**. b. To exit the program, press **[ENTER]**.
6.	`Enter # of Panels/Step:`	Press **[ENTER]** to accept current value, or type the number of panels in each step and press **[ENTER]**.
7.	`Enter # Of Output Values:`	Press **[ENTER]** to accept current value, or type the number of points to be output and press **[ENTER]**.
8.	`Enter Step Size:`	Press **[ENTER]** to accept current value, or type the step size and press **[ENTER]**.
9.	`Edit?`	a. To edit the input data, press **[Y]**. Go back to step 6. b. If you do not want to edit, press **[N]**.
10.	`Y(#)=`	Displays result. Press **[ENTER]** to display the next calculated value until all output values have been displayed.
11.	`Exit Program?`	a. If you want to exit the program, press **[Y]**. b. If you do not want to exit the program, press **[N]**. Go back to step 6.

Application: Finding the Output of a Linear System

Given the following conditions, find y(t) at t = .1, .2, . . . , .7, using four panels/step.

FX = 2t for t ≤ .3
= 0 for t > .3
FH = 10 × e^{-5t}

Entering the Subprograms

Before running the Convolution program, enter the following subprograms into program memory, adjusting the line numbers as needed.

```
1 SUB  FX(T,Y)
2 IF  T>.3  THEN  Y=0  ELSE  Y=2*T
3 SUBEND
4 SUB  FH(T,Y)
5 Y=10*EXP(-5*T)
6 SUBEND
```

After entering the subprograms, type **RUN "CON"** and press **[ENTER]**.

Step	Display	Procedure/Comment
1.	CONVOLUTION	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	Press **[N]**.
3.	Have You Entered Subs FX&FH?	Press **[Y]**.
4.	Enter # Of Panels/Step	Type **4** and press **[ENTER]**.
5.	Enter # Of Output Values	Type **7** and press **[ENTER]**.
6.	Enter Step Size	Type **.1** and press **[ENTER]**.
7.	Edit?	Press **[N]**.
8.	Y(.1)= .0861547694 Y(.2)= .2960028933 Y(.3)= .580874829 Y(.4)= .3978081927 Y(.5)= .2412828656 Y(.6)= .1463454556 Y(.7)= .0887630057	Displays the requested values, one at a time. Press **[ENTER]** after each output to continue.
9.	Exit Program?	Press **[Y]**.

The Differential Equations—Runge-Kutta (DIF) Program

The Differential Equations—Runge Kutta program uses a fifth-order Runge-Kutta method to solve a system of differential equations of the type y′ = f(x,y).

Inputs for the Program

The program finds approximate solutions at particular points $X_0, X_1, \ldots, X_N$ where the difference between successive X-values is a constant, h ($X_{N+1} - X_N = h$).

Before running the program, you must enter BASIC subprograms that define Y′. In the program, you must input the initial X value, the final X value, the h value (the step size), and the initial Y values.

To select a step size, subtract the initial X value from the final X value and divide by an integer number of steps.

By defining $Y'_N = Y_{N+1}$ and inputting initial values for $Y, Y', \ldots, Y_{N-1}$, a system of N first-order differential equations is used to solve an Nth-order differential equation.

For example, a third-order function $Y''' = X + Y$ is defined in such a manner that

If:	Then:
$Y = Y_1$	$Y'_1 = Y_2$
$Y' = Y_2$	$Y'_2 = Y_3$
$Y'' = Y_3$	$Y'_3 = X + Y$

Entering the Subprograms

The subprograms that define Y′ must be entered in the following format. The functions shown are from the example above.

```
NNN SUB F1(X, Y(), F(1))
NNN F(1)=Y(2)
NNN SUBEND
NNN SUB F2(X, Y(), F(2))
NNN F(2)=Y(3)
NNN SUBEND
NNN SUB F3(X, Y(), F(3))
NNN F(3)=X+Y(1)
NNN SUBEND
```

The program can use up to nine subprograms for the system of differential equations.

Method Used in the Program

The following derivations are used in the program.

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \cdot \\ \cdot \\ Y_N \end{pmatrix} \quad F = \begin{pmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ F_N \end{pmatrix} \quad Y' = F(X, Y)$$

$$Y_{N+1} = Y_N + \frac{1}{336}(14K_1 + 35K_4 + 162K_5 + 125K_6)$$

where

$$K_1 = hF(X, Y)$$

$$K_2 = hF(X + h/2, Y + K_1/2)$$

$$K_3 = hF(X + h/2, Y + K_1/4 + K_2/4)$$

$$K_4 = hF(X + h, Y - K_2 + 2K_3)$$

$$K_5 = hF(X + 2h/3, Y + 7K_1/27 + 10K_2/27 + K_4/27)$$

$$K_6 = hF(X + h/5, Y + 28K_1/625 - K_2/5 + 546K_3/625 + 54K_4/625 - 378K_5/625)$$

Reference

Numerical Solution of Ordinary Differential Equations, Lapidus and Seinfeld.

To execute the Differential Equations—Runge-Kutta program, type RUN "DIF" and press [ENTER].

Step	Display	Procedure/Comment
1.	DIFFERENTIAL EQUATIONS	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	Enter Device Code:	Type output device code and press **[ENTER]**.
4.	Have You Entered Subs F1->FN?	a. If you have entered subprograms F1 through FN, press **[Y]**. Go to step 6. b. If you have not entered the subprograms, press **[N]**.
5.	Enter Subs F1->FN... Refer To Manual For Aid.	a. To repeat prompt, press **[SHIFT] [PB]**. b. To exit the program, press **[ENTER]**.
6.	Enter Initial X:	Press **[ENTER]** to accept current value, or type initial X value and press **[ENTER]**.
7.	Enter Final X:	Press **[ENTER]** to accept current value, or type the X value and press **[ENTER]**.
8.	Enter Step Size:	Press **[ENTER]** to accept current value, or type the step size and press **[ENTER]**.
9.	Enter # Of Ftns. In System:	Press **[ENTER]** to accept current value, or type the number of functions in the system and press **[ENTER]**.
10.	Edit?	a. If you want to edit the input data, press **[Y]**. Go to step 6. b. If you do not want to edit, press **[N]**.
11.	Enter Yo(#):	Press **[ENTER]** to accept current value, or type the initial value for Y(#) and press **[ENTER]**. Repeat this step until all values have been entered.

Step	Display	Procedure/Comment
12.	`Edit?`	a. If you want to edit the initial Y values, press **[Y]**. b. If you do not want to edit, press **[N]**. Go to step 18.
13.	`Edit All Input?`	a. If you want to edit all the Y values, press **[Y]**. Go to step 11. b. If you want to edit single elements, press **[N]**.
14.	`Enter Row To Be Edited:`	Type number of the row to be edited and press **[ENTER]**.
15.	`Enter Yo(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**.
16.	`Edit Other Elements?`	a. If you want to edit other elements, press **[Y]**. Go to step 14. b. If you do not want to edit other elements, press **[N]**.
17.	`Print Final Result Only?`	a. To print only final results, press **[Y]**. b. To print intermediate and final results, press **[N]**.
18.	`X=`	Displays X value. Press **[ENTER]** to continue.
19.	`Y#=`	Displays the Y value(s), one at a time, corresponding to the particular X. Press **[ENTER]** until all the Y values have been displayed. a. If more X values remain, go to step 18. b. If output is complete, go to step 20.
20.	`Exit Program?`	a. To exit the program, press **[Y]**. b. To continue the program, press **[N]**. Go to step 6.

Application: Resolving a Differential Equation

This example finds the solution for the equation $Y'' = e^{2x}$ at Xf = 1 with the initial conditions $X_0 = 0$, $Y_0 = 1$, $Y'_0 = 2$, step size = .2.

Entering the Subprograms

Before running the Differential Equations—Runge Kutta program, enter the following subprograms into program memory, adjusting the line numbers as needed.

```
1 SUB F1(X,Y(),R)
2 R = Y(2)
3 SUBEND
4 SUB F2(X,Y(),R)
5 R = EXP(2*X)
6 SUBEND
```

After entering the subprograms, type **RUN "DIF"** and press **[ENTER]** to execute the Differential Equations—Runge-Kutta program.

Step	Display	Procedure/Comment
1.	DIFFERENTIAL EQUATIONS	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	Press **[N]**.
3.	Have You Entered Subs F1->FN?	Press **[Y]**.
4.	Enter Initial X: 0	Press **[ENTER]** to accept the current value.
5.	Enter Final X: 0	Type **1** and press **[ENTER]**.
6.	Enter Step Size: 0	Type **.2** and press **[ENTER]**.
7.	Enter # Of Ftns. In System: 0	Type **2** and press **[ENTER]**.
8.	Edit?	Press **[N]**.

