# The Calculator Edge

## *Using the HP28S in Precalculus*

John Paulling

# *The Calculator Edge*

*Using the HP28S in Precalculus*

John Paulling

*Nicholls State University*

Wm. C. Brown Publishers

ISBN 0-697-14963-3



Printed in the United States of America by Wm. C. Brown Publishers, 2460 Kerper Blvd., Dubuque, Iowa, 52001

10 9 8 7 6 5 4 3 2 1

## CONTENTS

## PREFACE

### Who Should Use This Manual?

**The Calculator Edge** is for students of precalculus who have HP28S calculators. Neither prior knowledge of precalculus nor prior experience with the HP28S calculator is assumed. This manual is intended to serve as a bridge between a precalculus text and the **Owner's Manual** and **Reference Manual** that come with the HP28S. Designed as a supplement to **Precalculus** by Murray Gechtman, it can be used under the direction of an instructor or by the individual student. There is sufficient flexibility to use this manual as a supplement to any precalculus course.

### What this Manual is Not

This manual is not a precalculus text. Neither is it a rewrite of the HP28S **Owner's Manual**. Detailed discussions are used only for the features of the HP28S that are most important for the precalculus student.

### How to Use Part I

In the first two chapters, series of examples are used to progressively introduce features of the HP28S. Repeated use of the most important features is intended for reinforcement. Consequently all examples should be worked through carefully. In fact, to develop a natural feel for the machine, it is advisable to habitually go through series of examples two or three times. People who already use the HP28S will probably go through Part I quickly. New users will have to go through Part I slowly.

Whether the HP28S is used by an individual adventurous student or an entire class, time will have to be devoted to studying **The Calculator Edge** as well as the HP28S **Owner's Manual** and **Reference Manual**. Most of this time needs to be at the very beginning.

Work through the chapters in Part I during the first two weeks of class. In these two chapters, references will be made to a few specific parts of the HP28S **Owner's Manual** and **Reference Manual**. If these referenced sections are studied carefully at the beginning, there will only be an occasional need to use the **Owner's Manual** or **Reference Manual** for the rest of your precalculus course.

**How to Use Part II**

In Part II the focus of **The Calculator Edge** will shift away from the HP28S to precalculus topics. If you are using Gechtman's **Precalculus** the symbol ***Gxxx** printed next to an example or explanation is to refer you to a related discussion on page **xxx** of the text.

Chapters 3 through 11 should be read as soon as the appropriate precalculus topic is discussed in class. You will find that **The Calculator Edge** will be easier to read after Chapter 2. Thereafter there will not be so much to learn about the HP28S and you will only need to use **The Calculator Edge** as a reference for your precalculus text.

The HP28S **Owner's Manual** and **Reference Manual** referred to throughout this book are the manuals that accompany the HP28S.

## Notational Conventions

Superscripts indicate use of the (red) shift key. For example, in applying the stroke sequence

CLEAR 16 ENTER 1/x

the shift key is used twice: just before CLEAR and again just before 1/x.

Large capital letters indicate single keystrokes. For example, in the six stroke sequence

Q ENTER 4 ENTER + DROP

Q, ENTER, and DROP refer to particular keys. In fact 4 and + refer to particular keys as well.

Small capital letters indicate use of a single cursor key to select menu items. A single such keystroke might recall the value of a variable, call up a subdirectory, perform an operation stored in the calculator, or run a program. For example, in the stroke sequence

51 ENTER SF

the SF means to press the key beneath the item SF in the TEST menu. In the stroke sequence

MEMORY HOME USER EDGE CH1 DIST X1

the four single keystrokes HOME, EDGE, CH1, and DIST select directories and the single keystroke X1 puts the value of the variable X1 on the stack.

# PART I THE CALCULATOR

# CHAPTER 1
# THE HP28S GRAPHING CALCULATOR AND PRECALCULUS

**Learning Objectives**

In this chapter you will:

- get an overview of the calculator features of the HP28S that are most useful in precalculus.
- be introduced to RPN (or stack logic).
- make a directory in your HP28S for precalculus.
- learn how to store variables.
- write and store a simple program.

## 1.1 Introduction

Often referred to as a graphing calculator, the HP28S is better described as a graphing-programmable calculator-computer. It has three general types of features that we will use:

(a) Calculator Features.

(b) Programming Features.

(c) Graphing Features.

In order to get the best use of the calculator in your precalculus course, you will need to

(a) become familiar with some of the calculator features.

(b) see a little programming.

(c) learn how to use most of the graphing features.

**The Calculator Edge** shows how to use *some* of the features of the HP28S to aid you in your study of precalculus. Patience and care are necessary, especially at the onset. Stay in touch with fellow students who are using the HP28S. And keep your HP28S **Owner's Manual** and **Reference Manual** handy. Work through the examples in **The Calculator Edge** carefully. When reviewing, repeat examples after a few days. You will be amazed how much easier some things will become. Work on getting used to the calculator *now*, so you can concentrate on precalculus later.

## 1.2 Getting Started

Begin your study of the graphing calculator by reading (with calculator in hand) Chapter 1, Chapter 2 and Appendix C in the HP28S **Owner's Manual**. This should take about an hour. After you have completed your reading, return to Section 1.2.

Most of the notation used in this manual will be introduced in this section. Remember to refer to the Notational Conventions at the front or to the Glossary in the appendix, if you need to. A word or phrase that is **bold and underlined** will be explained in the Glossary. Let us begin with an example on operational keys.

### EXAMPLE 1.1 OPERATION KEYS

Open the calculator. Turn it on. Press the red key and then the (letter) o key to get the ᵀᴱˢᵀ menu. Your screen should look like the figure. If not, press the red key and then the (numerical) 0 key to ᶜᴸᴱᴬᴿ.


Note: We will consistently use superscripted letters as in ᵀᴱˢᵀ and ᶜᴸᴱᴬᴿ above to indicate use of the (red) **shift key**.

Test out the + **operation key** by adding 2 to 2, using the **stack logic** **(RPN** logic) of the HP28S. We can write this as

2 ENTER 2 ENTER +

Note: We will refer to keys on the HP28S (like ENTER above) by capitalization.

You can think of + as taking the two 2's off the stack and operating on them, putting the sum of 4 back on the stack. The + key, like other operation keys, looks for its **arguments** on the **stack** and returns a result to the stack.

Clear the 4 off the stack by pressing DROP. Now try to add 2 and 2 as you would with an ordinary calculator. Your frustration at not seeing the sum 4 is only intensified by the unpleasant "beep" and message. As an example of another operation, we will silence this "beep".

Press ON and DROP until your screen again looks like the figure on the previous page. Next enter 51 on the stack and press the key directly above the red key. Let's write this as

51 ENTER SF

because you have run the "operation" or "program" SF.

Note: We will use small capital letters like the SF above to indicate **programs** and other **menu items** that are activated by pressing the keys in the first row below the screen (these keys are the **cursor keys** or **direction keys**).

You can think of SF as taking the 51 off the stack and operating on it. But where is the result? Not on the stack, but still somewhere. What you have done is *set* flag #51. With flag #51 set, the "beep" of an error message is silenced. With flag #51 *cleared*, the "beep" is heard. Slowly go through the stroke sequence

CLEAR 2 ENTER +

and think about what is happening. First you cleared the 4 off the stack, and then you put a 2 on the stack. Next + looked for two numbers on the stack to add, but found only one. The "beep" was silenced, although the **error message** "*Too Few Arguments*" still appears. For classroom use it is a good idea to keep flag #51 set. If you prefer to hear the sound use

51 ENTER CF

to clear flag #51. ▮

By the way, the fastest way to add 2 and 2 on a clear stack is

2 ENTER ENTER +

a four stroke sequence. Try it.

You have seen **Reverse Polish Notation (RPN)** as well as **algebraic notation** in the HP28S **Owner's Manual**. If you are encountering RPN or stack logic for the first time, you will need to get comfortable with it now. Stack logic is of immense value in operating the HP28S. It is only that your familiarity with algebraic notation makes it seem awkward at first. The following example is just a hint of the natural advantage of stack logic.

**EXAMPLE 1.2 STACK LOGIC**

Evaluate the fraction $\dfrac{\frac{1}{2} - \frac{1}{3}}{\frac{1}{4}}$ three ways:

(a) with an ordinary calculator,

(b) using algebraic notation on the HP28S, and

(c) using stack logic on the HP28S.

A RPN stroke sequence for (c) is

1 ENTER 2 ENTER / 1 ENTER 3 ENTER / - 1 ENTER 4 ENTER / /

Examine this stroke sequence step-by-step. Now using the multiplicative inverse key 1/x (the red key followed by the / key) try the stroke sequence

2 1/x 3 1/x - 4 1/x /

Aren't both of these easier to follow than

'((1/2)-(1/3))/(1/4)'

or even the stroke sequence

'(1/2-1/3)/(1/4)'

that takes advantage of hierarchy of operations? Practice entering the RPN stroke sequences above. Watch the stack logic at work.∎

## 1.3 Variables and Directories

Before going on, you should read Chapter 3 in the HP28S **Owner's Manual**. Look over the Menu Map in the appendix of the **Owner's Manual** and the Operations Index in the **Reference Manual**.

In this section we create directories and use them to store variables, programs, and subdirectories.

**<u>EXAMPLE 1.3 MAKING A DIRECTORY</u>**

Press the USER key to go to the USER directory. Your screen will appear as shown if you have not saved any variables.

Use the stroke sequence

MEMORY 'EDGE CRDIR

to create the directory EDGE in USER. Your USER menu should now appear as shown.


Now press the key underneath EDGE. At the bottom of your screen is space for all your menu items within EDGE. Of course there are none so far. Perform the stroke sequence

MEMORY 'CH1' CRDIR USER

which creates CH1 as a **<u>subdirectory</u>** (and first menu item) of EDGE. Use similar stroke sequences to add subdirectories CH2, CH3, CH4, and CH5 to correspond to chapters of **The Calculator Edge**.

Press CH1. We will store some things in the CH1 menu that correspond to items for this Chapter 1 of **The Calculator Edge** that you are reading. In particular we next store some variables.∎

**EXAMPLE 1.4 STORING VARIABLES**

Use CRDIR to create DIST as a subdirectory within CH1. Press DIST. You are now in the DIST menu (empty so far). If you are lost, use the stroke sequence

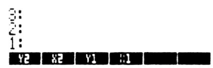
MEMORY HOME USER EDGE CH1 DIST

to orient yourself. You have created a PATH of subdirectories EDGE CH1 DIST in your HOME directory. In HOME EDGE CH1 DIST we are about to store variables associated with finding the distance between points in the Cartesian plane. Then we will use these stored variables to calculate the distance between two points.

Perform the stroke sequence

1 'X1 STO 6 CHS 'Y1 STO 4 'X2 STO 2 CHS 'Y2 STO

that stores the points (X1,Y1) = (1, -6) and (X2,Y2) = (4, -2) as the four variables X1, Y1, X2, and Y2 in DIST. Your screen should look like the figure.




Pressing X2 places its value of 4 on the stack. But pressing

'X2' ENTER

places its name 'X2' on the stack. Try it. Now enter the distance formula with the stroke sequence

'X1 ENTER 'X2 ENTER - 2 ^ 'Y1 ENTER 'Y2 ENTER - 2 ^ + √

Now 'D STO will store the distance formula. Do this and then press D EVAL. The distance between the stored points will be calculated and put on the stack. Next store another set of values as X1, Y1, X2, and Y2. Press D EVAL to get the distance between the "new" pair of points (X1, Y1) and (X2, Y2). Use the same procedure to find the distance between other pairs of points.∎

The above example is not the only (or even the best) way to calculate the distance between two points with the HP28S, but it does provide a good lesson in the use of some of the calculator features. The next example is an interactive program that finds the distance between two points. In Chapter 11 you will see yet another (and better) program for the distance formula.

**EXAMPLE 1.5 A PROGRAM TO FIND DISTANCE**

Enter the program D1 to find the distance between two points. It is given below along with a description of what it does. Type in the program exactly as shown. Then store it as D1 by the stroke sequence

ENTER 'D1 STO

D1 asks for two points, and halts. On continuing, the points are stored, the prompt is dropped and the distance is calculated.

```
D1
« "X1,Y1,X2,Y2=?"
HALT → X1 Y1 X2 Y2
  « DROP X2 X1 - 2 ^
Y2 Y1 - 2 ^ + √
  »
»
```

To run D1, press the key underneath its name, respond to the prompt by entering four numbers and then press CONT to continue.

Note: SST is a *very useful command* for analyzing programs. When a program is HALTed, you may use CONT to continue the program or SST to execute just the next step (and SST repeatedly to step through the rest of the program, one step at a time). Repeated use of SST while running a program is an extremely good analysis tool.

Run D1 one step at a time after the HALT by using SST in the CONTRL menu repeatedly. Try D1 for some other pairs of points.█

## 1.4 Summary

In this chapter you have created a directory structure of menus, programs, functions, and variables. To get a summary of what you have done, consider the stroke sequence

MEMORY HOME USER EDGE CH1 DIST

which first gets you into the HOME directory containing the **subdirectory** EDGE. EDGE in turn contains the **menu items** CH5, CH4, CH3, CH2, AND CH1, themselves subdirectories of EDGE. Within CH1 the above stroke sequence sends you into its subdirectory DIST, which contains the **program** D1, the **function** D and the **variables** X1, X2, Y1, and Y2.

You have seen almost all of the programming and calculator features of the HP28S that you will use in precalculus. The **stack logic** of the HP28S has been introduced as well. In the next chapter we turn to the important graphing capabilities of the HP28S.

# CHAPTER 2: FUNCTIONS AND GRAPHING

**Learning Objectives**

In this chapter you will:

- practice storing and evaluating functions.
- get an overview of the graphing features of the HP28S that are most useful in precalculus.
- make a lot of graphs.
- learn about "pixel accuracy".

**2.1 Introduction**

Before reading this chapter you should study your textbook's introduction to functions. It will be helpful also to study Chapters 3, 4, and 5 from the HP28S **Owner's Manual** before continuing. Note that the HP28S **Owner's Manual** does not use the word *function* precisely. What your textbook calls a function is often best thought of as an *expression* with one *variable* when reading the HP28S **Owner's Manual**. We begin with an example on entering and evaluating functions.

**2.2 Evaluating Functions**

With the HP28S you can quickly evaluate functions at many different values. Consider, for example, the function

$$f(X) = 2X/(1 + X^2) = 2*X/(1 + X\hat{}2\ )$$

which can be entered on the HP28S by the 12 stroke sequence

2,X*1,X,2^+/

and appears as '2*X/(1 + X^2 )' on the stack. Practice entering this on the stack a few times. Do you see the stack logic at work?

Store this expression under F using the three stroke sequence

'F STO

and then set the variable X equal to 3 by the four stroke sequence

3,X STO

Now the stroke sequence

F EVAL

will put .6 ( that is f(3) ) on the stack. To evaluate f(5), for example, you only need to enter

5,'X STO F EVAL

You could also use

5,'X STO F EVAL

but not

5,X STO F EVAL

since 3 has been stored under X. Try each of these three stroke sequences a few times to see what each is doing.

