HEWLETT-PACKARD

HP-25 Applications Programs

(Cover photograph courtesy of NASA.)

HEWLETT-PACKARD

HP-25 Applications Programs

00025-90011 Rev. E 7/76

INTRODUCTION

Welcome. You are about to step into a field that, ten years ago, was open only to users of large computer systems costing tens or hundreds of thousands of dollars, and even five years ago, required a several-thousand-dollar calculator that occupied the better part of a desktop. Today, the HP-25 puts programming into the hands of the individual. It is hoped that this book will allow you to realize some of the potential of this calculating instrument.

These HP-25 Applications Programs have been drawn from the varied fields of mathematics, statistics, finance, surveying, navigation, and games. They have been arranged in eight chapters which follow roughly the above classification. Each program is furnished with a full explanation which includes a description of the problem, any pertinent equations, a list of keystrokes to be entered into program memory, a set of instructions for running the program, and an example or two, with solutions. To use the programs does not require any proficiency in programming, but some familiarity with the HP-25 Owner's Handbook is assumed.

For users who want to enhance their understanding of programming principles and techniques, a number of programs are provided to help in this respect. The first program in each chapter contains, in addition to the usual explanations, a more detailed description of the problem, a commented list of the program keystrokes with a step-by-step tracing of the contents of the stack registers, and a list of the keystrokes required to solve the example problem. Whenever an interesting programming technique is used in one of these programs, it is described in a short section headed "Programming Remarks", which, if present, will immediately precede the list of program keystrokes.

Thus, whether your interest lies in solving a particular problem in a specific area, or in learning more about the programming power of your calculator, we hope that this book will help you get the most from your HP-25.

TABLE OF CONTENTS

A WORD ABOUT PROGRAM USAGE

Various kinds of information are provided to explain the use of each program. Besides a short description of the problem, a list of applicable equations, and an example problem with solution, there are two forms that deserve some explanation: the Program form and the User Instructions form.

Two different Program forms are provided, one of which is just a simplified version of the other. The detailed form is used for a total of eight programs, one per chapter, with the simpler form serving for the rest. A section of a detailed form, taken from the Plotting/Graphing program in Chapter 1, is shown below:

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS	REGISTERS
LINE	CODE							
00			v	θ				R_0 Δt
01	14 09	f →R	v_x	v_y			Use polar-to-rectangular for	
02	23 02	STO 2	v_x	v_y			$v_x = v \cos\theta$ = horiz. vel.	
03	21	x⇄y	v_y	v_x				R_1 g
04	23 03	STO 3	v_y	v_x			$v_y = v \sin\theta$ = vert. vel.	
05	00	0	0					
06	23 04	STO 4	0				Initialize: t = 0	R_2 v_x
07	24 00	RCL 0	Δt				Start of loop	
08	23 51 04	STO + 4	Δt				Next time interval:	
09	24 04	RCL 4	t				$t \leftarrow t + \Delta t$	R_3 v_y
10	15 02	g x^2	t^2					

The rightmost column, headed REGISTERS, explains what variables are stored in storage registers R_0 through R_7. The rest of the form is divided into eight columns. The first two columns describe the appearance of the display as the program is being keyed in: LINE shows the step number for the current instruction and CODE denotes the numeric keycodes corresponding to the keystrokes in the next column, KEY ENTRY. The entries in this column are the keys that must be pressed to enter the program into program memory. The **ENTER↑** key is denoted in this column as [↑]; all other key designations are identical to those appearing on the HP-25.

The next four columns, X, Y, Z, and T, trace the contents of the stack registers as they would change during execution of the program in RUN mode. Each entry under X, Y, Z, or T gives the contents of the respective register *after* the instruction on that line has been executed. The COMMENTS column contains additional step-by-step explanation of the program's calculations.

These last columns, X, Y, Z, T, and COMMENTS, are provided to help the interested user acquire a detailed, in-depth understanding of a particular program, or of programming techniques in general.

The simplified Program forms contain the same information as the detailed forms except for the omission of columns X, Y, Z, T, and COMMENTS.

The User Instructions form is the user's guide to operating the program to solve his own particular problem. This form, which is composed of five columns, is illustrated below for the same program from Chapter 1, Plotting/Graphing.

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store time interval	Δt	STO	0			
3	Store gravitational constant	g	STO	1			
4	Input angle and initial speed	θ	↑				
		v	f	PRGM			
5	Perform steps 5 and 6 any num-						
	ber of times: Display time and		R/S				(t)
	horizontal distance						x
6	Display height		R/S				y
7	To change θ or v, go to step 4.						
	To change Δt or g, go to						
	appropriate step, store new value,						
	then go to step 4.						

Reading from left to right, the STEP column gives the instruction step number. The INSTRUCTIONS column gives instructions and comments concerning the operations to be performed. Steps are executed in sequential order except where the INSTRUCTIONS column directs otherwise.

Normally, the first instruction is "Key in program", which means to store the keystrokes of the program in program memory (switch to PRGM mode, press [f] [PRGM], key in the program, then switch back to RUN mode).

Repeated processes, used in most cases for a long string of input/output data, are outlined with a bold border, as in steps 5 and 6 above. In this case, the steps are repeated in order to generate a number of (x,y) pairs for a graph.

The INPUT DATA/UNITS column specifies the input data to be supplied, and the units of data if applicable. The KEYS column specifies the keys to be pressed. [↑] is used for the ENTER↑ key, and all other key designations are identical to those appearing on the HP-25. Ignore any blank positions in the KEYS column.

Some programs are complex enough that users have to press additional keys to generate some results. Those keys are also shown in the KEYS column.

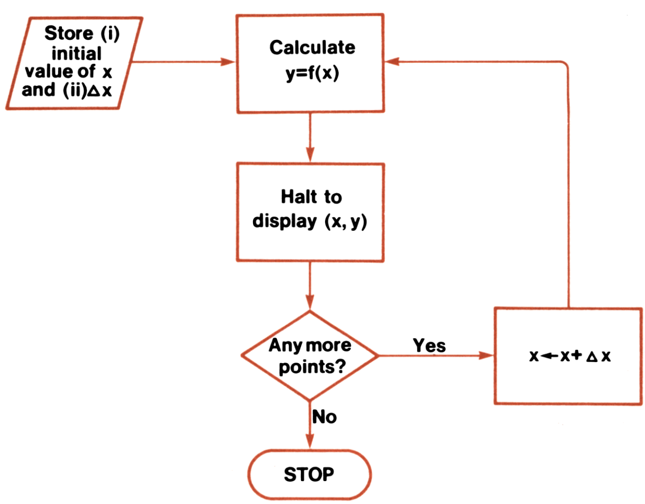
The OUTPUT DATA/UNITS column shows intermediate and final results that have been calculated either from the keyboard or from an executing program, and the units of data if applicable. Parentheses around an output variable, such as (t) in step 5, indicate that the result is displayed only briefly by a PAUSE instruction ([f] [PAUSE]).

CHAPTER 1 ALGEBRA AND NUMBER THEORY

PLOTTING/GRAPHING

Most people who have labored through a ninth-grade algebra course probably still respond with a shudder to the word "graph". Evidently the tedium of finding $y = 3x^2 - 4x + 4$, for integer values of x from $-\infty$ to $+\infty$, has etched permanent memories in us all. Fortunately, we need not endure this tedium any longer. The HP-25 lends itself perfectly to this kind of repetitive calculation.

The basic idea is to generate (x, y) pairs by keying into program memory the keystrokes required to calculate y, assuming x is given. Then the user need only return to the top of memory, enter a value for x, press [R/S], and see y displayed within seconds. The process may be repeated for as many values of x as desired. The programmer can take this process one step further into automation by also having the calculator generate each new value of x, for example, by adding 1 to the old value, or, in general, by adding a specified increment Δx. A flowchart of the process is shown below.



The program used here to illustrate this process takes a slightly different tack. We will consider the problem of plotting the trajectory of a stone which is hurled into the air with an initial velocity v at an angle to the horizontal of θ. Neglecting drag due to friction with the atmosphere, the following equations describe the stone's x- and y-coordinates as functions of the time t:

$$x = vt \cos \theta \qquad\qquad y = vt \sin \theta - \frac{1}{2} gt^2$$

where x = horizontal distance the stone has traveled

y = height of the stone

g = acceleration due to gravity

$\simeq 9.8\ m/s^2$

$\simeq 32\ ft/s^2$

These equations differ slightly from the usual graphing function in that y is not expressed directly as a function of x, but instead both x and y are expressed as functions of a third variable t. The points to be plotted are still the ordered pairs (x, y); but now it is the time t which should be incremented by an amount Δt.

Notes:

1. Any consistent set of units may be used.

2. This is *not* a general plotting/graphing program; it merely illustrates the method by application to a specific problem. However, some study of the program listing and the flowchart should enable the user to adapt the method to his own application.

Programming Remarks:

1. The components of the velocity in the horizontal and vertical directions, v_x and v_y, are computed in one step by a conversion of v and θ to rectangular coordinates ($\boxed{f}$ $\boxed{\to R}$). The values $v_x = v \cos \theta$ and $v_y = v \sin \theta$ are returned to the X- and Y-registers, respectively.

2. A pause ($\boxed{f}$ $\boxed{\text{PAUSE}}$) is used in this program in a very typical manner, to display briefly the output variable t, whose values are simple (0.25, 0.50, 0.75, etc.) and do not need to be written down.

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS
LINE	CODE						
00			v	θ			
01	14 09	f →R	v_x	v_y			Use polar-to-rectangular for
02	23 02	STO 2	v_x	v_y			$v_x = v \cos\theta$ = horiz. vel.
03	21	x⇄y	v_y	v_x			
04	23 03	STO 3	v_y	v_x			$v_y = v \sin\theta$ = vert. vel.
05	00	0	0				
06	23 04	STO 4	0				Initialize: $t = 0$
07	24 00	RCL 0	Δt				Start of loop
08	23 51 04	STO + 4	Δt				Next time interval:
09	24 04	RCL 4	t				$t \leftarrow t + \Delta t$
10	15 02	g x^2	t^2				
11	24 01	RCL 1	g	t^2			
12	61	×	$g\,t^2$				
13	02	2	2	$g\,t^2$			
14	71	÷	$1/2\,g\,t^2$				
15	32	CHS	$-1/2\,g\,t^2$				
16	24 04	RCL 4	t	$-1/2\,g\,t^2$			
17	24 03	RCL 3	v_y	t	$-1/2\,g\,t^2$		
18	61	×	$v_y\,t$	$-1/2\,g\,t^2$			
19	51	+	y				$y = v_y\,t - 1/2\,g\,t^2$
20	24 04	RCL 4	t	y			
21	24 02	RCL 2	v_x	t	y		
22	61	×	x	y			$x = v_x\,t$
23	24 04	RCL 4	t	x	y		
24	14 74	f PAUSE	t	x	y		Pause to display t
25	22	R↓	x	y		t	
26	74	R/S	x	y		t	Halt and display x
27	21	x⇄y	y	x		t	
28	74	R/S	y	x		t	Halt and display y
29	13 07	GTO 07	y	x		t	Branch back for next t
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							

REGISTERS	
R_0	Δt
R_1	g
R_2	v_x
R_3	v_y
R_4	t
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store time interval	Δt	STO	0			
3	Store gravitational constant	g	STO	1			
4	Input angle and initial speed	θ	↑				
		v	f	PRGM			
5	Perform steps 5 and 6 any num-						
	ber of times: Display time and		R/S				(t)
	horizontal distance						x
6	Display height		R/S				y
7	To change θ or v, go to step 4.						
	To change Δt or g, go to						
	appropriate step, store new value,						
	then go to step 4.						

Example:

Plot the trajectory of a stone cast upwards with a velocity of 20 m/s at an angle of 30° to the horizontal. Use intervals of ¼ second between points plotted. Let $g = 9.8\ m/s^2$.

Solution:

0.25 [STO] [0] 9.8 [STO] [1] 30 [↑] 20 [f] [PRGM] [R/S] ⟶ 0.25 (t_1)

4.33 (x_i)

[R/S] ⟶ 2.19 (y_1)

[R/S] ⟶ 0.5 (t_2)

8.66 (x_2)

[R/S] ⟶ 3.78 (y_2)

[R/S] ⟶ 0.75 (t_3)

12.99 (x_3)

[R/S] ⟶ 4.74 (y_3)

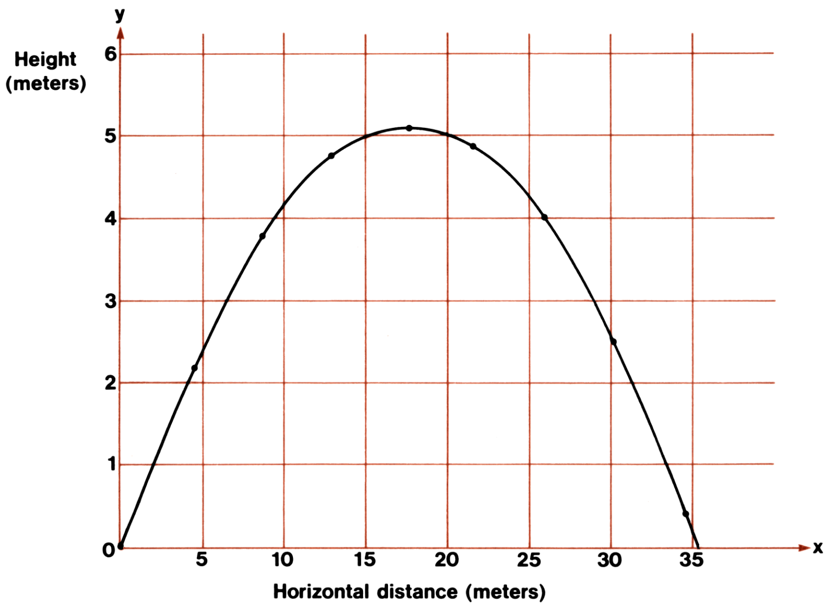
etc.

Continue until y becomes negative.

The table of these results is shown below:

t	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25
x	4.33	8.66	12.99	17.32	21.65	25.98	30.31	34.64	38.97
y	2.19	3.78	4.74	5.10	4.84	3.98	2.49	0.40	–2.31

The plot of these (x, y) values is made and the stone's trajectory is seen to be a parabola.



QUADRATIC EQUATION

The roots x_1, x_2 of

$$ax^2 + bx + c = 0$$

are given by

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

If

$$D = (b^2 - 4ac)/4a^2$$

is positive or zero, the roots are real. In these cases, better accuracy may sometimes be obtained by first computing the root with the larger absolute value:

If

$$-\frac{b}{2a} \geqslant 0, \qquad x_1 = -\frac{b}{2a} + \sqrt{D}$$

If

$$-\frac{b}{2a} < 0, \qquad x_1 = -\frac{b}{2a} - \sqrt{D}$$

In either case,

$$x_2 = \frac{c}{x_1\, a}.$$

If $D < 0$, the roots are complex, being

$$u \pm iv = \frac{-b}{2a} \pm \frac{\sqrt{4ac - b^2}}{2a}\, i$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	31	↑
02	22	R↓
03	71	÷
04	02	2
05	71	÷
06	32	CHS
07	31	↑
08	15 02	g x^2
09	22	R↓
10	22	R↓
11	21	x⇄y
12	71	÷
13	23 00	STO 0
14	41	–
15	14 74	f PAUSE
16	15 41	g x<0
17	13 31	GTO 31
18	14 02	f $\sqrt{x}$
19	21	x⇄y
20	15 41	g x<0
21	13 24	GTO 24
22	51	+
23	13 26	GTO 26
24	21	x⇄y
25	41	–
26	74	R/S
27	15 22	g 1/x
28	24 00	RCL 0
29	61	x
30	13 00	GTO 00
31	32	CHS
32	14 02	f $\sqrt{x}$
33	21	x⇄y
34	74	R/S
35	21	x⇄y
36	13 00	GTO 00
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 c/a
R_1
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM			
3	Enter coefficients and display D	c	↑				
		b	↑				
		a	R/S				(D)
4	If $D \geqslant 0$, roots are real						x_1
			R/S				x_2
	or						
	If $D < 0$, roots are complex of						
	form $u \pm iv$						u
			R/S				v
5	For new case, go to step 3.						

Example:

Find solutions to the three equations below:

1. $x^2 + x - 6 = 0$
2. $3x^2 + 2x - 1 = 0$
3. $2x^2 - 3x + 5 = 0$

Solutions:

1. $D = 6.25$

   $x_1 = -3.00$

   $x_2 = 2.00$

2. $D = 0.44$

   $x_1 = -1.00$

   $x_2 = 0.33$

3. $D = -1.94$

   $x_1, x_2 = 0.75 \pm 1.39\ i$

COMPLEX ARITHMETIC, +, –, x, ÷

Let $a_1 + ib_1$ and $a_2 + ib_2$ be two complex numbers. The arithmetic operations +, –, x, ÷ are defined as follows:

1. +, addition

$$(a_1 + ib_1) + (a_2 + ib_2) = (a_1 + a_2) + (b_1 + b_2)i$$

2. –, subtraction

$$(a_1 + ib_1) - (a_2 + ib_2) = (a_1 - a_2) + (b_1 - b_2)i$$

3. x, multiplication

$$(a_1 + ib_1) \times (a_2 + ib_2) = r_1\ r_2\ e^{i(\theta_1 + \theta_2)}$$

4. ÷, division

$$\frac{(a_1 + ib_1)}{(a_2 + ib_2)} = \frac{r_1}{r_2}\ e^{i(\theta_1 - \theta_2)},\ a_2 + ib_2 \neq 0$$

where $r_1\ e^{i\theta_1}$ is the polar representation of $a_1 + ib_1$ and $r_2\ e^{i\theta_2}$ is the polar representation of $a_2 + ib_2$. In each case let the answer be $x + iy$.

After a calculation is finished x is stored in R_o as well as the X-register and y is stored in R_1 as well as the Y-register. In this way arithmetic operations can be chained together.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	32	CHS
02	21	x⇄y
03	32	CHS
04	21	x⇄y
05	24 00	RCL 0
06	51	+
07	21	x⇄y
08	24 01	RCL 1
09	51	+
10	13 31	GTO 31
11	15 09	g →P
12	15 22	g 1/x
13	21	x⇄y
14	32	CHS
15	21	x⇄y
16	13 18	GTO 18
17	15 09	g →P
18	23 02	STO 2
19	22	R↓
20	24 01	RCL 1
21	24 00	RCL 0
22	15 09	g →P
23	24 02	RCL 2
24	61	x
25	23 02	STO 2
26	22	R↓
27	51	+
28	24 02	RCL 2
29	14 09	f →R
30	21	x⇄y
31	23 01	STO 1
32	21	x⇄y
33	23 00	STO 0
34	13 00	GTO 00
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 a_1, x
R_1 b_1, y
R_2 Used
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store first complex number	b_1	STO	1			
		a_1	STO	0			
3	Key in next number	b_2	↑				
		a_2					
4	For addition		GTO	05	R/S		x
	or						
	subtraction		f	PRGM	R/S		x
	or						
	multiplication		GTO	17	R/S		x
	or						
	division		GTO	11	R/S		x
5	For imaginary part		x⇄y				y
6	For next calculation in chain, go						
	to step 3.						
7	For new case, go to step 2.						

Examples:

1. $(1.2 + 3.7i) - (2.6 - 1.9i) = -1.4 + 5.6i$

2. $\dfrac{3 + 4i}{7 - 2i} = 0.25 + 0.64i$

3. $\left[\dfrac{(3 + 4i) + (7.4 - 5.6i)}{(7 - 2i)}\right] [3.1 + 4.6i] = 3.61 + 7.16i$

COMPLEX FUNCTIONS $|z|$, z^2, $^1/z$, $\sqrt{z}$

A complex number $z = a + ib$ has polar representation $re^{i\theta}$. The formulas used to evaluate the given functions are as follows:

1. $|z| = r$
2. $z^2 = r^2 e^{i2\theta}$
3. $^1/z = \dfrac{1}{r} e^{-i\theta}, z \neq 0$
4. $\sqrt{z} = \pm(\sqrt{r}\ e^{i\theta/2}) = \pm(x + iy)$

The answer is represented by x + iy.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	15 09	g →P
02	13 00	GTO 00
03	15 09	g →P
04	15 02	g x^2
05	21	x⇄y
06	31	↑
07	51	+
08	21	x⇄y
09	14 09	f →R
10	13 00	GTO 00
11	15 09	g →P
12	15 22	g 1/x
13	21	x⇄y
14	32	CHS
15	21	x⇄y
16	14 09	f →R
17	13 00	GTO 00
18	15 09	g →P
19	14 02	f $\sqrt{x}$
20	21	x⇄y
21	02	2
22	71	÷
23	21	x⇄y
24	14 09	f →R
25	13 00	GTO 00
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS	
R_0	
R_1	
R_2	
R_3	
R_4	
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Key in z	b	↑				
		a					
3	For $\lvert z\rvert$		f	PRGM	R/S		$\lvert z\rvert$
	or						
	z^2		GTO	03	R/S		x
			x⇄y				y
	or						
	$^1/z$		GTO	11	R/S		x
			x⇄y				y
	or						
	$\sqrt{z}$		GTO	18	R/S		x
			x⇄y				y
4	For new case, go to step 2.						

Examples:

1. $|12 - 5i| = 13.00$

2. $(6 - i)^2 = 35.00 - 12.00i$

3. $\dfrac{1}{2 + 5i} = 0.07 - 0.17i$

4. $\sqrt{3 + 4i} = \pm(2.00 + 1.00i)$

DETERMINANT AND INVERSE OF A 2 x 2 MATRIX

Let $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ be a 2 x 2 matrix.

The determinant of A denoted by Det A or |A| is evaluated by the following formula:

$$\text{Det } A = a_{22}\, a_{11} - a_{12}\, a_{21}$$

Also, the program evaluates the multiplicative inverse A^{-1} of A. The following formula is used:

$$A^{-1} = \begin{bmatrix} a_{22}/\text{Det } A & -a_{12}/\text{Det } A \\ -a_{21}/\text{Det } A & a_{11}/\text{Det } A \end{bmatrix}$$

DISPLAY		KEY ENTRY	DISPLAY		KEY ENTRY
LINE	CODE		LINE	CODE	
00			25	24 00	RCL 0
01	24 04	RCL 4	26	71	÷
02	24 01	RCL 1	27	13 00	GTO 00
03	61	x	28		
04	24 02	RCL 2	29		
05	24 03	RCL 3	30		
06	61	x	31		
07	41	–	32		
08	23 00	STO 0	33		
09	74	R/S	34		
10	24 04	RCL 4	35		
11	24 00	RCL 0	36		
12	71	÷	37		
13	74	R/S	38		
14	24 02	RCL 2	39		
15	24 00	RCL 0	40		
16	71	÷	41		
17	32	CHS	42		
18	74	R/S	43		
19	24 03	RCL 3	44		
20	24 00	RCL 0	45		
21	71	÷	46		
22	32	CHS	47		
23	74	R/S	48		
24	24 01	RCL 1	49		

REGISTERS
R_0 Det A
R_1 a_{11}
R_2 a_{12}
R_3 a_{21}
R_4 a_{22}
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store matrix	a_{11}	STO	1			
		a_{12}	STO	2			
		a_{21}	STO	3			
		a_{22}	STO	4			
3	Compute determinant		f	PRGM	R/S		Det A
4	Compute inverse		R/S				a_{11}^{-1}
			R/S				a_{12}^{-1}
			R/S				a_{21}^{-1}
			R/S				a_{22}^{-1}
5	For new case, go to step 2.						

Example:

Find the determinant and inverse of the matrix

$$A = \begin{bmatrix} 3 & 2 \\ 4 & -4 \end{bmatrix}$$

Solution:

$$\text{Det } A = -20$$

$$A^{-1} = \begin{bmatrix} 0.20 & 0.10 \\ 0.20 & -0.15 \end{bmatrix}$$

NUMBER IN BASE b TO NUMBER IN BASE 10

This program consists of two subprograms. The first changes the integer part of a number in base b to a number in base 10.

$$I_{10} = i_n i_{n-1} \ldots i_2 i_1 = i_n b^{n-1} + i_{n-1} b^{n-2} + \ldots + i_2 b + i_1$$

This is evaluated in the form

$$b(\ldots(b(b(i_n b + i_{n-1}) + i_{n-2}) + \ldots) + i_2) + i_1$$

The second subprogram changes the fraction part of a number in base b to a number in base 10.

$$F_{10} = f_1 f_2 \ldots f_m = f_1 b^{-1} + f_2 b^{-2} + \ldots + f_m b^{-m}$$

Together the two programs can convert any number in base b to a number in base 10. Zeros must be entered in their proper place.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	23 01	STO 1
02	24 00	RCL 0
03	31	↑
04	31	↑
05	31	↑
06	24 01	RCL 1
07	74	R/S
08	23 01	STO 1
09	34	CLX
10	51	+
11	61	x
12	24 01	RCL 1
13	51	+
14	13 07	GTO 07
15	24 00	RCL 0
16	15 22	g 1/x
17	23 02	STO 2
18	23 03	STO 3
19	61	x
20	74	R/S
21	24 02	RCL 2
22	24 03	RCL 3
23	61	x
24	23 03	STO 3
25	61	x
26	51	+
27	13 20	GTO 20
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 b
R_1 Used
R_2 b^{-1}
R_3 b^{-j}
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store base	b	STO	0			
3	For integer part, input left most						
	digit	i_n	f	PRGM	R/S		
4	Perform for j = n−1,..., 2:						
	Input next digit	i_j*	R/S				
5	Input final digit	i_1*	R/S				I_{10}
6	For fractional part, input digit						
	after decimal	f_1	GTO	15	R/S		
7	Perform for j = 2,..., m−1:						
	Input next digit	f_j*	R/S				
8	Input final digit	f_m*	R/S				F_{10}
9	For new case, go to step 2.						
	* The stack must be maintained						
	at these points.						