Step	Display	Procedure/Comment
9.	Enter Yo(1): 0	Type 1 and press **[ENTER]**.
10.	Enter Yo(2): 0	Type 2 and press **[ENTER]**.
11.	Edit?	Press **[N]**.
12.	Print Final Result Only?	Press **[Y]**.
13.	X=1	Displays final X. Press **[ENTER]** to continue.
14.	Y1= 4.097267833	Displays final Y1. Press **[ENTER]** to continue.
15.	Y2= 5.194528357	Displays final Y2. Press **[ENTER]** to continue.
16.	Exit Program?	Press **[Y]**.

The Gauss Quadrature (GAU) Program

The Gauss Quadrature program approximates the integral of a specified function over an interval a to b.

Entering a Subprogram

Before running the Gauss Quadrature program, you must enter a subprogram in program memory to evaluate the function f(x).

$$I = \int_{x_0}^{x_n} f(x)\,dx$$

The subprogram must be entered in the following form. (The function in this example is Y = MX + B.)

```
NNN SUB F1(X,Y)
NNN Y=MX+B
NNN SUBEND
```

where

NNN = a line number larger than any line number in any main program you have entered.
X = an independent input variable.
Y = a dependent output variable.

The integral is divided into a number of subintervals. The accuracy of the program is increased as the subintervals are shortened.

Method Used in the Program

This program uses the 7-point Gaussian Integration formula for an arbitrary interval.

$$\int_a^b f(y)\,dy = \frac{b-a}{2} \sum_{i=1}^{7} w_i f(y_i) + R_n$$

$$y_i = \left(\frac{b-a}{2}\right) x_i + \left(\frac{b+a}{2}\right)$$

where

$\pm X_i$	W_i
$X_4 = 0$	$W_4 = 0.41795\ 91836\ 735$
$X_5 = +0.40584\ 51513\ 774$	$W_{3,5} = 0.38183\ 00505\ 051$
$X_3 = -0.40584\ 51513\ 774$	
$X_6 = +0.74153\ 11855\ 394$	$W_{2,6} = 0.27970\ 53914\ 893$
$X_2 = -0.74153\ 11855\ 394$	
$X_7 = +0.94910\ 79123\ 428$	$W_{1,7} = 0.12948\ 49661\ 689$
$X_1 = -0.94910\ 79123\ 428$	

The error R_n is determined by

$$R_n = \frac{(b-a)^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3} f^{2n}(\zeta) \qquad a < \zeta < b$$

Reference

Handbook of Mathematical Functions, Abramowitz and Stegun, National Bureau of Standards, 1972, p. 887.

Using the Gauss Quadrature Program

To execute the Gauss Quadrature program, type RUN "GAU" and press [ENTER].

Step	Display	Procedure/Comment
1.	GAUSS QUADRATURE	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	Enter Device Code:	Type output device code and press **[ENTER]**.
4.	Have You Entered Sub FX?	a. If you have entered Sub FX, press **[Y]**. Go to step 6. b. If you have not entered Sub FX, press **[N]**.
5.	Enter Sub FX... Refer To Manual For Aid.	a. To repeat prompt, press **[SHIFT] [PB]**. b. To exit the program, press **[ENTER]**.
6.	Enter Lower Limit:	Press **[ENTER]** to accept current value, or type the lower limit for integration and press **[ENTER]**.
7.	Enter Upper Limit:	Press **[ENTER]** to accept current value, or type the upper limit for integration and press **[ENTER]**.
8.	Enter # Of Divisions:	Press **[ENTER]** to accept current value, or type the number of partitions for integration and press **[ENTER]**.
9.	Integral=	Displays the result. Press **[ENTER]** to continue.
10.	Exit Program?	a. To exit the program, press **[Y]**. b. To continue, press **[N]**. Go to step 7.

Application: Evaluating an Integral

This example uses 4 subintervals to evaluate the following equation.

$$\int_0^2 \frac{e^{2x}}{(x+2)^2}\,dx$$

Entering the Subprogram

Before running the program, enter the following subprogram in program memory, adjusting the line numbers as needed.

```
1 SUB FX(X,Y)
2 Y=EXP(2*X)/(X+2)^2
3 SUBEND
```

After entering the subprogram, type **RUN "GAU"** and press **[ENTER]** to execute the Gauss Quadrature program.

Step	Display	Procedure/Comment
1.	GAUSS QUADRATURE	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	Press **[N]**.
3.	Have You Entered Sub FX?	Press **[Y]**.
4.	Enter Lower Limit: 0	Type **0** and press **[ENTER]**.
5.	Enter Upper Limit: 0	Type **2** and press **[ENTER]**.
6.	Enter # Of Divisions: 0	Type **4** and press **[ENTER]**.
7.	Integral= 2.263036903	Displays result. Press **[ENTER]** to continue.
8.	Exit Program?	Press **[Y]**.

Chapter 5: Matrix Solution Programs

This chapter describes three programs that perform matrix operations.

Table of Contents

The Complex System (COMPS) Program

The Complex Systems program solves a system of n × n simultaneous equations with complex coefficients. In the program, rows and columns are assigned in a traditional matrix form.

Setting Up the Matrix

The order of the system establishes the number of rows in the matrix. The elements of the equations are input according to their row and column positions.

For complex numbers, the real part of the coefficient is entered as a, and the imaginary part is entered as b. The solution vector is input by rows, with the real part of this number entered as c and the imaginary part entered as d. The resulting matrix is similar to the following example.

Matrix			Solution Vector
(a_{11}, b_{11})	(a_{12}, b_{12})	(a_{13}, b_{13})	(c_1, d_1)
(a_{21}, b_{21})	(a_{22}, b_{22})	(a_{23}, b_{23})	(c_2, d_2)
(a_{31}, b_{31})	(a_{32}, b_{32})	(a_{33}, b_{33})	(c_3, d_3)

Method Used in the Program

The program expands an n×n complex system of equations into a 2n×2n system of real equations. Each complex coefficient is evaluated as

$$(a_{11}, b_{11}) = \begin{array}{cc} a_{11} & -b_{11} \\ b_{11} & a_{11} \end{array}$$

Using this method, the system

$$\begin{array}{cccc} (a_{11}, b_{11}) & (a_{12}, b_{12}) & (a_{13}, b_{13}) & (c_1, d_1) \\ (a_{21}, b_{21}) & (a_{22}, b_{22}) & (a_{23}, b_{23}) & (c_2, d_2) \\ (a_{31}, b_{31}) & (a_{32}, b_{32}) & (a_{33}, b_{33}) & (c_3, d_3) \end{array}$$

is expanded to

$$\begin{array}{cccccc} a_{11} & -b_{11} & a_{12} & -b_{12} & a_{12} & -b_{13} \\ b_{11} & a_{11} & b_{12} & a_{12} & b_{13} & a_{13} \\ a_{21} & -b_{21} & a_{22} & -b_{22} & a_{23} & -b_{23} \\ b_{21} & a_{21} & b_{22} & a_{22} & b_{23} & a_{23} \\ a_{31} & -b_{31} & a_{32} & -b_{32} & a_{33} & -b_{33} \\ b_{31} & a_{31} & b_{32} & a_{32} & b_{33} & a_{33} \end{array} \times \begin{array}{c} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \end{array} = \begin{array}{c} c_1 \\ d_1 \\ c_2 \\ d_2 \\ c_3 \\ d_3 \end{array}$$

When the expansion is complete, the program uses the lower-upper decomposition method to solve the matrix.

To execute the Complex Systems program, type RUN "COMPS" and press [ENTER].

Step	Display	Procedure/Comment
1.	`COMPLEX SYSTEMS`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**.
4.	`Enter Order:`	Press **[ENTER]** to accept current value, or type the order of system and press **[ENTER]**.
5.	`Enter A(#,#):`	Press **[ENTER]** to accept current value, or type the real part of the element and press **[ENTER]**.
6.	`Enter B(#,#):`	Press **[ENTER]** to accept current value, or type the imaginary part of the element and press **[ENTER]**. Repeat steps 5 and 6 until all the elements are entered.
7.	`Edit?`	a. To edit the values you have entered, press **[Y]**. b. To continue program, press **[N]**. Go to step 14.
8.	`Edit All Input?`	a. To edit all the input values, press **[Y]**. Go back to step 5. b. To edit individual elements, press **[N]**.
9.	`Enter Row To Be Edited:`	Type the number of the row to be edited and press **[ENTER]**.
10.	`Enter Column To Be Edited:`	Type the number of the column to be edited and press **[ENTER]**.