Naturally, more complicated functions can be evaluated just as quickly as our f(X). Furthermore, when the need arises, short programs can be written that evaluate functions at many different arguments, either automatically or by prompting. In fact the HP28S itself evaluates a function at 137 different arguments in order to produce the function's graph, as we see in the next section.

### 2.3 Graphing on the HP28S

Before working through this section you should read Chapter 7 from the HP28S **Owner's Manual.** Section 2.3 serves to demonstrate the versatility and limits of the graphing capability of the HP28S. A more careful step-by-step analysis of the graphs of precalculus begins in the next chapter, when we turn our focus away from the calculator and toward the subject matter of precalculus itself.

Start by plotting sin(x) as described in Chapter 7 of the HP28S **Owner's Manual**. With our notational conventions we can describe this as

'X ENTER TRIG SIN PLOT STEQ DRAW

or

'x ENTER TRIG SIN PLOT STEQ DRAW

Note how the two stroke sequences differ. In the first X is placed on the **command line** with the X key, in the second X is placed on the command line with a **cursor key**.

Now press

ON PPAR

to display the **default plot parameters**. You will find PPAR on the second **page** of the PLOT menu. Use the NEXT key to get there. Experiment with NEXT and PREV to determine what they do. The coordinate pairs

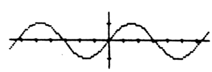
( -6.8, -1.5) and ( 6.8, 1.6)

give the position of the lower left and upper right corners of the screen displaying the graph of sin(x). The pairs

( -6.8, -1.5) and ( 6.8, 1.6)

are also stored as PMIN and PMAX. To produce the graph of sin(x) shown below DRAW determines 137 equally spaced x values between -6.8 and 6.8. The 137 corresponding functional values are calculated so that the 137 coordinate points can be darkened. More precisely, 137 of the **pixels** on the 137 by 32 **pixel grid** determined by the plot parameters are darkened.

The lack of perfect smoothness in the graph displayed by the HP28S is due to this (necessary) **rounding to pixel accuracy**.



Press ON (actually ATTN if a graph is displayed) to destroy the graph and return to the stack.

Note: Remember your ON key. After viewing a graph, use it to return to a view of the stack.

In the following examples we will see how to anticipate and overcome the problems caused by the necessities to **size the screen** and **round to pixel accuracy**.

**EXAMPLE 2.1 PIXEL ACCURACY**

Consider the function of Section 2.2

f(x) = 2*x/(1 + x^2 )

which can be stored as EQ for plotting by the stroke sequence

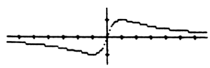
2,X*1,X,2^+/ STEQ

Try it now. IF the stroke sequence above does not work for you as before, it may be because X has been assigned a value. Use the stroke sequence

'X PURGE

to remove this stored value.

DRAW should produce the graph shown. Notice the lack of smoothness due to rounding to pixel accuracy.█



Not all your graphs are going to fit so nicely on the viewing screen. We address this problem with an example.

**EXAMPLE 2.2 SCREEN TOO SMALL**

The function in Chapter 7 of the the HP28S **Owner's Manual**

f(X) = X^3-X^2-X+3

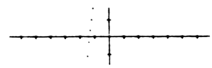
has a graph that doesn't fit the screen with the default plot parameters

( -6.8, -1.5) and ( 6.8, 1.6)

Store it as EQ for plotting by the stroke sequence

X,3^X,2^-X-3+ STEQ

DRAW should produce the graph you see to the right. This time the problem is that most of the 137 coordinates that make up the graph lie off the screen. ▮



By all means follow the **Owner's Manual** treatment of this problem, but try the following procedure for sizing the screen as well.

**EXAMPLE 2.3 SIZING THE SCREEN**

Use RCEQ to put

f(x) = X^3-X^2-X+3

on the stack. Perform the stroke sequence

7,'X STO EVAL

to evaluate f(7) = 290. Now evaluate f(-7) = -382 by means of the stroke sequence

7 CHS 'X STO RCEQ EVAL

This indicates that for X between -6.8 and 6.8, we should expect a much wider range of y-values than the -1.5 to 1.6 in PPAR. The stroke sequence

'PPAR VISIT

will display the plot parameters in the **editing mode**. Use the direction keys to change the coordinate pairs

( -6.8, -1.5) and ( 6.8, 1.6)

to

( -6.8, -300) and ( 6.8, 300)

and press the stroke sequence

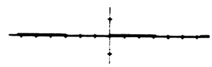
ENTER DRAW

to produce the graph shown.▮

Occasionally PPAR is such that we do not obtain a satisfactory graph due to the problem of pixel accuracy, as in the following example.

**EXAMPLE 2.4 PIXEL ACCURACY**

Retaining the plot parameters of the previous example, graph sin(x). If the result seems puzzling, think about the plot parameters. DRAW has done the best it can *up to pixel accuracy*.



Here the problem is that a single pixel represents a very large y range. The screen represents a vertical range of 600 (from y = -300 to y = 300), but sin(x) ranges only from -1 to +1. Actually, the default plot parameters are ideal for plotting sin(x). In general when you get graphs like the one above, you need to adjust the range of y values downward.■

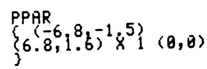
Often it is important to consider the domain of the function, as in the following.

**EXAMPLE 2.5 DOMAIN**

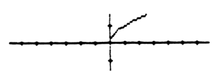
Use the stroke sequence

' PPAR PURGE

to return to the default plot parameters (as shown) and store the square root of X as EQ using the stroke sequence




X √X STEQ



Then graph EQ by pressing DRAW.

There are no pixels darkened for x < 0. Why? Not because of an inappropriate screen size, but because the function $\sqrt{x}$ is not defined for x < 0. Here a little reflection on the domain of the function is needed. ■

Note: If the function stored in EQ is not defined for any of the 137 x values determined by PPAR, DRAW will not plot points for these x values (but will still plot points for the remainder of the 137 x values).

We have been using the word *graph* to describe what the calculator produces with DRAW. More precisely, DRAW gives a *pixel approximation of part of a graph*. When you view such an *approximation of part of a graph* on your calculator screen, you will have to decide if you have a **complete graph**, a graph that shows all the significant features of a function. It will often be necessary to rescale PPAR and apply DRAW a second time to obtain a complete graph. Consequently it is helpful to know a little about how DRAW works.

DRAW will start by considering only the 137 equally-spaced x values determined by your plot parameters, and *try* to calculate 137 points on the graph of the function. If the function is undefined at some of these 137 x values, DRAW will ignore them (consider Example 2.5 for negative x). Even if DRAW can calculate all 137 coordinates, you may not see all (or even any) of them on the graph produced for two reasons:

1. The pixel grid does not extend far enough up or down (your y range is too limited). Look back at Example 2.2.

2. Parts of the graph lie on the x axis or lie so near the x axis that *up to pixel accuracy* they can be said to lie on the x axis (your y range is too wide). See Example 2.4.

Try graphing some of the functions from your textbook (even straight lines, if that's all you've encountered so far). Unfortunately most of the graphs that you will make under the default plot parameters of

( -6.8, -1.5) and ( 6.8, 1.6)

will miss some important aspect of the function. You will usually have to adjust the **plot parameters**. One good way to do this is to evaluate the function at a point or two if you find a graph unsuitable (as we did in Example 2.3), and use the information to edit the PPAR (or to edit PMIN and PMAX directly).

But editing PPAR is by no means the only way to adjust the plot parameters. You can rescale just one of the axes by using *W or *H, re-center the viewing screen without changing the scale with CENTR, and adjust the screen size quickly by using INS, PMAX, and PMIN. You can see how to use these and other commands in Chapter 7 of the **Owner's Manual** and in PLOT of your calculator's **Reference Manual**. Some of these commands will be used later in **The Calculator Edge**. As you draw more and more graphs using the features of the HP28S, you will develop your own favorite commands. Long after your precalculus course is completed, you can still be discovering things that your calculator can do.

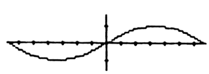
By combining the graphing capabilities of the HP28S with statistical commands, you can automatically adjust screen size. With the help of the string menu you can save (rather than redraw) graphs. Furthermore automatic or interactive programs can be written that plot points one at a time, plot several graphs on the same screen, plot polar graphs, or draw designs. To explore some of these possibilities now or later, you should see the HP28S **Owner's Manual** and **Reference Manual** or some of the books that Hewlett-Packard publishes for the HP28S.

The essential advantage in using DRAW is that you get 137 darkened pixels representing a graph quickly. Whatever the domain of the function stored as EQ, DRAW will represent a graph whose domain consists of the 137 equally spaced x values determined by PPAR. In fact, if EQ is not defined at some of these 137 x values, or if a pixel representing part of the x axis is redundantly darkened or if the y range is too narrow, you get even fewer pixels darkened. These next examples exhibit the danger of limiting yourself to a domain of 137 points or less.

**EXAMPLE 2.6 PIXEL PROBLEMS**

Let us look at the function sin(x) for x between $-137\pi$ and $+137\pi$. Sin(x) crosses the x axis 275 times on this interval, so a graph consisting of 137 pixels is bound to be insufficient. Store sin(X) as EQ and run DRAW with the PPAR shown.

```
PPAR
{ (-430.3982,-1.5)
(430.3982,1.6) X 1
(0,0) }
```



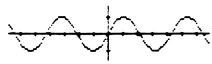
To do this you enter

'X ENTER SIN PLOT STEQ NEXT 'PPAR VISIT

using the direction keys, as well as INS and DEL, to VISIT the plot parameters and edit them. The graph produced by DRAW is deceptive.

Now edit PPAR to read as shown on the right. Then use DRAW again to see what the graph does between x = -10 and x = 10. In particular the graph crosses the x axis 7 times, as shown in the graph to the right. Compare this with the first graph.

```
PPAR
{ (-10,-1.5)
(10,1.6) X 1 (0,0) }
```



Pretty surprising, huh? Try using INS, PMIN, and PMAX to look at other parts of the graph up close.■

Fortunately polynomials can only 'wiggle' so much, so this kind of thing can't happen when graphing polynomials. With rational functions you only need to think about the asymptotes a bit to keep from going astray, as we see from this last example.

**EXAMPLE 2.7 ASYMPTOTES**

Store f(X) = X/(X -200) with the stroke sequence

X,X,200-/ STEQ 'PPAR PURGE

which sets the default plot parameters as well. Now DRAW appears to display no pixels at all. Actually to pixel accuracy, the graph lies on the x-axis. Use DRAW again with x going from 15 to 25 or -10 to 10. Use DRAW again and again until you think you know what the graph looks like.■

No matter how good you get at using DRAW, changing the plot parameters without thinking a little about the function is liable to take a lot of time, and may lead to a totally erroneous idea of the graph of f(X). The surest procedure for determining the graph with DRAW is the same as it is for determining the graph by hand. It consists of the following five points:

* consider any restrictions on the domain and range.
* find the intercepts.
* find the asymptotes.
* plot some points.
* connect the dots.

DRAW is helpful because it can plot some (actually 137) points quickly. In the examples we have seen occasions when DRAW actually deceived us in each of the other points.

DRAW is a fantastic aid in understanding functions through graphs. But you must never forget that the domain of the function stored in EQ and the domain of the graph displayed by DRAW are quite different and can lead to incorrect assumptions about the function stored in EQ. You can draw more graphs much faster with DRAW than by hand, but you must pay just as much attention to the domain of the function as ever.

**2.4 Summary**

In Chapter 2 you have seen almost all of the graphing features of the HP28S that you will use in precalculus. You have seen examples of rescaling by means of adjusting the plot parameters in PPAR. In other examples you have seen how to deal with the problem of "pixel accuracy" inherent in any machine grapher.

Do not go on to Part II until you feel you understand these first two chapters. If necessary, work through some of the examples again. If you are ready to go on, congratulations! You are well on the way to using a powerful tool to aid in your study of precalculus and beyond.

# PART II PRECALCULUS

# CHAPTER 3
# SYSTEMS OF LINEAR EQUATIONS

**Learning Objectives**

In this chapter you will:

- graph straight lines.
- "zoom in" on the point of intersection of two lines geometrically.
- use matrices to solve the linear system Ax = b.
- enter and use a program to test the system Ax = b for consistency.

**3.1 Introduction**

The focus of our attention now turns away from the calculator and to the subject of precalculus itself. We begin with straight lines (or just lines). If you are using Gechtman's **Precalculus**, recall that the symbol ***Gxyz** printed next to an example or explanation is to refer you to a related discussion on page **xyz** of that text. So ***G087** below refers you to page **87** of **Precalculus**.

For now, we will express lines in the slope-intercept form

$$y = mx + b$$

We begin by graphing some lines with the calculator.

## 3.2 Straight Lines

### EXAMPLE 3.1 A STRAIGHT LINE *G087

Use the stroke sequence

MEMORY HOME USER EDGE CH3

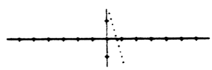
to get into your subdirectory for this chapter. Store the line

$$y = -3x + 2$$

as EQ by pressing

3 CHS ENTER X * 2 + PLOT STEQ DRAW

If your screen does not appear as shown use



'PPAR PURGE DRAW

to return to the default plot parameters, shown to the right, and then draw another graph.

```
PPAR
{ (-6.8,-1.5)
(6.8,1.6) X 1 (0,0)
}
```

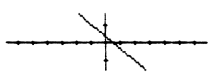
The default plot parameters have the advantage that each square pixel represents a .1 by .1 unit square in the x-y plane. The -3 slope of the graph of y = -3x + 2 is represented in true scale. To see more of the graph, the y intercept for example, it is necessary to **rescale**. We have already seen several ways to do this. This time use the stroke sequence

( 10,10 ENTER ENTER CHS PMIN PMAX

to change the plot parameters to those shown. Check PPAR to see what you have done. The stroke sequence above put (10,10) and (-10,-10) on the stack and then stored them in PPAR with PMIN and PMAX.

```
PPAR
{ (-10,-10) (10,10)
X 1 (0,0) }
```

Now press DRAW. You should produce the graph shown.

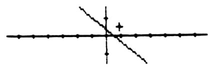


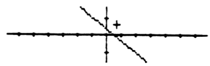
A lot of people find that the **viewing rectangle**

$$x \in [-10, 10], \quad y \in [-10, 10]$$

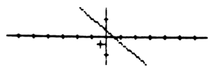
determined by the PPAR above is more convenient for their graphing. You may even want to store this viewing rectangle as PPAR1, say, if you see you are using it a lot.

Now with your last graph on the screen, press the four direction keys a few times. Observe how they move the cursor (now in the form of a small cross) across the graph. Move the cursor to the position shown to the right.





Press INS. Now move it just into the third quadrant, as shown. Press INS again.

