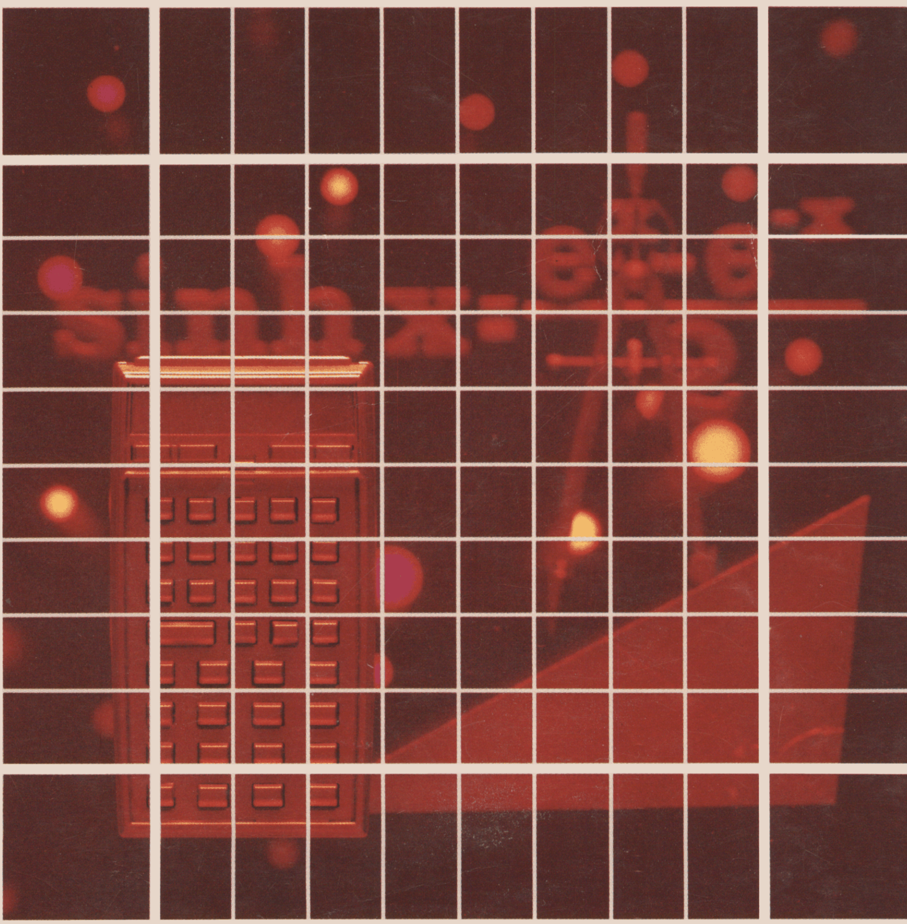
HEWLETT-PACKARD

HP-41C

MATH PAC



INTRODUCTION

The programs in the Math Pac have been drawn from the fields of calculus, numerical analysis, linear systems, analytical geometry and special functions. Each program in this pac is represented by one program in the Application Module and a section in this manual. The manual provides a description of the program with relevant equations, a set of instructions for using the program, and one or more example problems, each of which includes a list of the keystrokes required for its solution.

Before plugging in your Application Module, **turn your calculator off,** and be sure you understand the section Inserting and Removing Application Modules. And before using a particular program, take a few minutes to read Format of User Instructions and A Word About Program Usage.

You should first familiarize yourself with a program by running it once or twice while following the complete User Instructions in the manual. Thereafter, the program's prompting or the mnemonics on the overlays should provide the necessary instructions, including which variables are to be input, which keys are to be pressed, and which values will be output. A quick-reference card with a brief description of each program's operating instructions has been provided for your convenience.

We hope that Math Pac I will assist you in the solution of numerous problems in your discipline. We would appreciate knowing your reactions to the programs in this pac, and to this end we have provided a questionnaire inside the front cover of this manual. Would you please take a few minutes to give us your comments on these programs? It is from your comments that we learn how to increase the usefulness of our programs.

CONTENTS

INSERTING AND REMOVING APPLICATION MODULES

Before you insert an application module for the first time, familiarize yourself with the following information.

Up to four application modules can be plugged into the ports on the HP-41C. While plugged in, the names of all programs contained in the module can be displayed by pressing ■ [CATALOG] 2.

CAUTION

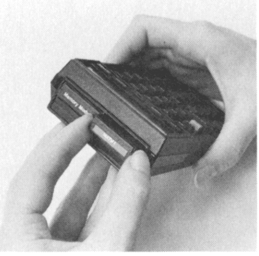
Always turn the HP-41C off before inserting or removing any plug-in extensions or accessories. Failure to turn the HP-41C off could damage both the calculator and the accessory.

Here is how you should insert application modules:

1. Turn the HP-41C off! Failure to turn the calculator off could damage both the module and the calculator.

2. Remove the port covers. Remember to save the port covers, they should be inserted into the empty ports when no extensions are inserted.

3. With the application module label facing downward as shown, insert the application module into any port **after** the last memory module presently inserted.



4. If you have additional application modules to insert, plug them into any port after the last memory module. For example, if you have a memory module inserted in port 1, you can insert application modules in any of ports 2, 3, or 4. **Never insert an application module into a lower numbered port than a memory module.** Be sure to place port covers over unused ports.
5. Turn the calculator on and follow the instructions given in this book for the desired application functions.

To remove application modules:

1. Turn the HP-41C off! Failure to do so could damage both the calculator and the module.
2. Grasp the desired module handle and pull it out as shown.

3. Place a port cap into the empty ports.

Mixing Memory Modules and Application Modules

Any time you wish to insert other extensions (such as the HP-82104A Card Reader, or the HP-82143 A Printer) the HP-41C has been designed so that the memory modules are in lower numbered ports.

So, when you are using both memory modules and application modules, the memory modules must always be inserted into the lower numbered ports and the application module into any port after the last memory module. When mixing memory and application modules, the HP-41C allows you to leave gaps in the port sequence. For example, you can plug a memory module into port 1 and an application module into port 4, leaving ports 2 and 3 empty.

FORMAT OF USER INSTRUCTIONS

The completed User Instruction Form—which accompanies each program—is your guide to operating the programs in this Pac.

The form is composed of five labeled columns. Reading from left to right, the first column, labeled STEP, gives the instruction step number.

The INSTRUCTIONS column gives instructions and comments concerning the operations to be performed.

The INPUT column specifies the input data, the units of data if applicable, or the appropriate alpha response to a prompted question. Data Input keys consist of 0 to 9 and the decimal point (the numeric keys), [EEX] (enter exponent), and [CHS] (change sign).

The FUNCTION column specifies the keys to be pressed after keying in the corresponding input data.

Whenever a statement in the INPUT or FUNCTION column is printed in gold, the [ALPHA] key must be pressed before the statement can be keyed in. After the statement is keyed in, press [ALPHA] again to return the calculator to its normal operating mode, or to begin program execution. For example, [XEQ] FOUR means press the following keys: [XEQ] [ALPHA] FOUR [ALPHA].

The DISPLAY column specifies prompts and intermediate and final answers and their units, where applicable.

Above the DISPLAY column is a box which specifies the minimum number of registers necessary to execute the program. Refer to pages 73 and 117 in the Owner's Handbook for a complete description of how to size calculator memory.

The following illustrates the User Instruction Form for *Fourier Series*.

				SIZE: 027
STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1.	Initialize program.		[XEQ] FOUR	NO. SAMPLES=?
2.	Key in number of samples in one period.	# samples	[R/S]	NO. FREQ=?
3.	Key in number of frequencies desired.	# freq.	[R/S]	1ST COEFF=?
4.	Key in order of first coefficient (J).	J	[R/S]	Y1=?
5.	Input y_n, n=1,...,N	y_n	[R/S]	Y2=?,...,RECT?
6.	Repeat step 5 until display shows RECT?			

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
7.	If the answer is yes, press [R/S] to display coefficients for J≤k≤J + # freqs. in rectangular form.		[R/S] [R/S]	a_k= b_k=
	If the answer is no (N), display coefficients in polar form.	N	[R/S] [R/S]	c_k= $\angle_k$=
	Pressing [R/S] displays successive coefficients.			
8.	To compute value of Fourier series at t, set USER mode and key in t.	t	[USER] [E]	f(t)

A WORD ABOUT PROGRAM USAGE

Catalog

When an Application Module is plugged into a port of the HP-41C, the contents of the Module can be reviewed by pressing [■] [CATALOG] 2 (the Extension Catalog). Executing the [CATALOG] function lists the name of each program or function in the module, as well as functions of any other extensions which might be plugged in.

Overlays

Overlays have been included for some of the programs in this pac. To run the program, choose the appropriate overlay, and place it on the calculator. The Mnemonics on the overlay are provided to help you run the program. The program's name is given vertically on the left side. Blue mnemonics are associated with the key they are directly below when the overlay is in place and the calculator is in USER mode. Gold mnemonics are similar to blue mnemonics, except that they are above the appropriate key and the shift (gold) key must be pressed before the re-defined key. Once again, USER mode must be set.

ALPHA and USER Mode Notation

This manual uses a special notation to signify ALPHA mode. Whenever a statement on the User Instruction Form is printed in gold, the [ALPHA] key must be pressed before the statement can be keyed in. After the statement is input, press [ALPHA] again to return the calculator to its normal operating mode, or to begin program execution. For example, [XEQ] FOUR means press the following keys: [XEQ] [ALPHA] FOUR [ALPHA].

In USER mode, when referring to the top two rows of keys (the keys have been re-defined), this manual will use the symbols [A] - [J] and [■] [A] - [■] [E] on the User Instruction Form and in the keystroke solutions to sample problems.

Using Optional Printer

When the optional printer is plugged into the HP-41C along with the Math Pac Applications Module, all results will be printed automatically. You may also want to keep a permanent record of the values input to a certain program. A convenient way to do this is to set the Print Mode switch to NORMAL before running the program. In this mode, all input values and the corresponding keystrokes will be listed on the printer, thus providing a record of the entire operation of the program.

Using Programs As Subroutines

The programs in the Math Pac may be called as subroutines for user programs in the HP-41C's program memory. Refer to Appendix B for information on special subroutine calling points.

Downloading Module Programs

If you wish to trace execution, to modify, to record on magnetic cards, or to print a program in this Application Module, it must first be copied into the HP-41C's program memory. For information concerning the HP-41C COPY function, see the Owner's Handbook. It is *not* necessary to copy a program in order to run it.

Program Interruption

These programs have been designed to operate properly when run from beginning to end, without turning the calculator off (remember, the calculator may turn itself off). If the HP-41C is turned off, it may be necessary to set flag 21 (SF 21) to continue proper execution.