Examples:

1. $1777_8 = 1023_{10}$

2. $143.2044_5 = 48.4384_{10}$

NUMBER IN BASE 10 TO NUMBER IN BASE b

This program will convert any positive number in base 10, N_{10}, to a number in base b, N_b, where $2 \leqslant b \leqslant 100$. The algorithm used is an iterative one which adds one more digit to N_b at each iteration. The program pauses as each new N_b is computed to display successive approximations to the final answer. When the displayed value of N_b has reached the accuracy desired by the user, he should press [R/S] to halt the program, then [RCL] [3] to display N_b.

Notes:

1. When the base b is such that $11 \leqslant b \leqslant 100$, two display positions are allocated to each digit of N_b. Begin partitioning to the right and to the left of the decimal point. For example, 41106.12 in base 16 stands for 4B6.C.

2. An error indication during execution means that the machine's accuracy has been exceeded. The value of N_b is in R_3.

LINE	CODE	KEY ENTRY
00		
01	24 00	RCL 0
02	01	1
03	00	0
04	14 51	f x≥y
05	13 09	GTO 09
06	01	1
07	00	0
08	00	0
09	23 02	STO 2
10	00	0
11	23 03	STO 3
12	24 01	RCL 1
13	14 07	f LN
14	24 00	RCL 0
15	14 07	f LN
16	71	÷
17	15 41	g x<0
18	13 21	GTO 21
19	14 01	f INT
20	13 24	GTO 24
21	14 01	f INT
22	01	1
23	41	−
24	23 04	STO 4

LINE	CODE	KEY ENTRY
25	24 02	RCL 2
26	21	x⇄y
27	14 03	f y^x
28	24 03	RCL 3
29	51	+
30	23 03	STO 3
31	14 74	f PAUSE
32	14 74	f PAUSE
33	24 00	RCL 0
34	24 04	RCL 4
35	14 03	f y^x
36	23 41 01	STO − 1
37	13 12	GTO 12
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 b
R_1 N_{10}
R_2 10 or 100
R_3 N_b
R_4 1 digit
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Set display format		f	FIX	9		
3	Store base and decimal number	b	STO	0			
		N_{10}	STO	1	f	PRGM	
4	Display successive approximat-						
	ions to N_b		R/S				(N_b)
5	When number is shown with						
	desired accuracy, press R/S to						
	halt, then		RCL	3			N_b
6	For new case, go to step 3.						

Examples:

1. $67.32_{10} = 403.050114_{16}$
   $= 43.51E_{16}$

2. $\pi = 3.141592654_{10} = 11.00100100_2$

VECTOR CROSS PRODUCT

If $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$ are two three dimensional vectors then the cross product of A and B is denoted by $A \times B$ and is calculated as follows:

$$A \times B = \left(\begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix}, - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix}, \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} \right) = (a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1)$$

Let the solution be represented by (c_1, c_2, c_3).

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	24 02	RCL 2
02	24 06	RCL 6
03	61	x
04	24 03	RCL 3
05	24 05	RCL 5
06	61	x
07	41	–
08	74	R/S
09	24 03	RCL 3
10	24 04	RCL 4
11	61	x
12	24 01	RCL 1
13	24 06	RCL 6
14	61	x
15	41	–
16	74	R/S
17	24 01	RCL 1
18	24 05	RCL 5
19	61	x
20	24 02	RCL 2
21	24 04	RCL 4
22	61	x
23	41	–
24	13 00	GTO 00

DISPLAY		KEY ENTRY
LINE	CODE	
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1 a_1
R_2 a_2
R_3 a_3
R_4 b_1
R_5 b_2
R_6 b_3
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store A	a_1	STO	1			
		a_2	STO	2			
		a_3	STO	3			
3	Store B	b_1	STO	4			
		b_2	STO	5			
		b_3	STO	6			
4	Compute cross-product		f	PRGM	R/S		c_1
			R/S				c_2
			R/S				c_3
5	For new case, go to step 2.						

Example:

Let $A = (2, 5, 2)$

$B = (3, 3, -4).$

Solution:

$A \times B = (-26, 14, -9)$

ANGLE BETWEEN, NORM, AND DOT PRODUCT OF VECTORS

Let $\vec{a} = (a_1, a_2 \ ..., a_n)$ and $\vec{b} = (b_1, b_2, ..., b_n)$ be two vectors.

The norm of $\vec{a}$ is denoted by $|\vec{a}|$ and is calculated by the following formula:

$$|\vec{a}| = \sqrt{a_1{}^2 + a_2{}^2 + ... + a_n{}^2}$$

similarly,

$$|\vec{b}| = \sqrt{b_1{}^2 + b_2{}^2 + ... + b_n{}^2}$$

The dot product of $\vec{a}$ and $\vec{b}$ is denoted by $\vec{a} \cdot \vec{b}$ and is calculated by the following formula:

$$\vec{a} \cdot \vec{b} = a_1 \ b_1 + a_2 \ b_2 + ... + a_n \ b_n$$

The angle between a and b is denoted by θ and is calculated by the following formula:

$$\theta = \cos^{-1} \left(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \right)$$

The angle is calculated in any angular mode. When calculated in degrees, decimal degrees are assumed.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	31	↑
02	15 02	g x^2
03	23 51 01	STO + 1
04	22	R↓
05	21	x⇄y
06	31	↑
07	15 02	g x^2
08	23 51 00	STO + 0
09	22	R↓
10	61	x
11	23 51 02	STO + 2
12	13 00	GTO 00
13	24 02	RCL 2
14	24 00	RCL 0
15	24 01	RCL 1
16	61	x
17	14 02	f $\sqrt{x}$
18	71	÷
19	15 05	g COS^{-1}
20	13 00	GTO 00
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS	
R_0	$\Sigma a_i{}^2$
R_1	$\Sigma b_i{}^2$
R_2	$\Sigma a_i b_i$
R_3	
R_4	
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Perform for i = 1,..., n:						
	Key in a_i and b_i	a_i	↑				
		b_i	R/S				
4	Find norm of $\vec{a}$		RCL	0	f	$\sqrt{x}$	$\lvert\vec{a}\rvert$
5	Find norm of $\vec{b}$		RCL	1	f	$\sqrt{x}$	$\lvert\vec{b}\rvert$
6	Find $\vec{a}\cdot\vec{b}$		RCL	2			$\vec{a}\cdot\vec{b}$
7	Compute angle between $\vec{a}$ and $\vec{b}$		GTO	13	R/S		θ

Example:

Let $A = (2, 5, 2)$
$B = (3, 3, -4)$

Solution:

$|\vec{a}| = 5.74$

$|\vec{b}| = 5.83$

$\vec{a} \cdot \vec{b} = 13.00$

$\theta = 67.16^\circ$

SIMULTANEOUS EQUATIONS IN TWO UNKNOWNS

Let $ax + by = e$

and $cx + dy = f$

be a system of two equations in two unknowns. Cramer's Rule is used to find the solution.

$$x = \frac{\begin{vmatrix} e & b \\ f & d \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{ed - bf}{ad - bc} \qquad y = \frac{\begin{vmatrix} a & e \\ c & f \end{vmatrix}}{\begin{vmatrix} a & b \\ c & d \end{vmatrix}} = \frac{af - ec}{ad - bc}$$

If $ad - bc = 0$ the calculator displays *Error*. In this case no solution or no unique solution exists.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	24 03	RCL 3
02	24 05	RCL 5
03	61	x
04	24 02	RCL 2
05	24 06	RCL 6
06	61	x
07	41	–
08	24 01	RCL 1
09	24 05	RCL 5
10	61	x
11	24 02	RCL 2
12	24 04	RCL 4
13	61	x
14	41	–
15	23 00	STO 0
16	71	÷
17	74	R/S
18	24 01	RCL 1
19	24 06	RCL 6
20	61	x
21	24 03	RCL 3
22	24 04	RCL 4
23	61	x
24	41	–

DISPLAY		KEY ENTRY
LINE	CODE	
25	24 00	RCL 0
26	71	÷
27	13 00	GTO 00
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 ad – bc
R_1 a
R_2 b
R_3 e
R_4 c
R_5 d
R_6 f
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store constants	a	STO	1			
		b	STO	2			
		e	STO	3			
		c	STO	4			
		d	STO	5			
		f	STO	6			
3	Find x and y		f	PRGM	R/S		x
			R/S				y
4	For new case, go to step 2.						

Example:

$5x - 3y = 12$

$2x + y = 9$

Solution:

$x = 3.55$

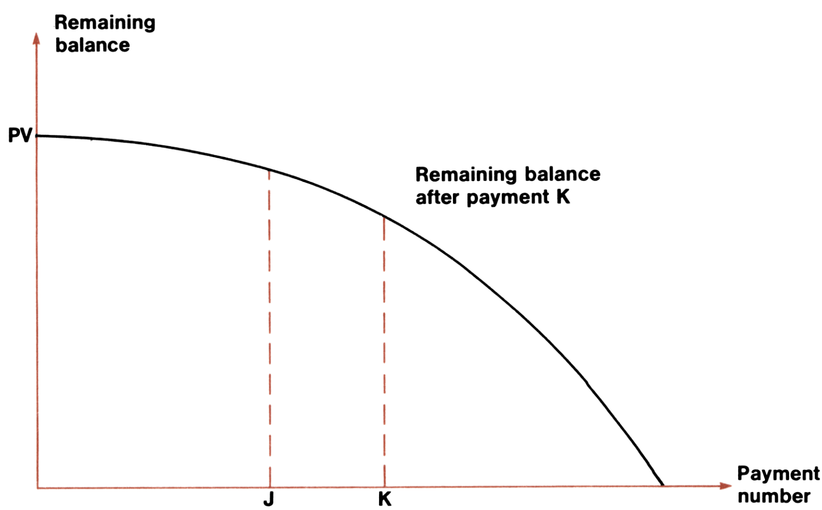
$y = 1.91$

CHAPTER 2 FINANCE

Because many of the finance programs have certain quantities in common, a word about these variables and the names used to refer to them may be helpful.

Five main variables recur in finance problems: n, i, PMT, PV, and FV. The first of these, n, denotes the total number of periods. The periodic interest rate i must be expressed in these programs as a decimal. Thus an annual interest rate of 6% is expressed as 0.06, which as a monthly rate would be $0.06/12 = 0.005$. PMT refers to the amount of the periodic payment. The present value, PV, is the value occurring at the beginning of the first period, while the future value, FV, is the value at the end of the last period.

MORTGAGE LOAN
ACCUMULATED INTEREST/REMAINING BALANCE



As one enters into the realm of financial calculations, one of the most striking revelations is how much of the repayment of a loan goes to interest. A new homeowner, for example, sends off his first monthly installment of \$220.13 toward repayment of a 30-year, \$30,000 mortgage assumed at 8% annual interest. With a proud sigh and a swelling chest, the homeowner mentally checks \$220 off the \$30,000 and figures he's well on his way. Right? Well, not quite. In fact, \$200 of that payment will go to interest, and only \$20.13 to reducing the principal of the loan.

This program will allow the user to calculate the amount paid to interest, for one payment or over a number of payments, as well as the amount of principal still unpaid, i.e., the remaining balance. The user must input the following values: the initial amount of the loan, the periodic interest rate, and the periodic payment amount. He must then key in a beginning payment number, J, and an ending payment number, K. The program will compute the accumulated interest charge from payment J through payment K, inclusive, and the balance remaining after payment K. If one wishes to find the amount of interest paid in a single payment, he can simply set K = J.

The program can also be used to generate a limited amortization schedule showing the balance remaining after successive payments. This can be done by leaving J = 1 and increasing K by 1 at each iteration. Outputs will be the total amount paid to interest over the first K payments, and the balance remaining after payment K.

Equations:

$$BAL_K = \frac{1}{(1+i)^{-K}} \left[PMT \, \frac{(1+i)^{-K} - 1}{i} + PV \right]$$

$$Int_{J-K} = BAL_K - BAL_{J-1} + (K - J + 1)\, PMT$$

where BAL_n = remaining balance after payment n

Int_{J-K} = accumulated interest, payments J through K

PV = initial loan amount

PMT = periodic payment amount

i = periodic interest rate

Notes:

1. The periodic interest rate i must be entered as a decimal. For example, for monthly payments with an annual interest rate of 9%, the periodic interest rate should be input as $i = \frac{.09}{12} = 0.0075$.

2. The use of this program is not restricted to mortgage loans, but applies equally well to any loan which is being repaid with equal periodic payments.

Programming Remarks:

In many finance programs, the expressions $(1 + i)$ and $(1 + i)^n$ are used several times per program. It is often simpler to calculate the quantity once and then store it for later use, rather than calculate it anew each time. In this program, the values of $(1+i)^{-K}$ and $(1+i)^{-J}$ are calculated once and then stored in R_7, thus saving both program steps and execution time. The same principle, of course, applies to other expressions in other problems.

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS
LINE	CODE						
00							
01	24 01	RCL 1	i				Calculate BAL_K
02	01	1	1	i			
03	51	+	1 + i				
04	24 05	RCL 5	K	1 + i			
05	32	CHS	-K	1 + i			
06	14 03	f y^x	$(1+i)^{-K}$				
07	23 07	STO 7	$(1+i)^{-K}$				
08	01	1	1	$(1+i)^{-K}$			
09	41	−	$(1+i)^{-K}-1$				
10	24 01	RCL 1	i	$(1+i)^{-K}-1$			
11	71	÷	s				Let $s = [(1+i)^{-K}-1] \div i$
12	24 02	RCL 2	PMT	s			
13	61	x	PMT s				
14	24 03	RCL 3	PV	PMT s			
15	51	+	PMT s + PV				
16	24 07	RCL 7	$(1+i)^{-K}$	PMT s + PV			
17	71	÷	BAL_K				
18	23 06	STO 6	BAL_K				
19	24 01	RCL 1	i	BAL_K			Calculate BAL_{J-1}
20	01	1	1	i	BAL_K		
21	51	+	(1 + i)	BAL_K			
22	24 04	RCL 4	J	(1 + i)	BAL_K		
23	01	1	1	J	(1 + i)	BAL_K	
24	41	−	J − 1	(1 + i)	BAL_K	BAL_K	
25	32	CHS	− (J − 1)	(1 + i)	BAL_K	BAL_K	
26	14 03	f y^x	$(1+i)^{-(J-1)}$	BAL_K	BAL_K	BAL_K	
27	23 07	STO 7	$(1+i)^{1-J}$	BAL_K	BAL_K	BAL_K	
28	01	1	1	$(1+i)^{1-J}$	BAL_K	BAL_K	
29	41	−	$(1+i)^{1-J}-1$	BAL_K	BAL_K	BAL_K	
30	24 01	RCL 1	i	$(1+i)^{1-J}-1$	BAL_K	BAL_K	
31	71	÷	s	BAL_K	BAL_K	BAL_K	Let $s = [(1+i)^{1-J}-1] \div i$
32	24 02	RCL 2	PMT	s	BAL_K	BAL_K	
33	61	x	PMT s	BAL_K	BAL_K	BAL_K	
34	24 03	RCL 3	PV	PMT s	BAL_K	BAL_K	
35	51	+	PMT s + PV	BAL_K	BAL_K	BAL_K	
36	24 07	RCL 7	$(1+i)^{1-J}$	PMT s + PV	BAL_K	BAL_K	
37	71	÷	BAL_{J-1}	BAL_K	BAL_K	BAL_K	
38	41	−	Diff	BAL_K	BAL_K	BAL_K	Diff = $BAL_K - BAL_{J-1}$
39	24 05	RCL 5	K	Diff	BAL_K	BAL_K	K − J + 1 gives no. PMT's
40	24 04	RCL 4	J	K	Diff	BAL_K	from J through K
41	41	−	K−J	Diff	BAL_K	BAL_K	
42	01	1	1	K−J	Diff	BAL_K	
43	51	+	K − J + 1	Diff	BAL_K	BAL_K	
44	24 02	RCL 2	PMT	m	Diff	BAL_K	m = K − J + 1
45	61	x	m PMT	Diff	BAL_K	BAL_K	m PMT is $ paid, J−K
46	51	+	Int_{J-K}	BAL_K	BAL_K	BAL_K	Display Int_{J-K}
47	74	R/S	Int_{J-K}	BAL_K	BAL_K	BAL_K	
48	21	x⇄y	BAL_K	Int_{J-K}	BAL_K	BAL_K	Display BAL_K
49	13 00	GTO 00	BAL_K	Int_{J-K}	BAL_K	BAL_K	

REGISTERS	
R_0	
R_1	i
R_2	PMT
R_3	PV
R_4	J
R_5	K
R_6	BAL_K
R_7	$(1+i)^{-n}$

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store the following variables:						
	Periodic interest (decimal)	i	STO	1			
	Periodic payment	PMT	STO	2			
	Initial loan amount	PV	STO	3			
	Starting payment number	J	STO	4			
	Ending payment number	K	STO	5	f	PRGM	
3	Compute accumulated interest						
	from payments J through K.		R/S				$Int_{J\text{-}K}$
4	Display remaining balance after						
	payment K		R/S				BAL_K
5	To change any variable, store						
	the new value in the appropriate						
	register and go to step 3.						

Example:

A mortgage is arranged so that the first payment is made at the end of October, 1974 (i.e., October is payment period 1). It is a $25,000 loan at 8% with monthly payments of $200. What is the accumulated interest for 1974 (periods 1–3) and for 1975 (periods 4–15) and what balance remains at the end of each year? Also, generate a schedule of interest paid and remaining balance for the first 5 years of the mortgage (periods 12, 24, 36, 48, 60).

Solution:

(Notice that i must be entered as a decimal, monthly rate.)

.08 [↑] 12 [÷] [STO] [1] 200 [STO] [2] 25000 [STO] [3] 1
[STO] [4] 3 [STO] [5] [f] [PRGM] [R/S] ——→ 499.33
(interest paid in 1974)

[R/S] ——→ 24899.33
(remaining balance at end of 1974)

4 [STO] [4] 15 [STO] [5] [R/S] ——→ 1976.65
(interest paid in 1975)

[R/S] ——→ 24475.98
(remaining balance at end of 1975)

Now, generate the amortization schedule:

1 STO 4 12 STO 5 R/S ⟶ 1985.00
(interest thru 1st year)

R/S ⟶ 24585.00
(remaining balance after 1st year)

24 STO 5 R/S ⟶ 3935.56
(interest thru 2nd year)

R/S ⟶ 24135.56
(remaining balance after 2nd year)

36 STO 5 R/S ⟶ 5848.81
(interest thru 3rd year)

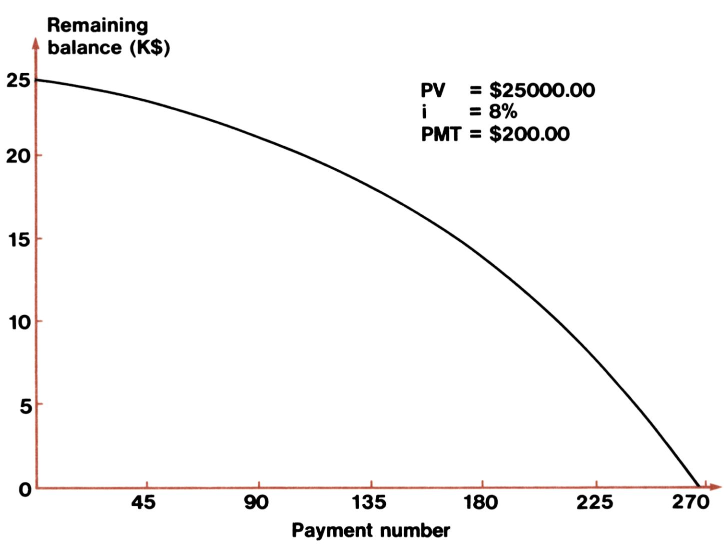
R/S ⟶ 23648.81
(remaining balance after 3rd year)

48 STO 5 R/S ⟶ 7721.67
(interest thru 4th year)

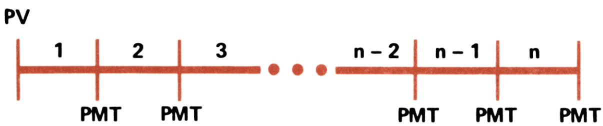
R/S ⟶ 23121.67
(remaining balance after 4th year)

60 STO 5 R/S ⟶ 9550.77
(interest thru 5th year)

R/S ⟶ 22550.77
(remaining balance after 5th year)



MORTGAGE LOAN
PAYMENT, PRESENT VALUE, NUMBER OF PERIODS



For a loan which is being repaid with equal periodic payments, this program will calculate the payment amount, the present value, or the number of periods of the loan, given the periodic interest rate and the two other variables.

Remember that the periodic interest rate i must be expressed as a decimal, e.g., 6% is represented as 0.06.

The equations used are as follows:

$$PMT = PV\left[\frac{i}{1-(1+i)^{-n}}\right] \qquad PV = PMT\left[\frac{1-(1+i)^{-n}}{i}\right]$$

$$n = -\frac{\ln(1 - i\,PV/PMT)}{\ln(1+i)}$$

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	01	1
02	24 02	RCL 2
03	01	1
04	51	+
05	24 01	RCL 1
06	32	CHS
07	14 03	f y^x
08	41	–
09	24 02	RCL 2
10	21	x⇄y
11	71	÷
12	24 04	RCL 4
13	61	x
14	13 00	GTO 00
15	01	1
16	24 02	RCL 2
17	01	1
18	51	+
19	24 01	RCL 1
20	32	CHS
21	14 03	f y^x
22	41	–
23	24 02	RCL 2
24	71	÷

DISPLAY		KEY ENTRY
LINE	CODE	
25	24 03	RCL 3
26	61	x
27	13 00	GTO 00
28	01	1
29	24 04	RCL 4
30	24 03	RCL 3
31	71	÷
32	24 02	RCL 2
33	61	x
34	41	–
35	14 07	f LN
36	24 02	RCL 2
37	01	1
38	51	+
39	14 07	f LN
40	71	÷
41	32	CHS
42	13 00	GTO 00
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1 n
R_2 i
R_3 PMT
R_4 PV
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	For payment	n	STO	1			
		i	STO	2			
		PV	STO	4			
			f	PRGM	R/S		PMT
3	For present value	n	STO	1			
		i	STO	2			
		PMT	STO	3			
			GTO	15	R/S		PV
4	For number of payments	i	STO	2			
		PMT	STO	3			
		PV	STO	4			
			GTO	28	R/S		n
5	For new case, go to step 2, 3, or						
	4.						

Examples:

1. What monthly payment is required to amortize a \$3000 loan at 9.5% (.095) in 36 months?

2. You are willing to pay \$175 per month for 24 months on a 9.5% loan. How much can you borrow?

3. How many months will it take to pay off a \$4000 loan if your monthly payment is \$200 and the annual interest rate is 9.5%?

Solutions:

Divide 0.095 by 12 to find the monthly interest rate expressed as a decimal.

1. \$96.10

2. \$3811.43

3. 21.86 months

MORTGAGE LOAN INTEREST RATE

This program will calculate the interest rate on a loan with equal periodic payments. The user must specify the number of periods, the present value or initial loan amount, and the payment amount.

The program performs an iterative solution for i using Newton's method:

$$i_{k+1} = i_k - \frac{f(i_k)}{f'(i_k)}$$

where

$$f(i) = \frac{1-(1+i)^{-n}}{i} - \frac{PV}{PMT}.$$

The initial guess for i is given by

$$i_o = \frac{PMT}{PV} - \frac{PV}{n^2\ PMT}.$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 03	RCL 3
02	31	↑
03	15 22	g 1/x
04	21	x⇄y
05	24 01	RCL 1
06	15 02	g x^2
07	71	÷
08	41	−
09	23 02	STO 2
10	24 03	RCL 3
11	24 02	RCL 2
12	61	x
13	01	1
14	24 02	RCL 2
15	01	1
16	51	+
17	24 01	RCL 1
18	32	CHS
19	14 03	f y^x
20	23 05	STO 5
21	41	−
22	41	−
23	24 01	RCL 1
24	24 02	RCL 2

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	15 22	g 1/x
26	01	1
27	51	+
28	71	÷
29	01	1
30	51	+
31	24 05	RCL 5
32	61	x
33	01	1
34	41	−
35	24 02	RCL 2
36	71	÷
37	71	÷
38	23 51 02	STO + 2
39	15 03	g ABS
40	33	EEX
41	06	6
42	32	CHS
43	14 41	f x<y
44	13 10	GTO 10
45	24 02	RCL 2
46	13 00	GTO 00
47		
48		
49		

REGISTERS
R_0
R_1 n
R_2 i
R_3 PV/PMT
R_4 $(1+i)^{-n}$
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store number of payments	n	STO	1			
3	Key in present value and pay-						
	ment amount	PV	↑				
		PMT	÷	STO	3		PV/PMT
4	Compute interest		f	PRGM	R/S		i (decimal)
			EEX	2	x		i (%)
5	For new case, go to step 2.						