Step	Display	Procedure/Comment
11.	Enter A(#,#):	Press **[ENTER]** to accept current value, or type the real part of the element and press **[ENTER]**.
12.	Enter B(#,#):	Press **[ENTER]** to accept current value, or type the imaginary part of the element and press **[ENTER]**.
13.	Edit Other Elements?	a. To edit other elements, press **[Y]**. Go to step 9. b. To continue the program, press **[N]**.
14.	Enter C(#):	Press **[ENTER]** to accept current value, or type the real part of the number (solution vector) and press **[ENTER]**.
15.	Enter D(#):	Press **[ENTER]** to accept current value, or type the imaginary part of the number (solution vector) and press **[ENTER]**. Repeat steps 14 and 15 until all solution vectors are entered.
16.	Edit?	a. To edit the input solution vectors, press **[Y]**. b. To continue the program, press **[N]**. Go to step 22.
17.	Edit All Input?	a. To edit all the input solution vectors, press **[Y]**. Go back to step 14. b. To edit individual elements, press **[N]**.
18.	Enter Row To Be Edited:	Type the number of the row to be edited and press **[ENTER]**.

Step	Display	Procedure/Comment
19.	`Enter C(#):`	Press **[ENTER]** to accept current value, or type the real part of the number (solution vector) and press **[ENTER]**.
20.	`Enter D(#):`	Press **[ENTER]** to accept current value, or type the imaginary part of the number (solution vector) and press **[ENTER]**. Repeat steps 19 and 20 until all solution vectors are entered.
21.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 18. b. To continue the program, press **[N]**.
22.	`X#=`	Displays real part of answer. Press **[ENTER]** to proceed.
23.	`Y#=`	Displays imaginary part of answer. Press **[ENTER]** to proceed. Repeat steps 22 and 23 until all results are displayed.
24.	`Exit Program?`	a. To exit the program, press **[Y]**. b. To continue the program, press **[N]**. Go to step 4.

Application: Solving a 3 × 3 System of Equations

This example solves the following 3 × 3 system of equations.

(1/2, 1/2) (1/3, 1/3) (1/4, 1/4) (−13/12, 59/12)
(1/3, 1/3) (1/4, 1/4) (1/5, 1/5) (−47/60, 73/20)
(1/4, 1/4) (1/5, 1/5) (1/6, 1/6) (−37/60, 35/12)

To execute the Complex Systems program, type RUN "COMPS" and press [ENTER].

Step	Display	Procedure/Comment
1.	COMPLEX SYSTEMS	Wait while the program name is displayed, or press [ENTER] to continue.
2.	Use Printer?	Press [N].
3.	Enter Order: 1	Type 3 and press [ENTER].
4.	Enter A(1,1): 0	Type 1/2 and press [ENTER].
5.	Enter B(1,1): 0	Type 1/2 and press [ENTER].
6.	Enter A(1,2): 0	Type 1/3 and press [ENTER].
7.	Enter B(1,2): 0	Type 1/3 and press [ENTER].
8.	Enter A(1,3): 0	Type 1/4 and press [ENTER].
9.	Enter B(1,3): 0	Type 1/4 and press [ENTER].
10.	Enter A(2,1): 0	Type 1/3 and press [ENTER].
11.	Enter B(2,1): 0	Type 1/3 and press [ENTER].
12.	Enter A(2,2): 0	Type 1/4 and press [ENTER].
13.	Enter B(2,2): 0	Type 1/4 and press [ENTER].
14.	Enter A(2,3): 0	Type 1/5 and press [ENTER].

Step	Display	Procedure/Comment
15.	Enter B(2,3): 0	Type **1/5** and press **[ENTER]**.
16.	Enter A(3,1): 0	Type **1/4** and press **[ENTER]**.
17.	Enter B(3,1): 0	Type **1/4** and press **[ENTER]**.
18.	Enter A(3,2): 0	Type **1/5** and press **[ENTER]**.
19.	Enter B(3,2): 0	Type **1/5** and press **[ENTER]**.
20.	Enter A(3,3): 0	Type **1/6** and press **[ENTER]**.
21.	Enter B(3,3): 0	Type **1/6** and press **[ENTER]**.
22.	Edit?	Press **[N]**.
23.	Enter C(1): 0	Type **−13/12** and press **[ENTER]**.
24.	Enter D(1): 0	Type **59/12** and press **[ENTER]**.
25.	Enter C(2): 0	Type **−47/60** and press **[ENTER]**.
26.	Enter D(2): 0	Type **73/20** and press **[ENTER]**.
27.	Enter C(3): 0	Type **−37/60** and press **[ENTER]**.

Step	Display	Procedure/Comment
28.	Enter D(3): 0	Type 35/12 and press **[ENTER]**.
29.	Edit?	Press **[N]** and press **[ENTER]**.
30.	X1=1. Y1=2. X2=2. Y2=3. X3=3. Y3=4.	Displays all requested values. Press **[ENTER]** after each output to proceed.
31.	Exit Program?	Press **[Y]**.

The Matrices program allows you to perform five different operations using matrices: matrix addition, matrix multiplication, matrix inversion, solution of linear simultaneous equations, and evaluation of the determinant of a matrix.

Addition

The program adds two matrices of m × n order (not to exceed a 19 × 19 matrix with the standard 8K byte memory capacity), using the following formula.

$$(M1 \times A) + (M2 \times B) = C$$

where A and B are m × n matrices, M1 and M2 are scalar multipliers, and C is the sum of the matrices.

The resulting matrix C replaces matrix A in memory and may be used in subsequent additions.

Multiplication

The program multiplies two matrices (not to exceed a 15 × 15 matrix with the standard 8K byte memory capacity), using the following formula.

$$(M1 \times A) \times (M2 \times B) = C$$

where M1 and M2 are scalar multipliers, A is an m × n matrix, B is an n × p matrix, and C is the resulting m × p matrix.

The procedure requires that matrix A be entered first. The next step is to enter matrix B. During multiplication, matrix A is lost and is not available for subsequent calculations.

Inversion

The program can invert a matrix A of N order (not to exceed an order of 15 with the standard 8K byte memory capacity), provided the determinant of A is not zero. The inverse of A may be stored with its columns permuted and must be reentered for subsequent calculations. The inversion is performed using the lower-upper decomposition method.

Simultaneous Equations

The program can solve a system of n linear equations with n unknowns, provided that the determinant of the matrix of coefficients is not zero. The solution uses the lower-upper decomposition method. After solution, the matrix of coefficients can be inverted without reentering the matrix. Also, you can solve the same system with different unknowns without reentering the matrix.

Determinant

The determinant is automatically calculated for the matrix inversion and simultaneous equations operations (not to exceed an order of 15 with the standard 8K byte memory capacity). However, you can compute the determinant independently. The determinant is calculated using the lower-upper decomposition method.

Method Used in the Program

The lower-upper (LU) decomposition method is used in the calculations, as described by the following equations. If the absolute value of A is not equal to zero, then A = LU where L is the lower triangular matrix and U is the upper triangular matrix. $U(u_{ij})$ and $L(l_{ij})$ are calculated using the following equations ($l_{kk} = 1$).

$$u_{kj} = a_{kj} - \sum_{p=1}^{k-1} l_{kp} u_{pj} \qquad j = (k, k+1, \ldots, n)$$

$$l_{ik} = \frac{a_{ik} - \sum_{p=1}^{k-1} l_{ip} u_{pk}}{u_{kk}} \qquad i = (k+1, \ldots, n)$$

The determinant of A is found as the product of the diagonal elements of U, and the inverse of A is calculated as:

$$A^{-1} = (LU)^{-1} = U^{-1} L^{-1}$$

$$y_{ij} = \frac{\delta_{ij} - \sum_{k=j}^{i-1} l_{ik} y_{kj}}{l_{ii}} \qquad i = (j, j+1, \ldots, n)$$

Method Used in the Program (Continued)

To complete this equation, if $L^{-1}=Y$ and $U^{-1}=Z$, then $\delta_{ij}=0$ for $i \neq j$; 1 for $i=j$:

$$z_{ij} = \frac{\delta_{ij} - \sum_{k=i+1}^{j} u_{ik} z_{kj}}{u_{ii}} \qquad i=(j, j-1, \ldots, 1)$$

If b is an $n \times 1$ column vector, then the system $Ax=b$ (not to exceed an order of 15 with the standard 8K byte memory capacity) is used during the following procedure. First, $Ly=b$ is solved for y where

$$y_{ij} = \frac{b_i - \sum_{k=1}^{i-1} I_{ik} x_k}{I_{ii}} \qquad i=(1, 2, \ldots, n)$$

Then $Ux=y$ is evaluated by

$$x_{ij} = \frac{b_i - \sum_{k=i+1}^{n} u_{ik} x_k}{u_{ii}} \qquad i=(n, n-1, \ldots, 1)$$

Partial pivoting is performed to improve accuracy.