Use of Labels

The user should be aware of possible problems when writing programs into calculator memory using Alpha labels identical with those in an Application Module.

MATRIX OPERATIONS

This program calculates the determinant and inverse of up to a 14 x 14 matrix, and gives the solution of a system of simultaneous equations in 14 unknowns.

The method used in this program is Gaussian elimination with partial pivoting. Space does not allow a full treatment of the pertinent equations, however, matrix theory references are provided.

Gaussian elimination is a series of row operations consisting of two parts: the forward elimination and the back solution. During forward elimination, the program takes an N x N matrix A and transforms it into an upper triangular matrix U, assuming A is nonsingular. The multipliers used to accomplish this transformation form a lower triangular matrix, L, which has 1's along its diagonal. If we disregard pivoting, a series of row interchanges that will increase accuracy for many systems of equations, the relationship among these matrices is U = LA. At the end of execution of this part of the program, the flag 4 annunciator turns off, and the original matrix A no longer exists in memory. The initial elements A_{ij} have been replaced by the elements of $U(i \leq j)$ and of $L(i > j)$. The back substitution part of the program uses the transformed matrices U and L to compute the determinant and inverse of A and to solve systems of simultaneous equations.

Equations: (using a matrix of order 5)

$$\text{Let } A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} \end{bmatrix}$$

The determinant of A, Det A, is found after its transformation to U by the product of diagonal elements:

$$\text{Det } A = (-1)^k U_{11} U_{22} U_{33} U_{44} U_{55},$$

where k is the number of row interchanges required by pivoting.

Let C be the inverse of A, i.e., the 5 x 5 matrix, such that AC = CA = I, where I is the 5 x 5 matrix such that

$$I_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad i, j = 1, 2, 3, 4, 5.$$

C is computed a column at a time in the following way:

let $\mathbf{c}^{(j)}$ be the jth column vector of C, i.e.,

$$\mathbf{c}^{(j)} = \begin{bmatrix} c_{1j} \\ c_{2j} \\ c_{3j} \\ c_{4j} \\ c_{5j} \end{bmatrix}, \quad j = 1, 2, 3, 4, 5$$

Then $\mathbf{c}^{(j)}$ is found by the solution of the equation

$$\mathbf{A}\mathbf{c}^{(j)} = \mathbf{I}^{(j)} \text{ where } \mathbf{I}^{(j)} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad i = 1, 2, 3, 4, 5$$

For example, $\mathbf{c}^{(1)}$ is found by solution of

$$\mathbf{A}\mathbf{c}^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

A set of 5 simultaneous equations in 5 unknowns may be written as

$$\begin{aligned}
A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + A_{14}x_4 + A_{15}x_5 &= B_1 \\
A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + A_{24}x_4 + A_{25}x_5 &= B_2 \\
A_{31}x_1 + A_{32}x_2 + A_{33}x_3 + A_{34}x_4 + A_{35}x_5 &= B_3 \\
A_{41}x_1 + A_{42}x_2 + A_{43}x_3 + A_{44}x_4 + A_{45}x_5 &= B_4 \\
A_{51}x_1 + A_{52}x_2 + A_{53}x_3 + A_{54}x_4 + A_{55}x_5 &= B_5,
\end{aligned}$$

where the $\{x_i\}$ are unknowns and the $\{B_i\}$ constants.

In matrix notation, this becomes $\mathbf{A}\mathbf{x} = \mathbf{B}$, where $\mathbf{x}$ and $\mathbf{B}$ are column vectors

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{bmatrix}$$

respectively.

This problem is solved (neglecting pivoting) as $\mathbf{U}\mathbf{x} = \mathbf{L}\mathbf{B}$.

Remarks:

- A halt during execution of the program with a display of ***NO SOLUTION*** indicates that the matrix is singular.

- The program is designed to solve a 14 × 14 matrix, however, memory modules will be required to go beyond a 6 × 6 matrix. Refer to the Matrix Register Chart for the register requirements of various matrices.
- The Math Module must be placed in a port after the Memory Module(s).
- If the elements of matrix A are already stored in the appropriate registers, the order is in register 14 (R_{14}), flag 04 is set, and flags 06-10 are cleared, the initial prompting may be skipped and the matrix pivoted by pressing [XEQ] PVT. To solve a system of equations, the column registers must be stored and flag 05 set.
- If DET is called as a subroutine, the value of the determinant will be returned to the Y-register.
- When keying in the elements of the matrix, the Y-register must not be disturbed.
- The best results will be obtained when the matrix is well-conditioned.

References:

Carnahan, Luther and Wilkes, *Applied Numerical Methods,* John Wiley and Sons, 1969.

George E. Forsythe, Michael A. Malcolm, and Cleve B. Moler, *Computer Methods in Mathematical Computation,* Computer Science Department, Stanford University, 1972.

G. Forsythe and C. Moler, *Computer Solution of Linear Algebraic Systems,* Prentice-Hall, 1967.

C. Moler, "Matrix Computations with Fortran and Paging," Comm. ACM, vol. 15, no. 4, pp. 268-270 (April, 1972).

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1	Initialize program		[XEQ] MATRIX	ORDER=?
2	Key in order (N) of matrix (N≤14)	N	[R/S]	SET SIZE nnn
3	Set size and continue		[XEQ] SIZE nnn [R/S]	A1, 1=?
4	Input elements of matrix in row order (A_{11}, A_{12}, A_{13}, …, A_{21}, A_{22}, A_{23},…, etc.)	A_{11} A_{12} ⋮	[R/S] [R/S] ⋮	A1, 2=? A1, 3=? ⋮
5	Repeat step 4 until all elements of matrix have been keyed in	A_{NN}	[R/S]	0.0000
6	(Optional) View the matrix		[XEQ] VMAT [R/S]	A1, 1= A1, 2=

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
			⋮	⋮
			[R/S]	AN, N=
7	(Optional) Edit the matrix		[XEQ] EDIT	ROW↑COL=?
7a	Input the row (I) and column (J) of element to be changed	I	[ENTER↑]	
		J	[R/S]	AI, J=?
7b	Key in new value	A_{IJ}	[R/S] *	AI, J=
			[R/S] *	ROW↑COL = ?
7c	Repeat 7a and 7b as needed			
7d	To stop editing		[R/S]	0.0000
8	Compute determinant		[XEQ] DET	DET=
9	Find the inverse (in column order)		[XEQ] INV	C1, 1=
			[R/S]	C2, 1=
			⋮	⋮
			[R/S]	CN, N=
10	For solution of simultaneous equations, input the column matrix		[XEQ] SIMEQ	B1=?
		B_1	[R/S]	B2=?
		⋮	⋮	⋮
		B_N	[R/S]	0.0000
11	Solve the system		[R/S]	X1=
			⋮	.
			[R/S]	XN=
11	(Optional) View the column		[XEQ] VCOL	B1=
			[R/S]	B2=
			⋮	
			[R/S]	BN=
	* An additional [R/S] is needed if the printer is not used			

Example 1:

Find the determinant and inverse of the matrix below.

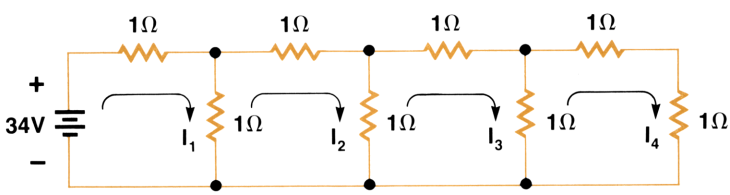
$$\begin{bmatrix} 6 & 3 & -2 & 2 & 3 \\ 1 & 4 & -3 & 4 & 2 \\ 2 & 3 & -1 & -2 & 9 \\ 4 & 3 & 0 & 2 & 1 \\ 3 & 5 & -6 & 6 & 2 \end{bmatrix}$$

Keystrokes:	Display:	
[XEQ] [ALPHA] MATRIX [ALPHA]	***ORDER=?***	
5 [R/S]	***SET SIZE 50***	(Assumes that less than 50 registers currently available)
[XEQ] [ALPHA] SIZE [ALPHA] 050		
[R/S]	***A1, 1=?***	
6 [R/S] 3 [R/S] 2 [CHS] [R/S] 2 [R/S]		
3 [R/S] 1 [R/S] 4 [R/S] 3 [CHS] [R/S]		
4 [R/S] 2 [R/S] 2 [R/S] 3 [R/S]		
1 [CHS] [R/S] 2 [CHS] [R/S] 9 [R/S]		
4 [R/S] 3 [R/S] 0 [R/S] 2 [R/S] 1 [R/S]		
3 [R/S] 5 [R/S] 6 [CHS] [R/S]		
6 [R/S] 2 [R/S]	***0.0000***	
[XEQ] [ALPHA] DET [ALPHA]	***DET=-200.0000***	(Det A)
[XEQ] [ALPHA] INV [ALPHA]	***C1, 1=0.2000***	
[R/S]	***C2, 1=-1.7500***	
[R/S]	***C3, 1=0.7000***	
[R/S]	***C4, 1=1.7250***	
[R/S]	***C5, 1=1.0000***	
[R/S]	***C1, 2=-0.1200***	
[R/S]	***C2, 2=-2.0000***	
[R/S]	***C3, 2=1.2800***	
[R/S]	***C4, 2=2.5400***	
[R/S]	***C5, 2=1.4000***	
[R/S]	***C1, 3=-0.0400***	
[R/S]	***C2, 3=0.5000***	
[R/S]	***C3, 3=-0.2400***	
[R/S]	***C4, 3=-0.5700***	
[R/S]	***C5, 3=-0.2000***	

Keystrokes	Display
R/S	C1, 4=-6.0000 E-10
R/S	C2, 4=1.7500
R/S	C3, 4=-0.5000
R/S	C4, 4=-1.6250
R/S	C5, 4=-1.0000
R/S	C1, 5=-1.0000 E-10
R/S	C2, 5=1.5000
R/S	C3, 5=-1.0000
R/S	C4, 5=-1.7500
R/S	C5, 5=-1.0000

Example 2:

By applying the technique of loop currents to the circuit below, find the currents I_1, I_2, I_3, and I_4.