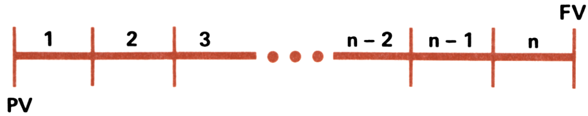
Example:

You recently obtained a $2500 car loan for 36 months. If your monthly payment is $86.67, what is the annual percentage rate?

Solution:

15.01%

COMPOUND AMOUNT



This program applies to an amount of principal that has been placed into an account and compounded periodically, with no further deposits. The important variables in this case are the number of compounding periods n, the periodic interest rate i, the principal or present value PV, the future value of the account FV, and the amount of interest accrued I. Any of these may be calculated from the others by these formulas:

$$n = \frac{\ln (FV/PV)}{\ln (1 + i)} \qquad i = \left(\frac{FV}{PV}\right)^{1/n} - 1 \qquad PV = FV\,(1 + i)^{-n}$$

$$FV = PV\,(1 + i)^{n} \qquad I = PV\,[(1 + i)^{n} - 1]$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 05	RCL 5
02	24 04	RCL 4
03	71	÷
04	14 07	f LN
05	24 02	RCL 2
06	01	1
07	51	+
08	14 07	f LN
09	71	÷
10	13 00	GTO 00
11	24 05	RCL 5
12	24 04	RCL 4
13	71	÷
14	24 01	RCL 1
15	15 22	g 1/x
16	14 03	f y^x
17	01	1
18	41	–
19	13 00	GTO 00
20	24 02	RCL 2
21	01	1
22	51	+
23	24 01	RCL 1
24	32	CHS

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	14 03	f y^x
26	24 05	RCL 5
27	61	x
28	13 00	GTO 00
29	24 02	RCL 2
30	01	1
31	51	+
32	24 01	RCL 1
33	14 03	f y^x
34	24 04	RCL 4
35	61	x
36	13 00	GTO 00
37	24 02	RCL 2
38	01	1
39	51	+
40	24 01	RCL 1
41	14 03	f y^x
42	01	1
43	41	–
44	24 04	RCL 4
45	61	x
46	13 00	GTO 00
47		
48		
49		

REGISTERS
R_0
R_1 n
R_2 i
R_3
R_4 PV
R_5 FV
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	To compute number of periods	i (decimal)	STO	2			
		PV	STO	4			
		FV	STO	5			
			f	PRGM	R/S		n
3	To compute periodic interest						
	rate	n	STO	1			
		PV	STO	4			
		FV	STO	5			
			GTO	11	R/S		i (decimal)
4	To compute principal	n	STO	1			
		i (decimal)	STO	2			
		FV	STO	5			
			GTO	20	R/S		PV
5	To compute future value	n	STO	1			
		i (decimal)	STO	2			
		PV	STO	4			
			GTO	29	R/S		FV
6	To compute accrued interest	n	STO	1			
		i (decimal)	STO	2			
		PV	STO	4			
			GTO	37	R/S		I
7	For new case, go to step 2, 3, 4,						
	5, or 6.						

Examples:

1. Assuming an annual inflation rate of 10%, how long will it take prices to double? (Suggestion: let PV = 1, FV = 2)

2. Find the rate of return on \$1000 compounded quarterly if it amounts to \$1500 in 5 years.

3. How much will you need to invest today at 5 3/4% compounded quarterly to have \$3000 in 5 years?

4. What is the future value of \$2000 invested at 5 3/4% compounded quarterly for 4 years (16 quarters)?

5. How much interest do you receive on \$1500 deposited for 10 years if interest at 5 1/2% is compounded annually?

Solutions:

1. 7.27 years
2. .0205 quarterly = 8.19% annually
3. \$2255.02 ($i = 0.0575/4$)
4. \$2513.08 ($i = 0.0575/4$)
5. \$1062.22 ($i = 0.055$)

PERIODIC SAVINGS
PAYMENT, FUTURE VALUE, NUMBER OF PERIODS

This program calculates payment, future value, or number of time periods for a schedule of periodic payments into a savings account, given the interest rate and two of the three other variables. Remember that i must be input as a decimal, e.g., 6% is expressed as 0.06.

Then n, PMT, or FV may be calculated from the following formulas:

$$n = \frac{\ln\left[\dfrac{FV\, i}{PMT} + (1+i)\right]}{\ln(1+i)} - 1 \qquad PMT = \frac{FV\, i}{(1+i)^{n+1} - (1+i)}$$

$$FV = \frac{PMT}{i}\left[(1+i)^{n+1} - (1+i)\right]$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 02	RCL 2
02	24 05	RCL 5
03	61	x
04	24 03	RCL 3
05	71	÷
06	24 02	RCL 2
07	01	1
08	51	+
09	23 00	STO 0
10	51	+
11	14 07	f LN
12	24 00	RCL 0
13	14 07	f LN
14	71	÷
15	01	1
16	41	–
17	13 00	GTO 00
18	24 05	RCL 5
19	24 02	RCL 2
20	61	x
21	24 02	RCL 2
22	01	1
23	51	+
24	71	÷

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	14 73	f LASTx
26	24 01	RCL 1
27	14 03	f y^x
28	01	1
29	41	–
30	71	÷
31	13 00	GTO 00
32	24 03	RCL 3
33	24 02	RCL 2
34	01	1
35	51	+
36	61	x
37	14 73	f LASTx
38	24 01	RCL 1
39	14 03	f y^x
40	01	1
41	41	–
42	61	x
43	24 02	RCL 2
44	71	÷
45	13 00	GTO 00
46		
47		
48		
49		

REGISTERS
R_0 (1 + i)
R_1 n
R_2 i
R_3 PMT
R_4
R_5 FV
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	To compute number of pay-						
	ments	i (decimal)	STO	2			
		PMT	STO	3			
		FV	STO	5			
			f	PRGM	R/S		n
3	To compute periodic payment						
	amount	n	STO	1			
		i (decimal)	STO	2			
		FV	STO	5			
			GTO	18	R/S		PMT
4	To compute future value	n	STO	1			
		i (decimal)	STO	2			
		PMT	STO	3			
			GTO	32	R/S		FV
5	For new case, go to step 2, 3,						
	or 4.						

Examples:

1. How long will it take to save \$15,000 if you are making quarterly deposits of \$400 at 6% annual interest?

2. You will need \$10,000 in 7 years. How large a monthly payment do you need to make if the annual interest rate is 6 1/2%?

3. How much money will a person have if he deposits \$150 at the end of each month for a period of 3 years? He receives 6% annual interest.

Solutions:

1. 29.62 quarters or 7.40 years (i = .06/4)
2. \$93.82 (n = 84, i = .065/12)
3. \$5929.92 (n = 36, i = .06/12)

DISCOUNTED CASH FLOW
NET PRESENT VALUE, INTERNAL RATE OF RETURN

The primary purpose of this program is to compute the net present value of a series of cash flows. In general, an initial investment V_0 is made in some enterprise which is expected to bring in periodic cash flows $C_1, C_2, \ldots, C_n$. Given a discount rate i, which must be entered as a decimal, then for each cash flow C_k, the program will compute the net present value at period k, NPV_k. A negative value for NPV_k indicates that the enterprise has not yet been profitable. A positive NPV_k means that the enterprise has been profitable, to the extent that a rate of return i on the original investment has been exceeded.

The program may also be used iteratively to calculate an internal rate of return. The objective here is to find the discount rate i which will make the final net present value, NPV_n, equal to zero. The procedure, then, is to store V_0 and a first guess at the rate of return i, input the cash flows C_1 through C_n, and thus find NPV_n. If NPV_n is negative, the estimated rate of return was too high; if NPV_n is positive, the estimate for i was too low. Adjust the estimate for i accordingly, store the new i, and input the cash flows again. Inspect the new value of NPV_n to obtain a new estimate for i and repeat the process. The entire procedure is repeated until NPV_n is zero, or very close to it. The last value of i input is then regarded as the internal rate of return.

Each figure for net present value is found by

$$NPV_k = -V_0 + \sum_{j=1}^{k} \frac{C_j}{(1+i)^j}$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 01	RCL 1
02	01	1
03	23 04	STO 4
04	51	+
05	23 02	STO 2
06	71	÷
07	24 00	RCL 0
08	41	−
09	24 04	RCL 4
10	14 74	f PAUSE
11	21	x⇄y
12	23 03	STO 3
13	74	R/S
14	24 02	RCL 2
15	24 04	RCL 4
16	01	1
17	51	+
18	23 04	STO 4
19	14 03	f y^x
20	71	÷
21	24 03	RCL 3
22	51	+
23	13 09	GTO 09
24		

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 V_0
R_1 i
R_2 (1 + i)
R_3 NPV_k
R_4 k
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store initial investment and						
	discount rate	V_0	STO	0			
		i (decimal)	STO	1	f	PRGM	
3	Perform for k = 1,..., n:						
	Input C_k and compute NPV_k	C_k	R/S				(k)
							NPV_k
4	For new case, go to step 2.						

Example:

You have been offered an investment opportunity for $150,000 at a capital cost of 10% after taxes. Based on the following cash flows, will this investment be profitable?

Year	Cash Flow
1	$30,000
2	26,300
3	50,000
4	55,600
5	45,200

Solutions:

Remember to enter i as 0.10.

NPV_1 = –$122,727.27

NPV_2 = –$100,991.74

NPV_3 = –$63,426.00

NPV_4 = –$25,450.45

NPV_5 = $2,615.20

Since C_5 is positive the cash flow is profitable to the extent that the cost of capital is 10%.

CALENDAR
DAY OF THE WEEK
DAYS BETWEEN TWO DATES

This program will compute the day of the week for a given date, or the number of days between two dates, for any dates from March 1, 1700, to February 28, 2100. The program works by assigning the number 1 to March 1, 1700, and a corresponding number to each succeeding day. When computing day of the week, a 0 represents Sunday, 1 Monday, 2 Tuesday, etc.

Thus for month m, day d, year y, the number N assigned to that date is

$$N(m, d, y) = [365.25\ g(y,m)] + [30.6\ f(m)] + D - 621049$$

where

$$g(y, m) = \begin{cases} y - 1 \text{ if } m = 1 \text{ or } 2 \\ y \text{ if } m > 2 \end{cases} \quad \text{and } f(m) = \begin{cases} m + 13 \text{ if } m = 1 \text{ or } 2 \\ m + 1 \text{ if } m > 2 \end{cases}$$

[m] represents the integer function, f INT. E.g., [6.34] = 6.

Note:

For days from March 1, 1700, to February 28, 1800, 2 days must be added to the value for N calculated by the program. For days from March 1, 1800, to February 28, 1900, 1 day must be added.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	03	3
02	24 01	RCL 1
03	14 41	f x<y
04	13 09	GTO 09
05	01	1
06	51	+
07	24 03	RCL 3
08	13 15	GTO 15
09	01	1
10	03	3
11	51	+
12	24 03	RCL 3
13	01	1
14	41	–
15	03	3
16	06	6
17	05	5
18	73	•
19	02	2
20	05	5
21	61	x
22	14 01	f INT
23	21	x⇄y
24	03	3

DISPLAY		KEY ENTRY
LINE	CODE	
25	00	0
26	73	•
27	06	6
28	61	x
29	14 01	f INT
30	51	+
31	24 02	RCL 2
32	51	+
33	06	6
34	02	2
35	01	1
36	00	0
37	04	4
38	09	9
39	41	–
40	74	R/S
41	07	7
42	71	÷
43	15 01	g FRAC
44	07	7
45	61	x
46	13 00	GTO 00
47		
48		
49		

REGISTERS
R_0
R_1 Month
R_2 Day
R_3 Year
R_4
R_5
R_6
R_7 Temporary

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store month	m	STO	1			
	day	d	STO	2			
	year	y	STO	3			
3	Compute N(m, d, y)		f	PRGM	R/S		N(m, d, y)
4	For day of week, go to step 8						
5	For days between dates, store						
	first N		STO	7			
6	Repeat steps 2 and 3 for second						
	date, then		RCL	7	–		# Days
7	For new case, go to step 2.						
8	For day of week (0 = Sunday)		R/S				Day (0,..., 6)
9	For new case, go to step 2.						

Examples:

1. What day of the week was July 4, 1776?

2. Find the number of days between March 27, 1948, and April 7, 1975.

Solutions:

1. Thursday (4). (Remember to add 2 days.)

2. 9872.

Notes

CHAPTER 3 GAMES

MOON LANDING SIMULATOR

Imagine for a moment the difficulties involved in landing a rocket on the moon with a strictly limited fuel supply. You're coming down tail-first, free-falling toward a hard rock surface. You'll have to ignite your rockets to slow your descent; but if you burn too much too soon, you'll run out of fuel 100 feet up, and then you'll have nothing to look forward to but cold eternal moon dust coming faster every second. The object, clearly, is to space your burns just right so that you will alight on the moon's surface with no downward velocity.

The game starts off with the rocket descending at a velocity of 50 feet/sec from a height of 500 feet. The velocity and height are shown in a combined display as –50.0500, the height appearing to the right of the decimal point and the velocity to the left, with a negative sign on the velocity to indicate downward motion. If a velocity is ever displayed with no fractional part, for example, –15., it means that you have crashed at a speed of 15 feet/sec. In game terms, this means that you have lost; in real-life, it signifies an even less favorable outcome.

You will start the game with 120 units of fuel. You may burn as much or as little of your available fuel as you wish at each step of your descent; burns of zero are quite common. A burn of 5 units will just cancel gravity and hold your speed constant. Any burn over 5 will act to change your speed in an upward direction. You must take care, however, not to burn more fuel than you have; for if you do, no burn at all will take place, and you will free-fall to your doom! The final velocity shown will be your impact velocity (generally rather high). You may display your remaining fuel at any time by recalling R_2.

Equations:

We don't want to get too specific, because that would spoil the fun of the game; but rest assured that the program is solidly based on some old friends from Newtonian physics:

$$x = x_0 + v_0 t + \frac{1}{2} at^2 \qquad v = v_0 + at \qquad v^2 = v_0{}^2 + 2ax$$

where x, v, a, and t are distance, velocity, acceleration, and time.

Notes:

1. If you crash before running out of fuel, the crash velocity shown will be the velocity before the burn, rather than the impact velocity.
2. Use only integer values for burns. Any decimal entry will cause an error in the display for V.X.

Programming Remarks:

An interesting feature of this program is the simultaneous display of both speed and altitude (V.X), as for example, –50.0500. This is accomplished by storing the speed V and the altitude X in their normal form (–50.00, 500.00), then dividing X by 10,000 (10^4) before combining them. An additional subtlety involves the question of the sign of V, and whether ($X/10^4$) is to be added to or subtracted from V. For example, if V = –50 and X = 500, we should subtract: V – ($X/10^4$), in order to generate a display of –50.0500. But if V = 10 and X = 50, we should add: V = V + ($X/10^4$) in order to display 10.0050. Inspection of the program listing, lines 2 through 12, will reveal how a conditional branch was used to resolve the dilemma.

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS
LINE	CODE						
00							
01	14 11 04	f FIX 4					Four-place display
02	24 00	RCL 0	X				Form display V.X
03	33	EEX	1. 00	X			
04	04	4	1. 04	X			
05	71	÷	$X/10^4$				Divide X by 10,000
06	24 01	RCL 1	V	$X/10^4$			
07	15 41	g x<0	V	$X/10^4$			Is V negative?
08	13 11	GTO 11	V	$X/10^4$			Yes, branch
09	51	+	$V + X/10^4$				No, add V and X
10	13 13	GTO 13	$V + X/10^4$				
11	21	x⇄y	$X/10^4$	V			V<0, add V and –X
12	41	–	$V - X/10^4$				
13	74	R/S	V.X				V.X is $V \pm (X/10^4)$
14	24 02	RCL 2	F	B			Burn B has been input
15	14 41	f x<y	F	B			Burn > Fuel?
16	13 34	GTO 34	F	B			Yes, prepare to crash
17	22	R↓	B			F	No, update A, X, V
18	23 41 02	STO – 2	B			F	Subtract burn from fuel
19	05	5	5	B			5 units cancels gravity
20	41	–	B – 5				Acceleration = B – 5
21	23 03	STO 3	A				
22	02	2	2	A			
23	71	÷	A/2				
24	24 00	RCL 0	X	A/2			
25	51	+	X + A/2				
26	24 01	RCL 1	V	X + A/2			
27	51	+	X + V + A/2				New altitude: X←X+V+A/2
28	23 00	STO 0	X				
29	15 41	g x<0	X				Is X below ground?
30	13 44	GTO 44	X				Yes, you've crashed
31	24 03	RCL 3	A	X			No, update V
32	23 51 01	STO + 1	A	X			New velocity: V ← V + A
33	13 02	GTO 02	A	X			Display V.X
34	24 01	RCL 1	V				All fuel gone, show
35	15 02	g x^2	V^2				crash velocity as
36	24 00	RCL 0	X	V^2			$V = (V^2 + 2gX)^{1/2}$
37	01	1	1	X	V^2		where g = gravity = 5
38	00	0	10	X	V^2		
39	61	x	10 X	V^2			
40	51	+	V^2 + 10 X				
41	14 02	f $\sqrt{x}$	V				
42	32	CHS	V				Show crash V down
43	23 01	STO 1	V				
44	24 01	RCL 1	V				Come here from line 30
45	14 11 00	f FIX 0	V				Display integer V to
46	13 00	GTO 00	V				show crash
47							
48							
49							

REGISTERS	
R_0	X
R_1	V
R_2	Fuel
R_3	Acceleration
R_4	
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize	X	500	STO	0		500.00
		V	50	CHS	STO	1	-50.00
		Fuel	120	STO	2		120.00
3	Display initial V.X		f	PRGM	R/S		-50.0500
4	Key in burn, compute new speed						
	and distance	Burn	R/S				V.X
5	Perform step 4 till you land or						
	crash						
6	To see remaining fuel at any						
	time		RCL	2			Fuel
7	To display speed and distance						
	at any time		f	PRGM	R/S		V.X
8	To start a new game, go to step 2.						

Example:

500 STO 0 50 CHS STO 1 120 STO 2

f PRGM R/S → –50.0500

0 R/S → –55.0448

5 R/S → –55.0393

(note constant V when burn = 5)

30 R/S → –30.0350

0 R/S → –35.0318

0 R/S → –40.0280

0 R/S → –45.0238

0 R/S → –50.0190

RCL 2 → 85.0000
(remaining fuel)

f PRGM R/S → –50.0190
(display V.X again)

10 R/S → –45.0143

0 R/S → –50.0095

RCL 2 → 75.0000

10 R/S → –45.0048

25 R/S → –25.0013

20 R/S → –25.

Oops.

NIMB

The game of Nimb begins with a collection of N objects, or as the calculator plays it, with the positive number N. Each player alternately subtracts one, two, or three from the total until only one is left. The player forced to take the last one loses.

To begin the game, you must tell the machine how many objects to start with, i.e., the value of N. A reasonable number is 15. After each move the machine will display the remaining total. A negative sign indicates that it is the user's move next, while a positive display indicates that it is the HP-25's move.

As the challenger you are allowed to make the first move. It is possible to win but of course the HP-25 is a master player: it will not let you make an error and win. (Not, that is, unless you cheat and take a number other than 1, 2, or 3—a contingency so far beyond the realm of the HP-25's naive faith in humankind that the unsuspecting calculator has no way of knowing if you do or don't.)

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	31	↑
02	01	1
03	23 02	STO 2
04	22	R↓
05	23 41 00	STO − 0
06	24 00	RCL 0
07	15 71	g x=0
08	13 42	GTO 42
09	23 61 02	STO x 2
10	24 02	RCL 2
11	74	R/S
12	21	x⇄y
13	15 51	g x ≥ 0
14	13 17	GTO 17
15	21	x⇄y
16	13 02	GTO 02
17	01	1
18	32	CHS
19	23 02	STO 2
20	00	0
21	23 01	STO 1
22	24 01	RCL 1
23	03	3
24	14 71	f x=y

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	13 40	GTO 40
26	01	1
27	23 51 01	STO + 1
28	32	CHS
29	24 00	RCL 0
30	51	+
31	24 01	RCL 1
32	41	−
33	04	4
34	71	÷
35	15 01	g FRAC
36	15 61	g x≠0
37	13 22	GTO 22
38	24 01	RCL 1
39	13 05	GTO 05
40	01	1
41	13 05	GTO 05
42	24 02	RCL 2
43	15 41	g x < 0
44	13 47	GTO 47
45	24 03	RCL 3
46	13 00	GTO 00
47	24 04	RCL 4
48	14 11 01	f FIX 1
49	13 00	GTO 00

REGISTERS
R_0 Total
R_1 Machine move
R_2 ± Total
R_3 55178
R_4 3507.1
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize	55178	STO	3			
		3507.1	STO	4	f	PRGM	
3	Store total number of objects						
	(usually 15) and set display	N	STO	0	CHS	f	
			FIX	0			–N.
4	If number in display is negative,						
	key in your move	Your move	R/S				+ Total
5	If number in display is positive,						
	let HP-25 move		R/S				– Total
6	Perform steps 4 and 5 until game						
	is over						
7	At end of game, turn calculator						
	upside down to read message						
8	For another game, go to step 3.						

Example:

Perform the initialization with N = 15.

User takes 3.

3 R/S → 12.

R/S → –9.

HP-25 takes 3.

User takes 2.

2 R/S → 7.

R/S → –5.

HP-25 takes 2.

User takes 3.

3 R/S → 2.

R/S → –1.

HP-25 takes 1.

User takes last 1.

1 R/S → 55178.

Turn calculator upside down for message (BLISS).

TEACH ARITHMETIC

We at Hewlett-Packard feel that the hand-held calculator, far from threatening the traditional tenets of a sound mathematics education, may be used creatively to reinforce learning in such areas as arithmetic, algebra, geometry, trigonometry, calculus, and numerical analysis. This program, which is designed to be used in teaching children the four operations of elementary arithmetic (+, –, x, ÷), demonstrates some of the (largely unexplored) potential of the HP-25 as an educational tool.

The basic flow of the program is to pose a problem in arithmetic, check the answer that the user keys in against the correct answer, and then do one of two things: if the user's answer was correct, the program will go on to pose a new problem; if the keyed-in answer was wrong, the program restates the original problem to give the learner a second chance.

To run the program, the user must store a value called Max in R_0. This tells the program not to use any numbers as large as Max in its problems. If you specify a Max of 12, for example, then all the problems will deal with numbers between 0 and 11. The user must then store in R_1 a "seed" s, a number between 0 and 1, which will determine the sequence of problems that will appear. Different seeds generate different problems, thus ensuring that the learning game doesn't get boring. With the display format set to [f] [FIX] [2], the execution of the program will cause the first problem to be displayed as follows: the display will show one number to the left of the decimal place, and one number to the right. For example, the numbers 8 and 2 would be displayed as 8.02. The user may then choose what operation to perform on the two numbers: he may add (8 + 2), subtract (8 – 2), multiply (8 x 2), or divide (8 ÷ 2). After he keys in his answer and re-initiates program execution, the program will either display a new problem, if his answer was right, or display the same two numbers again, but this time with a negative sign in front (–8.02). The negative sign is an indication that the answer was incorrect, and does not denote a negative number. (All numbers in the problems are positive, though of course the results of some subtractions may be negative). If the problem reappears with a negative sign, the user should key in a different answer and try again. As soon as the correct answer is given, the program will go to display a new problem.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	24 01	RCL 1
02	15 73	g π
03	15 02	g x^2
04	61	x
05	15 01	g FRAC
06	23 01	STO 1
07	24 00	RCL 0
08	61	x
09	14 01	f INT
10	23 03	STO 3
11	24 01	RCL 1
12	15 73	g π
13	15 02	g x^2
14	61	x
15	15 01	g FRAC
16	23 01	STO 1
17	24 00	RCL 0
18	61	x
19	14 01	f INT
20	23 02	STO 2
21	24 03	RCL 3
22	33	EEX
23	02	2
24	71	÷

DISPLAY		KEY ENTRY
LINE	CODE	
25	51	+
26	23 04	STO 4
27	74	R/S
28	24 02	RCL 2
29	24 03	RCL 3
30	51	+
31	13 43	GTO 43
32	24 02	RCL 2
33	24 03	RCL 3
34	41	–
35	13 43	GTO 43
36	24 02	RCL 2
37	24 03	RCL 3
38	61	x
39	13 43	GTO 43
40	24 02	RCL 2
41	24 03	RCL 3
42	71	÷
43	14 71	f x=y
44	13 01	GTO 01
45	24 04	RCL 4
46	32	CHS
47	13 27	GTO 27
48		
49		

REGISTERS
R_0 Max
R_1 Random #
R_2 Left #
R_3 Right #
R_4 Problem
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store Max ($0 < \text{Max} \leqslant 100$)	Max	STO	0			
3	Store seed ($0 < s < 1$)	s	STO	1			
4	Set display format		f	FIX	2		
5	Generate a problem		f	PRGM	R/S		$n_1 . n_2$
6	Choose an operation and key in						
	your answer:						
	For addition (+)	$n_1 + n_2$	R/S				
	For subtraction (−)	$n_1 - n_2$	GTO	32	R/S		
	For multiplication (x)	$n_1 \times n_2$	GTO	36	R/S		
	For division (÷)	$n_1 \div n_2$	GTO	40	R/S		
7	If you were right, program will						
	display new problem; go to step						
	6.						$n_3 . n_4$
8	If you were wrong, program will						
	show same problem again; go to						
	step 6 again.						$-n_1 . n_2$
9	Repeat steps 6–8 as many times						
	as desired						
10	To change Max, go to step 2,						
	then to step 5.						