Reference

Numerical Methods, Germund Dahlquist and Ake Bjork, Prentice Hall, 1974.

To execute the Matrices program, type RUN "MAT" and press [ENTER].

Step	Display	Procedure/Comment
1.	MATRICES	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	a. If you want to use a printer, press **[Y]**. b. If you do want to use a printer, press **[N]**. Go to step 4.
3.	Enter Device Code:	Type output device code and press **[ENTER]**.
4.	0-Menu 1-Add 2-Mult... 3-Det. 4-1/A 5-AX=B... 6-EXIT.	Displays the menu, pausing for 3 seconds while each line is displayed.
5.	Enter Option:	Type the number of the option corresponding to the desired operation and press **[ENTER]**. a. If you select option 0, go to step 4. b. If you select option 1 for the first time, go to step 7. If the previous problem used option 1, go to step 6. c. If you select option 2, go to step 7. d. If you select option 3, go to step 13. e. If you select option 4 or 5 for the first time, go to step 13. If the previous problem used option 5, go to step 12. f. If you select option 6, you exit the program.
6.	Chain With Previous Result?	a. To use the result of previous addition as matrix A, press **[Y]**. b. To enter a new A, press **[N]**.
7.	Enter MA:	Type the multiplier of matrix A and press **[ENTER]**.

Step	Display	Procedure/Comment
8.	`Enter MB:`	Type the multiplier of matrix B and press **[ENTER]**. a. If you are chaining the calculation with a previous result, go to step 21. b. If you are entering a new matrix, go to step 9.
9.	`Enter # Of Rows In A:`	Type the number of rows in matrix A and press **[ENTER]**.
10.	`Enter # Of Columns In A:`	Type the number of columns in matrix A and press **[ENTER]**. a. If you are using option 1, go to step 14. b. If you are using any other option, go to step 11.
11.	`Enter # Of Columns In B:`	Type the number of columns in matrix B. Go to step 14.
12.	`New A?`	a. To enter a new A, press **[Y]**. Go to step 13. b. To continue the program, press **[N]**. If you are using option 4, go to step 36. If you are using any other option, go to step 28.
13.	`Enter Order:`	Press **[ENTER]** to accept current value, or type the order of matrix A and press **[ENTER]**.
14.	`Enter A(#,#):`	Press **[ENTER]** to accept current value, or type the appropriate element of A and press **[ENTER]**. Repeat this step until all elements of matrix A are entered.

Step	Display	Procedure/Comment
15.	`Edit?`	a. To edit matrix A, press **[Y]**. Go to step 16. b. To continue the program, press **[N]**. If you are in option 1 or 2, go to step 21. If you are in option 3 or 4, go to step 35. If you are in option 5, go to step 28.
16.	`Edit All Input?`	a. To edit all elements of matrix A, press **[Y]**. Go back to step 14. b. To edit individual elements, press **[N]**.
17.	`Enter Row to Be Edited:`	Type the number of the row to be edited and press **[ENTER]**.
18.	`Enter Column To Be Edited:`	Type the number of the column to be edited and press **[ENTER]**.
19.	`Enter A(#,#):`	Press **[ENTER]** to accept current value, or type correct value and press **[ENTER]**.
20.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 17. b. To continue the program, press **[N]**. If you are in option 1 or 2, go to step 21. If you are in option 3 or 4, go to step 35. If you are in option 5, go to step 28.
21.	`Enter B(#,#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**. Repeat this step until all elements of matrix B are entered.

Step	Display	Procedure/Comment
22.	`Edit?`	a. To edit matrix B, press **[Y]**. Go to step 23. b. To continue the program, press **[N]**. Go to step 34.
23.	`Edit All Input?`	a. To edit all elements of matrix B, press **[Y]**. Go to step 21. b. To edit individual elements, press **[N]**.
24.	`Enter Row to Be Edited:`	Type the number of the row to be edited and press **[ENTER]**.
25.	`Enter Column To Be Edited:`	Type the number of the column to be edited and press **[ENTER]**.
26.	`Enter B(#,#):`	Press **[ENTER]** to accept current value, or type correct value and press **[ENTER]**.
27.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 24. b. To continue the program, press **[N]**.
28.	`Enter B(#):`	Press **[ENTER]** to accept current value, or type the correct element of the solution matrix B and press **[ENTER]**. Repeat this step until all elements are entered.
29.	`Edit?`	a. To edit solution matrix B, press **[Y]**. b. To continue the program, press **[N]**. Go to step 37.

Step	Display	Procedure/Comment
30.	`Edit All Input?`	a. To edit all elements of matrix B, press **[Y]**. Go to step 34. b. To edit individual elements, press **[N]**.
31.	`Enter Row to Be Edited:`	Type the number of the row to be edited and press **[ENTER]**.
32.	`Enter B(#):`	Press **[ENTER]** to accept current value, or type correct value and press **[ENTER]**.
33.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 31. b. To continue the program, press **[N]**.
34.	`C(#,#)=`	Displays result(s), one at a time. Continue to press **[ENTER]** until all results have been displayed. When output is complete, go to step 5.
35.	`Det=`	Displays determinant. Press **[ENTER]** to proceed. a. If you are in option 3, go to step 5. b. If you are in option 4, go to step 36. c. If you are in option 5, go to step 37.
36.	`Z(#,#)=`	Displays elements of inverse matrix, one at a time. Continue to press **[ENTER]** until all results have been displayed. When output is complete, go to step 5.
37.	`X(#)=`	Displays result(s) of AX = B, one at a time. Continue to press **[ENTER]** until all results have been displayed. When output is complete, go to step 5.

Application: Performing Matrix Arithmetic

This example performs the following matrix arithmetic:

$$(3) \times \left| (-2) \times \begin{matrix} 1 & 3 & 8 \\ 2 & 4 & 7 \end{matrix} + \begin{matrix} 5 & 6 & 9 \\ 3 & 0 & 8 \end{matrix} \right| + (1/3) \times \begin{matrix} -3 & 2 & -9 \\ 1 & -5 & 8 \end{matrix}$$

To execute the Matrices program, type RUN "MAT" and press [ENTER].

Step	Display	Procedure/Comment
1.	MATRICES	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	Use Printer?	Press **[N]**.
3.	0-Menu 1-Add 2-Mult... 3-Det. 4-1/A 5-AX=B.. 6-EXIT.	Displays the menu.
4.	Enter Option: 0	Type **1** and press **[ENTER]**.
5.	Enter MA: 1	Type **−2** and press **[ENTER]**.
6.	Enter MB: 1	Press **[ENTER]** to accept current value.
7.	Enter # Of Rows In A: 1	Type **2** and press **[ENTER]**.
8.	Enter # Of Columns In A: 1	Type **3** and press **[ENTER]**.
9.	Enter A(1,1): 0	Type **1** and press **[ENTER]**.
10.	Enter A(1,2): 0	Type **3** and press **[ENTER]**.
11.	Enter A(1,3): 0	Type **8** and press **[ENTER]**.
12.	Enter A(2,1): 0	Type **2** and press **[ENTER]**.
13.	Enter A(2,2): 0	Type **4** and press **[ENTER]**.

Step	Display	Procedure/Comment
14.	`Enter A(2,3): 0`	Type **7** and press **[ENTER]**.
15.	`Edit?`	Press **[N]**.
16.	`Enter B(1,1): 0`	Type **5** and press **[ENTER]**.
17.	`Enter B(1,2): 0`	Type **6** and press **[ENTER]**.
18.	`Enter B(1,3): 0`	Type **9** and press **[ENTER]**.
19.	`Enter B(2,1): 0`	Type **3** and press **[ENTER]**.
20.	`Enter B(2,2): 0`	Press **[ENTER]** to accept current value.
21.	`Enter B(2,3): 0`	Type **8** and press **[ENTER]**.
22.	`Edit?`	Press **[N]**.

23.	`C(1,1)= 3`
	`C(1,2)= 0`
	`C(1,3)= -7`
	`C(2,1)= -1`
	`C(2,2)= -8`
	`C(2,3)= -6`
		Displays all requested values, one at a time. Press **[ENTER]** after each output to continue.

24.	`Enter Option: 1`	Press **[ENTER]** to accept current value.
25.	`Chain With Previous Result?`	Press **[Y]**.