The equations to be solved are

$$
\begin{aligned}
2I_1 \quad -I_2 \quad\quad\quad\quad\quad\quad &= 34 \\
-I_1 \quad +3I_2 \quad -I_3 \quad\quad\quad &= 0 \\
-I_2 \quad +3I_3 \quad -I_4 &= 0 \\
-I_3 \quad +3I_4 &= 0
\end{aligned}
$$

In matrix form,

$$
\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 3 \end{bmatrix}
\begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix}
=
\begin{bmatrix} 34 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

Keystrokes:	**Display:**	
[XEQ] [ALPHA] MATRIX [ALPHA]	***ORDER=?***	
4 [R/S]	***A1,1=?***	(Assumes size still set at 050)
2 [R/S] 1 [CHS] [R/S] 0 [R/S] 0 [R/S]		
1 [CHS] [R/S] 3 [R/S] 1 [CHS] [R/S]		
0 [R/S] 0 [R/S] 1 [CHS] [R/S] 3 [R/S]		
1 [CHS] [R/S] 0 [R/S] 0 [R/S]		
1 [CHS] [R/S] 3 [R/S]	***0.0000***	
[XEQ] [ALPHA] SIMEQ [ALPHA]	***B1=?***	
34 [R/S] 0 [R/S] 0 [R/S] 0 [R/S]	***0.0000***	
[R/S]	***X1=21.0000***	
[R/S]	***X2=8.0000***	
[R/S]	***X3=3.0000***	
[R/S]	***X4=1.0000***	

SOLUTION TO f(x) = 0 ON AN INTERVAL

This program uses a modification of the secant iteration algorithm to find a real root of the equation $f(x) = 0$. The user must supply the function to be solved, and may supply two initial guesses (x_1 and x_2) intended to approximate the desired solution. If an initial interval is not specified, the program assumes initial values of 1 and 10.

If $f(x_1) \cdot f(x_2) \leq 0$ and the function is continuous on the interval, the program will always find a root. If $f(x_1) \cdot f(x_2) > 0$, the search for a root may not be successful. When more than one root exists in an interval, one root will be found, and the user may then choose a smaller interval and repeat the program.

The function f(x) may be keyed into program memory using any **global** label (maximum of 6 characters), and should assume that x will be in the X-register upon entry. Several functions may be loaded into program memory at the same time, as the program prompts the user for the name of the function to be evaluated. The program uses registers 00-06. The remaining registers and the stack are available for defining f(x).

				SIZE: 007
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1.	Prepare to key in function(s).		[GTO] [•] [•]	
2.	Switch to PRGM mode; load function under desired label; add [RTN]; switch out of PRGM mode.		[PRGM] [LBL] _ : [RTN] [PRGM]	
3.	Initialize program.		[XEQ] SOLVE	FUNCTION NAME?
4.	Key in the name of the function (ALPHA mode is set at step 3).	Name	[R/S]	GUESS 1=?
5.	If you wish to provide two initial guesses, key in first guess. Otherwise, go to step 7.	x_1	[R/S]	GUESS 2=?
6.	Key in second guess.	x_2	[R/S]	(See Note)
7.	Execute program.		[R/S]	(See Note)
8.	To search for another root, press [R/S]. For a new case, go to step 1 or 3. **NOTE:** There are three ways for this program to terminate normally. They result in the following messages: 1) NO ROOT FOUND 2) ROOT IS <VALUE> 3) ROOT IS BETWEEN <Value 1> AND <Value 2>			
9.	To use as a subroutine (when the initial guesses are already stored), steps 3-9 could be skipped by pressing [XEQ] SOL.			

Example 1:

Find the root of $\ln x + 3x - 10.8074 = 0$. Use [LBL] FF to define f(x).

Keystrokes:	Display:
[XEQ] [ALPHA] SIZE [ALPHA] 007	
[GTO] [•] [•]	
[PRGM]	
[LBL] [ALPHA] FF [ALPHA]	
[LN]	
[LASTx]	
3 [×] [+]	
10.8074 [−]	
[RTN]	
[PRGM]	
[XEQ] [ALPHA] SOLVE [ALPHA]	*FUNCTION NAME?*
FF [R/S]	*GUESS 1 =?*
[R/S]	*ROOT IS 3.2134*

Example 2:

Find an angle α between 100 and 101 radians such that $\sin\alpha = 0.01$. Hence, let $f(x) = \sin x - 0.01$ and use [LBL] ANGLE.

Keystrokes:	Display:
[GTO] [•] [•]	
[PRGM]	
[LBL] [ALPHA] ANGLE [ALPHA]	
[XEQ] [ALPHA] RAD [ALPHA]	
[SIN]	
.01 [−]	
[XEQ] [ALPHA] DEG [ALPHA]	
[RTN]	
[PRGM]	
[XEQ] [ALPHA] SOLVE [ALPHA]	*FUNCTION NAME?*
ANGLE [R/S]	*GUESS 1 =?*
100 [R/S]	*GUESS 2 =?*
101 [R/S]	*ROOT IS BETWEEN 100.5410*
[R/S]	*AND 100.5410*

To see the answers to more significant digits, press [FIX] 9 and [x≷y].

Example 3:

Find the roots of $x^2 + 1 = 0$ using [LBL] CC.

Keystrokes:	**Display:**
[GTO] [•] [•]	
[PRGM]	
[LBL] [ALPHA] CC [ALPHA]	
[x^2] 1 [+]	
[RTN]	
[PRGM]	
[XEQ] [ALPHA] SOLVE [ALPHA]	*FUNCTION NAME?*
CC [R/S]	*GUESS 1 =?*
[R/S]	*NO ROOT FOUND*

POLYNOMIAL SOLUTIONS/EVALUATION

This program may be used to find the roots of a polynomial with real coefficients of degree 5 and below, provided that the high-order coefficient is 1. The equation may be represented as

$$x^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0 = 0, \qquad n = 2, 3, 4 \text{ or } 5.$$

If the leading coefficient is not 1, it should be made 1 by dividing the entire equation by that coefficient.

Polynomials may also be evaluated for arbitrary values of x. This is of aid in plotting polynomials and using data correlations based on polynomials.

When the program is initialized, the user must specify the degree (n) of the polynomial. The calculator then prompts the user for the coefficients a_{n-1}, ··· a_1, a_0. Zero must be input for those coefficients which are zero. Registers 00–04 are used to store the coefficients.

Equations:

The routines for third and fifth degree equations use an iterative routine to find one real root of the equation. This routine requires that the constant term a_0 not be zero for these equations. (If $a_0 = 0$, then zero is a real root and by factoring out x, the equation may be reduced by one order.) After one root is found, synthetic division is performed to reduce the original equation to a second or fourth degree equation.

To solve a fourth degree equation, it is first necessary to solve the cubic equation

$$y^3 + b_2y^2 + b_1y + b_0 = 0$$

where $b_2 = -a_2$

$b_1 = a_3a_1 - 4a_0$

$b_0 = a_0(4a_2 - a_3^2) - a_1^2.$

Let y_0 be the largest real root of the above cubic.

Then the fourth degree equation is reduced to two quadratic equations:

$$x^2 + (A + C)x + (B + D) = 0$$

$$x^2 + (A - C)x + (B - D) = 0$$

where $A = \frac{a_3}{2}$, $B = \frac{y_0}{2}$, $D = \sqrt{B^2 - a_0}$, $C = \sqrt{A^2 - a_2 + y_0}$

Roots of the fourth degree equation are found by solving the two quadratic equations.

A quadratic equation $x^2 + a_1x + a_0 = 0$ is solved by the formula $x_{1,2} = -\frac{a_1}{2} \pm \sqrt{\frac{a_1^2}{4} - a_0}$. If $D = \frac{a_1^2}{4} - a_0 > 0$, the roots are real; if $D < 0$, the roots are complex, being $u \pm iv = -\frac{a_1}{2} \pm i\sqrt{-D}$.

A real root is output as a single number. Complex roots always occur in pairs of the form $u \pm iv$, and are labeled in the output.

Remarks:

- Long execution times may be expected for equations of degree 3, 4, or 5, as these use an interative routine once or more.
- Program uses registers 00-22.

				SIZE: 023
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1.	Initialize program.		XEQ POLY	DEGREE=?
2.	Key in degree of polynomial (n=2,3,4,5)	n	R/S	a(n−1)=?
3.	Input coefficient a_{n-1} of polynomial. Coefficients=0 must be so input. Repeat until display asks for a_0.	a_{n-1}	R/S	a(n−2)=?
		. . .		
		a_1	R/S	a0=?
4.	Input coefficient a_0.	a_0	R/S	ROOTS?
5.	To find the roots of the polynomial, press R/S to see successive roots appropriately labeled. Go to step 9.		R/S R/S R/S R/S R/S	ROOT= U= V= U= −V=
6.	To evaluate the polynomial, answer no (N)	N	R/S	X=?
7.	Input x and see f(x).	x	R/S	F<X>=
8.	For a new x, key in x, press R/S .	x	R/S	F<X>=
9.	For a new polynomial of same degree, go to step 1 or change appropriate coefficients (registers 00-04) and XEQ ROOTS.			

Example 1:

Find the roots of $x^5 - x^4 - 101x^3 + 101x^2 + 100x - 100 = 0$.

Keystrokes:	Display:	
[XEQ] [ALPHA] SIZE [ALPHA] 023		
[XEQ] [ALPHA] POLY [ALPHA]	DEGREE=?	
5 [R/S]	a4=?	
1 [CHS] [R/S]	a3=?	
101 [CHS] [R/S]	a2=?	
101 [R/S]	a1=?	
100 [R/S]	a0=?	
100 [CHS] [R/S]	ROOTS?	
[R/S]	ROOT=10.0000	(Root 1)
[R/S]	ROOT=1.0000	(Root 2)
[R/S]	ROOT=1.0000	(Root 3)
[R/S]	ROOT=-1.0000	(Root 4)
[R/S]	ROOT=-10.0000	(Root 5)

Example 2:

Solve $4x^4 - 8x^3 - 13x^2 - 10x + 22 = 0$.

Rewrite the equation as $x^4 - 2x^3 - \frac{13}{4}x^2 - \frac{10}{4}x + \frac{22}{4} = 0$.

Keystrokes:	Display:	
[XEQ] [ALPHA] POLY [ALPHA]	DEGREE=?	
4 [R/S]	a3=?	
2 [CHS] [R/S]	a2=?	
13 [ENTER↑] 4 [÷] [CHS] [R/S]	a1=?	
10 [ENTER↑] 4 [÷] [CHS] [R/S]	a0=?	
22 [ENTER↑] 4 [÷] [R/S]	ROOTS?	
[R/S]	U=-1.0000	
[R/S]	V=1.0000	(Roots 1 & 2 are
[R/S]	U=-1.0000	-1.00 ± 1.00i)
[R/S]	-V=-1.0000	
[R/S]	ROOT=3.1180	(Root 3)
[R/S]	ROOT=0.8820	(Root 4)

Example 3:

In the previous example, what would be the roots if the x^2 coefficient were changed from −13/4 to −5?