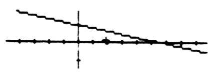


Now press ON to return to the stack. The pairs of numbers you see on the stack are the coordinates of the cursor when you pressed INS (probably not quite the same as those shown).


Now apply the stroke sequence

PMIN PMAX DRAW

to store the new plot parameters and draw the graph again. Your graph should be similar to the one to the right.



Notice that the slope seems to have changed along with PPAR. Of course, this can't really be the case. By changing PPAR, you have changed the relative scales of the x and y axes.

One advantage of using the default plot parameters is that slope is properly represented. This is because the 137:32 ratio of the number of square pixels in the grid matches the 13.7 by 3.2 size of the viewing rectangle

[-6.8, 6.8] by [-1.5, 1.6]

A major disadvantage of the default plot parameters is the small y range of only 3.2 units, compared to the x range of 13.7 units. It turns out that this size screen is perfect for trigonometric graphs, but usually unsuitable for polynomials. This explains why many users like to start with the viewing rectangle

[-10, 10] by [-10, 10]

Our next example shows how to graph two lines simultaneously.

**EXAMPLE 3.2 TWO STRAIGHT LINES** ***G131**

Consider the linear system

$$x - 4y = -8$$
$$3x - 2y = 6$$

or, in slope intercept form

$$y = x/4 + 2$$
$$y = 3x/2 - 3$$

Enter the functions on the stack and set them equal by

X,4/2+3,X*2/3-=ENTER

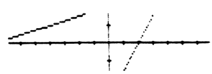
or

'X/4+2=3*X/2-3 ENTER

Now apply the stroke sequence

STEQ DRAW

to obtain the graph shown.
If your graph looks different from the figure, you are not using the default plot parameters.
Recall that the stroke sequence



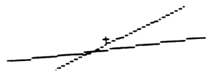
'PPAR PURGE

will get you back to the default plot parameters.

Next use

(1,1) ENTER CHS (8,8) PMAX PMIN DRAW

to set the viewing rectangle as [-1, 8] by [-1, 8] and get a better look at the graph. Now you can see where the lines cross.



Note: You may be wondering why you get graphs of both lines when DRAW operates on 'X/4+2=3*X/2-3'. After all, 'X/4+2=3*X/2-3' is actually an equation with the one solution (4,3). The reason is that your calculator is programmed to interpret 'X/4+2=3*X/2-3' as two different equations y=X/4+2=3 and y=X/2-3, so DRAW yields their two graphs. This unique interpretation of "=" allows us to draw two graphs simultaneously with a minimum of effort. As your precalculus course progresses, you will appreciate this feature more and more.

If you don't still have a graph of the two intersecting lines on your screen, draw the graph above again. In the next example, we use the **cursor keys** to investigate the point of intersection more closely.

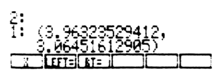
**EXAMPLE 3.3 CURSOR KEYS**

Use the **direction keys** or cursor keys just below the screen to move your **cross cursor** to one of the pixels where the lines overlap. Now INS the pixel coordinates as above.

Carefully perform the stroke sequence

ON SOLV SOLVR

Your screen should exhibit an ordered pair of points near (4,3), not necessarily exactly the pair shown.




Pressing the INS key sent the coordinates of the pixel location of your cross cursor to the stack. The stroke sequence

ON SOLV SOLVR

put you in the SOLV SOLVR menu. What SOLV SOLVR has done for you is to identify all the variables in your equation EQ (press SOLV RCEQ to see EQ). These variables consist of your independent variable X, the left hand side of your equation, and the right hand side of your equation. SOLV SOLVR is exhibiting each of these as a menu item.

Pressing the menu key for X (that is, pressing the key beneath the name X in the SOLV SOLVR menu) will set X equal to whatever the x coordinate in level 1 is. In this way, you can pass your *guess* of x = 3.89705882353 (or some other number close to 4) to SOLV SOLVR. Pressing X (that is pressing the red key and then the menu key for X) then runs a program that refines your initial guess for the x coordinate of a solution to EQ. The stroke sequence for all of this is

SOLV SOLVR X X

Perform this stroke sequence and you will see the value 4 appear on the stack. The "4" is the x value of the point of intersection of the graphs. To find the y value at the point of intersection, press LEFT= (to calculate X/4+2, the left side of EQ, when x=4) or RT= (to calculate 3*X/2-3, the right side of EQ, when x=4). In either case, you should get the value 3 for y. Thus the point of intersection of the two lines is (4,3). For more on how to use SOLV SOLVR, see the HP28S **Owner's Manual** and **Reference Manual**. We continue with this example on the cursor keys by looking at some more graphs.

Apply the stroke sequence

PLOT DRAW

to get your last graph on the screen again. As in Example 3.1, use INS and PMAX and PMIN to zoom in on the point of intersection. Now zoom in again. And again and again. Keep checking the coordinates that INS gives you. It should appear more and more likely that the intersection is the point (4,3). Roughly speaking, the procedure in SOLV SOLVR used by your calculator to determine that the lines cross when x is 4 is a quicker and more sophisticated version of the zooming in that you just did. Because the mathematics underlying SOLV SOLVR is beyond the scope of precalculus, and because the procedure can give you misleading results, you should verify anything SOLV SOLVR tells you by other methods. Still you will find SOLV SOLVR to be one of the most helpful parts of the calculator.

Graph your homework exercises in 2x2 Linear Systems(***G138**). Use INS, PPAR and other commands discussed above. If you like, try out some of the many other HP28S commands described in the HP28S **Owner's Manual** and **Reference Manual**. Some of these will appear later in **The Calculator Edge**, others will not.

Of course, a pair of equations may have graphs that cross at exactly one point or graphs that don't cross at all (parallel lines). A third possibility is that the pair of equations have the same graph. Next we analyze and run a program CST that, for a given pair of equations, determines in which of these three ways the graphs are related.

***G133** The program to the right, CST, performs a consistency test on a pair of equations

$$A_1X + B_1Y = C_1$$
$$A_2X + B_2Y = C_2$$

CST asks for the six coefficients and uses them to determine whether the system has exactly one solution, no solutions, or infinitely many solutions.

```
CST
« "AX + BY=C"
"A1,B1,C1,A2,B2,C2="
HALT → A1 B1 C1 A2
B2 C2
  « DROP2
    IF A1 A2 / B1 B2
/ ≠
    THEN
" One Solution "
    ELSE
      IF A1 A2 / C1
C2 / ≠
      THEN "No Sol."
      ELSE
"Infinite # of Solutions
"
      END
    END CLLCD 1 DISP
  »
»
```

Try CST out on the systems below.***G133**

a. $2x + y = 6$
   $4x + 2y = 9$

b. $2x + y = 6$
   $4x + 2y = 12$

c. $2x + y = 6$
   $4x + y = 12$

Use SST repeatedly after entering the coefficients in response to the prompt. SST allows you to see the consistency test as well as the calculator's stack logic in action.

Note: In stepping through the program notice that division by zero yields 9.99999999999E499. This large number (almost as large as $10^{500}$) is the HP28S's approximation of infinity. At some point almost all computers and calculators must approximate. In computing, the use of approximate values occasionally causes baffling errors. Even with the most sophisticated computing equipment, *analysis* plays a vital role. In fact, with more sophistication, analysis becomes even more important.

### 3.3 Matrix Algebra

Your HP28S is designed to do matrix algebra with the same keystrokes used to perform algebraic operations on numbers. We can see how this is done by looking at an example.

**EXAMPLE 3.4 MATRICES** ***G149**

Store the matrices

```
    [[ 2 -1 1]
A =  [ 1 2 -1]          and   B  = [ 5 -3 -3]
     [ 3 2 -2]]
```

with the stroke sequences

[[2,1CHS,1[1,2,1CHS[3,2,2CHS ENTER 'A STO

and

[5,3CHS,3CHS ENTER 'B STO

You calculator understands A to be the 3x3 coefficient matrix above and B to be the *3x1 column matrix (5 -3 -3)$^t$*. Now the simple stroke sequence

A B *

will yield the matrix product of A*B = [ 10 2 15 ] (meaning the 3x1 column matrix (10 2 15)$^t$). Similarly you can use + and - to add or subtract matrices or vectors just as you would add or subtract real numbers. But there is more. You can find the solution X to the matrix equation AX=B (algebraically $X=A^{-1}B$) by the stroke sequence

A 1/X B *

or even

B A /

Try both of these. Note that only the latter yields the exact solution [ 1 -1 2 ]. The former isn't as careful in rounding, but still comes close to [ 1 -1 2 ]. This seems too good to be true. And it is. Just try taking the inverse of the matrix

```
    [[ 2 -1 1]
M =  [ 3 2 -2]
     [ 3 2 -2]]
```

which you can get by editing A above. Although M has no inverse (why?) the calculator finds one if flag 59 is clear. For our purposes, it is better to keep flag 59 set. Do this with

TEST 59 SF

Again, *analysis* is important. You *can* depend on the procedure we used in Example 3.4 to solve Ax=b *if* there is a unique solution and *if* A and b are not *too unusual*. If you are interested, the HP28S **Owner's Manual** and **Reference Manual** contain a lot on the many operations that can be applied to matrices. Hewlett-Packard publishes a series of books for the HP28S that deal with various mathematical topics. One of these, **Vectors and Matrices**, deals exclusively with vectors and matrices.

In addition to the matrix capabilities of the HP28S, the programming capabilities can be exploited to fashion programs that

(a) allow symbolic matrix operations (see **HP28S Insights** by William Wickes),

(b) allow you to work in rational form (i.e. to use fractions like 2/3 instead of decimal approximations), or

(c) allow step-by-step row operations to be carried out.

Of course, these go well beyond the scope of your precalculus course, principally because of the programming involved. If you're curious, the secret to symbolic matrix operation is the use of a list like {{a,b},{c,d}} to represent the matrix [[a,b], [c,d]]. To work in rational form you can represent a/b as the complex number (a,b), and manipulate a and b with the complex arithmetic features of the calculator. Other ways to get rational results depend on mathematical methods to recover a fraction a/b from its decimal form (faster, but riskier than sticking with rational expressions all along).

### 3.4 Summary

In Chapter 3 we have concentrated on graphs of straight lines. We zoomed in on points of intersection and rescaled the axes by adjusting the plot parameters in PPAR.

You saw how to use SOLV SOLVR to find a point of intersection of two lines. Operations on matrices were employed to solve the linear system Ax = b. A program CST was given and explained that tests the system Ax = b for consistency.

You are now ready for the systematic treatment of polynomials in the next chapter.

# CHAPTER 4: POLYNOMIAL FUNCTIONS

**Learning Objectives**

In this chapter you will:

- graph polynomials.
- determine zeros of polynomials geometrically by "zooming in".
- determine zeros of polynomials algebraically with SOLV SOLVR.
- enter and use a program VRTEX that calculates the vertex of a parabola.
- do complex arithmetic and handle complex roots.
- enter and use a program SDIV that performs synthetic division.

## 4.1 Introduction

This chapter deals primarily with the graphing of polynomial functions. We will also be concerned with the algebra of polynomial functions and the evaluation of polynomial functions.

The graphing features of your calculator make it possible to create graphs of polynomials quickly and accurately. In Chapter 4 we start with graphs of quadratics and gradually consider graphs of more and more complicated polynomials. Together the many examples represent almost any kind of behavior that the polynomial functions of precalculus can exhibit. Work through them carefully.

## 4.2 Graphing Quadratic Functions

In this section we consider some important aspects of quadratic functions and their graphs with a series of examples.

**EXAMPLE 4.1 GRAPHING QUADRATICS** ***G098**

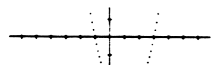
As a first example that involves adjusting the plot parameters, let's graph the quadratic function

$$y = x^2 - 2x - 3$$

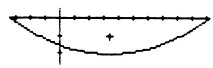
with the default plot parameters using the stroke sequence

X,2^2,X*-3- STEQ 'PPAR PURGE DRAW

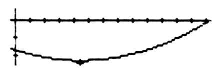
DRAW produces the graph shown.

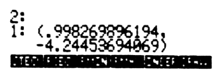


Change the PPAR until you get a graph (as in the second figure on the right) that shows the vertex and both x intercepts in the same viewing rectangle.



Use the cross cursor, INS, PMIN, and PMAX to create other views of the graph near the vertex. Try to get nearer and nearer the vertex (that is **zoom in**). Press INS with the cross cursor near the vertex (as in the third graph) to put the coordinates of a point near the vertex on the stack, as shown.






With enough persistence, you can reach the point (1, -4) itself. Even before reaching (1, -4), you probably will become convinced that (1, -4) is the vertex. The evidence is not leading you astray: the vertex really is (1, -4).∎

We can confirm that the vertex is (1, -4) with the program VRTEX given on the following page. VRTEX determines the vertex of a parabola, the most important point on the graph. Copy the program exactly as shown in the right column. Then save it as VRTEX in your CH4 subdirectory. The left column gives a step by step description of what VRTEX does.

***G099** VRTEX begins with a friendly display of the general quadratic function and asks for the parameters A, B, and C. Upon continuation of the program, the coordinates of the vertex (-B/2A, f(-B/2A)) are calculated and displayed.

```
VRTEX
«
"AX ^ 2 + B * X + C"
"A,B,C=" HALT → A B
C
  « DROP2 B NEG 2 A
* / 4 A C * * B B *
- 4 A * / R→C
  »
»
```

Try VRTEX out for the quadratic function above. Just respond to the prompt with

1,2CHS,3CHS CONT

and the vertex (1, -4) is displayed in level 1. Now use VRTEX to find vertices of some of the parabolas in your text.

If you like, alter VRTEX to suit your own tastes or needs. For example, try to come up with an IF statement that checks the sign of A and tells whether the parabola opens upward or downward. You may like to use the command DISP and string statements (see the program CST in Chapter 3) as well.

It is possible to zoom in on a *zero* of a function (an x value where the graph crosses the x axis) in the same way that we zoomed in on the vertex. But SOLV SOLVR does this for us algebraically (and much faster than we can zoom in ourselves), as the next example shows.

**EXAMPLE 4.2 ZEROS OF A QUADRATIC** ***G098**

You are going to find the zeros of

$$y = x^2 - 2x - 3$$

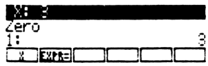
by using SOLV SOLVR. Begin by creating a graph of

$$y = x^2 - 2x - 3$$

that shows both x intercepts in the viewing rectangle. As in Example 2.2 move your cross cursor to one of the pixels along the negative x axis where the graph crosses the x axis. Use INS to put the approximate coordinates of this pixel on the stack. Now move your cross cursor to one of the pixels along the positive x axis where the graph crosses the x axis. Use INS to put the approximate coordinates of this pixel on the stack.
Now apply the stroke sequence

ATTN SOLV SOLVR X X

to go to the SOLV SOLVR menu, store the x coordinate of the point on level 1 as X, and have SOLV SOLVR "*algebraically zoom in*" on the zero. If you start SOLV SOLVR with a point near enough (3,0), it will produce the display shown to the right.