Step	Display	Procedure/Comment
26.	`Enter MA:  1`	Type **3** and press **[ENTER]**.
27.	`Enter MB:  1`	Type **1/3** and press **[ENTER]**.
28.	`Enter B(1,1): 5`	Type **−3** and press **[ENTER]**.
29.	`Enter B(1,2): 6`	Type **2** and press **[ENTER]**.
30.	`Enter B(1,3): 9`	Type **−9** and press **[ENTER]**.
31.	`Enter B(2,1): 3`	Type **1** and press **[ENTER]**.
32.	`Enter B(2,2): 0`	Type **−5** and press **[ENTER]**.
33.	`Enter B(2,3): 8`	Press **[ENTER]** to accept current value.
34.	`Edit?`	Press **[N]**.

35.	`C(1,1)= 8`
	`C(1,2)= .6666666667`
	`C(1,3)=-24`
	`C(2,1)=-2.666666667`
	`C(2,2)=-25.66666667`
	`C(2,3)=-15.33333333`
		Displays all requested values. Press **[ENTER]** after each output to continue.

36.	`Enter Option: 1`	To exit the program, type **6** and press **[ENTER]**.

The Tridiagonal Matrix (TRI) Program

The Tridiagonal Matrix program solves matrices that are in the tridiagonal form. It is appropriate to use the tridiagonal matrix program when a matrix has non-zero elements only along the main diagonal and the diagonals on either side.

Notation

A tridiagonal system of equations is a specialized form for the solution of matrix systems. The tridiagonal is represented as a square matrix of order n, not to exceed an order of 100 with the standard 8K byte memory capacity.

$$\text{for } n=4 \quad \begin{matrix} b_1 & c_1 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 \\ 0 & a_3 & b_3 & c_3 \\ 0 & 0 & a_4 & b_4 \end{matrix} \quad \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{matrix} = \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{matrix}$$

To run this program, you enter the coefficients $a_2, \ldots, a_n$, $b_1, \ldots, b_n$, $c_1, \ldots, c_{n-1}$, and $d_1, \ldots, d_n$, which are represented in the following form.

$$\begin{aligned} b_1 X_1 + c_1 X_2 &= d_1 \\ a_2 X_1 + b_2 X_2 + c_2 X_3 &= d_2 \\ a_3 X_2 + b_3 X_3 + c_3 X_4 &= d_3 \\ a_4 X_3 + b_4 X_4 &= d_4 \end{aligned}$$

This program utilizes Gaussian elimination for the solution of a tridiagonal set of equations.

For a tridiagonal arrangement of elements, the TI-74 can solve a larger matrix than the matrices program.

Method Used in the Program

The equations are

$$A_i U_{i-1} + b_i U_i + C_i U_{i+1} = d_i$$

for $1 \leqslant i \leqslant n$

with $A_1 = C_n = 0$

First compute

$$B_i = b_i - \frac{a_i C_{i-4}}{b_{i-1}} \text{ with } B_1 = b_1$$

and

$$G_i = \frac{d_i - A_i G_{i-1}}{B_i} \text{ with } G_1 = \frac{d_1}{b_1}$$

The values of the dependent variable are

$$U_n = G_n$$

and

$$U_i = G_i - \frac{C_i U_{i+1}}{b_i}$$

Reference

Numerical Methods. Robert W. Hornbeck, Quantum Publishers, Inc., 1975, pp. 93–98.

Using the Tridiagonal Matrix Program

To execute the Tridiagonal Matrix program, type RUN "TRI" and press [ENTER].

Step	Display	Procedure/Comment
1.	`TRIDIAGONAL MATRIX`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	a. If you want to use a printer, press **[Y]**. b. If you do not want to use a printer, press **[N]**. Go to step 4.
3.	`Enter Device Code:`	Type output device code and press **[ENTER]**.
4.	`Enter Order:`	Press **[ENTER]** to accept current value, or type the order of the matrix and press **[ENTER]**.
5.	`Enter A(#):`	Press **[ENTER]** to accept current value, or type the appropriate element and press **[ENTER]**. Repeat this step until all A elements are entered.
6.	`Edit?`	a. To edit data, press **[Y]**. b. To continue the program, press **[N]**. Go to step 11.
7.	`Edit All Input?`	a. To edit all of A, press **[Y]**. Go to step 5. b. To edit individual elements, press **[N]**.
8.	`Enter Element To Be Edited:`	Type the number of the element to be edited and press **[ENTER]**.
9.	`Enter A(#):`	Press **[ENTER]** to accept current value, or type the correct value.
10.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 8. b. To continue the program, press **[N]**.

Step	Display	Procedure/Comment
11.	`Enter B(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**. Repeat this step until all B elements are entered.
12.	`Edit?`	a. To edit data, press **[Y]**. b. To continue the program, press **[N]**. Go to step 17.
13.	`Edit All Input?`	a. To edit all of B, press **[Y]**. Go to step 11. b. To edit individual elements, press **[N]**.
14.	`Enter Element To Be Edited:`	Type the number of the element to be edited and press **[ENTER]**.
15.	`Enter B(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**.
16.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 14. b. To continue the program, press **[N]**.
17.	`Enter C(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**. Repeat this step until all C elements are entered.
18.	`Edit?`	a. To edit data, press **[Y]**. b. To continue the program, press **[N]**. Go to step 23.

Step	Display	Procedure/Comment
19.	`Edit All Input?`	a. To edit all of C, press **[Y]**. Go to step 17. b. To edit individual elements, press **[N]**.
20.	`Enter Element To Be Edited:`	Type the number of the element to be edited and press **[ENTER]**.
21.	`Enter C(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**.
22.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 20. b. To continue the program, press **[N]**.
23.	`Enter D(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**. Repeat this step until all D elements are entered.
24.	`Edit?`	a. To edit data, press **[Y]**. b. To continue the program, press **[N]**. Go to step 29.
25.	`Edit All Input?`	a. To edit all of C, press **[Y]**. Go to step 23. b. To edit individual elements, press **[N]**.
26.	`Enter Element To Be Edited:`	Type the number of the element to be edited and press **[ENTER]**.

Step	Display	Procedure/Comment
27.	`Enter D(#):`	Press **[ENTER]** to accept current value, or type the correct value and press **[ENTER]**.
28.	`Edit Other Elements?`	a. To edit other elements, press **[Y]**. Go to step 26. b. To continue the program, press **[N]**.
29.	`X#=`	Displays each solution. Continue to press **[ENTER]** until all have been displayed.
30.	`Exit Program?`	a. To exit the program, press **[Y]**. b. To continue the program, press **[N]**. Go to step 4.

Application: Solving a Tridiagonal Matrix

Find the solutions to the following tridiagonal matrix.

$$\begin{vmatrix} 3 & 2 & 0 \\ 4 & 7 & 3 \\ 0 & 1 & 9 \end{vmatrix} \qquad \begin{vmatrix} 27 \\ 86 \\ 78 \end{vmatrix}$$

To execute the program, type RUN "TRI" and press [ENTER].

Step	Display	Procedure/Comment
1.	`TRIDIAGONAL MATRIX`	Wait while the program name is displayed, or press **[ENTER]** to continue.
2.	`Use Printer?`	Press **[N]**.
3.	`Enter Order: 0`	Type **3** and press **[ENTER]**.
4.	`Enter A(2): 0`	Type **4** and press **[ENTER]**.
5.	`Enter A(3): 0`	Type **1** and press **[ENTER]**.
6.	`Edit?`	Press **[N]**.
7.	`Enter B(1): 0`	Type **3** and press **[ENTER]**.
8.	`Enter B(2): 0`	Type **7** and press **[ENTER]**.
9.	`Enter B(3): 0`	Type **9** and press **[ENTER]**.
10.	`Edit?`	Press **[N]**.
11.	`Enter C(1): 0`	Type **2** and press **[ENTER]**.
12.	`Enter C(2): 0`	Type **3** and press **[ENTER]**.
13.	`Edit?`	Press **[N]**.

Step	Display	Procedure/Comment
14.	`Enter D(1): 0`	Type **27** and press **[ENTER]**.
15.	`Enter D(2): 0`	Type **86** and press **[ENTER]**.
16.	`Enter D(3): 0`	Type **78** and press **[ENTER]**.
17.	`Edit?`	Press **[N]**.
18.	`X1= 5`	Displays result. Press **[ENTER]** to continue.
19.	`X2= 6`	Displays result. Press **[ENTER]** to continue.
20.	`X3= 8`	Displays result. Press **[ENTER]** to continue.
21.	`Exit Program?`	Press **[Y]**.

Application: Solving a Tridiagonal Matrix

Find the solutions to the following tridiagonal matrix:

5 1 0 [illegible]
4 7 3 [illegible]
[illegible]

To execute the program, type RUN "TRI" and press [ENTER].