Keystrokes:	Display:	
5 [CHS] [STO] 02		
[XEQ] [ALPHA] ROOTS [ALPHA]	*U=-1.1386*	(Roots 1 & 2 are
[R/S]	*V=0.8555*	-1.1386±.8555i)
[R/S]	*U=-1.1386*	
[R/S]	*-V=-0.8555*	
[R/S]	*ROOT=3.5031*	(Root 3)
[R/S]	*ROOT=0.7741*	(Root 4)

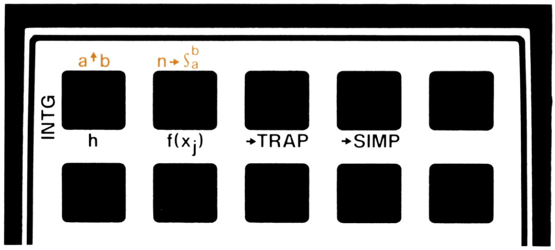
Example 4:

Evaluate the following polynomial at x = 2.5 and x = −5.

$$f(x) = x^5 + 5x^4 - 3x^2 - 7x + 11$$

Keystrokes:	Display:
[XEQ] [ALPHA] POLY [ALPHA]	*DEGREE=?*
5 [R/S]	*a4=?*
5 [R/S]	*a3=?*
0 [R/S]	*a2=?*
3 [CHS] [R/S]	*a1=?*
7 [CHS] [R/S]	*a0=?*
11 [R/S]	*ROOTS?*
N [R/S]	*X=?*
2.5 [R/S]	*F<X>=267.7188*
5 [CHS] [R/S]	*F<X>=-29.0000*

NUMERICAL INTEGRATION



This program will perform numerical integration whether a function is known explicitly or only at a finite number of equally spaced points (discrete case). The integrals of explicit functions are found using Simpson's rule; discrete case integrals may be approximated by either the trapezoidal rule or Simpson's rule.

Discrete case

Let $x_0, x_1, \ldots, x_n$ be n equally spaced points ($x_j = x_0 + jh$, $j = 1, 2, \ldots, n$) at which corresponding values $f(x_0), f(x_1), \ldots, f(x_n)$ of the function f(x) are known. The function itself need not be known explicitly. After input of the step size h and the values of $f(x_j)$, $j = 0, 1, \ldots, n$, then the integral

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

may be approximated using

1. The trapezoidal rule:

$$\int_{x_0}^{x_n} f(x)\,dx \simeq \frac{h}{2}\left[f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n)\right]$$

2. Simpson' rule:

$$\int_{x_0}^{x_n} f(x)\,dx \simeq \frac{h}{3}\Big[f(x_0) + 4f(x_1) + 2f(x_2) +$$

$$\ldots + 4f(x_{n-3}) + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\Big]$$

In order to apply Simpson's rule, n must be even. If n is not even, the calculator will halt displaying ***N NOT EVEN*** if [D] is pressed.

Explicit Functions

If an explicit formula is known for the function f(x), then the function may be keyed into program memory and numerically integrated by Simpson's rule. The user must specify the endpoints a and b of the interval over which integration is to be performed, and the number of subintervals n into which the interval (a, b) is to be divided. This n must be even; if it is not, ***N NOT EVEN*** will be displayed. The program will go on to compute

$$x_0 = a,\ x_j = x_0 + jh,\ j = 1, 2, \ldots, n - 1, \text{ and } x_n = b \text{ where}$$

$$h = \frac{b - a}{n}$$

The integral $\int_a^b f(x)\,dx$ is approximated by equation (2) above, Simpson's rule.

The function f(x) may be keyed into program memory using any **global** label (maximum of 6 characters), and should assume that x will be in the X-register upon entry. Several functions may be loaded into program memory at the same time, as the program prompts the user for the name of the function to be evaluated. The program uses Registers 00-07; the remaining registers are available for defining $f_i(x)$.

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: 008 DISPLAY
1.	Place overlay on calculator.			
2.	For explicit functions, go to step 9, for discrete case, go to step 3.			
	DISCRETE CASE			
3.	Initialize program		[XEQ] INTG	0.0000
4.	Key in the spacing between x-values.	h	[A]	h
5.	Key in the function value at x_j. Repeat this step for j=0,1,...,n.	$f(x_j)$	[B]	j
6.	Compute the area by the trapezoidal rule.		[C]	TRAP∫
7.	Compute the area by Simpson's rule (n must be even).		[D]	SIMP∫
8.	For a new case, go to step 2.			
	EXPLICIT FUNCTIONS			
9.	Prepare to load function.		[GTO] [•] [•]	

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
10.	Switch to PRGM mode; load function under desired label; add [RTN]; switch out of PRGM mode.		[PRGM] [LBL] _ ⋮ [RTN] [PRGM]	
11.	Initialize program.		[XEQ] INTG	
12.	Key in the beginning and final endpoints of the integration interval.	a b	[ENTER↑] [ ] [A]	 a
13.	Key in the number of subintervals (must be even), and compute the area by Simpson's rule.	n	[ ] [B]	FUNCTION NAME?
14.	Key in name of function.	Name	[R/S]	$\int_a^b f_i(x)dx$
15.	To change a, b, or n, go to the appropriate step; for a new case, go to step 2.			

Example 1:

Given the values below for $f(x_j)$, $j = 0, 1, \ldots, 8$, compute the approximations to the integral

$$\int_0^2 f(x)\,dx$$

by the trapezoidal rule and by Simpson's rule.

The value for h is 0.25.

i	0	1	2	3	4	5	6	7	8
x_i	0	.25	.5	.75	1	1.25	1.5	1.75	2
$f(x_i)$	2	2.8	3.8	5.2	7	9.2	12.1	15.6	20

Keystrokes:	Display:
[XEQ] [ALPHA] SIZE [ALPHA] 008	
[XEQ] [ALPHA] INTG [ALPHA]	***0.0000***

Keystrokes	Display	
.25 [A] 2 [B]		
2.8 [B] 3.8 [B]		
5.2 [B] 7 [B]		
9.2 [B] 12.1 [B]		
15.6 [B] 20 [B]		
[C]	*16.6750*	(Trapezoidal)
[D]	*16.5833*	(Simpson's)

Example 2:

Find the value of

$$\int_0^{2\pi} \frac{dx}{1 - \cos x + 0.25}$$

for n = 10 and then for n = 30. Note that x is assumed to be in radians. For safety, if you work mostly in degrees, it is good programming practice to set the angular mode to radians at the beginning of the routine, then back to degrees at the end. Key the function in under [LBL] FF.

Keystrokes	Display	
[GTO] [•] [•]		
[PRGM]		
[LBL] [ALPHA] FF [ALPHA]		
[XEQ] [ALPHA] RAD [ALPHA]		
[COS]		
1 [x≷y] [−]		
.25 [+] [1/X]		
[XEQ] [ALPHA] DEG [ALPHA]		
[RTN]		
[PRGM]		
[XEQ] [ALPHA] INTG [ALPHA]		
0 [ENTER↑] 2 [π] [×] [■] [A]		
10 [■] [B]	*FUNCTION NAME?*	
FF [R/S]	*8.2193*	(n=10)
30 [■] [B]	*FUNCTION NAME?*	
FF [R/S]	*8.3774*	(n=30)

The exact solution is $\frac{8\pi}{3} = 8.3776$

DIFFERENTIAL EQUATIONS

This program solves first- and second-order differential equations by the fourth-order Runge-Kutta method. A first-order equation is of the form $y' = f(x, y)$, with initial values x_0, y_0; a second-order equation is of the form $y'' = f(x, y, y')$, with initial values x_0, y_0, y_0'.

In either case, the function f(x) may be keyed into program memory using any **global** label (maximum of 6 characters), and should assume that x and y are in the X- and Y-registers respectively; y' will be in the Z-register for second-order equations. The Module program uses registers 00-07. The remaining registers are available for defining the function.

The solution is a numerical solution which calculates y_i for $x_i = x_0 + ih$ ($i = 1, 2, 3, \ldots$), where h is an increment specified by the user. The value for h may be changed at any time during the program's execution by storing h/2 in Register 01. This allows solution of the equation arbitrarily close to a pole ($y \rightarrow \pm\infty$).

Equations:

1st-order:

$$y_{i+1} = y_i + \frac{1}{6}(c_1 + 2c_2 + 2c_3 + c_4)$$

where

$$c_1 = hf(x_i, y_i)$$

$$c_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{c_1}{2}\right)$$

$$c_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{c_2}{2}\right)$$

$$c_4 = hf(x_i + h, y_i + c_3)$$

2nd-order:

$$y_{i+1} = y_i + h\left[y_i' + \frac{1}{6}(k_1 + k_2 + k_3)\right]$$

$$y'_{i+1} = y_i' + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf\,(x_i,\ y_i,\ y_i')$$

$$k_2 = hf\left(x_i + \frac{h}{2},\ y_i + \frac{h}{2}\,y_i' + \frac{h}{8}k_1,\ y_i' + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x_i + \frac{h}{2},\ y_i + \frac{h}{2}\,y_i' + \frac{h}{8}k_1,\ y_i' + \frac{k_2}{2}\right)$$

$$k_4 = hf\left(x_i + h,\ y_i + hy_i' + \frac{h}{2}\,k_3,\ y_i' + k_3\right)$$

Remarks:

- When inputting values for a second-order solution, the values for x_0 and y_0 must be input before the value of y_0'. All values must be input even if zero.

				SIZE: 008
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1.	Prepare to load function f(x, y, y′).		[GTO] [•] [•]	
2.	Switch to PRGM mode; load function under desired label; add [RTN]; switch out of PRGM mode.		[PRGM] [LBL] _ ⋮ [RTN] [PRGM]	
3.	Initialize program.		[XEQ] DIFEQ	FUNCTION NAME?
4.	Key in name of function.	Name	[R/S]	ORDER=?
5.	Key in the order of the differential equation (1 or 2)	1 or 2	[R/S]	STEP SIZE=?
6.	Key in step size (h).	h	[R/S]	X0=?
7.	Input initial value for x.	x_0	[R/S]	Y0=?
8.	Input initial value for y.	y_0	[R/S]	x_1 or Y0.=?
9.	For a second-order solution, key in initial value of y′.	y_0'	[R/S]	x_1
10.	Output successive values of x and y.		[R/S] [R/S] [R/S]	y_1 x_2 y_2 etc.

Example 1:

Using [LBL] FX, solve numerically the first-order differential equation

$$y' = \frac{\sin x + \tan^{-1}(y/x)}{y - \ln(\sqrt{x^2 + y^2})}$$

where $x_0 = y_0 = 1$. Let $h = 0.5$. The angular mode must be set to radians, and three additional storage registers are necessary to define the function.