Example:

Let Max = 12 and the seed s = 0.725

Solution:

f PRGM R/S → 6.01

(6 + 1 = 7)

7 R/S → 8.03

(8 x 3 = 25)

25 GTO 3 6 R/S → −8.03

(Try again: 8 x 3 = 24)

24 GTO 3 6 R/S → 3.11

(3 − 11 = −8)

8 CHS GTO 3 2 R/S → 9.00

(9 + 0 = 9)

9 R/S → 2.05

etc.

Notes

CHAPTER 4 NAVIGATION

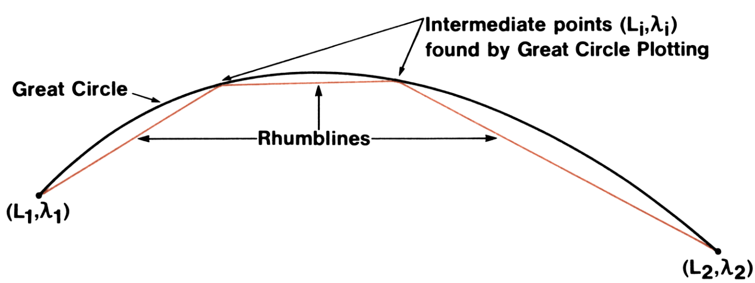
COURSE PLANNING—GREAT CIRCLE PLOTTING AND RHUMBLINE NAVIGATION

Long voyages by sea or air are generally made to follow one of two sorts of routes: a rhumbline or a great circle. The rhumbline is the path of constant heading between two points on the earth's surface; it intersects all lines of longitude at the same angle. It is also the course defined by the straight line between two points on a Mercator projection. It is a convenient course for navigation because its direction does not change, and for short distances at mid and low latitudes, the rhumbline is adequate for almost all calculations of course and distance.

Outside this range, a more efficient track is the great circle, which is always the shortest route between two points on a sphere. However, in order to follow a great circle, a vehicle must be continuously changing its course. Since this is at best inconvenient, if not impossible, several rhumblines are often used to approximate a great circle.

To plan a course using this technique, the navigator should first run the program Great Circle Plotting. For this program, the user must input the latitude and longitude of his starting point and his destination. Then, for any intermediate longitude λ_i that he specifies, the program will calculate the latitude L_i at which the great circle from source to destination will intersect the specified longitude. If several pairs of coordinates (L_i, λ_i) are calculated, then the next program, Rhumbline Navigation, may be used to find course and distance for the rhumblines linking these intermediate points along the great circle.

The inputs to Rhumbline Navigation are the coordinates of two points on the globe; outputs are the rhumbline course and distance from the first point to the second point. The program may be used alone, to determine the rhumbline from source to destination; or in conjunction with Great Circle Plotting, to compute several rhumblines to approximate a great circle.



GREAT CIRCLE PLOTTING

Equations:

$$L_i = \tan^{-1}\left[\frac{\tan L_2 \sin(\lambda_i - \lambda_1) - \tan L_1 \sin(\lambda_i - \lambda_2)}{\sin(\lambda_2 - \lambda_1)}\right]$$

where (L_1, λ_1) = coordinates of starting point

(L_2, λ_2) = coordinates of destination

(L_i, λ_i) = coordinates of intermediate point on great circle

Note:

The program does not compute along lines of longitude $(\lambda_1 = \lambda_2)$.

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS
LINE	CODE						
00			λ_i, D.MS				
01	15 00	g →H	λ_i, D.d				Convert λ_i to decimal deg.
02	23 04	STO 4	λ_i				
03	24 01	RCL 1	λ_1	λ_i			
04	41	–	$\lambda_i - \lambda_1$				
05	14 04	f SIN	$\sin_1$				$\sin_1 = \sin(\lambda_i - \lambda_1)$
06	24 02	RCL 2	L_2	$\sin_1$			
07	14 06	f TAN	$\tan_2$	$\sin_1$			$\tan_2 = \tan L_2$
08	61	x	$\tan_2 \sin_1$				
09	24 04	RCL 4	λ_i	$\tan_2 \sin_1$			
10	24 03	RCL 3	λ_2	λ_i	$\tan_2 \sin_1$		
11	41	–	$\lambda_i - \lambda_2$	$\tan_2 \sin_1$			
12	14 04	f SIN	$\sin_2$	$\tan_2 \sin_1$			$\sin_2 = \sin(\lambda_i - \lambda_2)$
13	24 00	RCL 0	L_1	$\sin_2$	$\tan_2 \sin_1$		
14	14 06	f TAN	$\tan_1$				$\tan_1 = \tan L_1$
15	61	x	$\tan_1 \sin_2$	$\tan_2 \sin_1$			
16	41	–	NUM				NUM = $\tan_2 \sin_1 - \tan_1 \sin_2$
17	24 03	RCL 3	λ_2	NUM			
18	24 01	RCL 1	λ_1	λ_2	NUM		
19	41	–	$\lambda_2 - \lambda_1$	NUM			
20	14 04	f SIN	DEN	NUM			DEN = $\sin(\lambda_2 - \lambda_1)$
21	71	÷	NUM/DEN				
22	15 06	g TAN^{-1}	L_i, D.d				
23	14 00	f →H.MS	L_i, D.MS				Display L_i in D.MS
24	14 11 04	f FIX 4					
25	13 00	GTO 00					
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							

REGISTERS	
R_0	L_1 (dec. deg.)
R_1	λ_1 (dec. deg.)
R_2	L_2 (dec. deg.)
R_3	λ_2 (dec. deg.)
R_4	λ_i (dec. deg.)
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Input coordinates of starting						
	point:						
	Latitude (CHS for S)	L_1, D.MS	g	→H	STO	0	L_1, dec. deg.
	Longitude (CHS for E)	λ_1, D.MS	g	→H	STO	1	λ_1, dec. deg.
3	Input coordinates of destination:						
	Latitude (CHS for S)	L_2, D.MS	g	→H	STO	2	L_2, dec. deg.
	Longitude (CHS for E)	λ_2, D.MS	g	→H	STO	3	λ_2, dec. deg.
4	Return to top of memory		f	PRGM			
5	Input the intermediate longitude						
	(CHS for S) and compute cor-						
	responding latitude	λ_i, D.MS	R/S				L_i, D.MS
6	For new intermediate longitude,						
	go to step 5; for new source (or						
	destination) go to step 2 (or						
	step 3).						

RHUMBLINE NAVIGATION

Equations:

$$C = \tan^{-1} \frac{\pi(\lambda_1 - \lambda_2)}{180\left[\ln\tan\left(45 + \frac{1}{2}L_2\right) - \ln\tan\left(45 + \frac{1}{2}L_1\right)\right]}$$

$$D = \begin{cases} 60(\lambda_2 - \lambda_1)\cos L; \cos C = 0 \\ 60\,\dfrac{(L_2 - L_1)}{\cos C}\ ; \text{ otherwise} \end{cases}$$

where (L_1, λ_1) = coordinates of initial point

(L_2, λ_2) = coordinates of final point

C = rhumbline course

D = rhumbline distance

Notes:

1. No course should pass through either the south or north pole.
2. The course may not go due east or due west across the 180° meridian (International Date Line).
3. Errors in distance calculations may be encountered as C approaches 90° or 270°.
4. Accuracy deteriorates for very short legs.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	COMMENTS
00			λ_2	λ_1			
01	41	–	$\lambda_1 - \lambda_2$				
02	23 06	STO 6	$\lambda_1 - \lambda_2$				
03	02	2	2	$\lambda_1 - \lambda_2$			
04	71	÷	α				Let $\alpha = 1/2\ (\lambda_1 - \lambda_2)$
05	14 04	f SIN	sin α				Normalize α so that
06	15 04	g SIN^{-1}	norm α				$-180 \leqslant \lambda_1 - \lambda_2 \leqslant 180$;
07	09	9	9	α			finds shortest route
08	00	0	90	α			round earth
09	71	÷	$\alpha/90$				
10	15 73	g π	π	$\alpha/90$			
11	61	x	$\pi\alpha/90$	$\pi\alpha/90$			
12	24 05	RCL 5	ln $\tan_2$	$\pi\alpha/90$			
13	24 04	RCL 4	ln $\tan_1$	y			Let $y = \pi\alpha/90$
14	41	–	x	y			Let $x = \ln \tan_2 - \ln \tan_1$
15	15 09	g →P	r	C			$C = \tan^{-1} y/x$
16	22	R↓	C			r	
17	15 03	g ABS	|C|			r	
18	23 07	STO 7	|C|			r	
19	24 06	RCL 6	$\lambda_1 - \lambda_2$	|C|			
20	14 04	f SIN	sin 2α	|C|			Normalize $\lambda_1 - \lambda_2$ so
21	15 04	g SIN^{-1}	norm 2α	|C|			that $-90 \leqslant \lambda_1 - \lambda_2 \leqslant 90$
22	15 41	g x<0	2α	|C|			$x < 0$ means East to West
23	13 26	GTO 26	2α	|C|			
24	21	x⇄y	|C|	2α			W to E, |C| is answer
25	13 31	GTO 31	|C|	2α			
26	03	3	3	2α	|C|		E to W, answer is
27	06	6	36	2α	|C|		360 – |C|
28	00	0	360	2α	|C|		
29	24 07	RCL 7	|C|	360	2α	|C|	
30	41	–	360 – |C|				
31	74	R/S	Course				Display course
32	06	6	6				Compute distance D
33	00	0	60				
34	24 07	RCL 7	|C|	60			
35	14 05	f COS	cos |C|	60			
36	15 61	g x≠0	cos |C|	60			If cos C ≠ 0,
37	13 45	GTO 45	cos |C|	60			go to line 45
38	34	CLX	0	60			Cos C = 0; heading is
39	24 06	RCL 6	$\lambda_1 - \lambda_2$				due E or due W
40	61	x	$60\ (\lambda_1 - \lambda_2)$				
41	24 02	RCL 2	L_2	$60\ (\lambda_1 - \lambda_2)$			
42	14 05	f COS	cos L_2	$60\ (\lambda_1 - \lambda_2)$			
43	61	x	Dist				$D = 60\ (\lambda_1 - \lambda_2) \cos L$
44	13 00	GTO 00	Dist				Halt and display Dist
45	71	÷	60/cos |C|				Heading is not due E or W
46	24 02	RCL 2	L_2				Apply formula:
47	24 00	RCL 0	L_1	L_2	60/cos |C|		$D = 60(L_2 - L_1)/\cos C$
48	41	–	$L_2 - L_1$	60/cos |C|			
49	61	x	Dist				Halt

REGISTERS	
R 0	L_1 (dec. deg.)
R 1	λ_1 (dec. deg.)
R 2	L_2 (dec. deg.)
R 3	λ_2 (dec. deg.)
R 4	ln tan $(45+L_1/2)$
R 5	ln tan $(45+L_2/2)$
R 6	$\lambda_1 - \lambda_2$
R 7	|C|

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Input the initial latitude (CHS						
	for S)	L_1, D.MS	g	→H	STO	2	
			2	÷	45	+	
			f	TAN	f	LN	
			STO	5			ln $\tan_1$
3	Input the initial longitude (CHS						
	for E)	λ_1, D.MS	g	→H	STO	3	λ_1, dec. deg.
4	Input the final latitude (CHS for						
	S)	L_2, D.MS	g	→H	RCL	2	
			STO	0	x⇄y	STO	
			2	2	÷	45	
			+	f	TAN	f	
			LN	RCL	5	STO	
			4	x⇄y	STO	5	ln $\tan_2$
5	Input the final longitude (CHS						
	for E)	λ_2, D.MS	g	→H	RCL	3	
			STO	1	x⇄y	STO	
			3				λ_2, dec. deg.
6	Compute course		f	PRGM	R/S		C
7	Compute distance		R/S				D
8	To continue the course, return to						
	step 4 and input a new final						
	position						

Example:

A ship sailing from San Francisco (L37° 49′N, λ 122° 25′W) to Tokyo (L35° 40′N, λ 139° 45′E) will follow three rhumblines to approximate the great circle route. The navigator chooses the two intermediate points to be at λ 155°W and λ 175°E. Find the rhumbline courses the ship should follow, and the distance covered on each leg.

Solution:

First key in Great Circle Plotting.

37.49 [g] [→H] [STO] [0] 122.25 [g] [→H] [STO] [1] 35.40 [g] [→H] [STO] [2] 139.45 [CHS] [g] [→H] [STO] [3] [f] [PRGM] 155 [R/S] ——→ 47.4606

175 [CHS] [R/S] ——→ 47.3610

Thus the two intermediate points are (L47° 46′N, λ 155°W) and (L47° 36′ N, λ 175°E).

Now key in Rhumbline Navigation.

Coordinates of starting point:

37.49 [g] [→H] [STO] [2] 2 [÷] 45 [+] [f] [tan] [f] [ln] [STO] [5]
122.25 [g] [→H] [STO] [3]

Find course, distance to first intermediate point:

47.4606 [g] [→H] [RCL] [2] [STO] [0] [x⇄y] [STO] [2] 2 [÷] 45 [+] [f] [tan] [f] [ln] [RCL] [5] [STO] [4] [x⇄y] [STO] [5] 155 [g] [→H] [RCL] [3] [STO] [1] [x⇄y] [STO] [3] [f] [PRGM]

[R/S] ⟶ 292.67 (course)

[R/S] ⟶ 1549.38 (distance)

Find course, distance to second intermediate point:

47.361 [g] [→H] [RCL] [2] [STO] [0] [x⇄y] [STO] [2] 2 [÷] 45 [+] [f] [tan] [f] [ln] [RCL] [5] [STO] [4] [x⇄y] [STO] [5] 175 [CHS] [g] [→H] [RCL] [3] [STO] [1] [x⇄y] [STO] [3] [f] [PRGM]

[R/S] ⟶ 269.53 (course)

[R/S] ⟶ 1211.80 (distance)

Find course, distance to destination:

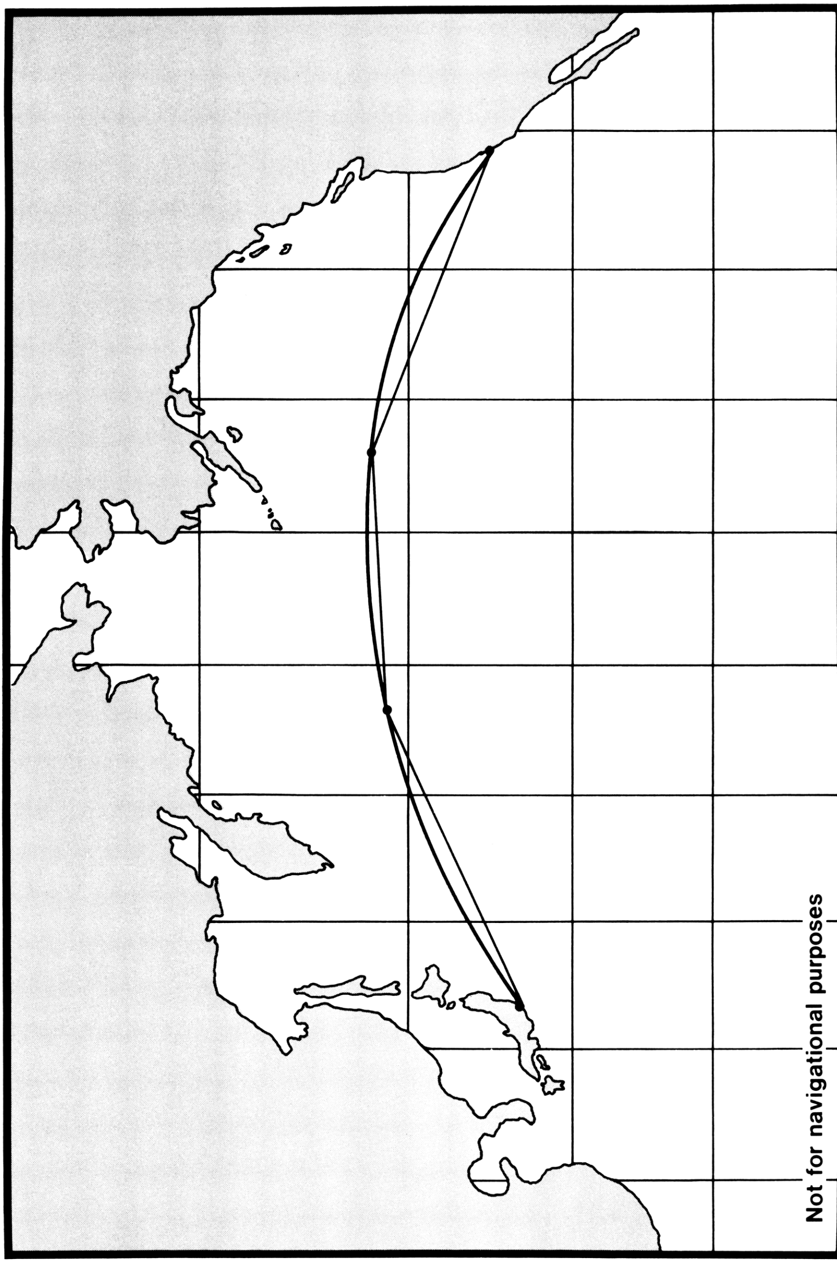
35.40 [g] [→H] [RCL] [2] [STO] [0] [x⇄y] [STO] [2] 2 [÷] 45 [+] [f] [tan] [f] [ln] [RCL] [5] [STO] [4] [x⇄y] [STO] [5] 139.45 [CHS] [g] [→H] [RCL] [3] [STO] [1] [x⇄y] [STO] [3] [f] [PRGM]

[R/S] ⟶ 245.53 (course)

[R/S] ⟶ 1728.66 (distance)

Summary:

Location	**Coordinates**	**Rhumbline Course**	**Rhumbline Distance**
San Francisco	L37° 49′N, λ 122° 25′W		
		292.7°	1549.38 n.m.
1st intermediate	L47° 46′N, λ 155°W		
		269.5°	1211.80 n.m.
2nd intermediate	L47° 36′N, λ 175°E		
		245.5°	1728.66 n.m.
Tokyo	L35° 40′N, λ 139° 45′E		



The total of the three rhumbline distances is 4489.8 nautical miles. The distance along the great circle from San Francisco to Tokyo may be found to be 4460 nautical miles. Even with just two intermediate points, the extra distance added by following rhumblines is less than 30 nautical miles.

SIGHT REDUCTION TABLE

This program calculates the computed altitude Hc and azimuth Zn of a celestial body given the observer's latitude L and the local hour angle LHA and declination d of the body. It thus becomes a replacement for the nine volumes of HO 214. However, the user need not bother with the distinctions of same name and contrary name; the program itself resolves all ambiguities of this type.

Equations:

$$Hc = \sin^{-1}\left[\sin d \sin L + \cos d \cos L \cos LHA\right]$$

$$Zn = \begin{cases} Z & ; \sin LHA < 0 \\ 360\text{-}Z; & \sin LHA \geq 0 \end{cases} \qquad Z = \cos^{-1}\left[\frac{\sin d - \sin L \sin Hc}{\cos L \cos Hc}\right]$$

Notes:

1. Southern latitudes and southern declinations must be entered as negative numbers.

2. The meridian angle t may be input in place of LHA, but if so, eastern meridian angles must be input as negative numbers.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 00	RCL 0
02	14 04	f SIN
03	24 01	RCL 1
04	14 04	f SIN
05	61	x
06	24 00	RCL 0
07	14 05	f COS
08	24 01	RCL 1
09	14 05	f COS
10	61	x
11	24 02	RCL 2
12	14 05	f COS
13	61	x
14	51	+
15	23 03	STO 3
16	15 04	g SIN^{-1}
17	23 04	STO 4
18	14 00	f →H.MS
19	74	R/S
20	24 01	RCL 1
21	14 04	f SIN
22	24 03	RCL 3
23	24 00	RCL 0
24	14 04	f SIN

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	61	x
26	41	–
27	24 00	RCL 0
28	14 05	f COS
29	71	÷
30	24 04	RCL 4
31	14 05	f COS
32	71	÷
33	15 05	g COS^{-1}
34	24 02	RCL 2
35	14 04	f SIN
36	15 41	g x<0
37	13 45	GTO 45
38	22	R↓
39	03	3
40	06	6
41	00	0
42	21	x⇄y
43	41	–
44	13 00	GTO 00
45	22	R↓
46	13 00	GTO 00
47		
48		
49		

REGISTERS
R_0 L
R_1 d
R_2 LHA
R_3 sin Hc
R_4 Hc
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Input the following:						
	Observer's latitude	L, D.MS	g	→H	STO	0	L, dec. deg.
	Declination	d, D.MS	g	→H	STO	1	d, dec. deg.
	Local hour angle	LHA, D.MS	g	→H	STO	2	LHA, dec. deg.
3	Compute altitude		f	PRGM	R/S		Hc, D.MS
4	Compute azimuth		R/S				Zn, dec. deg.
5	For new case, go to step 2.						

Example:

Compute the altitude and azimuth of the moon if its LHA is 2°39′54″W and its declination 13°51′06″S. The assumed latitude is 33°20′N.

Solution:

Hc = 42°44′47″

Zn = 183.5°

GREAT CIRCLE NAVIGATION

This program computes the great circle distance between two points and the initial heading from the first, given the latitude and longitude of the source (L_1, λ_1) and destination (L_2, λ_2).

Equations:

$$D = 60 \cos^{-1} [\sin L_1 \ \sin L_2 + \cos L_1 \cos L_2 \cos (\lambda_2 - \lambda_1)]$$

$$H = \cos^{-1} \left[\frac{\sin L_2 - \sin L_1 \cos (D/60)}{\sin (D/60) \cos L_1} \right]$$

$$H_i = \begin{cases} H & ; \sin (\lambda_2 - \lambda_1) < 0 \\ 360\text{-}H; & \sin (\lambda_2 - \lambda_1) \geqslant 0 \end{cases}$$

Notes:

1. Southern latitudes and eastern longitudes must be entered as negative numbers.
2. Truncation and round off errors occur when the source and destination are very close together (1 mile or less).
3. Do not use coordinates located at diametrically opposite sides of the earth.
4. Do not use latitudes of +90° or –90°.
5. Do not try to compute initial heading along a line of longitude $(L_1 = L_2)$.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 00	RCL 0
02	14 04	f SIN
03	24 01	RCL 1
04	14 04	f SIN
05	61	x
06	24 00	RCL 0
07	14 05	f COS
08	24 01	RCL 1
09	14 05	f COS
10	61	x
11	24 02	RCL 2
12	14 05	f COS
13	61	x
14	51	+
15	23 03	STO 3
16	15 05	g COS^{-1}
17	23 04	STO 4
18	06	6
19	00	0
20	61	x
21	74	R/S
22	24 01	RCL 1
23	14 04	f SIN
24	24 00	RCL 0

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	14 04	f SIN
26	24 03	RCL 3
27	61	x
28	41	–
29	24 00	RCL 0
30	14 05	f COS
31	71	÷
32	24 04	RCL 4
33	14 04	f SIN
34	71	÷
35	15 05	g COS^{-1}
36	24 02	RCL 2
37	14 04	f SIN
38	15 41	g $x<0$
39	13 47	GTO 47
40	22	R↓
41	03	3
42	06	6
43	00	0
44	21	x⇄y
45	41	–
46	13 00	GTO 00
47	22	R↓
48	13 00	GTO 00
49		

REGISTERS
R_0 L_1
R_1 L_2
R_2 $\lambda_2 - \lambda_1$
R_3 cos (D/60)
R_4 D/60
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Input the following:						
	Source latitude	L_1, D.MS	g	→H	STO	0	L_1, dec. deg.
	Destination latitude	L_2, D.MS	g	→H	STO	1	L_2, dec. deg.
	Destination longitude	λ_2, D.MS	g	→H			λ_2, dec. deg.
	Source longitude	λ_1, D.MS	g	→H	–	STO	
			2				$\lambda_2-\lambda_1$, dec. deg.
3	Compute great circle distance		f	PRGM	R/S		D, naut. mi.
4	Compute initial heading		R/S				H_i, dec. deg.
5	For new case, go to step 2.						

Example:

Find the great circle distance and initial heading from San Francisco (L37° 49′N, λ122°25′W) to Tokyo (L35°40′N, λ139°45′E).

Solution:

D = 4460.04

H_i = 303.29°

Notes

CHAPTER 5 NUMERICAL METHODS

NEWTON'S METHOD SOLUTION TO f(x) = 0

One of the most common and frustrating problems in algebra is the solution of an equation like

$$\ln x + 3x = 10.8074,$$

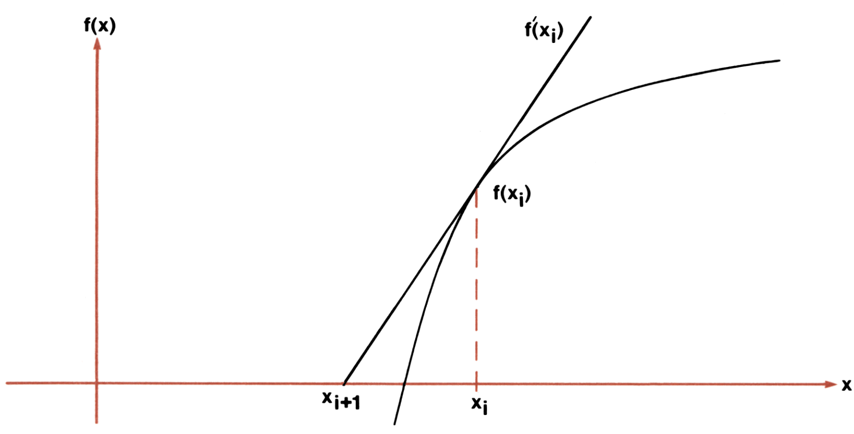
in which the x's refuse to conveniently migrate to one side of the equation and isolate themselves. That is, there is no simple algebraic solution. In this case, one of several root-finding algorithms may be employed to solve the equation $f(x) = 0$, where $f(x) = \ln x + 3x - 10.8074$. The following program uses Newton's method to find a solution for $f(x) = 0$, where $f(x)$ is specified by the user.