Interpret the word "*Zero*" displayed on level 2 to mean that, to the internal accuracy of the machine, the value of EQ is zero when x = 3.

For more on the meaning of this (and other) displayed messages, see the HP28S **Owner's Manual** and **Reference Manual.** For still more about SOLV SOLVR, see **HP28S Insights** by William Wickes. You should be able to use SOLV SOLVR to find zeros for quadratics or any other polynomials you will encounter in precalculus. However you may occasionally find a zero other than the one you anticipated. You may find it interesting to start SOLV SOLVR with various initial guesses for zeros and watch which zero it finds: x = -1 or x = 3.

In our next example we look at another way to find roots of a quadratic, the quadratic formula.

**EXAMPLE 4.3 THE QUADRATIC FORMULA** ***G099**

Practice what you have learned about graphing quadratics by producing a graph of the function

$$s = -2t^2 + 11t - 5$$

with your calculator and then doing each of the following:

(a) Alter PPAR in order to see the vertex and intercepts in the same viewing rectangle.

(b) Use the program VRTEX above to find the vertex.

(c) Find the zeros with SOLV SOLVR.

(d) Use the quadratic formula to find the zeros t = 1/2 and t = 5.

For repeated use of the quadratic formula, wouldn't it be easier just to write a program that asks for A, B, and C (like VRTEX on page 29) and then calculates the two roots? By all means, try this, if you are so inclined. However the HP28S already has QUAD (a program version of the quadratic formula) in the ALGEBRA menu. Try it out on the equation on page 30 by putting its arguments on the stack and then evaluating it with

RCEQ T ALGEBRA QUAD

The s1 in the result is like the ± in the quadratic formula

$$( -b \pm \sqrt{(b^2 - 4ac)})/2a$$

Let s1 be +1 to get one root and let s1 be -1 to get the other. Thus the stroke sequence

ENTER 1 ENTER 's1 STO EVAL SWAP 1 CHS 's1 STO EVAL

yields both zeros. Step through this stroke sequence carefully to see it at work.█

If you expect to use the quadratic formula a lot, no one can blame you for writing your own program. Another approach would be to use QUAD in a program that incorporates something like the stroke sequence above.

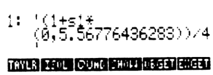
A double root causes no further hardship. You just get it twice; once with s1 = 1 and again with s1 = -1. We have only one more case to consider: a quadratic with complex roots.

**EXAMPLE 4.4 COMPLEX ROOTS** ***G037**

Find the zeros of

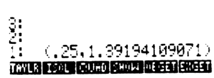
$$y = 2x^2 - x + 4$$

using QUAD. Remember that QUAD takes two arguments: the function and the independent variable. Your stack should appear as shown. Now set s1 equal to -1, say, with




1 CHS 's1 STO EVAL

to get the stack to the right.



The calculator is exhibiting the complex number

.25 - 1.39194109071 i

In general, the HP28S will exhibit the complex number a + bi as the ordered pair (a,b).█

Incidentally, with this convention, you may add, subtract, multiply, or divide complex numbers with the same keys that you use to add, subtract, multiply, or divide reals and matrices. If you have been doing complex arithmetic, you will find this example of complex division interesting:

**EXAMPLE 4.5 COMPLEX DIVISION**

To divide (1+i)/(1-i) use the stroke sequence

(1,1 ENTER (1,1CHS /

to get (0,1), which means 0 + 1i.█

### 4.3 Polynomial Algebra

It is possible to add, subtract or multiply two polynomials with the same +,-, and * keys that add, subtract or multiply two real numbers, two complex numbers, and two matrices (as we have seen). Unfortunately you must then simplify, expand, and otherwise manipulate polynomials to get a resulting expression in a usable form. The ALGEBRA menu has programs that do this, but they are time consuming to use.

In fact, you are much better off staying away from the ALGEBRA menu altogether. A useful trick is to think only of the coefficients of polynomials. For example, think of the polynomials

$11x^5 + 9x^3 - 11x^2 + 11x - 3$ and $x^2 + 10x + 3$

as the arrays

[ 11 0 9 -11 11 -3] and [ 0 0 0 1 10 3]

which you can enter by the stroke sequence

[11,0,9,11CHS,11,3CHS ENTER [0,0,0,1,10,3 ENTER

Now the + key can be used to add the vectors to yield

[ 11 0 9 -10 21 0]

which naturally represents the polynomial

$$11x^5 + 9x^3 - 10x^2 + 21x$$

Subtraction of polynomials can be similarly represented by subtraction of vectors, but multiplication is more complicated. Division is still more complicated. An excellent source for polynomial algebra is the book **Mathematical Applications** published by Hewlett-Packard. This book tells you all you need to know to do polynomial algebra quickly on the HP28S, giving and explaining several programs.

So that you don't feel cheated, below is a program SDIV to do synthetic division in the special important case of division of a general polynomial by a linear term of form x - z. First create a subdirectory PALG in the directory CH4. Then store SDIV, given below, in

HOME EDGE CH4 PALG

SDIV asks for the number Z (to represent the divisor x-Z) and the vector P of coefficients of the dividend polynomial. A vector Q is formed, initially set equal to P. Q is updated to represent the last row in the synthetic division tableau, and is finally used to represent the quotient polynomial. It is stored under Q. The remainder is stored under R.

```
SDIV
« "Z=?,P=[?...?]"
HALT DUP 'P' STO 'Q'
STO 'Z' STO DROP 2 Q
SIZE LIST→ DROP
  FOR I Q I 1 - GET
Z * P I GET + 'Q(I)'
STO
  NEXT Q ARRY→ SWAP
'R' STO LIST→ SWAP 1
- SWAP →LIST →ARRY
'Q' STO
»
```

Try out SDIV on the following example. If you SST through this program, you can see both the synthetic division algorithm and stack logic at work.

Note: The use of SST in the CONTR menu to "single step" through a program is very helpful in learning about programming on the HP28S. Even if you are not interested in programming, SST can help you to understand the procedure that has been programmed, in the present case the synthetic division algorithm.

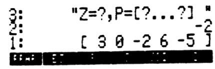
**EXAMPLE 4.6 SYNTHETIC DIVISION**

Use synthetic division to divide

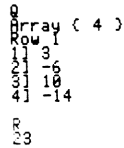
$$3x^4 - 2x^2 + 6x - 5$$

by

x + 2




Respond to SDIV's prompt as shown. SDIV will store the quotient polynomial (represented as a vector of coefficients) under Q and the remainder under R. Press Q and R to see the results.




You may want to change SDIV so that it automatically displays Q and R, perhaps even using a string to add comments. Once you understand SDIV you may like to modify it to handle more general divisors.

We have seen graphs of quadratics in Section 4.2. Now we look at graphs of cubic and higher degree polynomials.

### 4.4 Graphing Polynomials

This section does not *replace* (but rather *compliments*) your textbook's discussion of techniques for graphing polynomials (Section 4.4 in Gechtman's **Precalculus**). The more information you can gather about a function, the more likely you are to construct a good graph. The graphing calculator allows you to gather information quickly because it plots points (or more precisely, pixels) fast. You still may have to adjust the viewing rectangle several times before you get a **complete graph**. Good analysis is indispensable. An inappropriate viewing rectangle can easily lead you astray. For example, in a small enough viewing rectangle, the graph of *any* polynomial looks like a straight line . Graphs of most higher order polynomials look a lot like graphs of quadratics in some viewing rectangles. The best approach to graphing polynomials in general is to zoom out until you are satisfied that you have seen all the "wiggles" and then zoom in again as needed. Proper analysis of intercepts, asymptotic behavior, and rising and falling behavior should accompany this graphing process.

In the following three examples, we will always begin with the default viewing rectangle

[-6.8, 6.8] by [-1.5, 1.6]

This viewing rectangle is almost always too small for graphing the polynomials of precalculus. You may find it better to start with the viewing rectangle

[-10, 10] by [-10, 10]

suggested in Chapter 2.

Note: Recall that when we say that the viewing rectangle is

[-6.8, 6.8] by [-1.5, 1.6]

we mean that the screen will show a graph for x between -6.8 and +6.8 and y between -1.5 and +1.6. This is the same screen size described by the default plot parameters

( -6.8, -1.5) and ( 6.8, 1.6)

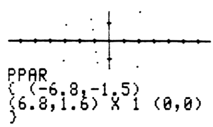
meaning that the lower left corner PMIN has coordinates ( -6.8, -1.5) and that the upper right corner PMAX has coordinates ( 6.8, 1.6).

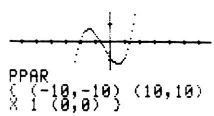
**EXAMPLE 4.7 A CUBIC** ***G190**

To the right the graph of the function

$f(x) = x^3 + 2x^2 - 5x - 6$

is displayed for two pairs of plot parameters (also shown). With the default plot parameters, we see *just* enough of the graph to understand how it behaves (if we note that a cubic can change directions at most twice). The second graph with the viewing rectangle [-10, 10] by [-10, 10] shows the complete graph.




The second graph does not in itself assure us that it displays the complete graph. We *know* that this is the complete graph only because we know something about the graph of a cubic.

Find the roots of the function

$$f(x) = x^3 + 2x^2 - 5x - 6$$

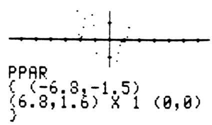
two different ways: by using SOLV SOLVR and by factoring (try factoring with the synthetic division algorithm SDIV). Remember that SOLV SOLVR is a numerical technique that, of necessity, approximates. You should always be a little uneasy with a SOLV SOLVR result.█

Often your first graph of a polynomial of degree four or more will consist of few or no darkened pixels. It is usually then best to **zoom out** once or twice to see the complete graph, and finally **zoom in** again. Two examples typical of this process follow. The first involves a polynomial of sixth degree.

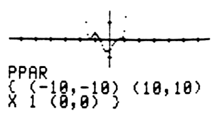
**EXAMPLE 4.8 A SIXTH DEGREE POLYNOMIAL** ***G191**

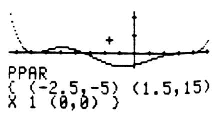
To the right the graph of the function

$f(x) = (x + 1)(x + 2)^2(x - 1)^3$




is displayed for several different plot parameters, starting with the default plot parameters. This time the mess we see with the default plot parameters is of little use, except that it inspires us to try a bigger viewing rectangle. Our standby viewing rectangle [-10, 10] by [-10, 10] reveals *just* enough (to go along with the fact that the function has the three roots -1, -2, and 1) to yield the complete graph. By looking at the second graph, a third set of plot parameters was determined and a third graph drawn. This is about the best representation the HP28S can give for




$f(x) = (x + 1)(x + 2)^2(x - 1)^3$ █

In the next example we see that even an eleventh degree polynomial like

$$f(x) = -x^2(x - 3)^5(x - 1)^4$$

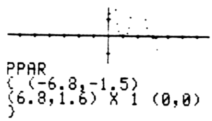
can be analyzed. Again, it is necessary to consider several plot parameters before obtaining a satisfactory graph.

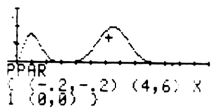
**EXAMPLE 4.9 AN ELEVENTH DEGREE POLYNOMIAL *G191**

As a final exercise in graphing higher order polynomials, consider the function

$f(x) = -x^2(x - 3)^5(x - 1)^4$

Graphs are displayed for two different plot parameters. The default plot parameters disappoint us as usual. Note that $f(x) \geq 0$ for $x \leq 3$ and $f(x) \leq 0$ for $x \geq 3$ so the viewing rectangle [-10, 10] by [-10, 10] is not appropriate either. Further analysis suggests the plot parameters used to make the second graph. Again we arrive at a complete graph.∎




For more complicated polynomials, complete graphs may just be impossible to get on the HP28S. However, none of the standard precalculus polynomials should require any more effort than we expended on

$$f(x) = -x^2(x - 3)^5(x - 1)^4$$

In the last three examples you have seen the basics for graphing polynomials with your calculator. Now you are ready to use the calculator to analyze the polynomial graphs of precalculus. Enhance your "natural feel" by working through Examples 4.7, 4.8, and 4.9 again. Recreate graphs pictured in your text. You will be pleased to see how fast your ability to graph improves.

When your text takes up the subject of *zeros* (or *roots* or *x intercepts*) of polynomials, you should study Section 4.5. In this section we see how the HP28S can be used to determine x intercepts of graphs.

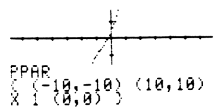
### 4.5 Zeros and Approximations to Zeros

If a polynomial function can be factored, the zeros can be determined. If not, some type of **numerical iteration** is often used to approximate (if not actually find) the zeros. One such method is explained in Gechtman's **Precalculus**, Section 4.5. The numerical iteration procedure in SOLV SOLVR of your HP28S is designed to accept a guess for a zero, and to improve on this guess. The method used is not foolproof, but should almost always find a zero. Using SOLV SOLVR and DRAW together, you can find all the zeros of any function you encounter in precalculus. We see how this is done by considering a pair of examples. In each example we are seeking the single real zero of a cubic.

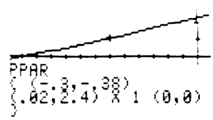
**EXAMPLE 4.10 ZERO OF A CUBIC** ***G199**

We are going to use both DRAW and SOLV SOLVR to determine the single real root of the cubic

$f(x) = x^3 + 3x^2 + 8x + 2$

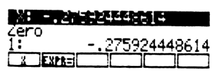



Look at different viewing rectangles to convince yourself that there is a single real root between x = -1 and x = 0. Use SOLV SOLVR to find the zero x = -.275924448614. Now evaluate f(x) at x = -.275924448614 with the stroke sequence




'X STO RCEQ EVAL

or, simply




EXPR=

In either case you get 0.∎

Note: The stroke sequence 'X STO RCEQ EVAL above does the following: It takes the value -.275924448614 off the stack and stores it under X. Then EQ (the equation $x^3 + 2x^2 - 5x - 6$) is recalled and evaluated as 0. On the other hand the two stroke sequence EXPR= evaluates $x^3 + 2x^2 - 5x - 6$ immediately.

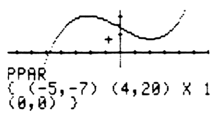
There is a variety of iterative techniques for approximating zeros, but the technique programmed into your HP28S should serve you well in precalculus and far beyond. We end this section and chapter with an example in which SOLV SOLVR appears to let us down.

**EXAMPLE 4.11 ZERO OF A CUBIC**

Start looking for a zero of the cubic

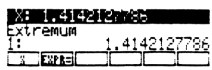
$f(x) = x^3 - 6x + 12$

by creating a graph. As before, look at different viewing rectangles until you have a complete graph. Use the cross cursor and SOLV SOLVR to find the single zero near x = -3.13. What if you had not had a graph to look at? Could you have still made an initial guess for the zero, and have SOLV SOLVR find the root for you? Some initial guesses further away from -3.13 would merely take SOLV SOLVR more time. But try an initial guess of x = 2 with the stroke sequence





2x^X




SOLV SOLVR takes longer than usual and yields the display shown.