Keystrokes:	Display:	
[XEQ] [ALPHA] SIZE [ALPHA] 011		
[GTO] [•] [•]		
[PRGM]		
[LBL] [ALPHA] FX [ALPHA]		
[XEQ] [ALPHA] RAD [ALPHA]		
[STO] 08		
[x≷y]		
[STO] 09		
[x≷y]		
[R-P]		
[LN]		
[STO] 10		
[R↓]		
[RCL] 08		
[SIN]		
[+]		
[RCL] 09		
[RCL] 10		
[−]		
[÷]		
[XEQ] [ALPHA] DEG [ALPHA]		
[RTN]		
[PRGM]		
[XEQ] [ALPHA] DIFEQ [ALPHA]	*FUNCTION NAME?*	
FX [R/S]	*ORDER=?*	
1 [R/S]	*STEP SIZE=?*	
.5 [R/S]	*XO=?*	
1 [R/S]	*YO=?*	
1 [R/S]	*1.5000*	(x_1)
[R/S]	*2.0553*	(y_1)
[R/S]	*2.0000*	(x_2)

Keystrokes	Display	
[R/S]	*2.7780*	(y_2)
[R/S]	*2.5000*	(x_3)
[R/S]	*3.2781*	(y_3)
	etc.	

Example 2:

Using [LBL] DIF, solve the second-order equation

$$(1 - x^2)y'' + xy' = x$$

where $x_0 = y_0 = y_0' = 0$ and $h = 0.1$.

Rewrite the equation as:

$$y'' = \frac{x(1 - y')}{1 - x^2} = \frac{x(y' - 1)}{x^2 - 1} \qquad x \neq 1$$

Keystrokes	Display	
[GTO] [•] [•]		
[PRGM]		
[LBL] [ALPHA] DIF [ALPHA]		
[STO] 08		
[R↓] [R↓]		
1 [−]		
[RCL] 08		
[×]		
[LASTX]		
[x²]		
1 [−] [÷]		
[RTN]		
[PRGM]		
[XEQ] [ALPHA] DIFEQ [ALPHA]	*FUNCTION NAME?*	
DIF [R/S]	*ORDER=?*	
2 [R/S]	*STEP SIZE=?*	
.1 [R/S]	*XO=?*	
0 [R/S]	*YO=?*	
0 [R/S]	*YO.=?*	
0 [R/S]	*0.1000*	(x_1)

Keystrokes:	**Display:**	
[R/S]	*0.0002*	(y_1)
[R/S]	*0.2000*	(x_2)
[R/S]	*0.0013*	(y_2)
[R/S]	*0.3000*	(x_3)
[R/S]	*0.0046*	(y_3)
[R/S]	*0.4000*	(x_4)
[R/S]	*0.0109*	(y_4)
	etc.	

FOURIER SERIES

Any periodic function may be written as a series of sines and cosines by the application of the following formulas.

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos \frac{2\pi tk}{T} + b_k \sin \frac{2\pi tk}{T} \right)$$

$$= \frac{a_0}{2} + \sum_{k=1}^{\infty} c_k \cos \left(\frac{2\pi tk}{T} - \theta_k \right)$$

$$a_k = \frac{2}{T} \int_0^T f(t) \cos \frac{2\pi tk}{T} \, dt, k = 0, 1, 2, \ldots$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin \frac{2\pi tk}{T} \, dt, k = 1, 2, \ldots$$

$$c_k = (a_k^2 + b_k^2)^{1/2}$$

$$\theta_k = \tan^{-1} \left(\frac{b_k}{a_k} \right)$$

$$T = \text{period of } f(t)$$

This program computes the Fourier coefficients from discrete versions of the above formulas given a large enough number of samples of the periodic function. Up to ten consecutive pairs of coefficients may be computed at one time from N equally spaced points. The coefficients may be displayed in either rectangular or polar form.

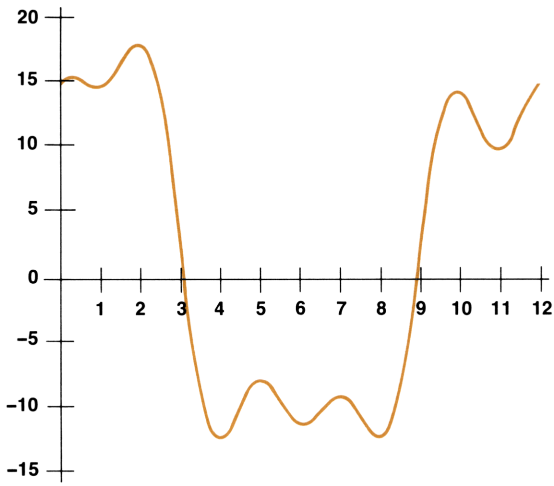
The value of N should be chosen to be more than twice the highest expected multiple of the fundamental frequency present in the function to be analyzed. A low estimate for N will cause energy above one-half the sampling rate to appear at a lower frequency (a phenomenon known as aliasing).

Registers 00–26 are used by the program.

				SIZE: 027
STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
1.	Initialize program.		[XEQ] FOUR	NO. SAMPLES=?
2.	Key in number of samples in one period.	# samples	[R/S]	NO. FREQ=?
3.	Key in number of frequencies desired.	# freq.	[R/S]	1ST COEFF=?
4.	Key in order of first coefficient (J)	J	[R/S]	Y1=?
5.	Input y_n, n=1,...,N	y_n	[R/S]	Y2=?,...,RECT?
6.	Repeat step 5 until display shows RECT?			
7.	If the answer is yes, press [R/S] to display coefficients for $J \leq k \leq J + \#freqs.$ in rectangular form.		[R/S] [R/S]	a_k b_k
	If the answer is no (N), display coefficients in polar form.	N	[R/S] [R/S]	c_k = $\angle_k$=
	Pressing [R/S] displays successive coefficients.			
8.	To compute value of Fourier series at t, set USER mode and key in t.	t	[USER] [E]	f(t)

Example:

Compute a discrete Fourier series representation for the waveform shown. Since there are 12 samples, select 7 frequencies (dc term plus 6 harmonics). Display coefficients in rectangular form.



t	f(t)
1	14.758
2	17.732
3	2
4	−12.
5	− 7.758
6	−11
7	− 9.026
8	−12.
9	2
10	14.268
11	10.026
12	15

Keystrokes:	**Display:**
[XEQ] [ALPHA] SIZE [ALPHA] 027	
[XEQ] [ALPHA] FOUR [ALPHA]	*NO. SAMPLES=?*
12 [R/S]	*NO. FREQ=?*
7 [R/S]	*1ST COEFF=?*
0 [R/S]	*Y1=?*

Keystrokes	Display
14.758 [R/S]	*Y2=?*
17.732 [R/S]	*Y3=?*
2 [R/S]	*Y4=?*
12 [CHS] [R/S]	*Y5=?*
7.758 [CHS] [R/S]	*Y6=?*
11 [CHS] [R/S]	*Y7=?*
9.026 [CHS] [R/S]	*Y8=?*
12 [CHS] [R/S]	*Y9=?*
2 [R/S]	*Y10=?*
14.268 [R/S]	*Y11=?*
10.026 [R/S]	*Y12=?*
15 [R/S]	*RECT?*
[R/S]	*a0=4.0000*
[R/S]	*b0=0.0000*
[R/S]	*a1=14.9998*
[R/S]	*b1=1.0000*
[R/S]	*a2=3.0000E-8*
[R/S]	*b2=1.0000*
[R/S]	*a3=-5.0000*
[R/S]	*b3=1.0000*
[R/S]	*a4=3.3333E-9*
[R/S]	*b4=3.2000E-9*
[R/S]	*a5=3.0002*
[R/S]	*b5=1.4673E-5*
[R/S]	*a6=0.0000*
[R/S]	*b6=2.3599E-8*

Thus $f(t) = 2 + 15\cos\frac{2\pi t}{12} + \sin\frac{2\pi t}{12}$

$$+ \sin\frac{4\pi t}{12}$$

$$- 5\cos\frac{6\pi t}{12} + \sin\frac{6\pi t}{12}$$

$$+ 3\cos\frac{10\pi t}{12}$$

COMPLEX OPERATIONS

This collection of programs allows for chained calculations involving complex numbers in rectangular form. The four operations of complex arithmetic (+, −, ×, ÷) are provided, as well as several of the most used functions of complex variables z and w ($|z|$, $1/z$, z^n, $z^{1/n}$, e^z, $\ln z$, $\sin z$, $\cos z$, $\tan z$, a^z, $\log_a z$, $z^{1/w}$ and z^w). Functions and operations may be mixed in the course of a calculation to allow evaluation of expressions like $z_3 / (z_1 + z_2)$, $e^{z_1 z_2}$, $|z_1 + z_2| + |z_2 - z_3|$, etc., where z_1, z_2, z_3 are complex numbers of the form $x + iy$.

For repeated use of these operations, the user might wish to reassign the individual programs to selected keys on the calculator, and create an appropriate overlay. One reasonable key reassignment might include:

[ASN] SINZ [SIN]
[ASN] LNZ [LN]
[ASN] C+ [+]
[ASN] C− [−]
[ASN] CINV [1/x]

The logic system for these programs may be thought of as a kind of Reverse Polish Notation (RPN) with a stack whose capacity is two complex numbers. Let the bottom register of the complex stack be ξ and the top register τ. These are analogous to the X- and T-registers in the calculator's own four-register stack.* A complex number z_1 is input to the ξ-register by the keystrokes y_1 [ENTER↑] x_1. Upon input of a second complex number z_2 ([ENTER↑] y_2 [ENTER↑] x_2), z_1 is moved to τ and z_2 is placed in ξ. The previous contents of τ are lost.

Functions operate on the ξ-register, and the result (except for $|z|$ which returns a real number) is left in ξ. Arithmetic operations involve both the ξ- and τ-registers; the result of the operation is left in ξ.

The Application Module program uses registers 00–04.

Equations:

Let

$$z_k = x_k + iy_k = r_k e^{i\theta_k}, k = 1, 2$$

$$z = x + iy = re^{i\theta}$$

* Each register of the complex stack must actually hold two real numbers: the real and the imaginary part of its complex contents. Thus it takes two of the calculator registers to represent one register in the complex stack. We will speak of the complex stack registers as though they were each just one register.