The user must define the function f(x) by keying into program memory the keystrokes required to find f(x), assuming x is in the X-register. Fourteen program steps are available for defining f(x); the stack registers and storage registers R_5 through R_7 are also available to the user. In addition, the user must provide the program with an initial guess, x_1, for the solution. The closer the initial guess is to the actual solution, the faster the program will converge to an answer. The program will halt when two successive approximations for x, say x_i and x_{i+1}, are within a tolerance ϵ, i.e., when $|x_{i+1} - x_i| < \epsilon$. The value for ϵ must be input by the user. In general a reasonable value for ϵ might be $10^{-6}\ x_1$.

Equations:

The basic formula used by Newton's method to generate the next approximation for the solution is

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



This program makes a numerical approximation for the derivative $f'(x)$ to give the following equation:

$$x_{i+1} = x_i - \delta_i \left[\frac{f(x_i + \delta_i)}{f(x_i)} - 1 \right]^{-1}$$

where $\delta_i = 10^{-5}\, x_i$

Notes:

1. After the routine has finished calculating, the last value of f(x) may be displayed by pressing RCL 4. If this value is not close enough to zero, the program may be run again with a smaller value for ϵ.

2. The user can watch the function converge to zero by making a slight change in the program. If the g NOP in line 07 is replaced by an f PAUSE, the program will pause during each iteration, displaying successive values of f(x) which should be converging to zero. To make this change to a program that has already been keyed in, perform the following operations:

   1. Press GTO 0 6
   2. Switch to PRGM
   3. Press f PAUSE
   4. Switch to RUN
   5. Press f PRGM

Programming Remarks

This is one of the more complex programs in the book. The main difficulty is that at each iteration both f(x) and $f(x + \delta)$ need to be calculated, but the function f is keyed in in only one place in program memory. Large computers handle this problem by the use of a subroutine. This program simulates that technique by a number stored in R_0 known as a flag. The flag is set to 0 to indicate that f(x) is to be calculated, or to 1 if $f(x + \delta)$ is to be found. After the calculation of f, a test is made on the flag. If it is 0, the program will branch to an instruction which will store f(x); if it is 1, the program will go on to calculate a derivative based on $f(x + \delta)$. All operations connected with the flag occupy a total of 9 program steps.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	COMMENTS
00							
01	34	CLX	0				Set flag to 0 for f(x)
02	23 00	STO 0	0				
03	24 01	RCL 1	x	0			Recall x and branch to
04	13 17	GTO 17	x	0			calculate f(x)
05	22	R↓	f(x)				Roll down to remove flag
06	23 04	STO 4	f(x)				
07	15 22	g NOP	f(x)				May Pause to see convergence
08	01	1	1	f(x)			Set flag to 1 for f(x + δ)
09	23 00	STO 0	1	f(x)			
10	24 01	RCL 1	x	1	f(x)		
11	24 01	RCL 1	x	x	1	f(x)	
12	33	EEX	1. 00	x	x	1	
13	05	5	1. 05	x	x	1	
14	71	÷	10^{-5} x	x	1	1	
15	23 03	STO 3	δ	x	1	1	
16	51	+	x + δ	1	1	1	
17							Lines 17 through 30 are
18							reserved for user
19							to define f(x)
20							
21							This section of pgm is
22							used to find f(x) and
23							f(x + δ). Flag in R_0 is
24							0 for f(x), 1 for
25							f(x + δ)
26							
27							
28							
29							
30							
31	15 71	g x = 0	f(x)/(x + δ)				Is function value = 0?
32	13 49	GTO 49	f(x)/(x + δ)				Yes, output solution
33	24 00	RCL 0	Flag	f(x)/(x + δ)			No, check flag
34	15 71	g x = 0	Flag	f(x)/(x + δ)			Flag = 0?
35	13 05	GTO 05	Flag	f(x)			Yes, have f(x)
36	22	R↓	f(x + δ)			Flag	No, flag = 1, have f(x + δ)
37	24 04	RCL 4	f(x)	f(x + δ)			
38	71	÷	R				R = f(x + δ)/f(x)
39	01	1	1	R			
40	41	−	R − 1				R−1 = [f(x+δ) − f(x)]/f(x)
41	15 22	g 1/x	$(R-1)^{-1}$				Approximate:
42	24 03	RCL 3	δ	$(R-1)^{-1}$			$f'(x) = [f(x+\delta) - f(x)]/\delta$
43	61	×	δ/(R − 1)				$\Delta = f(x)/f'(x)$
44	23 41 01	STO − 1	Δ				$x_{i+1} = x_i - \Delta$
45	15 73	g ABS	|Δ|				
46	24 02	RCL 2	ε	|Δ|			
47	14 41	f x<y	ε	|Δ|			$\|x_{i+1} - x_i\| > \epsilon$?
48	13 01	GTO 01	ε	|Δ|			Yes, iterate again
49	24 01	RCL 1	x	ε	|Δ|		No, display x and halt

REGISTERS	
R_0	Flag
R_1	x
R_2	ε
R_3	δ
R_4	f(x)
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in lines 1-16 of program						16 51
2	Key in function f(x)						
3	Key in a branch to line 31		GTO	31			
4	Press SST until display shows						
	line 30						
5	Key in lines 31-49 of program						
6	Switch to RUN						
7	Store initial guess for solution	x_1	STO	1			
8	Store tolerance	ϵ	STO	2			
9	Compute solution		f	PRGM	R/S		x_0
10	To change x_1 or ϵ go to appro-						
	priate step and store new value.						

Example:

An equation often solved by gear designers is

$$\tan x - x - I = 0$$

where x is an angle in radians and I is the *involute* of x. Find the angle x_0 corresponding to an involute of 0.0324.

Note:

Since a gear designer might want to calculate x for several values of I, it will be simpler to store I in R_7 for use by the function f(x).

Solution:

Example User Instructions

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in lines 1-16 of program						16 51
2	Key in steps for f(x) = tan x −						
	x − I		f	TAN			17 14 06
			f	LASTx			18 14 73
			−				19 41
			RCL	7			20 24 07
			−				21 41
3	Key in branch to 31		GTO	31			22 13 31
4	Press SST 8 times, until display						
	shows line 30						
5	Key in lines 31-49						49 24 01
6	Switch to RUN						
7	Set angular mode		g	RAD			
8	Store I	.0324	STO	7			
9	Guess x_1 = 1	1	STO	1			
10	Set tolerance $\epsilon = 10^{-6}$	10^{-6}	STO	2			
11	Compute solution x_0		f	PRGM	R/S		0.45
12	Convert the angle to degrees		180	x	g	π	
			÷				25.62
13	Display last value of f(x)		RCL	4			2.30 −09

$x_0 = 25.62°$

Last $f(x) = 2.30 \times 10^{-9}$

NUMERICAL INTEGRATION, SIMPSON'S RULE

Let x_0, x_1, ..., x_n be equally spaced points such that $x_i = x_0 + ih$ for $i = 0, 1, 2, ..., n$ at which corresponding values $f(x_0)$, $f(x_1)$, ..., $f(x_n)$ of a function f(x) are known. This function need not be known explicitly but if it is, these values can be found previously by writing the function into memory and evaluating at the various points. n must be an even positive integer.

Simpson's Rule is:

$$\int_{x_0}^{x_n} f(x)\,dx \cong \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + \ldots + 4f(x_{n-3}) + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)].$$

Let the solution be indicated by I.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 00	RCL 0
02	03	3
03	71	÷
04	23 00	STO 0
05	61	x
06	23 01	STO 1
07	74	R/S
08	24 00	RCL 0
09	61	x
10	24 01	RCL 1
11	51	+
12	23 01	STO 1
13	74	R/S
14	24 00	RCL 0
15	61	x
16	04	4
17	61	x
18	24 01	RCL 1
19	51	+
20	23 01	STO 1
21	74	R/S
22	24 00	RCL 0
23	61	x
24	02	2

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	61	x
26	24 01	RCL 1
27	51	+
28	23 01	STO 1
29	13 13	GTO 13
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 h/3
R_1 Σ
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store increment	h	STO	0			
3	Enter first function value	$f(x_0)$	f	PRGM	R/S		Partial sum
4	Enter last function value	$f(x_n)$	R/S				Partial sum
5	Enter values i = 1, 2,..., n − 2	$f(x_i)$	R/S				Partial sum
6	Enter value i = n − 1	$f(x_{n-1})$	R/S				I

Example

Compute $\int_0^\pi \sin^2 x \, dx$ using Simpson's rule with $h = \pi/8$.

The following data must be found first:

i	0	1	2	3	4	5	6	7	8
x_i	0	$\pi/8$	$\pi/4$	$3\pi/8$	$\pi/2$	$5\pi/8$	$3\pi/4$	$7\pi/8$	π
$f(x_i)$	0	0.1464	0.5	0.8536	1	0.8536	0.5	0.1464	0

Solution:

$$\int_0^\pi \sin^2 x \, dx \cong 1.5708$$

The exact solution is $\pi/2$.

NUMERICAL SOLUTION TO DIFFERENTIAL EQUATIONS

This program may be used to solve a wide variety of first order differential equations of the form

$$y' = f(x, y)$$

with initial values x_0, y_0.

The solution is a numerical solution which calculates y_i for $x_i = x_0 + ih$, where h is an increment specified by the user and $i = 1, 2, \ldots$.

The program uses a modified Euler method (predictor – corrector):

$$\hat{y}_{i+1} = y_i + h\, f(x_i, y_i) \qquad y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, \hat{y}_{i+1})\right]$$

f(x, y) is keyed into memory starting at line 18. The user has 13 program steps to write f(x, y); registers R_5, R_6, and R_7 are also available. The user should assume that x and y will be in the X- and Y-registers, respectively. The routine should return with the value of f(x, y) in the X-register and should end with a GTO 31.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	34	CLX
02	23 04	STO 4
03	24 02	RCL 2
04	24 01	RCL 1
05	13 18	GTO 18
06	22	R↓
07	23 03	STO 3
08	24 00	RCL 0
09	61	x
10	24 02	RCL 2
11	51	+
12	24 01	RCL 1
13	24 00	RCL 0
14	51	+
15	01	1
16	23 04	STO 4
17	22	R↓
18		
19		
20		
21		
22		
23		
24		

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25		
26		
27		
28		
29		
30		
31	24 04	RCL 4
32	15 71	g x=0
33	13 06	GTO 06
34	22	R↓
35	24 03	RCL 3
36	51	+
37	24 00	RCL 0
38	61	x
39	02	2
40	71	÷
41	24 02	RCL 2
42	51	+
43	23 02	STO 2
44	24 01	RCL 1
45	24 00	RCL 0
46	51	+
47	23 01	STO 1
48	14 74	f PAUSE
49	21	x⇄y

REGISTERS
R_0 h
R_1 x
R_2 y
R_3 y′
R_4 Flag
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in lines 1-17 of program						17 22
2	Key in function f(x, y)						
3	Key in branch to line 31		GTO	31			
4	Press SST repeatedly until dis-						
	play shows line 30						
5	Key in lines 31-49 of program						49 21
6	Switch to RUN						
7	Store increment	h	STO	0			
8	Store initial conditions	x_0	STO	1			
		y_0	STO	2	f	PRGM	
9	Display next x-value and cor-						
	responding y-value		R/S				(x_k)
							y_k
10	Repeat step 9 as often as desired						

Example:

Solve numerically the differential equation $y' = x\sqrt{y}$ with initial conditions $x_0 = 1, y_0 = 1$. Use a step size of $h = 0.1$.

Solution:

Key the function in as x⇄y f √x ×

x	1.0	1.1	1.2	1.3	1.4	1.5
y (by prgm)	1.0	1.1077	1.2319	1.3745	1.5372	1.7221
y (exact)	1.0	1.1078	1.2321	1.3748	1.5376	1.7227

LINEAR INTERPOLATION

If $(x_1, f(x_1))$ and $(x_2, f(x_2))$ are two points of a function $f(x)$, then the function at x_0 can be approximated by the following formula:

$$f(x_0) \cong \frac{(x_2 - x_0)\, f(x_1) + (x_0 - x_1)\, f(x_2)}{(x_2 - x_1)}$$

This is called the linear interpolation formula. Of course, x_2 cannot equal x_1.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	23 04	STO 4
02	24 00	RCL 0
03	41	−
04	24 03	RCL 3
05	61	x
06	24 02	RCL 2
07	24 04	RCL 4
08	41	−
09	24 01	RCL 1
10	61	x
11	51	+
12	24 02	RCL 2
13	24 00	RCL 0
14	41	−
15	71	÷
16	13 00	GTO 00
17		
18		
19		
20		
21		
22		
23		
24		

DISPLAY		KEY ENTRY
LINE	CODE	
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 x_1
R_1 $f(x_1)$
R_2 x_2
R_3 $f(x_2)$
R_4 x_0
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store first point	x_1	STO	0			
		$f(x_1)$	STO	1			
3	Store second point	x_2	STO	2			
		$f(x_2)$	STO	3	f	PRGM	
4	Key in x_0, find $f(x_0)$	x_0	R/S				$f(x_0)$
5	Repeat step 4 for as many x-						
	values as desired.						

Example:

Given

$$f(7.3) = 1.9879$$

$$f(7.4) = 2.0015,$$

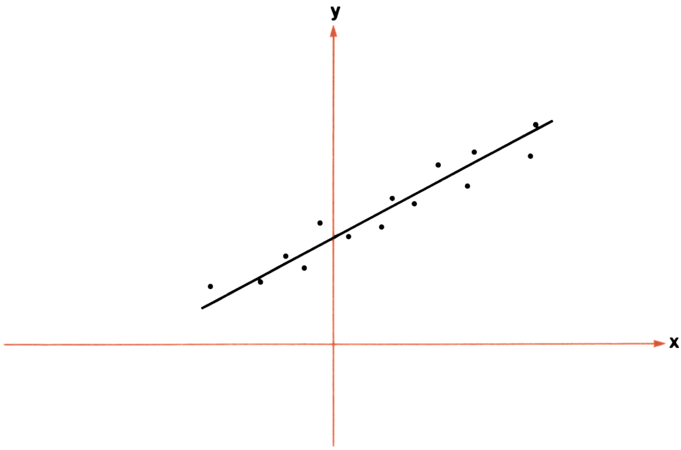
find by linear interpolation f(7.37).

Solution:

$$f(7.37) = 1.9974$$

CHAPTER 6 STATISTICS

CURVE FITTING–LINEAR REGRESSION



When investigating the relationship between two variables in the real world, it is a reasonable first step to make experimental observations of the system to gather paired values of the variables, (x, y). The investigator might then ask the question: What mathematical formula best describes the relationship between the variables x and y? His first guess will often be that the relationship is linear, i.e., that the form of the equation is $y = a_1 x + a_0$, where a_1 and a_0 are constants. The purpose of this program is to find the constants a_1 and a_0, which give the closest agreement between the experimental data and the equation $y = a_1 x + a_0$. The technique used is linear regression by the method of least squares.

The user must input the paired values of data he has gathered, (x_i, y_i), $i = 1, ..., n$. When all data pairs have been input, the regression constants a_1 and a_0 may be calculated. A third value may also be found, the coefficient of determination, r^2. The value of r^2 will lie between 0 and 1 and will indicate how closely the equation fits the experimental data: the closer r^2 is to 1, the better the fit.

Equations:

$$y = a_1 x + a_0$$

All summations below are performed for $i = 1, ..., n$.

Regression constants:

$$a_1 = \frac{\Sigma xy - \dfrac{\Sigma x\ \Sigma y}{n}}{\Sigma x^2 - \dfrac{(\Sigma x)^2}{n}}$$

$$a_0 = \bar{y} - a_1 \bar{x}$$

where $\bar{y} = \dfrac{\Sigma y}{n}$

$\bar{x} = \dfrac{\Sigma x}{n}$

Coefficient of determination:

$$r^2 = \frac{\left[\Sigma xy - \dfrac{\Sigma x \Sigma y}{n}\right]^2}{\left[\Sigma x^2 - \dfrac{(\Sigma x)^2}{n}\right]\left[\Sigma y^2 - \dfrac{(\Sigma y)^2}{n}\right]}$$

Note:

The values for a_0 and a_1 are stored in R_0 and R_1, respectively. After the calculation of a_0, a_1, and r^2, the estimated y-value, $\hat{y}$, corresponding to any x-value may be calculated by $y = a_1 x + a_0$.

Programming Remarks:

The intermediate value $C = \Sigma xy - (\Sigma x \Sigma y/n)$ is first calculated at line 14 but is also needed near the end of the program to find r^2. Since all registers R_0 through R_7 are in use, the only place to save this value is in the stack. Hence C is preserved in one or more of the stack registers from lines 14 through 36, when it is used. It is due to the presence of C in the stack that users are warned not to disturb the contents of the stack after calculation of a_0 and a_1 (see step 4 of User Instructions).

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS
LINE	CODE						
00			y	x			Steps 1-7 for summation
01	31	↑	y	y	x		
02	15 02	g x^2	y^2	y	x		
03	23 51 02	STO + 2	y^2	y	x		Σy^2
04	22	R↓	y	x		y^2	
05	21	x⇄y	x	y		y^2	
06	25	Σ+	n	y		y^2	n, Σy, Σxy, Σx^2, Σx
07	13 00	GTO 00	n	y		y^2	
08	24 05	RCL 5	Σxy				
09	24 07	RCL 7	Σx	Σxy			
10	24 04	RCL 4	Σy	Σx	Σxy		
11	61	x	$\Sigma x\ \Sigma y$	Σxy			
12	24 03	RCL 3	n	$\Sigma x\ \Sigma y$	Σxy		
13	71	÷	$\Sigma x\ \Sigma y/n$	Σxy			
14	41	−	C				$C = \Sigma xy - (\Sigma x\ \Sigma y/n)$
15	24 06	RCL 6	Σx^2	C			
16	24 07	RCL 7	Σx	Σx^2	C		
17	15 02	g x^2	$(\Sigma x)^2$	Σx^2	C		
18	24 03	RCL 3	n	$(\Sigma x)^2$	Σx^2	C	
19	71	÷	$(\Sigma x)^2/n$	Σx^2	C	C	
20	41	−	D	C	C	C	$D = \Sigma x^2 - [(\Sigma x)^2/n]$
21	71	÷	a_1	C	C	C	$a_1 = C/D$
22	23 01	STO 1	a_1	C	C	C	
23	24 07	RCL 7	Σx	a_1	C	C	
24	61	x	$a_1\ \Sigma x$	C	C	C	
25	32	CHS	$-a_1\ \Sigma x$	C	C	C	
26	24 04	RCL 4	Σy	$-a_1\ \Sigma x$	C	C	
27	51	+	$\Sigma y - a_1\ \Sigma x$	C	C	C	
28	24 03	RCL 3	n	$\Sigma y - a_1\ \Sigma x$	C	C	
29	71	÷	a_0	C	C	C	$a_0 = \bar{y} - a_1\ \bar{x}$
30	23 00	STO 0	a_0	C	C	C	
31	74	R/S	a_0	C	C	C	Halt to display a_0
32	24 01	RCL 1	a_1	a_0	C	C	
33	74	R/S	a_1	a_0	C	C	Halt to display a_1
34	21	x⇄y	a_0	a_1	C	C	
35	22	R↓	a_1	C	C	a_0	
36	61	x	a_1 C	C	a_0	a_0	
37	24 02	RCL 2	Σy^2	a_1 C	C	a_0	
38	24 04	RCL 4	Σy	Σy^2	a_1 C	C	
39	15 02	g x^2	$(\Sigma y)^2$	Σy^2	a_1 C	C	
40	24 03	RCL 3	n	$(\Sigma y)^2$	Σy^2	a_1 C	
41	71	÷	$(\Sigma y)^2/n$	Σy^2	a_1 C	a_1 C	
42	41	−	E	a_1 C	a_1 C	a_1 C	$E = \Sigma y^2 - [(\Sigma y)^2/n]$
43	71	÷	r^2	a_1 C	a_1 C	a_1 C	$r^2 = a_1\ C/E$
44	13 00	GTO 00	r^2	a_1 C	a_1 C	a_1 C	
45							
46							
47							
48							
49							

REGISTERS	
R_0	a_0
R_1	a_1
R_2	Σy^2
R_3	n
R_4	Σy
R_5	Σxy
R_6	Σx^2
R_7	Σx

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Perform for i = 1,..., n:						
	Input x-value and y-value	x_i	↑				
		y_i	R/S				i
4	Compute regression constants		GTO	08	R/S		a_0 *
			R/S				a_1 *
5	Compute coefficient of deter-						
	mination		R/S				r^2
6	To calculate a projected y-value,						
	input the x-value	x	RCL	1	x	RCL	
			0	+			$\hat{y}$
7	Perform step 6 as many times as						
	desired						
8	For a new case, go to step 2.						
	* The contents of the stack						
	should not be disturbed at these						
	points.						

Example:

An eccentric professor of numerical analysis wakes up one morning and feels feverish. A search through his medicine cabinet reveals one oral thermometer which, unfortunately, is in degrees centigrade, a scale he is not familiar with. As he stares disconsolately out his window, he spies the outdoor thermometer affixed to the windowframe. This thermometer, however, will not fit comfortably into his mouth. Still, with some ingenuity....

The professor suspects that the relationship is $F = a_1 C + a_0$. If he can get a few data pairs for F and C, he can run a linear regression program to find a_1 and a_0, then convert any reading in °C to °F through the equation. So tossing both thermometers into a sink of lukewarm water, he reads the following pairs of temperatures as the water cools:

C	40.5	38.6	37.9	36.2	35.1	34.6
F	104.5	102	100	97.5	95.5	94

If the relationship is indeed $F = a_1 C + a_0$, what are the values for a_1 and a_0? What is the coefficient of determination?

Solution:

[f] [PRGM] [f] [REG] 40.5 [↑] 104.5 [R/S] ⟶ 1.00

38.6 [↑] 102 [R/S] ⟶ 2.00

37.9 [↑] 100 [R/S] ⟶ 3.00

36.2 [↑] 97.5 [R/S] ⟶ 4.00

35.1 [↑] 95.5 [R/S] ⟶ 5.00

34.6 [↑] 94 [R/S] ⟶ 6.00

[GTO] [0] [8] [R/S] ⟶ 33.53

[R/S] ⟶ 1.76

[R/S] ⟶ 0.99

Thus, by the data above, F = 1.76 C + 33.53, with $r^2 = 0.99$. (The real equation, of course, is F = 1.8C + 32.)

Suppose the professor puts the centigrade thermometer in his mouth and finds he has a temperature of 37°C. Should he be worried?

37 [RCL] [1] [×] [RCL] [0] [+] ⟶ 98.65°F

It looks like he is safe.

EXPONENTIAL CURVE FIT

This program computes the least squares fit of n pairs of data points $\{(x_i, y_i), i = 1, 2, \ldots, n\}$, where $y_i > 0$, for an exponential function of the form

$$y = a\,e^{bx} \qquad (a > 0).$$

The equation is linearized into

$$\ln y = \ln a + bx.$$

The following statistics are computed:

1. Coefficients a, b

$$b = \frac{\Sigma x_i \ln y_i - \frac{1}{n}(\Sigma x_i)(\Sigma \ln y_i)}{\Sigma x_i^2 - \frac{1}{n}(\Sigma x_i)^2}$$

$$a = \exp\left[\frac{\Sigma \ln y_i}{n} - b\,\frac{\Sigma x_i}{n}\right]$$

2. Coefficient of determination

$$r^2 = \frac{\left[\Sigma x_i \ln y_i - \frac{1}{n}\Sigma x_i\,\Sigma \ln y_i\right]^2}{\left[\Sigma x_i^2 - \frac{(\Sigma x_i)^2}{n}\right]\left[\Sigma(\ln y_i)^2 - \frac{(\Sigma \ln y_i)^2}{n}\right]}$$

3. Estimated value $\hat{y}$ for a given x

$$\hat{y} = a\,e^{bx}$$

Note:

n is a positive integer and $n \neq 1$.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	14 07	f LN
02	31	↑
03	15 02	g x^2
04	23 51 02	STO + 2
05	22	R↓
06	21	x⇄y
07	25	Σ+
08	13 00	GTO 00
09	24 05	RCL 5
10	24 07	RCL 7
11	24 04	RCL 4
12	61	x
13	24 03	RCL 3
14	71	÷
15	41	–
16	24 06	RCL 6
17	24 07	RCL 7
18	15 02	g x^2
19	24 03	RCL 3
20	71	÷
21	41	–
22	71	÷
23	23 01	STO 1
24	24 07	RCL 7

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	61	x
26	32	CHS
27	24 04	RCL 4
28	51	+
29	24 03	RCL 3
30	71	÷
31	15 07	g e^x
32	23 00	STO 0
33	74	R/S
34	24 01	RCL 1
35	74	R/S
36	21	x⇄y
37	22	R↓
38	61	x
39	24 02	RCL 2
40	24 04	RCL 4
41	15 02	g x^2
42	24 03	RCL 3
43	71	÷
44	41	–
45	71	÷
46	13 00	GTO 00
47		
48		
49		

REGISTERS
R_0 a
R_1 b
R_2 $\Sigma (\ln y)^2$
R_3 n
R_4 $\Sigma \ln y$
R_5 $\Sigma x \ln y$
R_6 Σx^2
R_7 Σx

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Perform for i = 1,..., n:						
	Input x-value and y-value	x_i	↑				
		y_i	R/S				i
4	Compute constants		GTO	09	R/S		a*
			R/S				b*
5	Compute coefficient of deter-						
	mination		R/S				r^2
6	To calculate $\hat{y}$, input x	x	RCL	1	x	g	
			e^x	RCL	0	x	$\hat{y}$
7	Perform step 6 as many times						
	as desired						
8	For new case, go to step 2.						
	* The stack must be maintained						
	at these points.						