What happened? The action of SOLV SOLVR can best be thought of as an attempt to reach an x value for which f(x) changes signs (from positive to negative or visa versa). Somehow, it managed to home in on $x = \sqrt{2} \approx 1.41$, a place at the bottom of a "*valley*" on the graph.

You could make the same mistake hiking in fog-covered mountains. Trying to get down to the tree line, you head downhill, only to get hung up in a valley above the tree line. The geometric explanation for SOLV SOLVR'S problem is the same as the geometric explanation for your hiking problem. In graphing, as in hiking, there are ways around the problem.▮

Remember that your HP28S has several other graphing features. The commands *W and *H, allow you to rescale the axes one at a time. The command CENTR allows you to recenter the viewing screen without changing the scale. See Chapter 7 of the **Owner's Manual** and PLOT of the **Reference Manual** to see how to use these commands.

### 4.6 Summary

In Chapter 4 you have seen how to determine zeros of polynomials by graphing as well as by an algebraic procedure in SOLV SOLVR. You have entered and used two programs: SDIV that performs synthetic division and VRTEX that calculates the vertex of a parabola. You saw how to use QUAD, the HP28S version of the quadratic formula. But most importantly, you created complete graphs of a series of polynomial functions using the basic graphing features of your calculator.

# CHAPTER 5: RATIONAL FUNCTIONS

**Learning Objectives**

In this chapter you will:

- graph rational functions.
- see how to use your calculator to help find the vertical and horizontal asymptotes of rational functions.
- analyze a rational function whose graph has a "hole" rather than a vertical asymptote.

## 5.1 Introduction

Rational functions are functions of the form R(x)= P(x)/Q(x), where P(x) and Q(x) are polynomials. Unless stated otherwise, assume that P(x) and Q(x) have no factors in common. If you can graph polynomials, there are just three more geometric ideas you must consider in order to successfully graph rational functions:

* the location of any discontinuities.
* the vertical asymptotes.
* the horizontal and oblique asymptotes.

In this chapter's examples you will see how to use the algebra behind each of these geometric ideas to help you determine complete graphs of rational functions.

## 5.2 Graphs With Asymptotes

We begin by looking at a rational function whose graph has both vertical and horizontal asymptotes.

**EXAMPLE 5.1 ASYMPTOTES**

Store the function

f(x) = 1/(x-1)

in your menu for Chapter 5 with the stroke sequence

X,1-$^{1/X}$ STEQ

DRAW yields the graph shown under the default plot parameters.

From the graph it appears that x=1 might be a vertical asymptote. You can gather algebraic evidence that this is indeed the case by evaluation of f(x) for values of x "near" x=1. For example, use the stroke sequence

1.00001 ENTER 'X STO RCEQ EVAL

to store 1.00001 as x and evaluate 1/(1-x) there to get 100,000. Now evaluate f(x) for numbers even closer to x=1 than 1.00001 is. Be sure to pick some numbers larger than 1 and others smaller than 1. Convince yourself that |f(x)| tends to ∞ as |x-1| tends to 0. This is what makes x=1 a vertical asymptote. Before you graph a rational function you should already know where the vertical asymptotes are. Recall that x=a is a vertical asymptotes of R(x)= P(x)/Q(x) if Q(a)=0 and P(a) is not 0.

You can investigate the apparent horizontal asymptote y=0 by evaluating f(x) for very large or very small values of x. Do this until you are convinced that y=0 is indeed a horizontal asymptote.▮

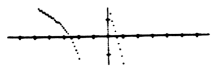
Note: The graphing and evaluation techniques described above can help you find vertical and horizontal asymptotes. However these techniques enhance, rather than replace, the *algebraic* techniques found in your precalculus text (Section 5.1 in Gechtman's **Precalculus**).

It is best to determine the vertical and horizontal asymptotes before you touch your graphing calculator. In particular the next example shows you how knowing the vertical and horizontal asymptotes can help you determine a suitable viewing rectangle quickly.

**EXAMPLE 5.2 ASYMPTOTES AND SIZING THE SCREEN I** *G212

If you graph the rational function

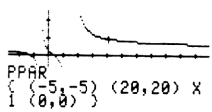
$(4x^2+8x-6)/(x^2-2x-3)$



under the default plot parameters your screen will appear as shown in the figure.

Such a graph should alert you to zoom out or, better, to *analyze* the vertical and horizontal asymptotes and change the plot parameters to get a suitable screen size. Zooming out without thinking about the asymptotes is quite tricky. Try it and see. For this function it is not too difficult to algebraically determine that the horizontal asymptote is y=4 and the two vertical asymptotes are x=-1 and x=3.

Evaluation yields f(5)≈11 and f(10)≈6, so plot parameters of (-5,-5) and (20,20) might be good to try. They yield the second graph illustrated. To be comfortable that this is a complete graph of the rational function, you really need to *know* the single horizontal and two vertical asymptotes.█




Note: When you graph rational functions, avoid the temptation to zoom out without some analysis. As in the previous example, a little bit of analysis can suggest a reasonable screen size.

As another exercise in the technique of sizing the screen consider the following example.

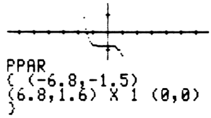
**EXAMPLE 5.3 ASYMPTOTES AND SIZING THE SCREEN II**

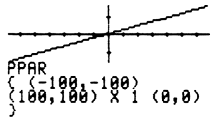
In this example you will see how useful it is to know the vertical asymptotes before you size the screen.

Begin by making the graph of the rational function

$(x^3+3)/(x^2-4)$

with the default plot parameters. Make a second graph with the plot parameters (-100,-100) and (100,100). Your graphs should be like those in the figures to the right.




The second graph looks like a straight line. How can this be? With the plot parameters

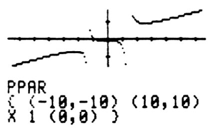
(-100,-100) and (100,100)

your calculator is showing you an important feature of the graph: the function's oblique asymptote y=x. But at this scale the vertical asymptotes are lost (either between pixels or beyond the screen). You can zoom in from the plot parameters

(-100,-100) and (100,100)

and find these asymptotes by trial and error, but it is much safer to analyze the function first.

Q(x) can be readily factored to reveal the vertical asymptotes x=2 and x=-2. Knowing that these are the two vertical asymptotes you can look at the graph shown and feel confident that it is indeed a complete graph.∎




Note: It was pointed out in Chapter 4 that if you just *zoom in* close enough, any polynomial graph looks like a straight line. Now by *zooming out* on the graph of a rational function we get the straight line look again. This is always the case for a rational function of form R(x)= P(x)/Q(x) if the degree of P(x) is 1 more than the degree of Q(x).

In a sense you can say the following: *far enough away* this type of rational function looks like a straight line. *Close enough* all polynomial functions look like straight lines.

If the degree of P(x) is equal to the degree of Q(x), there is a horizontal asymptote. If the degree of P(x) is 1 greater than the degree of Q(x) there is an oblique asymptote. Your textbook(*G217) may show how to find this oblique asymptote by long division or synthetic division on R(x). Recall that you may be able to use the program SDIV from Chapter 3 to do the synthetic division necessary to find the oblique asymptote.

We have been assuming that P(x) and Q(x) have no polynomial factors in common. In the next section we see what to expect if P(x) and Q(x) do have a polynomial factor in common.

### 5.3 Graphs Without Asymptotes

If P(x) and Q(x) have a polynomial factor in common, we can simplify R(x)= P(x)/Q(x) by dividing out the common factor. However the resulting polynomial may not be the same as R(x). In particular the rational function

$$R(x) = (x^2-1)/(x-1)$$

is not the same as

$$S(x) = x+1$$

because the domains are different. S(x) is defined for all real x. R(x) is defined and equal to S(x) for all x except x=1, but R(1) is not defined. In the following example we look at this difference geometrically.

**EXAMPLE 5.4 A GRAPH WITH A "HOLE"** *G207

Graph the rational function

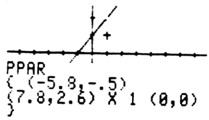
$$R(x) = (x^2-1)/(x-1)$$

and you should see what looks like the graph of the straight line

$$S(x) = x+1$$

As you know from the comments above, these two functions are not the same.

Now graph R(x) with the plot parameters (-5.8,-.5) and (7.8,2.6). Look closely at your graph. Like the graph to the right, it should look like a straight line of pixels with a single pixel "missing". Use the direction keys (as you did in Example 3.3) to move the cross cursor to the "missing" pixel.




Next apply the stroke sequence

INS ON

to place the coordinates (1,2) of the "missing" pixel on the stack. The plot parameters (-5.8,-.5) and (7.8,2.6) were selected so that one of the 137 equally spaced x values turned out to be x=1. In drawing the graph, the calculator tried to evaluate R(1) and found an undefined result, so no point was plotted. Of course, it is much simpler to notice that R(x) is undefined directly from the algebra. Once again we see the importance of analysis.█

Armed with the techniques of this chapter, you are ready to work on graphing the rational functions of precalculus. With a little prior analysis, you will produce complete graphs quickly.

### 5.4 Summary

In this chapter you have seen and worked through several examples of graphing rational functions on a calculator. Particular attention was given to vertical, horizontal, and oblique asymptotes and the algebra behind them. You also learned how to identify rational functions whose graphs have "holes" rather than vertical asymptotes.

# CHAPTER 6: EXPONENTIAL AND LOGARITHMIC FUNCTIONS

**Learning Objectives**

In this chapter you will:

- learn how to use the exponential and logarithmic function keys of your calculator.
- graph exponential and logarithmic functions.
- approximate the value of e using your calculator.
- enter time value of money equations and see how to use them.
- get some more practice with SOLV SOLVR.

**6.1 Introduction**

Besides its graphing capabilities, your calculator has a number of keys that perform exponential and logarithmic operations. You will see how to use these operational keys to help in the evaluation and analysis of the exponential and logarithmic functions of precalculus. Since the behavior of the exponential and logarithmic functions of precalculus ($e^x$, ln(x), and functions built on them) is not nearly as varied as the behavior of polynomials, graphing exponential and logarithmic functions is relatively easy.

In conclusion we will take a look at the important time value of money equations because they are

* examples of more complicated exponential functions.

* exponential functions that are practical in everyday terms.

* a means to learn the important features of the SOLV SOLVR menu.

We begin by looking at some keys and menu items on your calculator that apply to exponential and logarithmic functions.

## 6.2 Evaluation

You have probably already used the exponent key ^ to calculate quantities like $2^5$ with the stroke sequence

2,5 ENTER ^

Similarly the stroke sequence

2,X ENTER ^

will calculate the quantity $2^x$ if a value is stored under X. Otherwise this stroke sequence will put the expression '2^X' on the stack. Thus you can store the exponential function

$$f(x) = 2^x$$

with the stroke sequence

2 ENTER X ENTER ^ 'F STO

and evaluate it at x=5, say, by pressing

5,'X STO F EVAL

The two exponential functions $10^x$ and $e^x$ have their own keys on the HP28S as on most calculators. To see these (and other) exponential and logarithmic function keys, press

LOGS NEXT NEXT

to get a look at the LOGS menu. In precalculus you will most likely only need to use the first four menu items: LOG, ALOG, LN, and EXP. ALOG is the exponential function $10^x$. It takes a number off the stack and returns its common antilogarithm. Thus the stroke sequence 2 ALOG returns 100, which is $10^2$. Similarly EXP evaluates $e^x$. The menu keys LOG and LN evaluate the common logarithm and natural logarithm respectively. Try out these four keys before you go on to the next section on graphing.

As a final exercise in function evaluation, store

$$f(h)=(1+h)^{1/h}$$

in your calculator. It can be shown that the value of f(h) gets closer to e≈2.718281828459045 as h gets nearer and nearer 0. Try storing values of h just greater than 0 and see what f(h) yields on evaluation. See how close you can come to e.

## 6.3 Graphing

As long as you consider only the basic exponential functions of form

$$M(t) = M_0 e^{kt}$$

you can expect graphs that look very much like the graph of $e^x$ if $k>0$ and very much like the graph of $e^{-x}$ if $k<0$. We consider an example involving the decay of radium.

**EXAMPLE 6.1 EXPONENTIAL DECAY** ***G237**

The exponential function

$$M(t) = M_0 e^{kt} = 10e^{-.00041t}$$

gives the amount of radium in grams that remains of an original 10 grams of radium after t years. Store the function $10e^{-.00041t}$ as EQ for graphing. Now pressing

1000 'T STO RCEQ EVAL

we see that $M(1000) \approx 6.64$ grams. Similarly $M(10000) \approx .17$, so let's choose to look at $M(t)$ for $0 \le t \le 10000$.

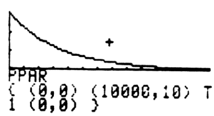
Since M(0)=10 a suitable viewing rectangle is

[0, 10000] by [0, 10]

Use the stroke sequence

(0,0 PMIN (10000,10 PMAX DRAW

and you should get the graph shown to the right.█




After working through the example above you should be able to use the evaluation and graphing features of your calculator to handle exponential growth and exponential decay problems. Do some of these problems from your text.

Note: In working with the exponential and logarithmic functions of precalculus you should not have to rely on the graphing capabilities of your calculator very much. However your calculator's ability to evaluate functions at many points quickly will turn out to be very useful.

There are many variations based on the formula

$$M(t) = M_0e^{kt}$$

that model natural phenomena that we will not go into. We do, however, look at a particular family of variations on this formula in the next section.

## 6.4 The Time Value of Money Equations

We will represent the *compound interest* formula by

$$A = P(1 + I)^N$$

In this formula and throughout this section the (standard) symbols A, P, I, and N will always mean the following:

A: the future value.

P: the present value.

I: the interest rate each compounding period.

N: the number of compounding periods.

The compound interest formula is ideal for showing the usefulness of SOLV SOLVR on the HP28S, as you are about to see.

**EXAMPLE 6.2 THE COMPOUND INTEREST FORMULA** ***G238**

Enter the compound interest formula as EQ for use in SOLV SOLVR with the stroke sequence

A,P,1,I+N^*= ENTER STEQ

Now the stroke sequence

SOLV SOLVR will make your screen look like the figure to the right. What you see is the solver menu for your equation $A = P(1 + I)^N$.


There are six items. One for each of your four variables, plus one for each side of the equation. The idea is for you to provide SOLV SOLVR with the values of three of the four variables and for SOLV SOLVR to respond by calculating and displaying the corresponding value of the fourth variable.

To see how this is done consider the problem:

> A bank offers 6% annual interest on savings accounts. How large does a $1000 account become after 10 years if interest is compounded monthly?

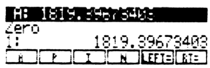
Enter the given values of P=1000, I=6%/12=.005, and N=120 months by means of the stroke sequence

1000 P .005 I 120 N

Now that SOLV SOLVR has been supplied with values for P, I, and N, press

A

(that is, the red key followed by A) to calculate and display A=1819.40. You screen should appear as shown.