Step	Display	Procedure/Comment
1.	[illegible]	[illegible]
2.	[illegible]	Press [N].
3.	[illegible]	Type 3 and press [ENTER].
4.	Enter A(2)? 0	Type 4 and press [ENTER].
5.	Enter A(3)? 0	[illegible] and press [ENTER].
6.	[illegible]	[illegible]
7.	Enter B(1)? 0	Type 5 and press [ENTER].
8.	Enter B(2)? 0	Type 7 and press [ENTER].
9.	Enter B(3)? 0	Type 8 and press [ENTER].
10.	Edit?	Press [N].
11.	Enter C(1)? 0	Type 2 and press [ENTER].
12.	Enter C(2)? 0	Type 2 and press [ENTER].
13.	[illegible]	Press [N].

Appendix A: User-Accessible Subprograms

This appendix describes the subprograms used by the Mathematics Library programs.

Table of Contents

Subprogram Information

Many of the routines used in the programs of this library are developed as independent subprograms. These subprograms may be accessed for use in your own BASIC programs by calling the subprogram with its required arguments.

The Information in this Appendix

This appendix provides the information you need for including the cartridge subprograms in your programs. Most of the subprograms are arranged in alphabetical order by their calling names; the last three pages cover mathematical and trigonometric subprograms for the complex number X, mathematical subprograms for the complex numbers X and Y, and polar-to-rectangular and rectangular-to-polar conversions for the complex number X.

Unless otherwise noted, subprograms that are passed an array name discard the array contents when the subprogram is called.

The discussion of each subprogram describes the routine that the subprogram performs and gives the following information, as appropriate:

- **Format**—illustrates the format necessary for calling the subprogram and identifies the requirements, restrictions, and purpose of each element in the argument list. Elements in all capital letters should be entered exactly as they appear in the format line.

- **Example**—demonstrates the procedure for executing the subprogram.

- **Special Requirements**—includes any additional information required for using the subprogram.

Calling a Subprogram

To call a subprogram in your BASIC program, you use the CALL statement with the name of the subprogram and its arguments.

Refer to the *TI-74 Programming Reference Guide* for details on the restrictions and requirements for using subprograms.

The AK subprogram allows you to input and edit a one-dimensional array.

Format CALL AK(*prompt$*,*array*(),*first*,*last*,*printer*)

prompt$ Passes the name or description of the array to be displayed preceding the subscript.
array() Passes the array name used in the calling program.
first Passes the first array element to be entered.
last Passes the last array element to be entered.
printer Passes the device number of the output device. If *printer* = 0, input is displayed, rather than printed.

After all array elements are input, the subprogram asks if you wish to edit the array. If you respond with **N** for no, the subprogram passes control back to the calling program. If you respond with **Y** for yes, the subprogram allows you to edit all of the elements previously entered or to edit only selected elements.

Example **100 CALL AK("NAME",A(),1,5,PN)**

When the statement is executed, the subprogram displays the following prompt to allow you to input into the array A() elements 1 through 5 inclusive:

```
Enter NAME(X):
```

where X ranges from 1 to 5. After you enter the elements, the subprogram prompts you for editing by displaying:

```
Edit?
```

An **N** response returns control to the calling program. A **Y** response prompts

```
Edit All Input?
```

A **Y** response allows you to review and, if necessary, reenter the previously entered elements.

Example (Continued)

An **N** response causes the following prompt to be displayed:

```
Enter Element To Be Edited:
```

Enter the specific element number to be edited, which must be one of the array elements previously entered. After you edit the element, the subprogram prompts

```
Edit Other Elements?
```

A **Y** response returns to the prompt

```
Enter Element To Be Edited:
```

An **N** response returns control to the calling program.

The AU subprogram allows you to input and edit a pair of one-dimensional arrays.

Format

CALL AU(*prompt1*,*prompt2*,*array1*(),*array2*(),*first*,*last*,*printer*)

prompt1 Passes the name or description of *array1*() to be displayed preceding the subscript.

prompt2 Passes the name or description of *array2*() to be displayed preceding the subscript.

array1() Passes the array name used in the calling program.

array2() Passes the array name used in the calling program.

first Passes the first array element to be entered.

last Passes the last array element to be entered.

printer Passes the device number of the output device. If *printer*=0, input is displayed, rather than printed.

After you input all array elements, the subprogram asks if you wish to edit the array. If you respond with **N** for no, the subprogram passes control back to the calling program. If you respond with **Y** for yes, the subprogram allows you to edit all of the elements previously entered or to edit only selected elements.

Example

100 CALL AU("ONE","TWO",A(),B(),1,5,PN)

When the statement is executed, the subprogram displays the following prompt to allow you to input into arrays A() and B() elements 1 through 5 inclusive:

```
Enter One(X):

Enter Two(X):
```

where X ranges from 1 to 5. After you enter the elements, the subprogram prompts you for editing by displaying

```
Edit?
```

An **N** response returns control to the calling program.

Example (Continued)

A **Y** response prompts

```
Edit All Input?
```

A **Y** response allows you to review and, if necessary, reenter the previously entered elements. An **N** response causes the following prompt to be displayed:

```
Enter Element To Be Edited:
```

Enter the specific element number to be edited, which must be one of the array elements previously entered. After you edit the element, the subprogram prompts

```
Edit Other Elements?
```

A **Y** response returns to the prompt

```
Enter Element To Be Edited:
```

An **N** response returns control to the calling program.

BISS and CMAT

The BISS subprogram finds the root of a function. The CMAT subprogram finds the solutions to a system of equations of complex numbers.

BISS Format

CALL BISS(*bound1*,*bound2*,*epsilon*)

bound1 Passes one of the points bounding the root.
bound2 Passes the other point bounding the root.
epsilon Passes the error limit for the estimate, and returns the result.

This subprogram finds the root dependent upon the test ABS(*bound2* – *bound1*) < *epsilon*.

Special Requirements

Before calling this subprogram, you must enter your own function, using the following format and adjusting line numbers as necessary.

```
1 SUB  FX(input,output)
2 output = f(input)
3 SUBEND
```

Everything in capitals must be exactly as shown.

CMAT Format

CALL CMAT(*order*,*array*(,))

order Passes the order of the system, and returns 0 if the system is singular.
array(,) Passes the array to be solved, and returns the solutions to the system.

Refer to the instructions for the Complex Systems (COMPS) program to see the proper format for the array that is passed.

The solutions are returned in the following manner:

X1 is in ARRAY(1,1),
Y1 is in ARRAY(2,1),
X2 is in ARRAY(3,1),
Y2 is in ARRAY(4,1),
etc.

The CONVO subprogram finds the convolution integral using the trapezoidal rule. The D3 subprogram finds the solutions to a tridiagonal system of equations.

CONVO Format

CALL CONVO(*panels*,*outputs*,*stepsize*,*array*())

panels Passes the number of panels per step.
outputs Passes the number of outputs desired.
stepsize Passes the step size to be used for integration
array() Passes an empty array dimensioned at least as big as *outputs*, and returns solutions to the convolution integral. The first value is in *array*(1), the second is in *array*(2), etc.

Special Requirements

Before calling this subprogram, you must enter your own functions, using the following format and adjusting line numbers as necessary.

1 SUB FX(*input,output*)
2 *output* = f(*input*)
3 SUBEND
4 SUB FH(*input,output*)
5 *output* = f(*input*)
6 SUBEND

Everything in capitals must be exactly as shown.

D3 Format

CALL D3(*order*,*array1*(),*array2*(),*array3*(),*array4*())

order Passes the order of the system, and returns 0 if the system is singular.
array1() Passes the A diagonal values.
array2() Passes the B diagonal values.
array3() Passes the C diagonal values.
array4() Passes the D values, and returns the solutions. X1 in *array4*(1), etc.

The DER2 subprogram, given several points of a function, returns the values of the second derivatives at those points. The GAMS subprogram finds the natural log of the gamma function for a given number.

DER2 Format CALL DER2(*number*,*array1*(),*array2*(),*array3*())

number Passes the number of data points minus 2. Must be at least 1.

array1() Passes the X values of the data points. Must be in sequential order.

array2() Passes the Y values of the corresponding X values.

array3() Passes an empty array dimensioned as *number*+2, and returns the second derivatives in order.

GAMS Format CALL GAMS(*input*,*output*)

input Passes the number to be solved for.

output Returns the value LN(gamma(*input*)).

The GAUS subprogram finds the integral of a function using Gauss Quadrature. The II subprogram inputs a positive integer.

GAUS Format

CALL GAUS(*lower*,*upper*,*number*,*output*)

lower	Passes the lower limit of the integral.
upper	Passes the upper limit of the integral.
number	Passes the number of partitions to be used.
output	Passes a dummy variable that returns the solution to the integral.