Let the result in each case be u + iv.

$$z_1 + z_2 = (x_1 + x_2) + i(y_1 + y_2)$$

$$z_1 - z_2 = (x_1 - x_2) + i(y_1 - y_2)$$

$$z_1 z_2 = r_1 r_2 e^{i(\theta_1 + \theta_2)}$$

$$z_1/z_2 = \frac{r_1}{r_2} e^{i(\theta_1 - \theta_2)}$$

$$|z| = \sqrt{x^2 + y^2}$$

$$1/z = \frac{x}{r^2} - i\frac{y}{r^2}$$

$$z^n = r^n e^{in\theta}$$

$$z^{1/n} = r^{1/n} e^{i\left(\frac{\theta}{n} + \frac{360k}{n}\right)}, k = 0, 1, \ldots, n-1$$

(All n roots will be output, k = 0, 1, ..., n−1.)

$$e^z = e^x (\cos y + i\sin y)\text{, where y is in radians}$$

$$\ln z = \ln r + i\theta\text{, where } z \neq 0$$

$$a^z = e^{z\ln a}\text{, where } a > 0 \text{ and real}$$

$$\log_a z = \frac{\ln z}{\ln a}\text{, where } a > 0 \text{ and real, } z \neq 0$$

$$z^w = e^{w\ln z}\text{, where } z \neq 0\text{, w is complex}$$

$$z^{1/w} = e^{\ln z/w}\text{, where } z \neq 0\text{, w is complex and } w \neq 0$$

$$\sin z = \sin x\cosh y + i\cos x\sinh y\text{, angles in radians}$$

$$\cos z = \cos x\cosh y - i\sin x\sinh y\text{, angles in radians}$$

$$\tan z = \frac{\sin 2x + \sinh 2y}{\cos 2x + \cosh 2y}\text{, angles in radians}$$

SIZE: 005

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	ARITHMETIC			
1.	Key in first complex number ($x_1 + iy_1$)	y_1 x_1	[ENTER↑] [ENTER↑]	
2.	Key in second complex number ($x_2 + iy_2$)	y_2 x_2	[ENTER↑]	
3.	Select one of four operations:			
	• Add (+)		[XEQ] C+ [R/S]	U= V=
	• Subtract (−)		[XEQ] C− [R/S]	U= V=
	• Multiply (×)		[XEQ] C× [R/S]	U= V=
	• Divide (÷)		[XEQ] C÷ [R/S]	U= V=
4.	The result of the operation is in the stack; go to step 5 for a function or to step 2 for further arithmetic.			
	FUNCTIONS			
5.	Select one of these functions:			
	• Magnitude (|z|)	y_1 x_1	[ENTER↑] [XEQ] MAGZ	 R=
	• Reciprocal (1/z)	y_1 x_1	[ENTER↑] [XEQ] CINV [R/S]	 U= V=
	• Raise z to an integer power (z^n)	y_1 x_1 n	[ENTER↑] [ENTER↑] [XEQ] Z↑N [R/S]	 U= V=
	• Find the n^{th} root of z ($z^{1/n}$) **NOTE:** n roots (u+iv) will be found.	y_1 x_1 n	[ENTER↑] [ENTER↑] [XEQ] Z↑1/N [R/S]	 U= V=
	• Raise e to the power z (e^z)	y_1 x_1	[ENTER↑] [XEQ] e↑Z [R/S]	 U= V=
	• Natural logarithm of z (lnz)	y_1 x_1	[ENTER↑] [XEQ] LNZ [R/S]	 U= V=
	• Raise real number to the power z (a^z)	y_1 x_1 a	[ENTER↑] [ENTER↑] [XEQ] a↑Z [R/S]	 U= V=

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
	• Logarithm base a of z ($\log_a z$)	y_1	ENTER↑	
		x_1	ENTER↑	
		a	XEQ LOGZ	U=
			R/S	V=
	• Raise z to complex power $w=x_2 + iy_2$ (z^w)	y_2	ENTER↑	
		x_2	ENTER↑	
		y_1	ENTER↑	
		x_1	XEQ Z↑W	U=
			R/S	V=
	• Find the w^{th} complex root of z ($z^{1/w}$)	y_2	ENTER↑	
		x_2	ENTER↑	
		y_1	ENTER↑	
		x_1	XEQ Z↑1/W	U=
			R/S	V=
	• Find sin z	y_1	ENTER↑	
		x_1	XEQ SINZ	U=
			R/S	V=
	• Find cos z	y_1	ENTER↑	
		x_1	XEQ COSZ	U=
			R/S	V=
	• Find tan z	y_1	ENTER↑	
		x_1	XEQ TANZ	U=
			R/S	V=
6.	Go to step 2 for arithmetic or to step 5 for another function.			

Example 1:

Evaluate the expression

$$\frac{z_1}{z_2 + z_3},$$

where $z_1 = 23 + 13i$, $z_2 = -2 + i$, $z_3 = 4 - 3i$.

(Suggestion: since the program can remember only two numbers at a time, perform the calculation as

$$z_1 \times \left[1/(z_2 + z_3)\right].)$$

Keystrokes	Display	
[XEQ] [ALPHA] SIZE [ALPHA] 005		
1 [ENTER↑] 2 [CHS] [ENTER↑]		
3 [CHS] [ENTER↑] 4		
[XEQ] [ALPHA] C+ [ALPHA]	*U*=2.0000	real ($z_2 + z_3$)
[R/S]	*V*=−2.0000	imag ($z_2 + z_3$)
[XEQ] [ALPHA] CINV [ALPHA]	*U*=0.2500	$1/(z_2 + z_3)$
[R/S]	*V*=0.2500	
13 [ENTER↑] 23		
[XEQ] [ALPHA] C× [ALPHA]	*U*=2.500	$(z_1/(z_2 + z_3))$
[R/S]	*V*=9.0000	

Example 2:

Find the 3 cube roots of 8.

Keystrokes	Display
0 [ENTER↑] 8 [ENTER↑] 3	
[XEQ] [ALPHA] Z↑1/N [ALPHA]	*U*=2.0000
[R/S]	*V*=0.0000
[R/S]	*U*=−1.0000
[R/S]	*V*=1.7321
[R/S]	*U*=−1.0000
[R/S]	*V*=−1.7321

Example 3:

Evaluate $e^{z^{-2}}$, where $z = (1+i)$.

Keystrokes	Display	
1 [ENTER↑] 1 [ENTER↑] 2		
[XEQ] [ALPHA] Z↑N [ALPHA]	*U*=0.0000	(z^2)
[R/S]	*V*=2.0000	
[XEQ] [ALPHA] CINV [ALPHA]	*U*=0.0000	(z^{-2})
[R/S]	*V*=−0.5000	
[XEQ] [ALPHA] e↑Z [ALPHA]	*U*=0.8776	$(e^{z^{-2}})$
[R/S]	*V*=−0.4794	

Example 4:

Evaluate sin(2 + 3i).

Keystrokes	Display
3 [ENTER↑] 2	
[XEQ] [ALPHA] SINZ [ALPHA]	*U=9.1545*
[R/S]	*V=-4.1689*

HYPERBOLICS

This program computes hyperbolic functions and their inverses. The user might wish to reassign the individual programs to selected keys on the calculator, and create an appropriate overlay. A reasonable key reassignment might be:

ASN SINH SIN
ASN COSH COS
ASN TANH TAN
ASN ASINH ■ SIN
ASN ACOSH ■ COS
ASN ATANH ■ TAN

Equations:

Hyperbolic Functions

$$\sinh x = \tfrac{1}{2}\left[e^x - 1 + \frac{e^x - 1}{e^x}\right]$$

$$\cosh x = \frac{e^x + e^{-x}}{2}$$

$$\tanh x = \frac{\sinh x}{\cosh x}$$

Inverse Hyperbolic Functions

$$\sinh^{-1}x = \ln\left[x + (x^2 + 1)^{1/2}\right]$$

$$\cosh^{-1}x = \ln\left[x + (x^2 - 1)^{1/2}\right] \qquad x \geq 1$$

$$\tanh^{-1}x = \tfrac{1}{2}\ln\left[\frac{1+x}{1-x}\right] \qquad x^2 < 1$$

Remarks:

- The module program uses Register 00.
- The printer flag (flag 21) is not set by the module program.

STEP	INSTRUCTIONS	INPUT	FUNCTION	SIZE: 001 DISPLAY
1.	For hyperbolics, go to step 2; for inverse hyperbolics, go to step 3.			
	HYPERBOLIC FUNCTIONS			
2.	Key in argument and compute			
	• hyperbolic sine	x	[XEQ] SINH	sinh x
	• hyperbolic cosine	x	[XEQ] COSH	cosh x
	• hyperbolic tangent	x	[XEQ] TANH	tanh x
	INVERSE HYPERBOLIC FUNCTIONS			
3.	Key in argument and compute			
	• inverse hyperbolic sine	x	[XEQ] ASINH	$\sinh^{-1}x$
	• inverse hyperbolic cosine	x	[XEQ] ACOSH	$\cosh^{-1}x$
	• inverse hyperbolic tangent	x	[XEQ] ATANH	$\tanh^{-1}x$

Example 1:

Evaluate the following hyperbolic functions:

sinh 2.5; cosh 3.2; tanh 1.9.

Keystrokes	Display	
[XEQ] [ALPHA] SIZE [ALPHA] 001		
2.5 [XEQ] [ALPHA] SINH [ALPHA]	*6.0502*	(sinh 2.5)
3.2 [XEQ] [ALPHA] COSH [ALPHA]	*12.2866*	(cosh 3.2)
1.9 [XEQ] [ALPHA] TANH [ALPHA]	*0.9562*	(tanh 1.9)

Example 2:

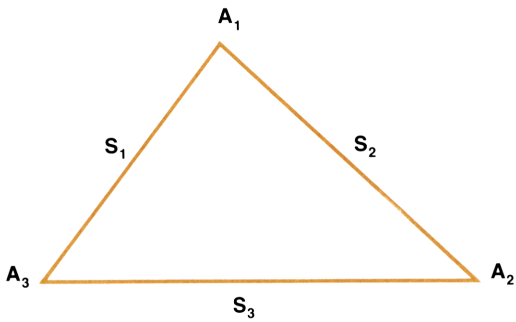
Evaluate the following inverse hyperbolic functions:

$\sinh^{-1} 2.4$; $\cosh^{-1} 90$; $\tanh^{-1} -0.65$.

Keystrokes	Display	
2.4 [XEQ] [ALPHA] ASINH [ALPHA]	*1.6094*	($\sinh^{-1} 2.4$)
90 [XEQ] [ALPHA] ACOSH [ALPHA]	*5.1929*	($\cosh^{-1} 90$)
.65 [CHS]		
[XEQ] [ALPHA] ATANH [ALPHA]	*-0.7753*	($\tanh^{-1} -0.65$)

TRIANGLE SOLUTIONS

These programs can be used to find the area, the dimensions of the sides (S_1, S_2, S_3) and the angles (A_1, A_2, A_3) of a triangle.