Now what if you want to see what happens if interest is compounded daily instead of monthly? You just enter the new values for I and N and calculate A by means of

.06,365 / I 10,365 * N A

to find out that A=1822.03 for daily compounding. With only a short amount of practice you can make "what if" comparisons of different values for any of the variables A, P, I, and N.∎

Once entered into the calculator, more complicated equations are just as easily used as the compound interest formula

$$A = P(1 + I)^N$$

Such an equation that you may be familiar with is the *present value of an annuity* formula

P=R*(1-(1+I)^(-N))/I

that can be derived from the compound interest formula. This formula involves the same P, I, and N as above. R (the "periodic payment") represents the payment that must be made at the end of each of N compounding periods to be equivalent to the present value P, assuming the use of money is worth a rate of I per period.

For example, if you borrow $68000 at 12% compounded monthly for 30 years, P=68000, I=.01, and N=360. If you store the formula P=R*(1-(1+I)^(-N))/I as EQ, and then enter the values for P, I, and N, SOLV SOLVR will calculate R=699.46 when R is pressed.

Another time value of money formula that you may find interesting is the *future value of an annuity* formula

A=R*(((1+I)^N-1)/I)

that can be derived from the previous two formulas (Try it). Some precalculus texts treat these time value of money equations in detail, most texts do not. But you can certainly see how they are useful in everyday life. Furthermore the time value of money equations are the perfect way to get familiar with SOLV SOLVR, one of the most useful features of the calculator.

If your text has time value of money problems you will certainly want to practice using these equations and SOLV SOLVR. If your text doesn't, you might still want to use them (now or in the future) in analyzing your own finances.

**6.5 Summary**

In Chapter 6 you have seen how to use the exponential and logarithmic function keys of your calculator. You worked through an example in graphing an exponential function. Three of the time value of money equations were discussed as examples of more complicated exponential functions. You saw how the time value of money equations are not only practical for your personal finances, but also helpful in learning the important SOLV SOLVR features of the HP28S.

# CHAPTER 7: TRIGONOMETRIC FUNCTIONS

**Learning Objectives**

In this chapter you will:

- learn how to use the trigonometric function keys of your calculator.
- graph the basic trigonometric functions.
- see how to graph the inverse trigonometric functions.
- solve trigonometric equations numerically.

**7.1 Introduction**

There are three features of your calculator that are indispensable to the study of trigonometric functions and trigonometric equations. These are the evaluation operations, the graphing capabilities, and the numerical method in SOLV SOLVR. You will see how to apply these calculator features by working through a few examples.

After a brief tour of the trigonometric function evaluation keys you will graph the basic trigonometric functions. Using just the default plot parameters you will make good graphs quickly. Through graphs you will visualize the behavior of more complicated trigonometric functions. Most importantly, you will see how to use graphs and SOLV SOLVR together to analyze and then find approximate numerical solutions to trigonometric equations.

**7.2 Evaluation of Trigonometric Functions**

Look at the TRIG menu on your HP28S. Use NEXT to see all three pages. On the first page you see SIN, COS, and TAN as well as their inverse functions ASIN, ACOS, and ATAN. There are no keys for the cotangent, secant, and cosecant. If you want, you can define your own USER functions for the cotangent, secant, and cosecant as '1/TAN(X)', '1/COS(X)', and '1/SIN(X)' respectively. Otherwise you can evaluate the secant, say, of a number on the stack with the stroke sequence

COS 1/X

which evaluates sec(X)=1/cos(X). The other 11 functions in the TRIG menu are for polar/rectangular coordinate conversion, degree/radian conversion, and calculations with degree-minutes-seconds. You can read about these in detail in the HP28S **Reference Manual** under TRIG. Note that there are keys for converting from degrees to radians and from radians to degrees. You can usually avoid the use of these by setting the **angle mode** in the MODE menu as described below.

On the first page of the MODE menu, either DEG or RAD will have a small square following its name to indicate that the calculator is in the DEG or RAD angle mode. To change angle mode from RAD (as in the top figure) to DEG (as in the lower figure) press DEG. You may select RAD similarly.

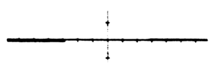
With the DEG angle mode and 30 on the stack, press SIN. You should get .5, the sine of 30° to appear on the stack. Now with .5 on the stack, press ASIN. The inverse sine of .5 (that is, 30) appears on the stack. Practice these six trigonometric function keys SIN, COS, TAN, ASIN, ACOS, and ATAN in both radian mode and degree mode. Check the values you get in your text or in a table of trigonometric function values. When you feel comfortable with these keys you are ready to graph some trigonometric functions.

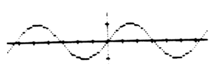
## 7.3 Graphing of Trigonometric Functions

Start by graphing sin(x) under the default plot parameters with the stroke sequence

X ENTER SIN STEQ 'PPAR PURGE DRAW

Your graph will look like one of the figures to the right. If you got the familiar sine shape of the lower figure, you are done. If you got the uninteresting figure on the top, you need to change the angular mode and draw another graph with





MODE RAD DRAW

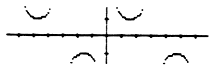
Note: The most common mistake in graphing trigonometric functions is being in the wrong angular mode.

Now try the stroke sequence

RCEQ 1/x STEQ DRAW

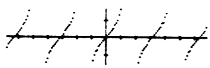
to produce the graph of

csc(x)

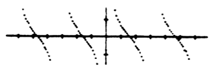


(the inverse of sin(x)). Without changing from the default plot parameters produce the graphs of

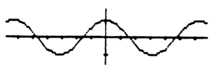
tan(x)



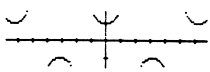
cot(x)



cos(x)



and



sec(x)

respectively. Your graphs should appear as shown on the right.

The same default plot parameters that rarely satisfied you when graphing polynomial functions are ideal for graphing the basic trigonometric functions. In your precalculus course it is likely that you will soon need to see graphs of functions of the form f(x) = Asin(Bx) as in the following example.

**EXAMPLE 7.1 GRAPHING TRIGONOMETRIC FUNCTIONS**

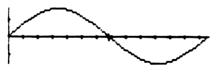
To see how to graph a trigonometric function with an amplitude different from 1 and a period different from $2\pi$ consider

$f(x) = 12\sin(8\pi x)$

To accommodate the amplitude of 12 we must allow y to vary between -12 and 12. To see one period of the graph we need to let x vary by at least $2\pi/8\pi=.25$, so we choose the plot parameters

(0,-12) and (.25, 12)

which yield the graph shown in the top figure. Of course



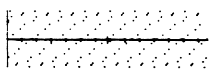
$f(x) = A\sin(Bx)$

*always* has this graph (for A positive) if we pick the plot parameters so well.

On the other hand a poor choice of plot parameters can give a confusing graph. Recall Example 2.6 in which a graph with 275 zeros appeared to have only three. As another example suppose we had chosen the plot parameters

(0,-12) and (4, 12)

for our graph above. DRAW yields the confusing graph to the right. Experiment with other plot parameters of form



(0,-12) and ($x_1$, 12)

for $x_1>4$. Even with a good choice of scale for the y axis, the graph can be difficult to analyze. Try making some graphs of this function with the y range vary large (100 or so) and very small (1 or so) to see what can happen with poor choices of the size of y.■

Note: When graphing trigonometric functions first determine a reasonable y range by looking at the amplitude. Select x values that cover one period or so.

Try your hand at graphing some of the functions of form Asin(Bx+C), Atan(Bx+C), Acot(Bx+C), and Acos(Bx+C) that appear in your textbook. Your analysis should proceed much like the analysis in Example 7.1. If your text has graphs of Asec(Bx+C) and Acsc(Bx+C) the procedure is again similar (just be sure that your y range is sufficient).

The inverse trigonometric functions may need some resizing of the screen, but are not very difficult to graph, as we are about to see in the case of the function $\cos^{-1}(x)$.

**EXAMPLE 7.2 GRAPHING INVERSE TRIGONOMETRIC FUNCTIONS**

As an example of a graph of an inverse trigonometric function consider

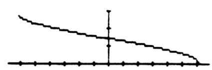
$$f(x) = \cos^{-1}(x)$$

Store $\cos^{-1}(x)$ as EQ and graph it under the default plot parameters by means of

X, ACOS STEQ 'PPAR PURGE DRAW

to obtain the first graph shown. The graph itself may make you suspect that there are some more pixels to be seen in the second quadrant if you extend the y range a bit. A proper analysis tells you this and more. From the definition of inverse function and properties of cos(x), it can be shown that the graph of $\cos^{-1}(x)$ goes from $(-1,\pi)$ to (1,0) monotonically.Thus an appropriate set of plot parameters is

(-1.1, -.1) and (1.1, 3.3)



These produce the second graph to the right.■

The graphing and evaluation capabilities of the calculator that you have covered so far are helpful in investigating identities (or determining *if* a trigonometric equation is an identity). They are most helpful, however, in approximating (if not actually finding) solutions to trigonometric equations, as we see in the next section.

### 7.4 Numerical Techniques for Trigonometric Equations

In this section you will see how to use the calculator to find (or numerically approximate) solutions of trigonometric equations using SOLV SOLVR. If your textbook discusses numerical techniques for solving trigonometric equations (as in Section 8.4 of Gechtman's **Precalculus**) by all means study them. Although the iterative procedure of the calculator is somewhat more advanced, it is still based on simple algebraic and geometric ideas. It is fast and accurate because it utilizes the calculator's speed, programming capabilities, and precision to advantage. As an example we choose a trigonometric equation that cannot be solved by algebraic means.

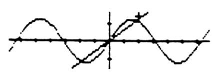
**EXAMPLE 7.3 NUMERICAL APPROXIMATION** ***G337**

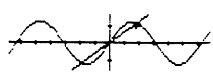
As an example of the solution of a trigonometric equation by the numerical method in SOLV SOLVR consider the simple looking equation

x = 2sin(x)

Begin by noting that since 2sin(x) oscillates between -2 and +2 the plot parameters (-6.8, -2.5) and (6.8, 2.6) ought to be suitable.

Do this and you should obtain the graph in the top figure. Now as in Example 3.1, move the cross cursor to a point just above and to the right of the point of intersection in the first quadrant (as shown in the top figure). Press INS to send the coordinates of the point to the stack (the stack is not visible at the moment). Now move the cross cursor to a point just below and to the left of the point of intersection (as in the second figure). Press INS to send the coordinates of this point to the stack. Next use the stroke sequence

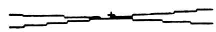


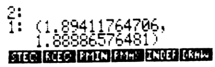


ON PMIN PMAX DRAW

to zoom in on the point of

intersection. Your graph should be similar to the graph shown. Zoom in more if you like. Then press INS with the cross cursor as near the point of intersection as you can. After zooming in just once the cross cursor and INS yielded the approximation shown in the figure. This gives us the approximation to the point of intersection






(x, y) = (1.89411764706, 1.88886576481)

For practice try zooming in again and again relentlessly. See how close you can approximate the point of intersection of

f(x)=x and g(x)=2sin(x)

When you think you know the point of intersection (up to the limit of the calculator) or when you tire of the chase, put a "guess" for the x value of the point of intersection on the stack (you may leave a coordinate pair for the point of intersection on the stack instead). Now as in Example 3.3 press

SOLV SOLVR X X

to have SOLV SOLVR do the zooming for you. Your screen should proudly display the approximation 1.89549426704 for the x value of the point of intersection. The point itself is approximately


(1.89549426704, 1.89549426704)

of course. You can say that

2sin(1.89549426704) ≈ 1.89549426704.█

Use the combination graphing and SOLV SOLVR procedure above to approximate solutions to trigonometric equations from your text. Algebraic methods may be possible on some of these equations. When possible compare algebraic and numerical results.

## 7.5 Solving Triangles

The programming capability of the HP28S is advanced enough that a single program can be written to solve triangles. If you would like an ambitious project in programming, try writing such a program. It would be best to start with special case programs and ultimately combine them into a large user friendly program. A much less ambitious project would be to store the Law of Sines or Law of Cosines as EQ and apply SOLV SOLVR.

## 7.6 Summary

In Chapter 7 you saw how to use the basic trigonometric function keys of the HP28S in order to evaluate trigonometric functions. You graphed all the basic trigonometric functions as well as an inverse trigonometric function. Working through a long example using both the graphing and numerical approximation capabilities of the calculator you learned how to solve trigonometric equations numerically.

# CHAPTER 8: SEQUENCES AND SERIES

**Learning Objectives**

In this chapter you will:

- use calculator keys to evaluate permutations and combinations.
- learn how to evaluate terms of sequences with your calculator.
- learn how to sum series with your calculator.
- generate terms of a sequence using a recursion formula.

## 8.1 Introduction

Primarily this chapter deals with *evaluation*: evaluation of sequences and series and evaluation of permutations and combinations. The expressions P(n,r) and C(n,r) that come up so often in counting problems are often difficult to evaluate with an ordinary calculator. The HP28S has menu items that perform these calculations delightfully fast. The world of sequences and series is too varied for single key evaluation, but the programming capabilities of your calculator allow you to evaluate sequences and series with a minimum of keystrokes. You will have the chance to learn a little about the programming language of the HP28S at the same time that you learn how to evaluate sequences and series.

## 8.2 Counting Formulas

Combination and permutation counting problems became a lot more fun when the ! (factorial) key became standard on calculators. Not only does your HP28S have such a key, but it has two others that take combinations and permutations with a single keystroke. In this section we see how to use these keys to evaluate factorials, combinations, and permutations.

The factorial key is REAL FACT. It takes an argument n off the stack and returns n!. For example

10, REAL FACT

returns 3628800, which is 10!.

The other two counting keys are in the STAT directory, whose four pages of menu items perform a tremendous number of the standard operations of elementary statistics. We will content ourselves by doing two counting problems using the keys COMB and PERM (for combination and permutation).

A Combination Problem:

> In how many ways can five senators be chosen from a group of 100 senators to appear before the Senate Ethics Committee?

The solution is given by the combination formula

$$C(n,r) = C(100,5)$$

which on the HP28S is evaluated by placing the arguments on the stack and running COMB with the stroke sequence

100,5 COMB

yielding 75,287,520 with minimal effort.

A Permutation Problem: ***G528**

> How many four digit numerals having no repeating digits can be formed from the elements in {1,2,3,4,5,6,7,8,9}?

The solution is given by the permutation formula

$$P(n,r) = P(9,4)$$

which is evaluated by placing the arguments on the stack and running PERM with the stroke sequence

9,4 PERM

yielding 3024.

For more advanced problems the advantage of these keys increases. Formulas that involve the evaluation of several combinations (like the Hypergeometric Probability Formula) can be programmed if they are used a lot. Expressions with lots of combinations and other operations (for example, those that arise when determining the probability of certain poker hands) can be quickly evaluated using combinations in conjunction with other keys.