Example:

x_i	.72	1.31	1.95	2.58	3.14
y_i	2.16	1.61	1.16	.85	0.5

Solution:

$a = 3.45, b = -0.58$

$y = 3.45\ e^{-0.58x}$

$r^2 = 0.98$

For $x = 1.5, \hat{y} = 1.44$

LOGARITHMIC CURVE FIT

This program fits a logarithmic curve

$$y = a + b \ln x$$

to a set of data points

$$\left\{ (x_i, y_i), i = 1, 2, \ldots, n \right\}$$

where $x_i > 0$.

Program computes:

1. Regression coefficients

$$b = \frac{\Sigma y_i \ln x_i - \frac{1}{n} \Sigma \ln x_i \Sigma y_i}{\Sigma (\ln x_i)^2 - \frac{1}{n} (\Sigma \ln x_i)^2}$$

$$a = \frac{1}{n} (\Sigma y_i - b \Sigma \ln x_i)$$

2. Coefficient of determination

$$r^2 = \frac{\left[\Sigma y_i \ln x_i - \frac{1}{n} \Sigma \ln x_i \Sigma y_i\right]^2}{\left[\Sigma (\ln x_i)^2 - \frac{1}{n} (\Sigma \ln x_i)^2\right] \left[\Sigma y_i^2 - \frac{1}{n} (\Sigma y_i)^2\right]}$$

3. Estimated value $\hat{y}$ for given x

$$\hat{y} = a + b \ln x$$

Note:

n is a positive integer and $n \neq 1$.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	31	↑
02	15 02	g x^2
03	23 51 02	STO + 2
04	22	R↓
05	21	x⇄y
06	14 07	f LN
07	25	Σ+
08	13 00	GTO 00
09	24 05	RCL 5
10	24 07	RCL 7
11	24 04	RCL 4
12	61	x
13	24 03	RCL 3
14	71	÷
15	41	–
16	24 06	RCL 6
17	24 07	RCL 7
18	15 02	g x^2
19	24 03	RCL 3
20	71	÷
21	41	–
22	71	÷
23	23 01	STO 1
24	24 07	RCL 7

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	61	x
26	32	CHS
27	24 04	RCL 4
28	51	+
29	24 03	RCL 3
30	71	÷
31	23 00	STO 0
32	74	R/S
33	24 01	RCL 1
34	74	R/S
35	21	x⇄y
36	22	R↓
37	61	x
38	24 02	RCL 2
39	24 04	RCL 4
40	15 02	g x^2
41	24 03	RCL 3
42	71	÷
43	41	–
44	71	÷
45	13 00	GTO 00
46		
47		
48		
49		

REGISTERS
R_0 a
R_1 b
R_2 Σy^2
R_3 n
R_4 Σy
R_5 $\Sigma y \ln x$
R_6 $\Sigma (\ln x)^2$
R_7 $\Sigma \ln x$

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Perform for i = 1,..., n:						
	Input x-value and y-value	x_i	↑				
		y_i	R/S				i
4	Compute constants		GTO	09	R/S		a*
			R/S				b*
5	Compute coefficient of deter-						
	mination		R/S				r^2
6	To calculate $\hat{y}$, input x	x	f	ln	RCL	1	
			x	RCL	0	+	$\hat{y}$
7	Perform step 6 as many times as						
	desired						
8	For new case, go to step 2.						
	* The stack must be maintained						
	at these points						

Example:

x_i	3	4	6	10	12
y_i	1.5	9.3	23.4	45.8	60.1

Solution:

$a = -47.02$, $b = 41.39$

$y = -47.02 + 41.39 \ln x$

$r^2 = 0.98$

For $x = 8$, $\hat{y} = 39.06$

For $x = 14.5$, $\hat{y} = 63.67$

POWER CURVE FIT

This program fits a power curve

$$y = ax^b \qquad (a > 0)$$

to a set of data points

$$\{(x_i, y_i),\ i = 1, 2, \ldots, n\}$$

where $x_i > 0$, $y_i > 0$.

By writing this equation as

$$\ln y = b \ln x + \ln a$$

the problem can be solved as a linear regression problem.

Output statistics are:

1. Regression coefficients

$$b = \frac{\Sigma (\ln x_i)(\ln y_i) - \dfrac{(\Sigma \ln x_i)(\Sigma \ln y_i)}{n}}{\Sigma (\ln x_i)^2 - \dfrac{(\Sigma \ln x_i)^2}{n}}$$

$$a = \exp\left[\frac{\Sigma \ln y_i}{n} - b\,\frac{\Sigma \ln x_i}{n}\right]$$

2. Coefficient of determination

$$r^2 = \frac{\left[\Sigma (\ln x_i)(\ln y_i) - \dfrac{(\Sigma \ln x_i)(\Sigma \ln y_i)}{n}\right]^2}{\left[\Sigma (\ln x_i)^2 - \dfrac{(\Sigma \ln x_i)^2}{n}\right]\left[\Sigma (\ln y_i)^2 - \dfrac{(\Sigma \ln y_i)^2}{n}\right]}$$

3. Estimated value $\hat{y}$ for given x

$$\hat{y} = ax^b$$

Note:

n is a positive integer and $n \neq 1$.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	14 07	f LN
02	31	↑
03	15 02	g x^2
04	23 51 02	STO + 2
05	22	R↓
06	21	x⇄y
07	14 07	f LN
08	25	Σ+
09	13 00	GTO 00
10	24 05	RCL 5
11	24 07	RCL 7
12	24 04	RCL 4
13	61	x
14	24 03	RCL 3
15	71	÷
16	41	–
17	24 06	RCL 6
18	24 07	RCL 7
19	15 02	g x^2
20	24 03	RCL 3
21	71	÷
22	41	–
23	71	÷
24	23 01	STO 1

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	24 07	RCL 7
26	61	x
27	32	CHS
28	24 04	RCL 4
29	51	+
30	24 03	RCL 3
31	71	÷
32	15 07	g e^x
33	23 00	STO 0
34	74	R/S
35	24 01	RCL 1
36	74	R/S
37	21	x⇄y
38	22	R↓
39	61	x
40	24 02	RCL 2
41	24 04	RCL 4
42	15 02	g x^2
43	24 03	RCL 3
44	71	÷
45	41	–
46	71	÷
47	13 00	GTO 00
48		
49		

REGISTERS
R_0 a
R_1 b
R_2 $\Sigma (\ln y)^2$
R_3 n
R_4 $\Sigma \ln y$
R_5 $\Sigma (\ln x)(\ln y)$
R_6 $\Sigma (\ln x)^2$
R_7 $\Sigma \ln x$

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Perform for i = 1,..., n:						
	Input x-value and y-value	x_i	↑				
		y_i	R/S				i
4	Compute constants		GTO	10	R/S		a*
			R/S				b*
5	Compute coefficient of determination						
			R/S				r^2
6	Input x-value and compute $\hat{y}$	x	RCL	1	f	y^x	
			RCL	0	x		$\hat{y}$
7	Perform step 6 as many times as desired						
8	For new case, go to step 2.						
	* The stack must be maintained at these points.						

Example:

x_i	10	12	15	17	20	22	25	27	30	32	35
y_i	0.95	1.05	1.25	1.41	1.73	2.00	2.53	2.98	3.85	4.59	6.02

Solution:

$a = .03$, $b = 1.46$

$y = .03x^{1.46}$

$r^2 = 0.94$

For $x = 18$, $\hat{y} = 1.76$

$x = 23$, $\hat{y} = 2.52$

COVARIANCE AND CORRELATION COEFFICIENT

For a set of given data points $\{(x_i, y_i), i = 1, 2, ..., n\}$, the covariance and the correlation coefficent are defined as:

$$\text{covariance } s_{xy} = \frac{1}{n-1}\left(\Sigma x_i y_i - \frac{1}{n}\Sigma x_i \Sigma y_i\right)$$

$$\text{or } s_{xy}' = \frac{1}{n}\left(\Sigma x_i y_i - \frac{1}{n}\Sigma x_i \Sigma y_i\right)$$

$$\text{correlation coefficient } r = \frac{s_{xy}}{s_x s_y}$$

where s_x and s_y are standard deviations

$$s_x = \sqrt{\frac{\Sigma x_i^2 - (\Sigma x_i)^2/n}{n-1}} \qquad s_y = \sqrt{\frac{\Sigma y_i^2 - (\Sigma y_i)^2/n}{n-1}}$$

Note:

$-1 \leqslant r \leqslant 1$

DISPLAY		KEY
LINE	CODE	ENTRY
00		
01	31	↑
02	15 02	g x^2
03	23 51 02	STO + 2
04	22	R↓
05	21	x⇄y
06	25	Σ+
07	13 00	GTO 00
08	24 05	RCL 5
09	24 04	RCL 4
10	24 07	RCL 7
11	61	x
12	24 03	RCL 3
13	71	÷
14	41	–
15	24 03	RCL 3
16	01	1
17	41	–
18	23 00	STO 0
19	71	÷
20	23 01	STO 1
21	74	R/S
22	24 00	RCL 0
23	61	x
24	24 03	RCL 3

DISPLAY		KEY
LINE	CODE	ENTRY
25	71	÷
26	74	R/S
27	14 22	f s
28	23 71 01	STO ÷ 1
29	24 02	RCL 2
30	24 04	RCL 4
31	15 02	g x^2
32	24 03	RCL 3
33	71	÷
34	41	–
35	24 00	RCL 0
36	71	÷
37	14 02	f $\sqrt{x}$
38	23 71 01	STO ÷ 1
39	24 01	RCL 1
40	13 00	GTO 00
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 n – 1
R_1 Used
R_2 Σy^2
R_3 n
R_4 Σy
R_5 Σxy
R_6 Σx^2
R_7 Σx

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM	f	REG	
3	Perform this step for i = 1, 2,...,n	x_i	↑				
		y_i	R/S				i
4	Compute covariance s_{xy}		GTO	08	R/S		s_{xy}
5	Compute s_{xy}'		R/S				s_{xy}'
6	Compute correlation coefficient		R/S				r
7	For new case, go to step 2.						

Example:

x_i	26	30	44	50	62	68	74
y_i	92	85	78	81	54	51	40

Solution:

$s_{xy} = -354.14$

$s_{xy}' = -303.55$

$r = -0.96$

MOMENTS AND SKEWNESS

This program computes the following statistics for a set of given data $\{x_1, x_2, ..., x_n\}$:

1st moment $\quad \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

2nd moment $\quad m_2 = \frac{1}{n} \Sigma x_i^2 - \bar{x}^2$

3rd moment $\quad m_3 = \frac{1}{n} \Sigma x_i^3 - \frac{3}{n} \bar{x} \Sigma x_i^2 + 2\bar{x}^3$

moment coefficient of skewness

$$\gamma_1 = \frac{m_3}{m_2^{3/2}}$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY	DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00			25	71	÷
01	31	↑	26	03	3
02	15 02	g x^2	27	61	x
03	25	Σ+	28	41	–
04	13 00	GTO 00	29	24 02	RCL 2
05	24 04	RCL 4	30	31	↑
06	24 03	RCL 3	31	15 02	g x^2
07	71	÷	32	61	x
08	23 02	STO 2	33	02	2
09	74	R/S	34	61	x
10	24 07	RCL 7	35	51	+
11	24 03	RCL 3	36	23 00	STO 0
12	71	÷	37	74	R/S
13	24 02	RCL 2	38	24 00	RCL 0
14	15 02	g x^2	39	24 01	RCL 1
15	41	–	40	01	1
16	23 01	STO 1	41	73	•
17	74	R/S	42	05	5
18	24 05	RCL 5	43	14 03	f y^x
19	24 03	RCL 3	44	71	÷
20	71	÷	45	13 00	GTO 00
21	24 07	RCL 7	46		
22	24 02	RCL 2	47		
23	61	x	48		
24	24 03	RCL 3	49		

REGISTERS
R_0 m_3
R_1 m_2
R_2 $\bar{x}$
R_3 n
R_4 Σx
R_5 Σx^3
R_6 Σx^4
R_7 Σx^2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM	f	REG	
3	Perform for i = 1, 2,..., n:						
	Input x-value	x_i	R/S				i
4	Delete erroneous data	x_k	↑	g	x^2	f	
			Σ−				
5	Compute the mean		GTO	05	R/S		$\bar{x}$
6	Compute the second and third						
	moments		R/S				m_2
			R/S				m_3
7	Compute the moment coefficient						
	of skewness		R/S				γ_1
8	For new case, go to step 2.						

Example:

i	1	2	3	4	5	6	7	8	9
x_i	2.1	3.5	4.2	6.5	4.1	3.6	5.3	3.7	4.9

Solution:

$\bar{x} = 4.21$

$m_2 = 1.39$

$m_3 = 0.39$

$\gamma_1 = 0.24$

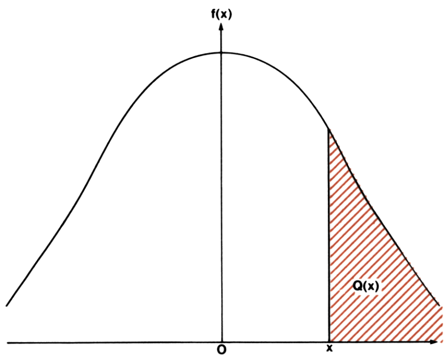
NORMAL DISTRIBUTION

The density function for a standard normal variable is

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} .$$

The upper tail area is

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt .$$



For $x \geqslant 0$, polynomial approximation is used to compute Q(x):

$$Q(x) = f(x)\,(b_1\, t + b_2\, t^2 + b_3\, t^3 + b_4\, t^4 + b_5\, t^5) + \epsilon(x)$$

where $|\epsilon(x)| < 7.5 \times 10^{-8}$

$$t = \frac{1}{1 + rx}, \quad r = 0.2316419$$

$b_1 = .31938153$, $b_2 = -.356563782$

$b_3 = 1.781477937$, $b_4 = -1.821255978$

$b_5 = 1.330274429$

Note:

The program only works for $x \geqslant 0$. Equations $f(-x) = f(x)$, $Q(-x) = 1-Q(x)$, where $x \geqslant 0$, can be used to find f and Q for negative numbers.

Reference:

Abramowitz and Stegun, *Handbook of Mathematical Functions,* National Bureau of Standards, 1968.

LINE	CODE	KEY ENTRY
00		
01	31	↑
02	23 06	STO 6
03	61	x
04	02	2
05	71	÷
06	32	CHS
07	15 07	g e^x
08	15 73	g π
09	02	2
10	61	x
11	14 02	f $\sqrt{x}$
12	71	÷
13	23 07	STO 7
14	74	R/S
15	24 00	RCL 0
16	24 06	RCL 6
17	61	x
18	01	1
19	51	+
20	15 22	g 1/x
21	31	↑
22	31	↑
23	31	↑
24	24 05	RCL 5

LINE	CODE	KEY ENTRY
25	61	x
26	24 04	RCL 4
27	51	+
28	61	x
29	24 03	RCL 3
30	51	+
31	61	x
32	24 02	RCL 2
33	51	+
34	61	x
35	24 01	RCL 1
36	51	+
37	61	x
38	24 07	RCL 7
39	61	x
40	13 00	GTO 00
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 r
R_1 b_1
R_2 b_2
R_3 b_3
R_4 b_4
R_5 b_5
R_6 x
R_7 f(x)

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM			
3	Store constants	r	STO	0			
		b_1	STO	1			
		b_2	STO	2			
		b_3	STO	3			
		b_4	STO	4			
		b_5	STO	5			
4	Input x and compute f(x)	x	R/S				f(x)
5	Compute Q(x)		R/S				Q(x)
6	For a new case, go to 4.						

Examples:

1. $x = 1.18$
2. $x = 2.28$

Solutions:

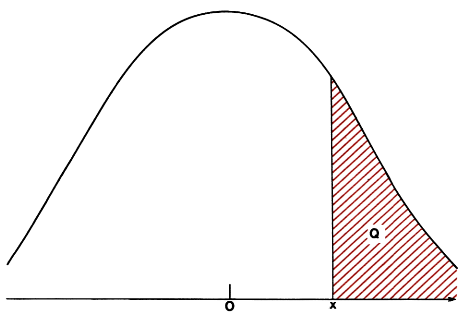
1. $f(x) = 0.20$
   $Q(x) = 0.12$
2. $f(x) = 0.03$
   $Q(x) = 0.01$

INVERSE NORMAL INTEGRAL

This program determines the value of x such that

$$Q = \int_{x}^{\infty} \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}} dt$$

where Q is given and $0 < Q \leqslant 0.5$.



The following rational approximation is used:

$$x = t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3} + \epsilon(Q)$$

where $|\epsilon(Q)| < 4.5 \times 10^{-4}$

$$t = \sqrt{\ln \frac{1}{Q^2}}$$

$c_0 = 2.515517$ $\quad d_1 = 1.432788$

$c_1 = 0.802853$ $\quad d_2 = 0.189269$

$c_2 = 0.010328$ $\quad d_3 = 0.001308$

Reference:

Abramowitz and Stegun, *Handbook of Mathematical Functions,* National Bureau of Standards, 1968.

LINE	CODE	KEY ENTRY
00		
01	31	↑
02	61	x
03	15 22	g 1/x
04	14 07	f LN
05	14 02	f $\sqrt{x}$
06	23 06	STO 6
07	31	↑
08	31	↑
09	31	↑
10	24 05	RCL 5
11	61	x
12	24 04	RCL 4
13	51	+
14	61	x
15	24 03	RCL 3
16	51	+
17	61	x
18	01	1
19	51	+
20	23 07	STO 7
21	34	CLX
22	24 02	RCL 2
23	61	x
24	24 01	RCL 1
25	51	+
26	61	x
27	24 00	RCL 0
28	51	+
29	24 07	RCL 7
30	71	÷
31	41	−
32	13 00	GTO 00
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 c_0
R_1 c_1
R_2 c_2
R_3 d_1
R_4 d_2
R_5 d_3
R_6 t
R_7 $1 + d_1 t + d_2 t^2 + d_3 t^3$

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM			
3	Store constants	c_0	STO	0			
		c_1	STO	1			
		c_2	STO	2			
		d_1	STO	3			
		d_2	STO	4			
		d_3	STO	5			
4	Input Q	Q	R/S				
5	For a new case, go to 4.						

Examples:

1. Q = 0.12
2. Q = 0.05

Solutions:

1. x = 1.18
2. x = 1.65

FACTORIAL

This program will compute factorials for positive integers between 2 and 69.

$$n! = n\,(n-1)\,(n-2)\,...\,(2)\,(1)$$

Notes:

1. For large values of n, the program will take some time to arrive at a result, up to a maximum of about 20 seconds for n = 69.

2. The program does not check input values and will return incorrect answers for values of $n < 2$ or $n > 69$ or n non-integer.

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	31	↑
02	01	1
03	23 00	STO 0
04	21	$x \rightleftarrows y$
05	23 61 00	STO × 0
06	01	1
07	41	–
08	14 61	f $x \neq y$
09	13 05	GTO 05
10	24 00	RCL 0
11	13 00	GTO 00
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

DISPLAY		KEY ENTRY
LINE	CODE	
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 Used
R_1
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM			
3	Key in n ($2 \leqslant n \leqslant 69$)	n	R/S				n!
4	For a new n, go to step 3.						

Examples:

1. 5! = 120.00
2. 10! = 3628800.00

PERMUTATION

A permutation is an ordered subset of a set of distinct objects. The number of possible permutations, each containing n objects, that can be formed from a collection of m distinct objects is given by

$$_mP_n = \frac{m!}{(m-n)!} = m(m-1)\ldots(m-n+1)$$

where m, n are integers and $0 \leqslant n \leqslant m$.

Notes:

1. $_mP_n$ can also be denoted by P_n^m, $P(m,n)$ or $(m)_n$.
2. $_mP_0 = 1$, $_mP_1 = m$, $_mP_m = m!$

LINE	CODE	KEY ENTRY
00		
01	24 00	RCL 0
02	24 00	RCL 0
03	24 01	RCL 1
04	15 71	g x=0
05	13 29	GTO 29
06	14 71	f x=y
07	13 31	GTO 31
08	14 51	f x≥y
09	13 39	GTO 39
10	01	1
11	14 71	f x=y
12	13 41	GTO 41
13	22	R↓
14	41	–
15	01	1
16	51	+
17	61	x
18	14 73	f LASTx
19	24 00	RCL 0
20	01	1
21	41	–
22	14 71	f x=y
23	13 26	GTO 26
24	22	R↓

LINE	CODE	KEY ENTRY
25	13 15	GTO 15
26	22	R↓
27	22	R↓
28	13 00	GTO 00
29	01	1
30	13 00	GTO 00
31	01	1
32	41	–
33	15 71	g x=0
34	13 37	GTO 37
35	23 61 00	STO x 0
36	13 31	GTO 31
37	24 00	RCL 0
38	13 00	GTO 00
39	00	0
40	71	÷
41	22	R↓
42	22	R↓
43	13 00	GTO 00
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 m
R_1 n
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store m, n	m	STO	0			
		n	STO	1			
3	Compute permutations		f	PRGM	R/S		$_mP_n$
4	For new case, go to step 2.						

Examples:

1. $_{43}P_3 = 74046.00$

2. $_{73}P_4 = 26122320.00$

COMBINATION

A combination is a selection of one or more of a set of distinct objects without regard to order. The number of possible combinations, each containing n objects, that can be formed from a collection of m distinct objects is given by

$$_mC_n = \frac{m!}{(m-n)!\,n!} = \frac{m(m-1)\ldots(m-n+1)}{1\cdot 2\cdot\ldots\cdot n}$$

where m, n are integers and $0 \leqslant n \leqslant m$.

This program computes $_mC_n$ using the following algorithm:

1. If $n \leqslant m - n$

$$_mC_n = \frac{m-n+1}{1}\cdot\frac{m-n+2}{2}\cdot\ldots\cdot\frac{m}{n}\ .$$

2. If $n > m - n$, program computes $_mC_{m-n}$.

Notes:

1. $_mC_n$, which is also called the binomial coefficient, can be denoted by C_n^m, $C(m,n)$, or $\binom{m}{n}$.
2. $_mC_n = {}_mC_{m-n}$
3. $_mC_0 = {}_mC_m = 1$
4. $_mC_1 = {}_mC_{m-1} = m$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	41	−
02	14 73	f LASTx
03	14 41	f x<y
04	21	x⇄y
05	23 00	STO 0
06	01	1
07	23 01	STO 1
08	51	+
09	23 02	STO 2
10	22	R↓
11	15 71	g x=0
12	13 30	GTO 30
13	01	1
14	24 01	RCL 1
15	51	+
16	23 01	STO 1
17	21	x⇄y
18	14 51	f x≥y
19	13 22	GTO 22
20	24 02	RCL 2
21	13 00	GTO 00
22	21	x⇄y
23	24 00	RCL 0
24	51	+
25	24 01	RCL 1
26	71	÷
27	23 61 02	STO x 2
28	22	R↓
29	13 13	GTO 13
30	01	1
31	13 00	GTO 00
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 max (n, m − n)
R_1 Used
R_2 Used
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Key in m and n	m	↑				
		n	f	PRGM	R/S		$_mC_n$
3	For new case, go to step 2.						

Examples:

1. $_{73}C_4 = 1088430.00$

2. $_{43}C_3 = 12341.00$

RANDOM NUMBER GENERATOR

This program calculates uniformly distributed pseudo random numbers u_i in the range

$$0 \leqslant u_i \leqslant 1$$

using the following formula:

$$u_i = \text{Fractional part of } [(\pi + u_{i-1})^5].$$

The user has to specify the starting value u_0 (the "seed" of the sequence) such that

$$0 \leqslant u_0 \leqslant 1.$$

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	15 73	g π
02	24 00	RCL 0
03	51	+
04	05	5
05	14 03	f y^x
06	15 01	g FRAC
07	23 00	STO 0
08	13 00	GTO 00
09		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

DISPLAY		KEY ENTRY
LINE	CODE	
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 u_i
R_1
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store seed	u_0	STO	0	f	PRGM	
3	Generate random number		R/S				u_i
4	Repeat step 3 as many times as						
	desired						
5	For new sequence, go to step 2.						

Example:

Find the sequence of random numbers generated from a seed of 0.192743568.

Solution:

0.14, 0.76, 0.15, 0.35, 0.62, 0.54, 0.62, 0.91, 0.48, 0.24, ...

CHI-SQUARE EVALUATION

This program calculates the value of the χ^2 statistic for the goodness of fit test by the equation

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

where O_i = observed frequency

E_i = expected frequency.

The χ^2 statistic measures the closeness of the agreement between the observed frequencies and expected frequencies.