Simply key in three known values and execute the appropriate program. The calculator will output the values of the sides, the angles, and the area. The order of output is determined by the order of input. If input values are selected in a clockwise order around the triangle, the outputs will also follow a clockwise order around the triangle. The order is as follows:

First side input	(S_1)
Adjacent angle	(A_1)
Adjacent side	(S_2)
Adjacent angle	(A_2)
Adjacent side	(S_3)
Adjacent angle	(A_3)
Area	

Equations:

S_1, S_2, S_3 (all sides of triangle are known)

$$A_3 = 2 \cos^{-1} \sqrt{\frac{P(P - S_2)}{S_1 S_3}}$$

where $P = (S_1 + S_2 + S_3)/2$

$$A_2 = 2 \cos^{-1} \sqrt{\frac{P(P - S_1)}{S_2 S_3}}$$

$$A_1 = \cos^{-1} (-\cos(A_3 + A_2))$$

A_3, S_1, A_1 (Two angles and the included side are known)

$$A_2 = \cos^{-1}(-\cos(A_3 + A_1))$$

$$S_2 = S_1 \frac{\sin A_3}{\sin A_2}$$

$$S_3 = S_1 \cos A_3 + S_2 \cos A_2$$

S_1, A_1, A_2 (side and following two angles known)

$$A_3 = \cos^{-1}(-\cos(A_1 + A_2))$$

Problem has been reduced to the A_3, S_1, A_1 configuration.

S_1, A_1, S_2 (Two sides and included angle are known)

$$S_3 = \sqrt{S_1^2 + S_2^2 - 2 S_1 S_2 \cos A_1}$$

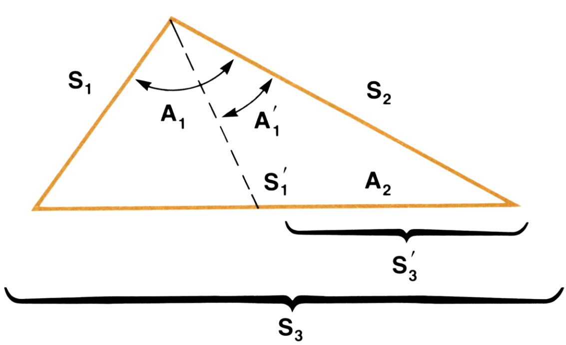
The problem has been reduced to the S_1, S_2, S_3 configuration.

S_1,S_2,A_2 (Two sides and the adjacent angle known)

$$A_3 = \sin^{-1}\left[\frac{S_2}{S_1} \sin A_2\right]^*$$

$$A_1 = \cos^{-1}[-\cos(A_2 + A_3)]$$

The problem has been reduced to the A_3,S_1,A_1 configuration.



Area = 1/2 S_1 S_3 sin A_3

* Note that two possible solutions exist if S_2 is greater than S_1 and A_3 does not equal 90°. Both possible answer sets are calculated.

Remarks:

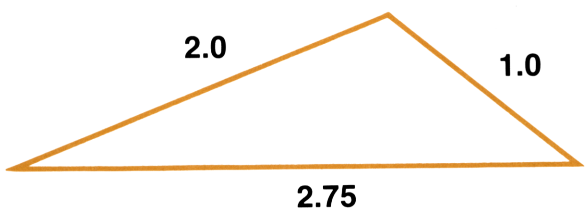
- Program uses registers R_{00}–R_{07}.
- Angles must be in units corresponding to the angular mode of the machine.
- Note that the triangle described by the program does not conform to the standard triangle notation, i.e., A_1 is not opposite S_1.
- Angles must be entered as decimals. The [HR] conversion can be used to convert degrees, minutes, and seconds to decimal degrees.
- Accuracy of solution may degenerate for triangles containing extremely small angles.

				SIZE: 008
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1.	Choose appropriate angular mode.			
2.	Find applicable case in the list below and input indicated values:			
	All sides known		[XEQ] SSS	S1=?
		S_1	[R/S]	S2=?
		S_2	[R/S]	S3=?
		S_3	[R/S]	S1=
	Two angles and included side known.		[XEQ] ASA	A3=?
		A_3	[R/S]	S1=?
		S_1	[R/S]	A1=?
		A_1	[R/S]	S1=
	Two angles and adjacent side known.		[XEQ] SAA	S1=?
		S_1	[R/S]	A1=?
		A_1	[R/S]	A2=?
		A_2	[R/S]	S1=
	Two sides and included angle known.		[XEQ] SAS	S1=?
		S_1	[R/S]	A1=?
		A_1	[R/S]	S2=?
		S_2	[R/S]	S1=
	Two sides and adjacent angle known.		[XEQ] SSA	S1=?
		S_1	[R/S]	S2=?
		S_2	[R/S]	A2=?
		A_2	[R/S]	S1=

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
3.	After step 2, the values of the sides and angles are output with successive use of [R/S]. The last output is the triangle's area. For the last case (SSA), two possible solutions may exist and both will be output.		[R/S] [R/S] [R/S] [R/S] [R/S] [R/S]	A1= S2= A2= S3= A3= AREA=

Example 1:

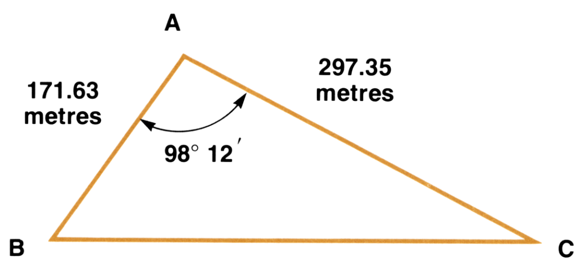
Find the angles (in degrees) and the area for the following triangle.



Keystrokes	Display
[XEQ] [ALPHA] SIZE [ALPHA] 008	
[XEQ] [ALPHA] DEG [ALPHA]	
[XEQ] [ALPHA] SSS [ALPHA]	*S1=?*
2 [R/S]	*S2=?*
1 [R/S]	*S3=?*
2.75 [R/S]	*S1=2.0000*
[R/S]	*A1=129.8384*
[R/S]	*S2=1.0000*
[R/S]	*A2=33.9479*
[R/S]	*S3=2.7500*
[R/S]	*A3=16.2136*
[R/S]	*AREA=0.7679*

Example 2:

A surveyor is to find the area and dimensions of a triangular land parcel. From point A, the distances to B and C are measured with an electronic distance meter. The angle between AB and AC is also measured. Find the area and other dimensions of the triangle.



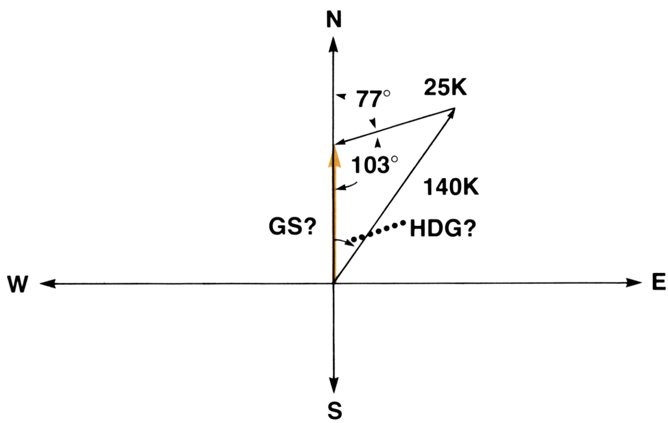
This is a side-angle-side problem where:

$$S_1 = 171.63, A_1 = 98°12' \text{ and } S_2 = 297.35.$$

Keystrokes	Display
[XEQ] [ALPHA] SAS [ALPHA]	*S1=?*
171.63 [R/S]	*A1=?*
98.12 [XEQ] [ALPHA] HR [ALPHA]	
[R/S]	*S2=?*
297.35 [R/S]	*S1=171.6300*
[R/S]	*A1=98.2000*
[R/S]	*S2=297.3500*
[R/S]	*A2=27.8270*
[R/S]	*S3=363.9118*
[R/S]	*A3=53.9730*
[R/S]	*AREA=25,256.2094*

Example 3:

A pilot wishes to fly due north. The wind is reported as 25 knots at 77°. Because winds are reported opposite to the direction they blow, this is interpreted as 77 + 180 or 257°. The true airspeed of the aircraft is 140 knots. What heading (HDG) should be flown? What is the ground speed (GS)?

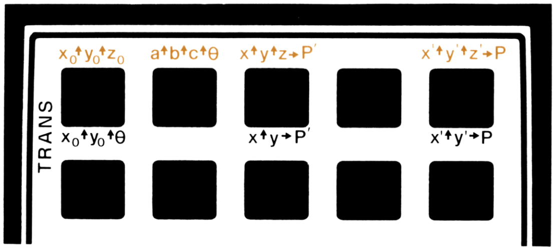


By subtracting the wind direction from 180 (yielding an angle of 103°), the problem reduces to a S_1,S_2,A_2 triangle.

Keystrokes	Display	
[XEQ] [ALPHA] SSA [ALPHA]	*S1=?*	
140 [R/S]	*S2=?*	
25 [R/S]	*A2=?*	
103 [R/S]	*S1=140.0000*	(TAS)
[R/S]	*A1=66.9798*	
[R/S]	*S2=25.0000*	(Wind velocity)
[R/S]	*A2=103.0000*	
[R/S]	*S3=132.2407*	(GS)
[R/S]	*A3=10.0202*	(HDG)
[R/S]	*AREA=1,610.6428*	

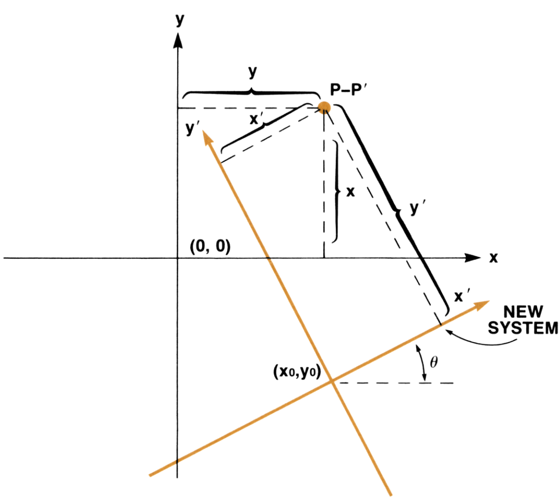
Thus, the pilot should fly a heading 10.02° east of due north. His ground speed equals 132.24 knots.

COORDINATE TRANSFORMATIONS



This program provides 2-dimensional and 3-dimensional coordinate translation and/or rotation.

For the 2-dimensional case, input the coordinates of the origin of the translated system (x_0, y_0), the rotation angle (θ) relative to the original system, and specify the new coordinate axis. These quantities are input with the [A] key. Subsequently, points specified in the original system (x,y) may be converted to the translated rotated system (x′, y′) using the [C] key. Points in the new (x′,y′) system may be converted to points in the original (x,y) system using the [E] key.