### 8.3 Terms of a Sequence

In Section 8.2 and 8.3 we will use the sequences

{1/(n*(n+1))} = 1/1*2, 1/2*3, 1/3*4,..., 1/(n*(n+1)),...

and

{2n} = 2, 4, 6, 8,...

to demonstrate how to work with sequences and series on your calculator. First we see how to evaluate terms of a sequence.

**EXAMPLE 8.1 EVALUATING TERMS OF A SEQUENCE** ***G510**

To evaluate $a_n$, the $n^{th}$ term of the sequence

{1/(n*(n+1))}

for several values of n, we first store the expression for $a_n$ by means of the stroke sequence

N ENTER ENTER 1+ * 1/x 'AN STO

Then we evaluate $a_{15}$, say, with the stroke sequence

15,'N STO AN EVAL

This will put 4.16666666667E-3 on the stack, representing the decimal approximation

.00416666666667 ≈ 1/15*16

This evaluation process can be done quickly for a lot of values for N using the program TERM given to the right. TERM begins by prompting for a term. At the HALT the stroke sequence

```
TERM
« "N=?" HALT 'N' STO
DROP AN EVAL
»
```

15 CONT

instructs TERM to continue as shown in the figure to the lower right. Upon continuing, the prompt is dropped, and $a_{15}$ is evaluated and displayed.

```
TERM
1: "N=?"

15 CONTINUE
1: 4.16666666667E-3
```

Now suppose you want to evaluate terms of another sequence, $a_n$=2n, for example. First use

'N PURGE 2,N* 'AN STO

to enter the $n^{th}$ term of the new sequence. Now you can use TERM to evaluate terms of the sequence $a_n$=2n. With this sequence now stored under AN, pressing

TERM 15 CONT

yields 30 instead of .00416666666667.█

You are about to see how programming can help even more in the next section, where we calculate the sum of a series.

## 8.4 Series

To see how the power of programming can help us in evaluating the partial sums of a series

$$S_n = a_1 + a_2 + \ldots + a_n$$

we look at a particular case.

### EXAMPLE 8.2 CALCULATING PARTIAL SUMS *G510

If you *know* that n/(n+1) is the $n^{th}$ partial sum of the sequence

{1/(n*(n+1))}

it is not too difficult to *prove* that it is by mathematical induction. But how do you come to believe that n/(n+1) is the partial sum $S_n$? Often you begin looking for a formula for $S_n$ by evaluating $S_1$, $S_2$, $S_3$, $S_4$,... and looking for a pattern. The program SS given to the right below helps to do this. The program SS calculates and displays the first 15 partial sums of the sequence {1/(n*(n+1))}. If you would like to save it, create a CH8 directory and store it there.

SS begins by storing 0 under SUM. Then the first 15 terms of $a_n$=1/(n(n+1)) are added to SUM one by one and the partial sums $S_1$, $S_2$,...,$S_{15}$ are put on the stack (See

```
SS
« 0 'SUM' STO 1 15
  FOR I '1/(I*(I+1))
' EVAL 'SUM' STO+
SUM
  NEXT
»
```

the second figure to the right for a sample output of ss). $S_{15}$ remains stored under SUM as well. You may or may not be able to tell that the $n^{th}$ partial sum $S_n$ is n/(n+1), but by replacing the "15" in the program by "50" you can learn more about the partial sums.

```
15: .5
14: .666666666667
13: .75
12: .8
11: .833333333333
10: .857142857142
9: .874999999999
8: .888888888888
7: .899999999999
6: .909090909090
5: .916666666666
4: .923076923076
3: .928571428571
2: .933333333333
1: .9375
```

Do this by visiting ss with the stroke sequence

'ss VISIT

and then using the direction keys to move the cursor to "15" and type "50". Next ENTER and run the revised program ss to see what the first 50 partial sums are. You'd have to be very lucky to notice that the value

$$S_{50} \approx .980392156861$$

given by the program is a signal that $S_{50}$ might be

$$50/51 \approx .980392156863$$

But other properties of the sequence are easier to see. For example, it appears that each $S_n$ is bigger than $S_{n-1}$ but still less than 1.

To look at the partial sums of the sequence $a_n$=2n, use the VISIT key to edit ss as before. This time move the direction key to the first "1" of '1/(I*(I+1))' and replace it by '2*I' with the stroke sequence

2*I DEL DEL DEL DEL DEL DEL DEL DEL

Change the "50" to a "5" to get just the first five terms. See if you can come up with a simple formula for the general partial sum $S_n$.∎

Try using the program ss to investigate the sequences and series that you encounter in precalculus. To do so, you will have to customize ss as in Example 8.2 above.

We conclude this section with an example that shows how the stack logic of the HP28S can be used to generate terms of a recursively defined sequence.

**EXAMPLE 8.3 TERMS OF THE FIBONACCI SEQUENCE** ***G515**

Many mathematics texts have interesting stories associated with the Fibonacci sequence defined recursively as

$$f_0 = 0, \; f_1 = 1, \; f_n = f_{n-1} + f_{n-2}$$

We give two short programs that calculate the terms $f_n$ of the Fibonacci sequence to demonstrate how to calculate the terms of any recursively defined sequence. Enter the programs FIB3 and FIB4 exactly as given below on the right.

FIB3 is very simple. If $f_{n-2}$ and $f_{n-1}$ are in level 2 and level 1 of the stack, the program calculates $f_n$ and puts $f_n$ on level 1 (moving $f_{n-2}$ and $f_{n-1}$ up to level 3 and level 2.

```
FIB3
« DUP2 +
»
```

FIB4 begins by putting $f_0$=0 and $f_1$=1 on the stack, and by prompting for the number of terms you want to see. After you enter your response for "n", FIB4 continues by performing a loop that puts $f_2$, $f_3$, $f_4$,...,$f_n$ on the stack.

```
FIB4
« 0 1 2 " # Terms? "
HALT SWAP DROP
  START DUP2 +
  NEXT
»
```

Notice that in putting FIB3 on your calculator you have essentially defined a new key, a key that takes two consecutive terms of a Fibonacci sequence off the stack and returns them, along with the next term of the Fibonacci sequence. Use each of FIB3 and FIB4 to determine the first few terms of the Fibonacci sequence

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...$$

The reason that these programs above are designated FIB3 and FIB4 is that there are two other programs FIB1 and FIB2 provided and explained in the HP28S **Owner's Manual**. All four of these programs calculate terms of the Fibonacci sequence, but in different ways. You can learn a lot about programming on the HP28S by studying these four programs. You can also learn a lot about the Fibonacci sequence. Investigate the Fibonacci sequence using these programs. For example look at the sequence of ratios $f_{n+1}/f_n$. ∎

**8.5 Summary**

In Chapter 8 the menu keys FACT, COMB, and PERM that calculate factorials, combinations, and permutations were demonstrated by means of some examples. However, most of this chapter dealt with four programs helpful in working with sequences and series:

* TERM that calculates terms of a sequence from the formula for the general term.

* SS that calculates the partial sums of a sequence.

* FIB3 and FIB4 that calculate terms of the Fibonacci sequence from the recursion formula.

You have seen how to modify these programs to calculate terms and partial sums of other sequences. If you choose, you can study these programs to begin to learn a little about the programming capability of the HP28S.

# CHAPTER 9: POLAR COORDINATES AND PARAMETRIC EQUATIONS

**Learning Objectives**

In this chapter you will:

- learn how to make polar coordinate graphs.
- learn how to make parametric equation graphs.

## 9.1 Introduction

This chapter deals primarily with the graphing of polar coordinate and parametric equation graphs. The HP28S does not make either of these type graphs readily. Consequently we will write and use programs that utilize some of the graphing features of the HP28S to produce polar coordinate and parametric equation graphs. You might like to study these programs closely in order to improve your programming skills, but you need not understand them completely in order to get the graphs you want. However you can learn some things about the mathematics of polar and parametric equations if you study these programs.

Look in your COMPLEX menu for a number of keys associated with polar coordinates. They are explained in COMPLEX in the HP28S **Reference Manual**. In particular note that P→R transforms polar coordinates into rectangular coordinates and R→P transforms rectangular coordinates into polar coordinates.

## 9.2 Graphs in Polar Coordinates

Upon seeing the beautiful polar graphs in your text, you were doubtlessly eager to draw your own. Plotting points of even the most basic polar coordinate graphs is slow going. The program POLAR below will allow you to speed this process up considerably. Enter it exactly as shown in your CH9 menu.

POLAR begins by asking you for r in terms of T (there is no θ key on the HP28S). After you enter the function r(T), POLAR clears the screen, draws axes, and evaluates the *polar* coordinate pairs (r,T) for

T = 0°,4°,8°,...,360°

Then these *polar* coordinate pairs are converted to *rectangular* coordinate pairs and plotted, one point at a time.

```
POLAR
« " r(T)=? " HALT →
r
  « CLLCD DRAX DROP
0 360
    FOR t t 'T' STO
r EVAL T R→C P→R
PIXEL 4
    STEP { T } PURGE
DGTIZ
  »
»
```

Since most people like to think in terms of degrees rather than radians when dealing with polar coordinates, POLAR is written to be used in DEG angular mode. Use the stroke sequence

MODE DEG

to get into DEG angular mode whenever you use POLAR. Now you are ready to use POLAR to make some polar graphs.

**EXAMPLE 9.1 A POLAR PLOT** ***G389**

Store r=1+cos(θ) under EX1 in CH9 for graphing by means of the stroke sequence

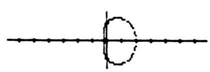
1,T COS + 'EX1 STO

Now press

POLAR EX1

```
POLAR
1: " r(T)=? "
EX1
1: '1+COS(T)'
```

to run POLAR and respond to the prompt (as shown to the right). Your screen should exhibit the graph shown. Your graph consists of the 91 pixels approximating the coordinates



(1+cos(0°),0°), (1+cos(4°),4°),...,(1+cos(360°),360°)

The first and last pixels plotted are the same, since

(1+cos(0°),0°)

is the same as

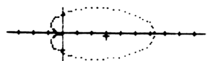
(1+cos(360°),360°).

That leaves 90 pixels to be darkened, but a close count reveals only about 60. Where are the other 30? Some lie on the (already darkened) axes. Others are redundantly darkened a second time due to *rounding to pixel accuracy* (see Chapter 2). None of the 91 points correspond to pixels out of range of the screen.

Use the cross cursor, INS, PMIN and PMAX to get a more suitable screen size (as in the figure to the right). Run

POLAR EX1

again, watching the pixels as they are darkened. This will give you a better feel for the behavior of the graph than the still picture alone does.█



```
PPAR
{ (-1.2,-1.5)
(3.1,1.3) constant 1
(0,0) }
```

Now that you see how to use POLAR, you are going to have some fun plotting a family of interesting polar graphs quickly in the next example.

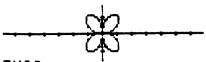
**EXAMPLE 9.2 SIN(Kθ)**

At the top of the next page you see graphs of sin(kθ) for k = 1, 2, 3, 4, 5, and 6. PPAR is as shown to the right. POLAR has been changed slightly as shown. The "4 STEP" in the program POLAR has been changed to read "2 STEP". This causes a pixel to be plotted for each 2° rather than each 4°. The picture is better, but it takes twice as much time to draw.

```
PPAR
{ (-4.6,-1.1)
(4.6,1.1) constant 1
(0,0) }
```

```
POLAR
« " r(T)=? " HALT →
r
  « CLLCD DRAX DROP
0 360
    FOR t t 'T' STO
r EVAL T R→C P→R
PIXEL 2
    STEP { T } PURGE
DGTIZ
  »
»
```


EX21
'SIN(T)'


EX22
'SIN(T*2)'


EX23
'SIN(T*3)'


EX24
'SIN(T*4)'


EX25
'SIN(T*5)'


EX26
'SIN(T*6)'

In this example we have taken advantage of the calculator's speed to plot several graphs quickly. This leaves more time for analysis. For example, look at the "loops" as they are formed in each of the graphs and see if you can come up with a conjecture for

* the number of "loops" in the graph of $\sin(k\theta)$.

* the values of k for which the "loops" are tangent to one another.

* the values of k for which the "loops" are formed twice as $\theta$ goes from $0^\circ$ to $360^\circ$.

* the reason some of the "loops" are formed twice as $\theta$ goes from $0^\circ$ to $360^\circ$. ■

Now make graphs for the polar coordinate functions in your text. Try graphing whole families of graphs, as you did above for the family $\sin(k\theta)$.

As a last example on polar graphs, we consider $r^2 = 4\cos(2\theta)$, a standard polar equation not of form $r=r(\theta)$.

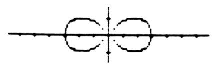
**EXAMPLE 9.3 A LEMNISCATE** *G393

One way to handle an equation like $r^2 = 4\cos(2\theta)$ is to modify POLAR slightly as we did to get the program POL2 shown on the right. The logical statement

IF r EVAL IM 0 == THEN ... END

prevents the calculator from giving an error message in trying to deal with a complex number. There are other ways to modify POLAR to accomplish the same purpose (you may like to try). The graph produced by POL2 is shown as well.■

```
POL2
« " r(T)=? " HALT →
r
  « CLLCD DRAX DROP
0 360
    FOR t t 'T' STO
      IF r EVAL IM 0
==
      THEN r EVAL T
R→C P→R PIXEL
      END 1
    STEP { T } PURGE
DGTIZ
  »
»
```



Now that you know how to use POLAR and POL2 to make polar graphs, it is time to look at the graphs of parametric equations.

### 9.3 Graphs of Parametric Equations

The program PARA below will help you produce graphs of equations given parametrically. The program PARA plots 100 (x,y) coordinates for 100 equally spaced values of the parameter t. Store it in your CH9 menu.

PARA begins by asking you for the extreme values

$t_0$ and $t_n$

of the parameter t, as well as for the parametric equations x(t) and y(t). When these four items are entered by the user, they are stored and 101 points $(x(t_0),y(t_0))$, $(x(t_1),y(t_1))$, ..., $(x(t_{100}),y(t_{100}))$ are plotted.

```
PARA
«
"T0,TN,X(T),Y(T) =?"
HALT 'YT' STO 'XT'
STO 'TN' STO 'T0'
STO CLLCD DRAX DROP
0 100
  FOR t T0 TN 100 /
t * + 'T' STO XT
EVAL YT EVAL R→C
PIXEL
  NEXT DGTIZ { T }
PURGE
»
```

Since it is customary to think in terms of radians rather than degrees when dealing with parametric equations, PARA is written to be used in RAD angular mode. Use the stroke sequence

MODE RAD

to get into RAD angular mode so you can try PARA out on the two examples that follow. If you like, you could insert RAD at the beginning of PARA and DEG at the beginning of POLAR to assure yourself that the proper angular mode is being used for these two programs.