Special Requirements

Before calling the subprogram, you must enter your own function, using the following format and adjusting line numbers as necessary.

```
1 SUB  FX(input,output)
2 output = f(input)
3 SUBEND
```

Everything in capitals must be exactly as shown.

II Format

CALL II(*name$*,*variable*,*printer*)

name$	Passes the name of the input variable in quotes.
variable	Passes the variable which is to receive the integer, and returns the input value.
printer	Passes the number of the output device. If *printer* = 0, the input is displayed, rather than printed.

Special Requirements

This subprogram does not allow non-integer or non-positive input.

The INTERP subprogram performs a cubic splines interpolation after finding the second derivatives. The MATS subprogram multiplies or adds two matrices and finds the determinant of, inverts, or finds the solutions to a matrix.

INTERP Format

CALL INTERP(*number*,*inputX*,*array1*(),*array2*(),
array3(),*output*)

number	Passes *number* from DER2.
inputX	Passes the X to be interpolated for.
array1()	Passes *array1*() from DER2.
array2()	Passes *array2*() from DER2.
array3()	Passes *array3*() from DER2.
output	Passes a dummy variable that returns the interpolated value.

Special Requirements

This subprogram must be run after DER2.

MATS Format

CALL MATS(*array1*(,),*array2*(,),*array3*(),*multA*,*multB*,
option,*row*,*number*,*col*,*test*)

array1(,)	Passes matrix A.
array2(,)	Passes matrix B.
array3()	Passes the solution matrix to solve AX=B.
multA	Passes the multiplier of matrix A.
multB	Passes the multiplier of matrix B.
option	Passes the option number to be performed: 1-ADD, 2-MULT, 3-DETERMINANT, 4-INVERT, 5-SOLVE AX=B.
row	Passes the number of rows in matrix A (options 1 and 2 only).
number	Passes the number of columns in matrix A (options 1 and 2 only). Passes the order of matrix A (options 3, 4, and 5 only).
col	Passes the number of columns in B (option 2 only).
test	Passes a dummy variable, and returns a 0 if the matrix is singular.

Special Requirements

You must pass dummy variables for any parameters not used in the program you are performing.

The MI subprogram allows you to input and edit a two-dimensional array.

Format CALL MI(*prompt$*,*array*(,),*first*,*lastrow*,*lastcol*,*printer*)

prompt$ Passes the name or description of the array to be displayed preceding the subscript.
array(,) Passes the array name used in the calling program.
first Passes the first array element to be entered.
lastrow Passes the last row number to be entered.
lastcol Passes the last column number to be entered.
printer Passes the device number of the output device. If *printer*=0, input is displayed, rather than printed.

After you have input all array elements, the subprogram asks if you wish to edit the array. If you respond with **N** for no, the subprogram passes control back to the calling program. If you respond with **Y** for yes, the subprogram allows you to edit all of the elements previously entered or to edit only selected elements.

Example **100 CALL MI("NAME",A(,),1,5,5,PN)**

When the statement is executed, the subprogram displays the following prompt to allow you to input into the array A(,) elements (1,1) through (5,5) inclusive.

```
Enter NAME(X,Y):
```

where X ranges from 1 to 5 and Y ranges from 1 to 5 for each X. After you enter the elements, the subprogram prompts you for editing by displaying:

```
Edit?
```

An **N** response returns control to the calling program. A **Y** response prompts

```
Edit All Input?
```

A **Y** response allows you to review and, if necessary, reenter the previously entered elements.

Example (Continued)

An **N** response causes the following prompts to be displayed:

```
Enter Row To Be Edited:
```

and

```
Enter Column To Be Edited:
```

Enter the specific element number to be edited, which must be one of the array elements previously entered. After you edit the element, the subprogram prompts

```
Edit Other Elements?
```

A **Y** response returns to the prompt

```
Enter Row To Be Edited:
```

An **N** response returns control to the calling program.

MINS and NROOT

The MINS subprogram finds the relative minimum of a function between points A and B. The NROOT subprogram finds the root of a function using Newton's method.

MINS Format

CALL MINS(*bound1*,*bound2*,*epsilon*)

bound1 Passes one of the boundary values.
bound2 Passes the second boundary value.
epsilon Passes the value for epsilon, and returns the X value of the minimum point. The epsilon test is ABS(*bound2* − *bound1*) < *epsilon*.

Special Requirements

Before calling this subprogram, you must enter your own function, using the following format and adjusting line numbers as necessary.

```
1 SUB FX(input,output)
2 output = f(input)
3 SUBEND
```

Everything in capitals must be exactly as shown.

NROOT Format

CALL NROOT(*initialX*,*epsilon*)

initialX Passes the initial guess for X.
epsilon Passes the value for epsilon. The epsilon test is $SQR(Y^2 + D^2) \leqslant$ *epsilon* where Y is the value of the function at X and D is −Y ÷ (value of the derivative at X).

Special Requirements

Before calling this subprogram, you must enter your own functions, using the following format and adjusting line numbers as necessary.

```
1 SUB FX(input,output)
2 output = f(input)
3 SUBEND
4 SUB FD(input,output)
5 output = f'(input)
6 SUBEND
```

Everything in capitals must be exactly as shown.

PLYMLT, PRIS, AND RI

The PLYMLT subprogram multiplies two polynomials. The PRIS subprogram finds all the prime factors of a given positive integer. The RI subprogram inputs a real number.

PLYMLT Format CALL PLYMLT(*order1*,*order2*,*array1*(),*array2*())

order1	Passes the order of *array1*.
order2	Passes the order of *array2*.
array1()	Passes the first polynomial, and returns the resulting polynomial. The subscript corresponds to the exponent associated with the particular coefficient.
array2()	Passes the second polynomial. The subscript corresponds to the exponent associated with the particular coefficient.

PRIS Format CALL PRIS(*integer*,*printer*)

integer	Passes the number whose prime factors are to be found.
printer	Passes the number of the output device which will receive the prime factors. If *printer* = 0, the output is displayed, rather than printed.

RI Format CALL RI(*name$*,*variable*,*printer*)

name$	Passes the name of the input variable in quotes.
variable	Passes the variable that is to receive the number, and returns the input value.
printer	Passes the number of the output device. If *printer* = 0, the input is displayed, rather than printed.

The RK subprogram finds the values of a system of differential equations at particular points.

Format CALL RK(*initial*,*final*,*step*,*number*,*array*( ),*test*,*printer*)

initial	Passes the initial value for X.
final	Passes the final value for X.
step	Passes the step size for computation.
number	Passes the number of equations in the system.
array( )	Passes the initial values for the Y's.
test	Passes a number telling the subprogram to either display the final values only or to display all the intermediate results as well. To display only the final results, pass a value of 1. To display all intermediate results, pass a value of 0, in which case the final result is printed as well. If a value of 1 is passed, the results are returned in *array*( ).
printer	Passes the number of the output device to be used if you passed a 0 for *test*. If *printer* = 0, the output is displayed, rather than printed. If you passed a value of 1 for *test*, this is a dummy variable.

Special Requirements Before calling this subprogram, you must enter your own functions, using the following format and adjusting line numbers as necessary.

```
1 SUB  F1(in,array(),out)
2 out = f(in,array(1), . . . , array(n))
3 SUBEND
4 SUB  F2(in,array(),out)
5 out = f(in,array(1), . . . , array(n))
6 SUBEND
.
.
# SUB  Fn(in,array(),out)
# + 1 out = f(in,array(1), . . . , array(n))
# + 2 SUBEND
```

The number *n* is the number of functions in the system. Everything in capitals must be exactly as shown.

The UP subprogram prompts you for printer use.

Format CALL UP(*name$*,*printer*)

name$ Passes the name of the program to be displayed.

printer Returns the number of the opened output device. A device number of 0 indicates output will appear on the display, while a device number of 1 indicates the requested output device was opened as device #1.

Example **100 CALL UP("CUBIC SPLINES",PN)**

When the line is executed, the following actions occur.

1. The name CUBIC SPLINES appears in the display, followed by the prompt

   ```
   Use Printer?
   ```

2. If you enter an **N** for no, control is returned to the calling program. If you enter a **Y** for yes, the subprogram prompts

   ```
   Enter Device Name:
   ```

   Enter the device code for the printer. PN is then set to the device code for the printer, the device is opened for output, and the subprogram then returns control to the calling program.

The WR subprogram prints a message surrounded by five asterisks. The YN subprogram prompts for a yes/no response.

WR Format

CALL WR(*message$*,*printer*,*pause-time*)

message$	Passes the message, in quotes, to be printed.
printer	Passes a number to the output device where the message is to be printed. If *printer*=0, the message is displayed rather than printed.
pause-time	Specifies the number of seconds to pause.