Notes:

1. In order to apply this test to a set of given data, it may be necessary to combine some classes to make sure that each expected frequency is not too small (say, not less than 5).

2. If the expected frequencies E_i are all equal to some value E, then E should be computed beforehand as

$$E = \frac{\Sigma O_i}{n}$$

and then input at each step as the expected frequency E_i.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	00	0
02	23 00	STO 0
03	23 01	STO 1
04	74	R/S
05	23 02	STO 2
06	41	–
07	15 02	g x^2
08	24 02	RCL 2
09	71	÷
10	23 51 01	STO + 1
11	24 00	RCL 0
12	01	1
13	51	+
14	23 00	STO 0
15	13 04	GTO 04
16	23 02	STO 2
17	41	–
18	15 02	g x^2
19	24 02	RCL 2
20	71	÷
21	23 41 01	STO – 1
22	24 00	RCL 0
23	01	1
24	41	–
25	23 00	STO 0
26	13 04	GTO 04
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 n
R_1 χ^2
R_2 E_i
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	PRGM	R/S		0.00
3	Perform for i = 1,..., n:						
	Input observed and expected						
	frequencies	O_i	↑				
		E_i	R/S				i
4	Delete erroneous data	O_k	↑				
		E_k	GTO	16	R/S		
5	Display χ^2		RCL	1			χ^2
6	For new case, go to step 2.						

Example:

O_i	8	50	47	56	5	14
E_i	9.6	46.75	51.85	54.4	8.25	9.15

Solution:

$\chi^2 = 4.84$

PAIRED t STATISTIC

Given a set of paired observations from two normal populations with means μ_1, μ_2 (unknown)

x_i	x_1	x_2	...	x_n
y_i	y_1	y_2	...	y_n

let

$$D_i = x_i - y_i$$

$$\overline{D} = \frac{1}{n} \sum_{i=1}^{n} D_i$$

$$s_D = \sqrt{\frac{\Sigma D_i^2 - \frac{1}{n} (\Sigma D_i)^2}{n - 1}}$$

$$s_{\overline{D}} = \frac{s_D}{\sqrt{n}}$$

The test statistic

$$t = \frac{\overline{D}}{s_{\overline{D}}},$$

which has n − 1 degrees of freedom (df), can be used to test the null hypothesis

$$H_0: \mu_1 = \mu_2.$$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	41	−
02	25	Σ+
03	13 00	GTO 00
04	14 22	f s
05	24 03	RCL 3
06	14 02	f $\sqrt{x}$
07	71	÷
08	14 21	f $\bar{x}$
09	21	x⇄y
10	71	÷
11	74	R/S
12	24 03	RCL 3
13	01	1
14	41	−
15	13 00	GTO 00
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1
R_2
R_3 n
R_4 Used
R_5 Used
R_6 ΣD_i
R_7 ΣD_i^2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Perform for i = 1,..., n:						
	Input one pair of observations	x_i	↑				
		y_i	R/S				i
4	Delete erroneous data	x_k	↑				
		y_k	−	f	Σ−		
5	Compute t and df		GTO	04	R/S		t
			R/S				df
6	For new case, go to step 2.						

Example:

x_i	14	17.5	17	17.5	15.4
y_i	17	20.7	21.6	20.9	17.2

Solution:

t = –7.16

df = 4.00

t STATISTIC FOR TWO MEANS

Suppose $\{x_1, x_2, ..., x_{n_1}\}$ and $\{y_1, y_2, ..., y_{n_2}\}$ are independent random samples from two normal populations having means μ_1, μ_2 (unknown) and the same unknown variance σ^2.

We want to test the null hypothesis

$$H_0 : \mu_1 - \mu_2 = D$$

where D is a given number.

Define

$$\bar{x} = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i$$

$$\bar{y} = \frac{1}{n_2} \sum_{i=1}^{n_2} y_i$$

$$t = \frac{\bar{x} - \bar{y} - D}{\sqrt{\dfrac{1}{n_1} + \dfrac{1}{n_2}} \sqrt{\dfrac{\Sigma x_i^2 - n_1 \bar{x}^2 + \Sigma y_i^2 - n_2 \bar{y}^2}{n_1 + n_2 - 2}}}$$

We can use this t statistic, which has the t distribution with $n_1 + n_2 - 2$ degrees of freedom, to test the null hypothesis H_0.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 03	RCL 3
02	23 00	STO 0
03	24 06	RCL 6
04	23 01	STO 1
05	14 21	f $\bar{x}$
06	23 02	STO 2
07	34	CLX
08	23 03	STO 3
09	23 06	STO 6
10	23 07	STO 7
11	74	R/S
12	31	↑
13	14 21	f $\bar{x}$
14	51	+
15	24 02	RCL 2
16	21	$x \rightleftarrows y$
17	41	–
18	24 00	RCL 0
19	15 22	g 1/x
20	24 03	RCL 3
21	15 22	g 1/x
22	51	+
23	14 02	f $\sqrt{x}$
24	71	÷

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	24 01	RCL 1
26	24 02	RCL 2
27	15 02	g x^2
28	24 00	RCL 0
29	61	x
30	41	–
31	24 06	RCL 6
32	51	+
33	14 21	f $\bar{x}$
34	15 02	g x^2
35	24 03	RCL 3
36	61	x
37	41	–
38	24 00	RCL 0
39	24 03	RCL 3
40	51	+
41	02	2
42	41	–
43	71	÷
44	14 02	f $\sqrt{x}$
45	71	÷
46	13 00	GTO 00
47		
48		
49		

REGISTERS	
R_0	n_1
R_1	Σx^2
R_2	$\bar{x}$
R_3	n_2
R_4	Used
R_5	Used
R_6	Σy^2
R_7	Σy

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG			
3	Perform for i = 1,..., n_1 :						
	Input x-value	x_i	Σ+				i
4	Initialize for y		f	PRGM	R/S		0.00
5	Perform for i = 1,..., n_2 :						
	Input y-value	y_i	Σ+				i
6	Input D and compute t	D	R/S				t
7	To find the means of x- and y-						
	values		RCL	2			$\bar{x}$
			f	$\bar{x}$			$\bar{y}$
8	For a new case, go to step 2.						

Example:

x: 79, 84, 108, 114, 120, 103, 122, 120

y: 91, 103, 90, 113, 108, 87, 100, 80, 99, 54

$n_1 = 8$

$n_2 = 10$

D = 0 (i.e., $H_0 : \mu_1 = \mu_2$)

Solution:

t = 1.73

$\overline{x} = 106.25$

$\overline{y} = 92.50$

ONE SAMPLE TEST STATISTICS FOR THE MEAN

For a normal population (x_1, x_2 ..., x_n) with a known variance σ^2, a test of the null hypothesis

$$H_0: \text{mean } \mu = \mu_0$$

is based on the z statistic (which has a standard normal distribution)

$$z = \frac{\sqrt{n}\ (\bar{x} - \mu_0)}{\sigma}.$$

If the variance σ^2 is unknown, then

$$t = \frac{\sqrt{n}\ (\bar{x} - \mu_0)}{s}$$

is used instead. This t statistic has the t distribution with n – 1 degrees of freedom. $\bar{x}$ and s are the sample mean and standard deviation.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	14 21	f $\bar{x}$
02	21	x⇄y
03	41	–
04	24 03	RCL 3
05	14 02	f $\sqrt{x}$
06	61	x
07	23 00	STO 0
08	34	CLX
09	74	R/S
10	24 00	RCL 0
11	14 22	f s
12	71	÷
13	74	R/S
14	24 00	RCL 0
15	21	x⇄y
16	71	÷
17	13 00	GTO 00
18		
19		
20		
21		
22		
23		
24		

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0 $\sqrt{n}\ (\bar{x} - \mu_0)$
R_1
R_2
R_3 n
R_4 Used
R_5 Used
R_6 Σx
R_7 Σx^2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG			
3	Perform for i = 1,..., n:						
	Input value	x_i	Σ+				i
4	Input μ_0	μ_0	f	PRGM	R/S		0.00
5	Compute t		GTO	10	R/S		t
	or						
5	Input σ and compute z	σ	GTO	14	R/S		z
6	For new case, go to step 2.						

Example:

Suppose $\mu_0 = 2$, for the following set of data

$\{2.73, 0.45, 2.52, 1.19, 3.51, 2.75, 1.79, 1.83, 1, 0.87, 1.9, 1.62, 1.74, 1.92, 1.24, 2.68\}$

Solution:

test statistic t = –.69

or z = –.57 if $\sigma = 1$.

CHAPTER 7 SURVEYING

FIELD ANGLE TRAVERSE

A traverse is a series of line segments joined with specific lengths and angular relations to each other. With many applications in surveying, the field angle traverse may be used in establishing boundary lines, road layout, and in numerous construction situations. The transit and "chain" (commonly a length of steel tape) are often used to establish the angles and distances involved in a field angle traverse.

Starting at a known point from a given reference direction, the transit man establishes the direction of a new line by measuring the angle or deflection turned to align the scope of the transit to the new line. With a measured distance to the end point on the new line and its direction, coordinates of the end point relative to the origin may be established. The transit is then moved to the new "origin", the reference direction is the line just determined, and the process continues.

To run this program, the user must input the northing and easting of his starting point, the reference azimuth, and then the direction and distance from each point in the traverse to the next point. The direction may be input either as a deflection right or left, or as an angle right or left. The distance may be input either as horizontal distance, or as slope distance with zenith angle.

Equations:

$$\text{H Dist} = \text{S Dist} \sin(\text{Znth ang})$$

$$N_{i+1} = N_i + \text{H Dist} \cos \text{Az}$$

$$E_{i+1} = E_i + \text{H Dist} \sin \text{Az}$$

$$\text{Area} = \tfrac{1}{2}[(N_2 + N_1)(E_2 - E_1) + (N_3 + N_2)(E_3 - E_2) + \\ \ldots + (N_n + N_1)(E_1 - E_n)]$$

where: N, E = Northing, easting of a point

Subscript i refers to current point

Subscript n refers to next to last point

Numeric subscript refers to point number

Az = Azimuth of a course

H Dist = Horizontal distance

S Dist = Slope distance

Znth ang = Zenith angle

Notes:

1. The calculation for area of a closed traverse may be inaccurate for cases in which the coordinates of the figure are quite large, such as in state plane coordinate systems. In such cases, the user may run the Area By Double Meridian Distance program to calculate an accurate value for area once the bearings and distances have been established by this program.

2. All angular inputs and outputs are in the form degrees, minutes, and seconds (D.MS).

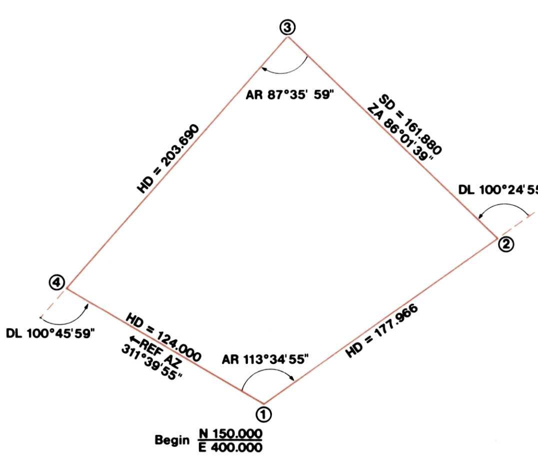
DISPLAY LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	COMMENTS
00							
01	15 00	g →H	Ref Az				Convert to decimal degrees
02	01	1	1	Ref Az			
03	08	8	18	Ref Az			
04	00	0	180	Ref Az			
05	51	+	180 + Az				
06	23 00	STO 0	180 + Az				
07	24 01	RCL 1	N_1	180 + Az			
08	23 05	STO 5	N_1	180 + Az			Initialize "previous N"
09	00	0	0	N_1	180 + Az		Clear R_3, R_4, for
10	23 03	STO 3	0	N_1	180 + Az		accumulation
11	23 04	STO 4	0	N_1	180 + Az		
12	74	R/S	0	N_1	180 + Az		
13	15 00	g →H	Angle				Convert to decimal degrees
14	01	1	1	Angle			
15	08	8	18	Angle			
16	00	0	180	Angle			
17	51	+	180 + Ang				
18	14 00	f →H.MS	(D.MS)				
19	15 00	g →H	Defl				Deflection comes in here
20	24 00	RCL 0	Az	Defl			
21	51	+	Az + Defl				Find new azimuth
22	23 00	STO 0	Az_i				
23	14 00	f →H.MS	Az_i				Convert to D.MS for
24	74	R/S	Az_i				display
25	13 29	GTO 29	H Dist				
26	21	x⇄y	Zn Ang	S Dist			
27	14 04	f SIN	sin Zn	S Dist			
28	61	x	H Dist				H Dist = sin Zn (S Dist)
29	23 51 03	STO + 3	H Dist				Accumulate H Dist
30	24 00	RCL 0	Az	H Dist			
31	21	x⇄y	H Dist	Az			
32	14 09	f →R	ΔN	ΔE			
33	23 51 01	STO + 1	ΔN	ΔE			ΔN = H Dist (cos Az)
34	21	x⇄y	ΔE	ΔN			
35	23 51 02	STO + 2	ΔE	ΔN			ΔE = H Dist (sin Az)
36	24 05	RCL 5	N_{i-1}	ΔE	ΔN		
37	24 01	RCL 1	N_i	N_{i-1}	ΔE	ΔN	
38	23 05	STO 5	N_i	N_{i-1}	ΔE	ΔN	Update "previous N"
39	51	+	$(N_i + N_{i-1})$	ΔE	ΔN		
40	61	x	ΔA	ΔN			ΔA = $(N_i + N_{i-1})$ ΔE
41	02	2	2	ΔA	ΔN		
42	71	÷	1/2 ΔA	ΔN			
43	23 51 04	STO + 4	1/2 ΔA	ΔN			Accumulate Area
44	24 01	RCL 1	N_i	1/2 ΔA	ΔN		
45	74	R/S	N_i	1/2 ΔA	ΔN		Display Northing
46	24 02	RCL 2	E_i	N_i	1/2 ΔA	ΔN	
47	13 12	GTO 12	E_i	N_i	1/2 ΔA	ΔN	Display Easting
48							
49							

REGISTERS	
R_0	Az
R_1	Current N
R_2	Current E
R_3	Σ H Dist
R_4	Area
R_5	Previous N
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Input the starting point coord-						
	inates	N_1	STO	1			
		E_1	STO	2			
3	Input the reference azimuth	Ref Az, D.MS	f	PRGM	R/S		0.00
4a.	If angle right	AR, D.MS	R/S				Az_i, D.MS
4b.	If angle left	AL, D.MS	CHS	R/S			Az_i, D.MS
4c.	If deflection right	DR, D.MS	GTO	19	R/S		Az_i, D.MS
4d.	If deflection left	DL, D.MS	CHS	GTO	19	R/S	Az_i, D.MS
5a.	If horizontal distance	H Dist	R/S				N_i
			R/S				E_i
5b.	If slope distance, input Zenith						
	Angle and Slope Distance	Zn, Ang, D.MS	↑				
		S Dist	GTO	26	R/S		N_i
			R/S				E_i
6	Repeat steps 4-5 for successive						
	courses.						
7	Display total horizonal distance						
	traversed		RCL	3			Σ H Dist
8	Display area for closed traverse						
	(ignore sign)		RCL	4			Area

Example:

The diagram below shows measurements taken for a closed traverse. Find the coordinates of points 2, 3, and 4, the total horizontal distance traversed, and the area of the figure.



Solution:

150 **STO** [1] 400 **STO** [2] 311.3955 [f] [PRGM] [R/S] → 0.00

113.3455 [R/S] 177.966 [R/S] → 224.515 (N_2)

[R/S] → 561.615 (E_2)

100.2455 **CHS** **GTO** [1] [9] [R/S] 86.0139 [↑] 161.880 **GTO**

[2] [6] [R/S] → 356.526 (N_3)

[R/S] → 468.601 (E_3)

87.3559 [R/S] 203.690 [R/S] → 232.335 (N_4)

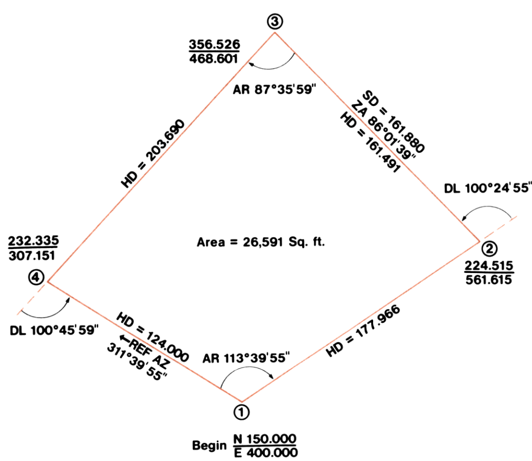
[R/S] → 307.151 (E_4)

100.4559 **CHS** **GTO** [1] [9] [R/S] 124.0 [R/S] → 149.903 (N_1)

[R/S] → 399.784 (E_1)

RCL [3] → 667.144 (ΣH Dist)

RCL [4] → 26590.68 (Area)



Calculated ending coordinates $\frac{\mathbf{N = 149.903}}{\mathbf{E = 399.784}}$

AREA BY DOUBLE MERIDIAN DISTANCE

This program computes the area of a straight-sided closed figure from the bearings and lengths of its sides. It is generally more accurate than methods which calculate area from the coordinates of the figure.

$$\text{Area} = \frac{1}{2} \sum_{i} \text{DMD}_i \times \text{Latitude}_i$$

$$\text{DMD}_i = \text{DMD}_{i-1} + \text{Departure}_{i-1} + \text{Departure}_i$$

where

$$\text{Departure}_i = \text{Dist}_i \sin \text{Az}_i \qquad \text{Latitude}_i = \text{Dist}_i \cos \text{Az}_i$$

Note:

Angles are input as bearing and quadrant code. The quadrant code is 1 for NE, 2 for SE, 3 for SW, and 4 for NW.

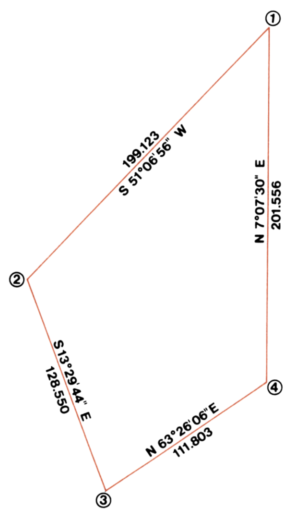
DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	15 00	g→H
02	23 00	STO 0
03	74	R/S
04	31	↑
05	31	↑
06	02	2
07	71	÷
08	14 01	f INT
09	01	1
10	08	8
11	00	0
12	61	x
13	21	x⇄y
14	14 73	f LAST X
15	61	x
16	14 05	f COS
17	24 00	RCL 0
18	61	x
19	41	–
20	23 00	STO 0
21	14 00	f→H.MS
22	74	R/S
23	24 00	RCL 0
24	21	x⇄y
25	14 09	f→R
26	21	x⇄y
27	24 02	RCL 2
28	21	x⇄y
29	23 02	STO 2
30	51	+
31	24 01	RCL 1
32	51	+
33	23 01	STO 1
34	61	x
35	02	2
36	71	÷
37	23 51 03	STO + 3
38	24 03	RCL 3
39	13 00	GTO 00
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS	
R_0	Brg, Az_i
R_1	DMD_{i-1}
R_2	$Departure_{i-1}$
R_3	Area
R_4	
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Initialize		f	REG	f	PRGM	
3	Input bearing	Brg, D.MS	R/S				Brg, dec. deg.
4	Input quadrant code	Quad	R/S				Az, D.MS
5	Input distance	Dist	R/S				Area
6	Repeat steps 3, 4, 5 for succes-						
	sive courses. Area is displayed						
	after last distance has been input.						

Example:

Compute the area of the figure below.



Solution:

Area = 20937.44 sq. ft.

INVERSE FROM COORDINATES

This program uses coordinates to calculate distance and bearing between points of a traverse. The area in square feet and a summation of distance inversed are also computed.

$$\text{H Dist} = \sqrt{(N_i - N_{i-1})^2 + (E_i - E_{i-1})^2} \qquad \text{Az} = \tan^{-1} \frac{E_i - E_{i-1}}{N_i - N_{i-1}}$$

$$\text{Area} = \tfrac{1}{2}[(N_2 + N_1)(E_2 - E_1) + (N_3 + N_2)(E_3 - E_2) + \ldots .(N_n + N_1)(E_1 - E_n)]$$

where N, E = Northing, easting of a point

Subscript i referes to current point

Subscript n refers to next to last point

Numeric subscript refers to point number

H Dist = Horizontal distance

Az = Azimuth of a course

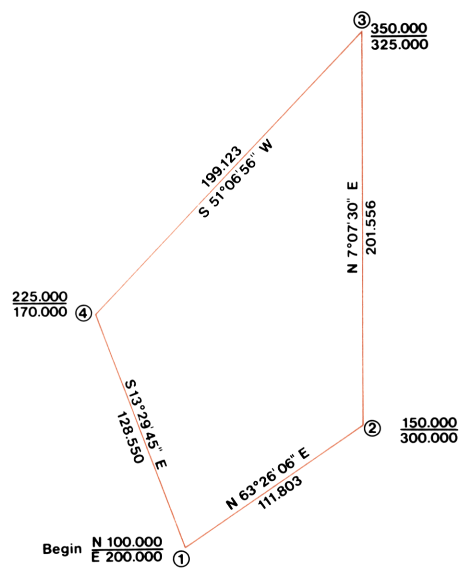
DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	14 33	f REG
02	23 02	STO 2
03	21	x⇄y
04	23 00	STO 0
05	23 01	STO 1
06	74	R/S
07	24 02	RCL 2
08	41	−
09	23 51 02	STO + 2
10	23 05	STO 5
11	21	x⇄y
12	24 01	RCL 1
13	41	−
14	23 51 01	STO + 1
15	15 09	g →P
16	23 51 03	STO + 3
17	74	R/S
18	21	x⇄y
19	15 51	g x≥0
20	13 25	GTO 25
21	03	3
22	06	6
23	00	0
24	51	+

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	31	↑
26	31	↑
27	09	9
28	00	0
29	71	÷
30	01	1
31	51	+
32	14 01	f INT
33	21	x⇄y
34	14 04	f SIN
35	15 04	g SIN⁻¹
36	15 41	g x<0
37	32	CHS
38	14 00	f →H.MS
39	24 00	RCL 0
40	24 01	RCL 1
41	23 00	STO 0
42	51	+
43	24 05	RCL 5
44	61	x
45	02	2
46	71	÷
47	23 51 04	STO + 4
48	22	R↓
49	13 06	GTO 06

REGISTERS	
R_0	Previous N
R_1	Current N
R_2	Current E
R_3	Σ H Dist
R_4	Area
R_5	ΔE
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Input starting coordinates	N_1	↑				
		E_1	f	PRGM	R/S		
3	Input next coordinates and						
	display distance	N_i	↑				
		E_i	R/S				H Dist
4	Compute bearing and quadrant						
	code		R/S				Brg, D.MS
			R↓				Quad code
5	Repeat steps 3-4 for successive						
	courses						
6	Display total distance inversed		RCL	3			Σ H Dist
7	Display area of closed figure						
	(ignore the sign)		RCL	4			Area

Example:



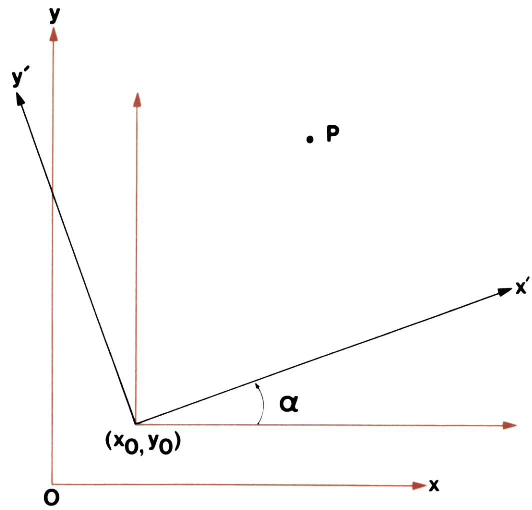
Area = 20937.5 Sq. ft.

Total distance inversed = 641.033

CHAPTER 8
TRIGONOMETRY AND ANALYTICAL GEOMETRY

COORDINATE TRANSLATION AND ROTATION

There are occasions, for instance in cartography or metalworking, when it is necessary or advantageous to shift one's frame of reference. In mathematical terms, the occasion calls for a translation and/or rotation of the coordinate system. The origin is translated from (0, 0) to a new point, (x_0, y_0), and the x and y axes are then rotated through an angle α to give new axes, x' and y'. Suppose that a point P has coordinates (x, y) with respect to the old system of x and y axes. The problem then is to find the coordinates (x', y') of P with respect to the new system whose axes are x' and y'. The diagram below illustrates this situation.



Equations:

$$x' = (x - x_0) \cos \alpha + (y - y_0) \sin \alpha$$

$$y' = -(x - x_0) \sin \alpha + (y - y_0) \cos \alpha$$

Notes:

1. The program may be used to solve a problem of translation only, or of rotation only, or of combined translation and rotation. If the problem involves translation alone, a value of $\alpha = 0$ must be input. For rotation alone, the values $x_0 = y_0 = 0$ must be input.

2. The program assumes the following sign convention: α should be input as a positive number if the rotation is counterclockwise, and negative if clockwise.