The 3-dimensional case is analogous to the 2-dimensional case. The only important difference is the specification of the rotation. The rotation axis passes through the translated origin (x_0, y_0, z_0) and is parallel to an arbitrary direction vector $(a\vec{i}, b\vec{j}, c\vec{k})$. The sign of the rotation angle (θ) is determined by the right-hand rule and the direction of the rotation vector. For instance, the special case of 2-dimensional rotation (rotation in the (x,y) plane) could be achieved using a direction vector of (0,0,1) and a positive rotation angle for counter-clockwise rotations. The coordinates of the translated origin (x_0, y_0, z_0) are input using ■ [A]. The direction vector and angle are input using ■ [B]. Conversions from the original system (x,y,z) to the new system (x′,y′,z′) are initiated using ■ [C] while the inverse conversion is performed with ■ [E].

Equations:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \ell_1 & m_1 & n_1 \\ \ell_2 & m_2 & n_2 \\ \ell_3 & m_3 & n_3 \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

where

$$\begin{bmatrix} \ell_1 & \ell_2 & \ell_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} = \begin{bmatrix} a^2(1-\cos\theta)+\cos\theta & ab(1-\cos\theta)-c\sin\theta & ac(1-\cos\theta)+b\sin\theta \\ ba(1-\cos\theta)+c\sin\theta & b^2(1-\cos\theta)+\cos\theta & bc(1-\cos\theta)-a\sin\theta \\ ca(1-\cos\theta)-b\sin\theta & cb(1-\cos\theta)+a\sin\theta & c^2(1-\cos\theta)+\cos\theta \end{bmatrix}$$

Two-dimensional transformations are handled as a special case of three-dimensional transformation with (a,b,c) set to (0,0,1).

Remarks:

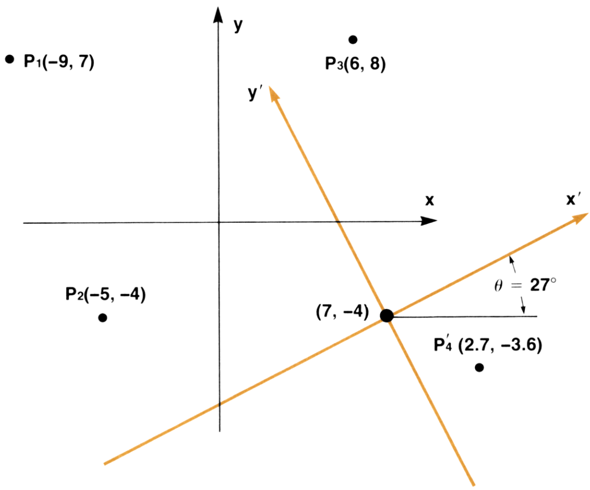
- For pure translation, input zero for θ.
- For pure rotation, input zeros for x_0, y_0, and z_0.
- Program uses registers 00-24.

				SIZE: 025
STEP	**INSTRUCTIONS**	**INPUT**	**FUNCTION**	**DISPLAY**
1.	Initialize program and position overlay.		[XEQ] TRANS	0.0000
2.	For 2-dimensional transformations, go to step 3. For 3-dimensional transformations, go to step 6.			

STEP	INSTRUCTIONS	INPUT	FUNCTION	DISPLAY
3.	Input the origin of the translated system and the rotation angle.	x_0 y_0 θ	[ENTER↑] [ENTER↑] [A]	 1.0000
4.	Transform coordinates from the original system to the translated-rotated system.	x y	[ENTER↑] [C] [R/S]	 x′ y′
	or From the translated-rotated system to the original system.	x′ y′	[ENTER↑] [E] [R/S]	 x y
5.	For a new set of coordinates, go to step 4. For a new 2-dimensional transformation, go to step 3.			
6.	Input the origin of the translated system.	x_0 y_0 z_0	[ENTER↑] [ENTER↑] ■ [A]	 x_0
	and Input the rotation direction vector and angle.	a b c θ	[ENTER↑] [ENTER↑] [ENTER↑] ■ [B]	 $\sqrt{a^2+b^2+c^2}$
7.	Transform coordinates from original system to translated-rotated system.	x y z	[ENTER↑] [ENTER↑] ■ [C] [R/S] [R/S]	 x′ y′ z′
	or From the translated-rotated system to the original system.	x′ y′ z′	[ENTER↑] [ENTER↑] ■ [E] [R/S] [R/S]	 x y z
8.	For a new set of coordinates, go to step 7. For a new 3-dimensional transformation, go to step 6 (either (x_0,y_0,z_0) or (a,b,c,θ) may be changed independently.)			

Example 1:

The coordinate systems (x, y) and (x′, y′) are shown below:



Convert the points P_1, P_2 and P_3 to equivalent coordinates in the (x′, y′) system. Convert the point P_4' to equivalent coordinates in the (x, y) system. (Degrees mode is being used.)

Keystrokes	Display	
[XEQ] [ALPHA] SIZE [ALPHA] 025		
[XEQ] [ALPHA] TRANS [ALPHA]	0.0000	
7 [ENTER↑] 4 [CHS] [ENTER↑]		
27 [A]	1.0000	
9 [CHS] [ENTER↑] 7 [C]	-9.2622	(x_1')
[R/S]	17.0649	(y_1')
5 [CHS] [ENTER↑] 4 [CHS] [C]	-10.6921	(x_2')
[R/S]	5.4479	(y_2')
6 [ENTER↑] 8 [C]	4.5569	(x_3')
[R/S]	11.1461	(y_3')
2.7 [ENTER↑] 3.6 [CHS] [E]	11.0401	(x_4)
[R/S]	-5.9818	(y_4)

Example 2:

A 3-dimensional coordinate system is translated to (2.45, 4.00, 4.25). After translation, a 62.5 degree rotation occurs about the (0, −1, −1) axis. In the original system, a point had the coordinates (3.9, 2.1, 7.0). What are the coordinates of the point in the translated rotated system?

Keystrokes	Display	
[XEQ] [ALPHA] TRANS [ALPHA]		
2.45 [ENTER↑] 4 [ENTER↑]		
4.25 ■ [A]	**2.4500**	
0 [ENTER↑] 1 [CHS] [ENTER↑]		
1 [CHS] [ENTER↑] 62.5 ■ [B]	**1.4142**	
3.9 [ENTER↑] 2.1 [ENTER↑]		
7 ■ [C]	**3.5861**	(x′)
[R/S]	**0.2609**	(y′)
[R/S]	**0.5891**	(z′)

In the translated-rotated system above, a point has the coordinate (1,1,1). What are the corresponding coordinates in the original system?

Keystrokes	Display	
1 [ENTER↑] 1 [ENTER↑]		
1 ■ [E]	**2.9117**	(x)
[R/S]	**4.3728**	(y)
[R/S]	**5.8772**	(z)

APPENDIX A

PROGRAM DATA

Program	# Regs. to COPY	Data Registers	Flags	Display Format	Angular Mode
Matrix Operations	138	(See Register Chart)	02-10 21-22 25 29		
Solution to f(x)=0	43	00-06	00 21-22		
Polynomial Solutions/ Evaluation	84	00-22	00 02-03 21	FIX 4	
Numerical Integration	27	00-07	21 27		
Differential Equations	35	00-07	01 21		
Fourier Series	50	00-26	01-02 21 29	FIX 4	
Complex Operations	59	00-04	21		Rad/Deg.
Hyperbolics	17	00			
Triangle Solutions	46	00-07	21		
Coordinate Transformations	50	00-24	00-02 21 27		

MATRIX REGISTER CHART

Order	Number of Memory Modules (M)	Number of Registers (R)	Location of Registers	Location of Matrix*	Location of Pivots	Location of Column	Registers to Record
N	$\mathrm{INT}\left(\frac{N^2+2N+15}{64}\right)$	$N^2+2N+15$	00 to $N^2+2N+14$	15 to N^2+14	N^2+15 to N^2+N+14	N^2+N+15 to $N^2+2N+14$	13 to N^2+N+14
1	0	18	00- 17	15- 15	16- 16	17- 17	13- 16
2	0	23	00- 22	15- 18	19- 20	21- 22	13- 20
3	0	30	00- 29	15- 23	24- 26	27- 29	13- 26
4	0	39	00- 38	15- 30	31- 34	35- 38	13- 34
5	0	50	00- 49	15- 39	40- 44	45- 49	13- 44
6	0	63	00- 62	15- 50	51- 56	57- 62	13- 56
7	1	78	00- 77	15- 63	64- 70	71- 77	13- 70
8	1	95	00- 94	15- 78	79- 86	87- 94	13- 86
9	1	114	00-113	15- 95	96-104	105-113	13-104
10	2	135	00-134	15-114	115-124	125-134	13-124
11	2	158	00-157	15-135	136-146	147-157	13-146
12	2	183	00-182	15-158	159-170	171-182	13-170
13	3	210	00-209	15-183	184-196	197-209	
14	3	239	00-238	15-210	211-224	225-238	

* The matrix is stored in *row* order. Any element A(I,J) can be located using the following formula: Register address = $N(I-1)+J+14$

SUBROUTINES

This table provides information necessary to use various portions of the Math Application Module as subroutines.

SUBROUTINE	LABEL	INITIAL REGISTERS	FLAG STATUS	FINAL REGISTERS	REMARKS
Matrix Pivoting	PVT	15 to N^2+14 (See Register Chart) R_{14} Order	SF 04 CF 06 CF 07 CF 08 CF 09 CF 10 SF 21	00 to $N^2+2N+14$ (See Register Chart)	Allows the user to skip the initial matrix prompting.
Simultaneous Equations	SIMEQ	15 to N^2+14 and N^2+N+15 to $N^2+2N+14$ (See Register Chart) R_{14} Order	SF 04 SF 05 CF 06 CF 07 CF 08 CF 09 CF 10 SF 21	00 to $N^2+2N+14$ (See Register Chart)	Skips the initial matrix prompting. Assumes column vector is already stored.
Solution of f(x) on Interval	SOL	R_{01} Guess 1 R_{02} Guess 2 R_{06} Function Name	SF 21	R_{00}-R_{06} are used	Remember to key in function to be evaluated. Flag 00 is used in the program.
Roots of a Polynomial	ROOTS	R_{00} a_0 R_{01} a_1 R_{02} a_2 R_{03} a_3 R_{04} a_4 R_{22} Degree	SF 21 SF 00	R_{00}-R_{22} are used	Finds all roots of a polynomial with real coefficients; highest order coefficient must be 1.

HEWLETT PACKARD

1000 N.E. Circle Blvd., Corvallis, OR 97330

For additional sales and service information contact your local Hewlett-Packard Sales Office or call 800/648-4711. (In Nevada call 800/992-5710.)

00041-90034

Printed in U.S.A.