Example

100 CALL WR("MESSAGE",PN,1)

When the line is executed, the message MESSAGE is displayed for one second and then printed.

YN Format

CALL YN(*question$*,*answer*,*printer*)

question$	Passes the question to be asked. (Do not include a question mark; it is added automatically.)
answer	Returns *answer*=0 if the response is **N** or *answer*= −1 if the response is **Y**.
printer	Passes the device number. If *printer*=0, *question$* is displayed, rather than printed. Otherwise, *question$* is printed.

Example

100 CALL YN("Edit",D,PN)

When the line is executed, the question

```
Edit?
```

is displayed. A **Y** response sets D equal to −1 and prints EDIT on device # PN. An **N** response sets D equal to 0. Control is then passed back to the calling program.

Math and Trig Subprograms for a Complex Number

Several subprograms perform mathematical and trigonometric operations on a complex number.

The Subprograms The mathematical and trigonometric subprograms that operate on a single complex number are listed in the following table.

Subprogram	Operation
SQR	Computes the square of a complex number.
SQRT	Computes the square root of a complex number.
RECIP	Computes the reciprocal of a complex number.
CLN	Computes the natural log of a complex number.
CXP	Computes the exponential value of a complex number (e^x).
CSIN	Computes the sine of a complex number.
CCOS	Computes the cosine of a complex number.
CTAN	Computes the tangent of a complex number.
ASIN	Computes the inverse sine of a complex number.
ACOS	Computes the inverse cosine of a complex number.
ATAN	Computes the inverse tangent of a complex number.

Sample Format The following format applies to all these subprograms.

CALL SQR(*real*,*imaginary*)

real Passes the real part of X, and returns the real part of the result.

imaginary Passes the imaginary part of X, and returns the imaginary part of the result.

Mathematical Subprograms for Two Complex Numbers

Several subprograms perform mathematical operations on two complex numbers.

The Subprograms The mathematical subprograms that operate on two complex numbers are listed in the following table.

Subprogram	Operation
ADD	Adds the complex numbers X and Y.
SUB .	Subtracts the complex number Y from the complex number X.
MULT	Multiplies the complex number Y by the complex number X.
DIV	Divides the complex number X by the complex number Y.
YTOX	Raises the complex number Y to the power of the complex number X.
XRTY	Takes the complex number Xth root of the complex number Y.
LOGYX	Computes the log base complex number Y of the complex number X.

Sample Format The following format applies to all these subprograms.

CALL ADD(*realX*,*imaginaryX*,*realY*,*imaginaryY*)

realX	Passes the real part of X, and returns the real part of the result.
imaginaryX	Passes the imaginary part of X, and returns the imaginary part of the result.
realY	Passes the real part of Y.
imaginaryY	Passes the imaginary part of Y.

Polar/Rectangular Conversions for a Complex Number

Two subprograms allow you to convert a complex number from polar to rectangular coordinates or from rectangular to polar coordinates.

The Subprograms

The two subprograms are described below.

Subprogram	Operation
PTOR	Converts a complex number from polar to rectangular coordinates.
RTOP	Converts a complex number from rectangular to polar coordinates.

Formats

CALL PTOR(*magnitude*,*angle*)
CALL RTOP(*real*,*imaginary*)

magnitude	Passes the magnitude of the polar number X, and returns the real part of the result.
angle	Passes the angle of the polar number X, and returns the imaginary part of the result.
real	Passes the real part of X, and returns the magnitude of the result.
imaginary	Passes the imaginary part of X, and returns the angle of the result.

[illegible] operations on two complex numbers are listed in the following table.

Subprogram	Operation
[illegible]	[illegible] complex numbers X and Y.
[illegible]	[illegible] the complex number Y from the complex number X.
MULT	Multiplies the complex number X by the [illegible]
DIV	Divides the complex number X by the [illegible]
[illegible]	[illegible] complex number Y to the power [illegible] complex number X.
[illegible]	[illegible] complex number Xth root of the complex number Y.
[illegible]	[illegible] the log base complex number Y of the complex number X.

Sample Format The following format applies to all these subprograms.

CALL ADD(*realX*, *imaginaryX*, *realY*, *imaginaryY*)

realX	Passes the real part of X, and returns the real part of the result.
imaginaryX	Passes the imaginary part of X, and returns the imaginary part of the result.
realY	Passes the real part of Y.
imaginaryY	Passes the imaginary part of Y.

Appendix B: Service and Warranty Information

This appendix describes the service provided by Texas Instruments and the warranty for the cartridge.

Table of Contents

Service Information

If you experience a problem with the cartridge, you can call or write Consumer Relations to discuss the problem.

For Service and General Information

If you have questions about service or the general use of the cartridge, please call Consumer Relations **toll-free** within the United States at:

1–800–TI CARES (842–2737).

From outside the United States, call 1–806–741–4800. (We cannot accept collect calls at this number.)

You may also write to the following address:

Texas Instruments Incorporated
Consumer Relations
P.O. Box 53
Lubbock, Texas 79408

Please contact Consumer Relations:

- Before returning the cartridge for service
- For general information about using the cartridge

For Technical Information

If you have technical questions about cartridge operation or programming applications, call 1–806–741–2663. We regret that we cannot accept collect calls at this number. As an alternative, you can write Consumer Relations at the address given above.

Express Service

Texas Instruments offers an express service option for fast return delivery. Please call Consumer Relations at 1–800–TI CARES (842–2737) for information.

Accessories

If you are unable to purchase calculator accessories (such as the CI–7 cassette interface cable) from your local dealer, you may order them from Texas Instruments. Please call Consumer Relations at 1–800–TI CARES (842–2737) for information.

Returning a Cartridge for Service

A defective cartridge will be either repaired or replaced with the same or comparable reconditioned model (at TI's option) when it is returned, postage prepaid, to a Texas Instruments Service Facility.

Texas Instruments cannot assume responsibility for loss or damage during incoming shipment. For your protection, carefully package the cartridge for shipment and insure it with the carrier. Be sure to enclose the following items with the cartridge:

- Your full return address
- Any accessories related to the problem
- A note describing the problem you experienced
- A copy of your sales receipt or other proof of purchase to determine warranty status

Please ship the cartridge postage prepaid; COD shipments cannot be accepted.

In-Warranty Repair

For a cartridge covered under the warranty period, no charge is made for service.

Out-of-Warranty Repair

For an out-of-warranty cartridge, a flat-rate fee by model is charged for service. Estimates are not provided prior to repair; to obtain the service charge for a particular model, please call Consumer Relations before returning the cartridge to the Service Facility.

Texas Instruments Service Facilities

U.S. Residents (U.S. Postal Service)
Texas Instruments
P.O. Box 2500
Lubbock, Texas 79408

U.S. Residents (other carriers)
Texas Instruments
2305 N. University
Lubbock, Texas 79415

Canadian Residents Only
Texas Instruments
41 Shelley Road
Richmond Hill, Ontario L4C 5G4

One-Year Limited Warranty

This Texas Instruments software cartridge warranty extends to the original consumer purchaser of the product.

Warranty Duration

This cartridge is warranted to the original consumer purchaser for a period of one (1) year from the original purchase date.

Warranty Coverage

This cartridge is warranted against defective materials and construction. This warranty covers the electronic and case components of the software cartridge. These components include all semiconductor chips and devices, plastics, boards, wiring, and all other hardware contained in this cartridge ("the Hardware"). This limited warranty does not extend to the programs contained in the cartridge and the accompanying book materials ("the Programs"). **The warranty is void if the cartridge has been damaged by accident or unreasonable use, neglect, improper service, or other causes not arising out of defects in materials or construction.**

Warranty Disclaimers

Any implied warranties arising out of this sale, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, are limited in duration to the above one year period. Texas Instruments shall not be liable for loss of use of the cartridge or other incidental or consequential costs, expenses, or damages incurred by the consumer or any other user.

Some states do not allow the exclusion or limitations of implied warranties or consequential damages, so the above limitations or exclusions may not apply to you.

Legal Remedies

This warranty gives you specific legal rights, and you may also have other rights that vary from state to state.

Warranty Performance

During the above one-year warranty period, your TI cartridge will be either repaired or replaced with a reconditioned comparable model (at TI's option) when the cartridge is returned, postage prepaid, to a Texas Instruments Service Facility. The repaired or replacement cartridge will be in warranty for the remainder of the original warranty period or for six months, whichever is longer. Other than the postage requirement, no charge will be made for such repair or replacement. Texas Instruments strongly recommends that you insure the product for value, prior to mailing.

TEXAS INSTRUMENTS

Printed in Taiwan

1060309-0101