Programming Remarks:

This program demonstrates a particularly powerful application of the polar-to-rectangular conversion ($\boxed{\text{f}}$ $\boxed{\to\text{R}}$) when combined with the capabilities of the four-register stack. The subterms $(x - x_0) \cos \alpha$, $(x - x_0) \sin \alpha$, $(y - y_0) \cos \alpha$, and $(y - y_0) \sin \alpha$ are all generated through $\boxed{\text{f}}$ $\boxed{\to\text{R}}$ and stored in the stack until needed. A more straightforward program using $\boxed{\text{f}}$ $\boxed{\text{sin}}$ and $\boxed{\text{f}}$ $\boxed{\text{cos}}$ would have required 30 program steps (as compared to 19) and one more storage register.

DISPLAY		KEY ENTRY	X	Y	Z	T	COMMENTS
LINE	CODE						
00			y	x			
01	23 03	STO 3	y	x			
02	22	R↓	x			y	
03	24 02	RCL 2	α	x			
04	21	x⇄y	x	α			
05	24 00	RCL 0	x_0	x	α		
06	41	–	Δx	α			$\Delta x = x - x_0$
07	14 09	f →R	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$			
08	24 03	RCL 3	y	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$		
09	24 01	RCL 1	y_0	y	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$	
10	41	–	Δy	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$	$\Delta x \sin \alpha$	$\Delta y = y - y_0$
11	24 02	RCL 2	α	Δy	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$	
12	21	x⇄y	Δy	α	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$	
13	14 09	f →R	$\Delta y \cos \alpha$	$\Delta y \sin \alpha$	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$	
14	22	R↓	$\Delta y \sin \alpha$	$\Delta x \cos \alpha$	$\Delta x \sin \alpha$	$\Delta y \cos \alpha$	
15	51	+	x'	$\Delta x \sin \alpha$	$\Delta y \cos \alpha$	$\Delta y \cos \alpha$	$x' = \Delta x \cos \alpha + \Delta y \sin \alpha$
16	74	R/S	x'	$\Delta x \sin \alpha$	$\Delta y \cos \alpha$	$\Delta y \cos \alpha$	
17	22	R↓	$\Delta x \sin \alpha$	$\Delta y \cos \alpha$	$\Delta y \cos \alpha$	x'	
18	41	–	y'	$\Delta y \cos \alpha$	x'	x'	$y' = -\Delta x \sin \alpha + \Delta y \cos \alpha$
19	13 00	GTO 00	y'	$\Delta y \cos \alpha$	x'	x'	
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							

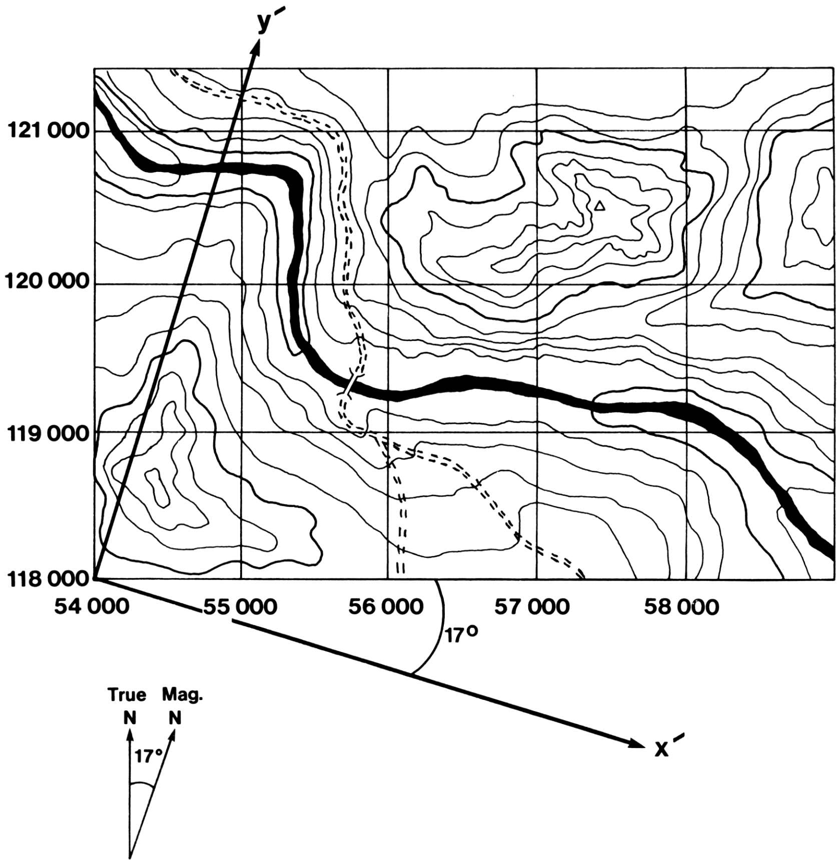
REGISTERS	
R_0	x_0
R_1	y_0
R_2	α
R_3	y
R_4	
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store origin of new coordinate						
	system	x_0	STO	0			
		y_0	STO	1			
3	Store angle of rotation	α	STO	2	f	PRGM	
4	Convert coordinates from old to						
	new system	x	↑				
		y	R/S				x'
			R/S				y'
5	Perform step 4 for as many						
	points as desired						
6	For a new case, go to step 2.						

Example:

A backpacker's route will take him cross-country away from the marked trails of an area. He knows that he will have to check his compass frequently against his map over this terrain, and regrets that the map is in such an inconvenient format for his purposes. In the first place, the grid lines on his map represent distances in feet from an origin about 25 miles away, which are such large numbers that they are hard to calculate with. Secondly, the map's grid is based on true north while his compass readings are relative to magnetic north, a variation of 17°.

Before he leaves home, the packer decides to draw a rough version of the map for his own convenience, locating his origin at the grid point (54 000, 118 000) and rotating his axes by 17° in a clockwise direction. As a first step, he wants to find the new coordinates of the bridge and the peak of the hill, whose coordinates in the old system are (55 750, 119 300) and (57 450, 120 500) respectively.



Solution:

54000 STO 0 118000 STO 1 17 CHS STO 2 f PRGM

55750 ↑ 119300 R/S ⟶ 1293.45

R/S ⟶ 1754.85

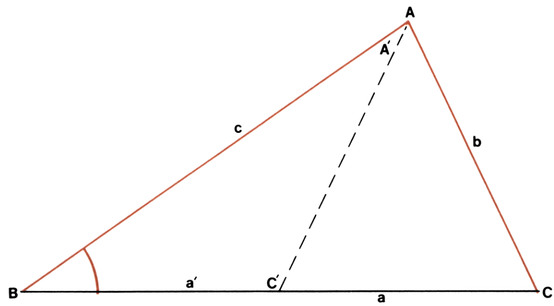
The new coordinates of the bridge are (1293, 1755).

57450 ↑ 120500 R/S ⟶ 2568.32

R/S ⟶ 3399.44

The new coordinates of the peak are (2568, 3399).

TRIANGLE SOLUTION B, b, c



Given two sides and a non-included angle, this program solves the triangle for the remaining parameters by the following formulas:

1. $C = \sin^{-1}\left(\dfrac{c \sin B}{b}\right)$

2. $A = 2 \sin^{-1} 1 - (B + C) = \pi \text{ radians} - (B + C) = 180^\circ - (B + C)$

   $= 200 \text{ grads} - (B + C)$

3. $a = \dfrac{b \sin A}{\sin B}$

If B is acute ($< 90^\circ$) and $b < c$, a second set of solutions exists and is calculated by the following formulas:

4. $C' = 2 \sin^{-1} 1 - C$

5. $A' = 2 \sin^{-1} 1 - (B + C')$

6. $a' = \dfrac{b \sin A'}{\sin B}$

The area is computed with the formula

$$\text{Area} = \frac{1}{2} \, ac \sin B$$

This program works in any angular mode. However, if in degrees, decimal degrees are assumed.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 03	RCL 3
02	24 01	RCL 1
03	14 04	f SIN
04	61	x
05	24 02	RCL 2
06	71	÷
07	15 04	g SIN^{-1}
08	23 05	STO 5
09	74	R/S
10	24 01	RCL 1
11	51	+
12	01	1
13	15 04	g SIN^{-1}
14	02	2
15	61	x
16	23 04	STO 4
17	21	x⇄y
18	41	–
19	74	R/S
20	14 04	f SIN
21	24 02	RCL 2
22	61	x
23	24 01	RCL 1
24	14 04	f SIN

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	71	÷
26	74	R/S
27	24 03	RCL 3
28	61	x
29	24 01	RCL 1
30	14 04	f SIN
31	61	x
32	02	2
33	71	÷
34	74	R/S
35	24 04	RCL 4
36	24 05	RCL 5
37	41	–
38	74	R/S
39	13 10	GTO 10
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1 B
R_2 b
R_3 c
R_4 $2 \sin^{-1} 1$
R_5 C
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store B, b, and c	B	STO	1			
		b	STO	2			
		c	STO	3			
3	Solve the triangle		f	PRGM	R/S		C*
			R/S				A*
			R/S				a*
			R/S				Area
4	If B < 90° and b < c, find al-						
	ternate solution		R/S				C′*
			R/S				A′*
			R/S				a′*
			R/S				Area′
	* The stack must be maintained						
	at these positions.						

Example:

Given the following two sides and non-included angle:

$B = 42.3°$
$b = 25.6$
$c = 32.8$

Solve the triangle.

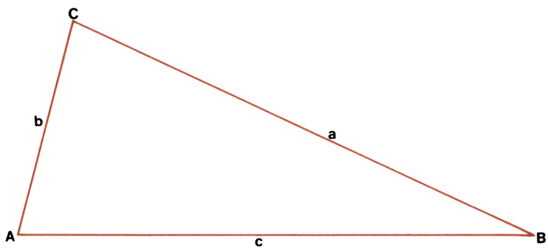
Solution:

Since B is less than 90° and b < c, two sets of solutions exist.

$C = 59.58°$
$A = 78.12°$
$a = 37.22$
Area = 410.85

$C' = 120.42°$
$A' = 17.28°$
$a' = 11.30$
Area′ = 124.68

TRIANGLE SOLUTION a, b, c



Given three sides of a triangle this program solves the triangle for the remaining parameters by the following formulas:

$$C = \cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

$$B = \sin^{-1}\left(\frac{b \sin C}{c}\right) \qquad A = \sin^{-1}\left(\frac{a \sin C}{c}\right)$$

This program also computes the area by the following formula:

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{where } s = \frac{1}{2}(a + b + c)$$

Reletter if necessary to make c the largest side. The program works in any angular mode. However, if in degree mode decimal degrees are assumed.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 01	RCL 1
02	24 02	RCL 2
03	15 09	g →P
04	15 02	g x^2
05	24 03	RCL 3
06	15 02	g x^2
07	41	–
08	24 01	RCL 1
09	24 02	RCL 2
10	61	x
11	02	2
12	61	x
13	71	÷
14	15 05	g COS^{-1}
15	74	R/S
16	14 04	f SIN
17	24 03	RCL 3
18	71	÷
19	23 00	STO 0
20	24 02	RCL 2
21	61	x
22	15 04	g SIN^{-1}
23	74	R/S
24	24 00	RCL 0

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	24 01	RCL 1
26	61	x
27	15 04	g SIN^{-1}
28	74	R/S
29	24 01	RCL 1
30	24 02	RCL 2
31	51	+
32	24 03	RCL 3
33	51	+
34	02	2
35	71	÷
36	31	↑
37	23 00	STO 0
38	24 01	RCL 1
39	41	–
40	61	x
41	24 00	RCL 0
42	24 02	RCL 2
43	41	–
44	61	x
45	24 00	RCL 0
46	24 03	RCL 3
47	41	–
48	61	x
49	14 02	f $\sqrt{x}$

REGISTERS
R_0 Used
R_1 a
R_2 b
R_3 c
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store sides (c is the largest)	a	STO	1			
		b	STO	2			
		c	STO	3			
3	Solve the triangle		f	PRGM	R/S		C*
			R/S				B*
			R/S				A
			R/S				Area
4	If only the area is needed:	a	STO	1			
		b	STO	2			
		c	STO	3			
			GTO	29	R/S		Area
	* The stack must be maintained						
	at these points.						

Example:

Let a = 5.43, b = 10.46, c = 14.87

Solution:

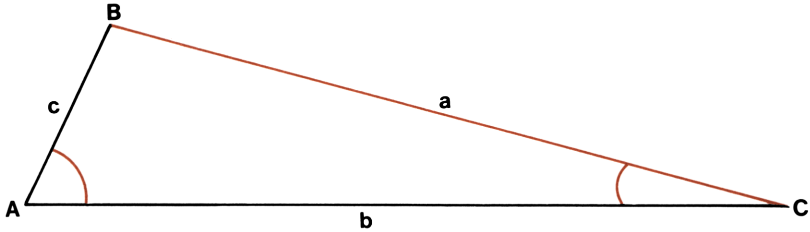
C = 136.37°

B = 29.04°

A = 14.59°

Area = 19.60

TRIANGLE SOLUTION a, A, C



Given two angles and an opposite side this program solves the triangle for the remaining parameters by the following formulas:

$$B = 2 \sin^{-1} 1 - (A + C) = \pi \text{ radians} - (A + C) = 180^\circ - (A + C)$$
$$= 200 \text{ grads} - (A + C)$$

$$b = \frac{a \sin B}{\sin A}$$

$$c = \frac{a \sin C}{\sin A}$$

The area is computed with the following formula:

$$\text{Area} = \frac{1}{2} \text{ ab } \sin C$$

The program works in any angular mode. However, if in degree mode all angles are assumed to be in decimal degrees.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	01	1
02	15 04	g SIN^{-1}
03	02	2
04	61	x
05	24 02	RCL 2
06	24 03	RCL 3
07	51	+
08	41	–
09	74	R/S
10	14 04	f SIN
11	24 01	RCL 1
12	61	x
13	24 02	RCL 2
14	14 04	f SIN
15	71	÷
16	23 04	STO 4
17	74	R/S
18	24 01	RCL 1
19	14 73	f LASTx
20	71	÷
21	24 03	RCL 3
22	14 04	f SIN
23	61	x
24	74	R/S

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	24 01	RCL 1
26	24 04	RCL 4
27	61	x
28	24 03	RCL 3
29	14 04	f SIN
30	61	x
31	02	2
32	71	÷
33	13 00	GTO 00
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1 a
R_2 A
R_3 C
R_4 b
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store a, A, and C	a	STO	1			
		A	STO	2			
		C	STO	3			
3	Solve the triangle		f	PRGM	R/S		B*
			R/S				b*
			R/S				c
			R/S				Area
	* The stack must be maintained						
	at these points.						

Example:

Let a = 19.6, A = 40.25°, C = 61.06°

Solution:

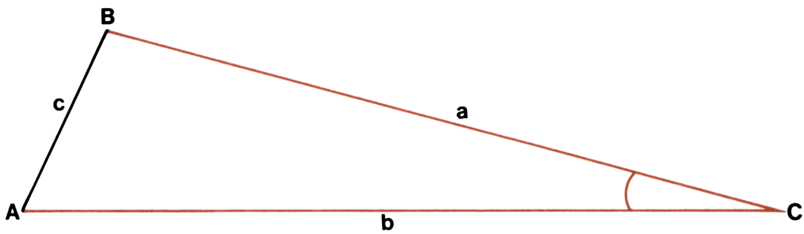
B = 78.69°

b = 29.75

c = 26.55

Area = 255.11

TRIANGLE SOLUTION a, b, C



Given two sides and their included angle this program solves the triangle for the remaining parameters by the following formulas:

$$c = \sqrt{a^2 + b^2 - 2ab \cos C} \qquad A = \sin^{-1}\left(\frac{a \sin C}{c}\right)$$

$$B = 2 \sin^{-1} 1 - (A + C) = \pi \text{ radians} - (A + C) = 180^\circ - (A + C)$$
$$= 200 \text{ grads} - (A + C)$$

The area is calculated by

$$\text{Area} = \frac{1}{2} ab \sin C$$

Reletter if necessary, to make a the smaller of a and b.

This program works in any angular mode. However, if in degrees decimal degrees are assumed.

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	24 01	RCL 1
02	24 02	RCL 2
03	15 09	g →P
04	15 02	g x^2
05	24 01	RCL 1
06	24 02	RCL 2
07	61	x
08	02	2
09	61	x
10	24 03	RCL 3
11	14 05	f COS
12	61	x
13	41	–
14	14 02	f $\sqrt{x}$
15	74	R/S
16	24 01	RCL 1
17	24 03	RCL 3
18	14 04	f SIN
19	61	x
20	21	x⇄y
21	71	÷
22	15 04	g SIN^{-1}
23	74	R/S
24	01	1

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	15 04	g SIN^{-1}
26	02	2
27	61	x
28	21	x⇄y
29	24 03	RCL 3
30	51	+
31	41	–
32	74	R/S
33	24 03	RCL 3
34	14 04	f SIN
35	24 01	RCL 1
36	61	x
37	24 02	RCL 2
38	61	x
39	02	2
40	71	÷
41	13 00	GTO 00
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS	
R_0	
R_1	a
R_2	b
R_3	C
R_4	
R_5	
R_6	
R_7	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store a, b, and C (a is smaller						
	of a and b)	a	STO	1			
		b	STO	2			
		C	STO	3			
3	Solve the triangle		f	PRGM	R/S		c*
			R/S				A*
			R/S				B
			R/S				Area
4	If only the area is needed:	a	STO	1			
		b	STO	2			
		C	STO	3			
			GTO	33	R/S		Area
	* The stack must be maintained						
	at these points.						

Example:

Let a = 146, b = 227, C = 31.49°

Solution:

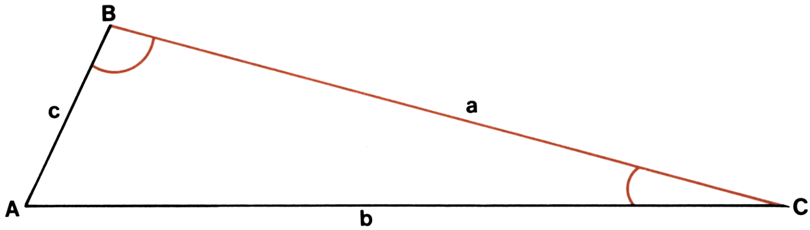
c = 127.76

A = 36.65°

B = 111.86°

Area = 8655.86

TRIANGLE SOLUTION a, B, C



Given two angles and their included side this program solves the triangle for the remaining parameters by the following formulas:

$$A = 2 \sin^{-1} 1 - (B + C) = \pi \text{ radians} - (B + C) = 180^\circ - (B + C)$$

$$= 200 \text{ grads} - (B + C)$$

$$b = \frac{a \sin B}{\sin A}$$

$$c = \frac{a \sin C}{\sin A}$$

The area is found using the formula:

$$\text{Area} = \frac{a^2 \sin B \sin C}{2 \sin (B + C)}$$

The program works in any angular mode. However, if in degrees the program assumes decimal degrees.

DISPLAY		KEY
LINE	CODE	ENTRY
00		
01	01	1
02	15 04	g SIN^{-1}
03	02	2
04	61	x
05	24 02	RCL 2
06	24 03	RCL 3
07	51	+
08	41	–
09	23 04	STO 4
10	74	R/S
11	24 01	RCL 1
12	24 04	RCL 4
13	14 04	f SIN
14	71	÷
15	23 04	STO 4
16	24 02	RCL 2
17	14 04	f SIN
18	61	x
19	74	R/S
20	24 04	RCL 4
21	24 03	RCL 3
22	14 04	f SIN
23	61	x
24	74	R/S

DISPLAY		KEY
LINE	CODE	ENTRY
25	24 01	RCL 1
26	15 02	g x^2
27	02	2
28	71	÷
29	24 02	RCL 2
30	14 04	f SIN
31	61	x
32	24 03	RCL 3
33	14 04	f SIN
34	61	x
35	24 02	RCL 2
36	24 03	RCL 3
37	51	+
38	14 04	f SIN
39	71	÷
40	13 00	GTO 00
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1 a
R_2 B
R_3 C
R_4 A, (a/sin A)
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	Store a, B, C	a	STO	1			
		B	STO	2			
		C	STO	3			
3	Solve the triangle		f	PRGM	R/S		A*
			R/S				b*
			R/S				c
			R/S				Area
4	If only the area is needed:	a	STO	1			
		B	STO	2			
		C	STO	3			
			GTO	25	R/S		Area
	* The stack must be maintained						
	at these points.						

Example:

Let a = 20.96, B = 64°32′, C = 35°06′.

Solution:

First convert B and C to decimal degrees.

A = 80.37°

b = 19.19

c = 12.22

Area = 115.66

HYPERBOLIC FUNCTIONS

This program evaluates the six hyperbolic functions by the following formulas:

1. $\sinh x = \dfrac{e^x - e^{-x}}{2}$

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

3. $\tanh x = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$

4. $\operatorname{csch} x = \dfrac{1}{\sinh x} \qquad (x \neq 0)$

5. $\operatorname{sech} x = \dfrac{1}{\cosh x}$

6. $\coth x = \dfrac{1}{\tanh x} \qquad (x \neq 0)$

DISPLAY		KEY ENTRY
LINE	CODE	
00		
01	15 07	g e^x
02	31	↑
03	15 22	g 1/x
04	41	–
05	02	2
06	71	÷
07	13 00	GTO 00
08	15 07	g e^x
09	31	↑
10	15 22	g 1/x
11	51	+
12	13 05	GTO 05
13	15 07	g e^x
14	31	↑
15	15 22	g 1/x
16	41	–
17	31	↑
18	31	↑
19	14 73	f LASTx
20	02	2
21	61	x
22	51	+
23	71	÷
24	13 00	GTO 00

DISPLAY		KEY ENTRY
LINE	CODE	
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	sinh x	x	f	PRGM	R/S		sinh x
	or						
	cosh x	x	GTO	08	R/S		cosh x
	or						
	tanh x	x	GTO	13	R/S		tanh x
	or						
	csch x	x	f	PRGM	R/S		
			g	$1/x$			csch x
	or						
	sech x	x	GTO	08	R/S		
			g	$1/x$			sech x
	or						
	coth x	x	GTO	13	R/S		
			g	$1/x$			coth x

Examples:

1. sinh 2.5 = 6.05
2. cosh 3.2 = 12.29
3. tanh 1.9 = 0.96
4. csch 4.6 = 0.02
5. sech (–.25) = 0.97
6. coth (–2.01) = –1.04

INVERSE HYPERBOLIC FUNCTIONS

This program evaluates the inverse hyperbolic functions by the following formulas:

1. $\sinh^{-1} x = \ln [x + (x^2 + 1)^{1/2}]$
2. $\cosh^{-1} x = \ln [x + (x^2 - 1)^{1/2}]$ $\quad x \geqslant 1$
3. $\tanh^{-1} x = \frac{1}{2} \ln \left[\frac{1+x}{1-x}\right]$ $\quad x^2 < 1$
4. $\operatorname{csch}^{-1} x = \sinh^{-1} \left[\frac{1}{x}\right]$ $\quad x \neq 0$
5. $\operatorname{sech}^{-1} x = \cosh^{-1} \left[\frac{1}{x}\right]$ $\quad 0 < x \leqslant 1$
6. $\coth^{-1} x = \tanh^{-1} \left[\frac{1}{x}\right]$ $\quad x^2 > 1$

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
00		
01	31	↑
02	31	↑
03	61	x
04	01	1
05	51	+
06	14 02	f $\sqrt{x}$
07	51	+
08	14 07	f LN
09	13 00	GTO 00
10	31	↑
11	31	↑
12	61	x
13	01	1
14	41	−
15	14 02	f $\sqrt{x}$
16	51	+
17	14 07	f LN
18	13 00	GTO 00
19	31	↑
20	31	↑
21	01	1
22	51	+
23	21	x⇄y
24	32	CHS

DISPLAY LINE	DISPLAY CODE	KEY ENTRY
25	01	1
26	51	+
27	71	÷
28	14 07	f LN
29	02	2
30	71	÷
31	13 00	GTO 00
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		

REGISTERS
R_0
R_1
R_2
R_3
R_4
R_5
R_6
R_7

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS				OUTPUT DATA/UNITS
1	Key in program						
2	$\sinh^{-1} x$	x	f	PRGM	R/S		$\sinh^{-1} x$
	or						
	$\cosh^{-1} x$	x	GTO	10	R/S		$\cosh^{-1} x$
	or						
	$\tanh^{-1} x$	x	GTO	19	R/S		$\tanh^{-1} x$
	or						
	$\operatorname{csch}^{-1} x$	x	g	$^1/x$	f	PRGM	
			R/S				$\operatorname{csch}^{-1} x$
	or						
	$\operatorname{sech}^{-1} x$	x	g	$^1/x$	GTO	10	
			R/S				$\operatorname{sech}^{-1} x$
	or						
	$\coth^{-1} x$	x	g	$^1/x$	GTO	19	
			R/S				$\coth^{-1} x$

Example:

1. $\sinh^{-1} (2.4) = 1.61$
2. $\cosh^{-1} (90) = 5.19$
3. $\tanh^{-1} (-.65) = -0.78$
4. $\operatorname{csch}^{-1} (2) = 0.48$
5. $\operatorname{sech}^{-1} (.4) = 1.57$
6. $\coth^{-1} (3.4) = 0.30$

INDEX

HEWLETT PACKARD

Sales and service from 172 offices in 65 countries.
19310 Pruneridge Avenue, Cupertino, California 95014

00025-90011 Rev. E 7/76