**EXAMPLE 9.4 MOTION OF A PROJECTILE** ***G395**

Under certain conditions a projectile will move through the air over time t such that the projectile's coordinates relative to a fixed point on the surface of the earth are given parametrically by

$$x(t) = 10t \quad \text{and} \quad y(t) = 10t-t^2$$

$$\text{for } 0 \le t \le 10 \text{ sec.}$$

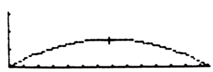
Do the following to produce a graph showing the flight of the particle over the 10 second period.

* Set the plot parameters as (0,0) and (100,50).
* Store the four arguments for this problem as ARG1 (as shown to the right).

```
ARG1
« 0 10 '10*T' '10*T-
T^2'
»
```

* Press PARA ARG1 to produce the graph below.

PARA produces a graph simulating the flight of the projectile as time passes.



If you modify PARA by replacing each "100" by "40", the graph will be a real time simulation. That is, the calculator will take 10 seconds to draw the graph of the 10 second flight of the projectile.

We conclude this section on graphs of parametric functions by considering another problem of motion.

**EXAMPLE 9.5 AN ELLIPTICAL PATH** ***G396**

As another example of the use of PARA, graph the parametric equations

$$x(t) = 3\cos(t) \quad \text{and} \quad y(t) = 2\sin(t)$$

$$\text{for } o \le t \le 2\pi \text{sec.}$$

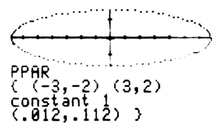
Do this similar to the way you did the previous example.

* Set the plot parameters as (-3,-2) and (3,2).
* Store the four arguments for this problem as ARG2 (as shown to the right).

```
ARG2
« 0 6.3 '3*COS(T)' '
2*SIN(T)'
»
```

* Press PARA ARG2 to produce the graph below.

This time PARA simulates the motion of an object around an elliptical path. Again, with a parametric equation you don't just get a graph, you get a moving particle.█




You should have no trouble using the procedure of the two problems above to draw graphs of the parametric equations of precalculus.

**9.4 Summary**

In Chapter 9 you were given the coding for a program POLAR that allows you to plot polar coordinate graphs easily. You also were given the coding for PARA, a similar program for graphing parametric equations. Through several examples you saw how to apply these programs in particular situations. Some modifications of both of these programs were made (and other modifications suggested) that give them greater flexibility. Still in their original form, both POLAR and PARA produce good graphs quickly for the polar coordinate and parametric equation graphs of precalculus.

# CHAPTER 10: THE CONIC SECTIONS

**Learning Objectives**

In this chapter you will:

- learn how to make graphs of conics in rectangular coordinates.
- learn how to make graphs of conics in polar coordinates.

## 10.1 Introduction

A discussion of conic sections (circles, ellipses, parabolas, and hyperbolas) often begins with the geometric definition of a conic section as the intersection of a plane and a cone **(*G406)**. Pictures are used to show how such intersections yield circles, ellipses, parabolas, and hyperbolas (or degenerate forms of these such as single points, no points at all, straight lines, or two intersecting straight lines).

In rectangular coordinates the (equivalent) algebraic definition of a conic section is given as the collection of solutions of

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

for A, B, C, D, E, and F real. Once A, B, C, D, E, and F are given, a particular conic is determined, so it is possible to find its graph by substituting x values and determining corresponding y values. It has long been necessary to look for algebraic relationships between the coefficients A, B, C, D, E, and F before attempting to make a graph. In Section 10.2 you will see how to use a program CONIC that prompts for the coefficients A, B, C, D, E, and F and then provides you with a graph of

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

without prior algebraic analysis. This does not mean that you can avoid algebraic analysis; it just means that you can get a look at a graph quickly. You will still need to analyze algebraic relationships to determine features like asymptotes, intercepts, directrices, domain of definition, and foci. Your graphs can aid in this analysis, but they can't replace it.

In Section 10.3 you will see how to use the program POLAR of Chapter 9 to graph conics in polar coordinate form

$$r(\theta) = ed\ /\ (1 \pm e\cos(\theta)) \quad \text{or} \quad r(\theta) = ed\ /\ (1 \pm e\sin(\theta))$$

As in the case of rectangular coordinates, you can get graphs quickly that can help you in your algebraic analysis of conics in polar form.

## 10.2 Rectangular Coordinate Form

You can't use DRAW directly to graph

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

because y is not expressed (and usually not expressible) as a function of x.

The equation EQ1 and the program CONIC given below allow you to use DRAW indirectly to graph conics in rectangular coordinates. To make a place for EQ1 and CONIC and some other menu items associated with graphing conics in rectangular coordinates, create a CH10 subdirectory and within CH10 create the subdirectory RECT. Then store EQ1 and CONIC as given below in the subdirectory RECT.

Carefully store the expression to the right as EQ1. Notice that the two sides of EQ1 are the two roots of $x^2$+Bxy+C$y^2$+Dx+Ey+F=0, a quadratic equation in y.

```
EQ1
'(-(B*X+E)+√((B*X+E)
^2-4*C*(A*X^2+D*X+F)
))/(2*C)=(-(B*X+E)-√
((B*X+E)^2-4*C*(A*X^
2+D*X+F)))/(2*C)'
```

Now enter the program CONIC as shown to the right. This program begins by asking for the six coefficients of A$x^2$+Bxy+C$y^2$+Dx+Ey+F=0. Upon continuing, CONIC stores the coefficients and uses DRAW to graph A$x^2$+Bxy+C$y^2$+Dx+Ey+F=0.

```
CONIC
« "A,B,C,D,E,F=? "
HALT 'F' STO 'E' STO
'D' STO 'C' STO 'B'
STO 'A' STO DROP
CLLCD DRAW DGTIZ
»
```

When EQ1 is stored under EQ, CONIC will use DRAW to simultaneously produce two graphs that together constitute the graph of

$$Ax^2+Bxy+Cy^2+Dx+Ey+F=0.$$

We'll see how this is done by looking at three examples, beginning with an ellipse.

## EXAMPLE 10.1 AN ELLIPSE *G424

For this example use the plot parameters shown to the right. The "4" following the "X" will cause evaluation of EQ for only every fourth column of pixels.

```
PPAR
{ (-4,-4) (4,4) X 4
(0,0) }
```

The loss of "smoothness" is compensated by a shorter drawing time. Be sure to store EQ1 as EQ before using CONIC.

Rewrite the expression

$$x^2/9 + y^2/16 = 1$$

for an ellipse as

$$16x^2 + 9y^2 -144 = 0$$

and run CONIC. At the prompt respond by entering

16,0,9,0,0,144 CHS CONT

to get the graph shown.█

Now use the stroke sequence

(6,6) ENTER ENTER CHS PMIN PMAX

to get the viewing rectangle

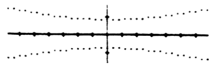
[-6, 6] by [-6, 6]

for the graph of the hyperbola in the next example.

**EXAMPLE 10.2 A HYPERBOLA** ***G432**

Graph the hyperbola

$-9x^2 + 16y^2 - 144 = 0$



by pressing

CONIC 9CHS,0,16,0,0,144CHS CONT

to get the graph to the right.█

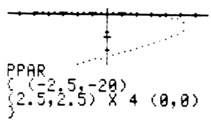
As a final example of CONIC we look at a parabola whose axis is not parallel to either of the coordinate axes.

**EXAMPLE 10.3 A Parabola** ***G419**

Graph the parabola

$4x^2 - 4xy + y^2 + 8x + 16y - 16 = 0$

with the plot parameters shown.




It is clear that the axis of the graph is not parallel to either coordinate axis.█

You are now ready to use CONIC to investigate graphs of conic sections given in your text. As you use CONIC, keep the following points in mind:

* Be sure to store EQ1 as EQ.

* Investigate the equation algebraically to determine information on the conic.

* Set a reasonable PPAR. Make another graph if your original choice of PPAR gives an unsuitable graph.

Next we look at the graphing of conic sections expressed in polar coordinate form.

## 10.3 Polar Coordinate Form

As you should have seen from your textbook by now (Section 11.6 in Gechtman's **Precalculus**), conic sections have a simpler form when expressed in polar coordinates. Ellipses, parabolas, and hyperbolas share the same simple general form

$$r(\theta) = ed / (1 \pm e \cos(\theta)) \quad \text{or} \quad r(\theta) = ed / (1 \pm e \sin(\theta))$$

with just the two parameters e and d (besides the choices between + and - and sin(θ) and cos(θ) ). Furthermore r is expressed as a function of θ directly, which allows us to use the program POLAR of Chapter 9. A single example will suffice to illustrate this.

**EXAMPLE 10.4 A HYPERBOLA** ***G456**

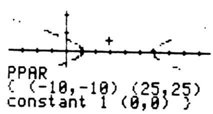
Sketch the conic

$$r(\theta) = 15 / (2 + 3\cos(\theta))$$

There is no need to put r(θ) in standard polar form. Run polar and enter r(θ) at the prompt with the stroke sequence

POLAR 15,2,3,T COS * + / CONT

Your graph should appear as to the right if you use the plot parameters shown. If you find the order in which pixels are darkened puzzling, remember that T takes on the values




0°, 2°, 4°,..., 360°

in order and that r(θ) changes from positive to negative and back to positive as θ goes from 0° to 360°.

If your graph does not appear as shown, check the following:

- Are you in DEG angular mode?
- Did you set PPAR as shown above?
- Did you set PPAR in the Chapter 9 subdirectory?■

Of course you usually want to know more about a polar coordinate function of the form

$$r(\theta) = ed / (1 \pm e \cos(\theta)) \quad \text{or} \quad r(\theta) = ed / (1 \pm e \sin(\theta))$$

than the graph can tell you. To find the eccentricity in the previous example, you would divide numerator and denominator by 2 to get e=1.5. To find the distance between the directrix and the pole, you must calculate d=5. Algebraic methods are also necessary to find asymptotes and intercepts of conics.

### 10.4 Summary

In Chapter 10 you have entered and learned how to use the program CONIC to graph a conic section given in its general rectangular form

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

You also saw that the program POLAR from Chapter 9 can be used to graph a conic section given in its general polar form

$$r(\theta) = ed / (1 \pm e\cos(\theta)) \quad \text{or} \quad r(\theta) = ed / (1 \pm e\sin(\theta))$$

After going through the examples of producing graphs of conics in this chapter, you should be able to obtain good graphs of the conics you encounter in precalculus.

# CHAPTER 11: VECTORS

**Learning Objectives**

In this chapter you will:

- see how to do the operations of vector algebra using keys on the HP28S.
- enter and use several programs that perform multiple step calculations on vectors.

**11.1 Introduction**

A vector is represented on the HP28S as a *one-dimensional array*, numbers within brackets separated by spaces. For example the vector (2, 3, 4) is entered on the stack with the stroke sequence

[2,3,4 ENTER

Chapter 11 in the HP28S **Owner's Manual** and ARRAY in the HP28S **Reference Manual** explain the many operations that the HP28S can perform on vectors and matrices. In this chapter we look at the vector operations most important for the precalculus student. Some of these operations can be done by single keystrokes on the HP28S. Programs that perform some other vector operations are given and explained in Section 11.3.

**11.2 Vector Operation Keys**

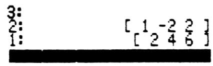
Vector addition is the most basic algebraic operation on vectors. You can add two vectors just as you add two numbers: put both of them on the stack and press + .

As an example place the vectors

(1, -2, 2) and (2, 4, 6)

on the stack with the stroke sequence

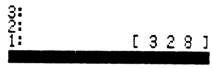
[1,2 CHS,2 ENTER [2,4,6 ENTER




to make your screen appear as shown to the right. Now press

+

to get the sum of (3, 2, 8) to appear on the stack as shown.




Vector subtraction is done similarly, using the - key. Multiplication and division are not defined for arrays. However *scalar* multiplication and division can be done using the * and ÷ keys.

The keys that perform vector functions are found on the third page of the ARRAY menu.

Use the stroke sequence

ARRAY NEXT NEXT

to see the menu items DOT and ABS. These are the only two vector function keys that you will need in precalculus.


The key ABS calculates the length of a vector as in the following example.

**EXAMPLE 11.1 LENGTH OF VECTORS** ***G470**

Find the length || A || of the vector

$$A = (3, 4)$$

Recall that $|| A || = \sqrt{x^2+y^2} = \sqrt{3^2+4^2} = 5$. Perform the stroke sequence

[3,4 ABS

to do this calculation automatically. Think of ABS as a program that squares the entries of A, adds the result, and then takes the square root.

This is a lot of effort to get 5, but ABS works just as well for any numbers x and y. In fact ABS will find the length

$$|| A || = \sqrt{a_1^2+a_2^2+a_3^2+a_4^2+...+a_n^2}$$

of any vector. For example press

[1,2,3,4,5,6,7,8,9 ABS

to calculate

$$|| (1, 2, 3, 4, 5, 6, 7, 8, 9) || \approx 16.88$$ ■

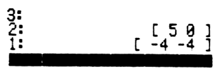
In the next example we turn to DOT that calculates the dot product of two vectors.

**EXAMPLE 11.2 DOT PRODUCTS *G477**

Use the stroke sequence

[5,0 ENTER [4,4 ENTER CHS

to place the vectors (5, 0) and (-4, -4) on the stack as shown. Now press DOT which puts -20 on the stack.■




The dot product is involved in the formula

$$\theta = \cos^{-1}(A \cdot B/||A|| \cdot ||B||)$$

for the angle θ between two vectors. To calculate the angle between two vectors A and B, you must calculate the lengths of A and B, take the dot product of A and B, perform a multiplication, take a quotient, and finally evaluate an inverse cosine. In the next section we see how to combine all of these operations into a program.

### 11.3 Vector Operation Programs

Suppose you want to find the angle θ between two vectors A and B for many pairs of vectors. The formula

$$\theta = \cos^{-1}(A \cdot B/||A|| \cdot ||B||)$$

that gives this angle takes about a dozen keystrokes, even if A and B are stored as menu items. A program made up of these dozen or so keystrokes can serve as a single key for finding the angle between two vectors.

Two such programs, ANG1 and ANG2, are given below. Enter these programs as shown.

ANG1 begins by putting the calculator in DEG angular mode and asking for two vectors A and B. Upon continuing, this program uses stack operations to calculate the angle between A and B.

```
ANG1
« DEG "A,B=?" HALT
DUP2 DOT SWAP ABS /
SWAP ABS / ACOS SWAP
DROP
»
```

ANG2 begins by putting the calculator in DEG angular mode and asking for two vectors A and B, which are stored under the names A and B. Upon continuing, the angle between A and B is calculated, taking advantage of stack logic.

```
ANG2
« DEG "A,B=?" HALT
'B' STO 'A' STO A B
DOT A ABS B ABS * /
ACOS SWAP DROP
»
```

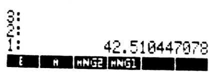
Now you are ready to test these programs with an example.

**EXAMPLE 11.3 THE ANGLE BETWEEN TWO VECTORS** ***G479**

Find the angle between the vectors A=(7,-2) and B=(6,3). Use the stroke sequence

ANG2 [7,2 CHS ENTER [6,3 CONT

and your screen will appear as show. The angle between the vectors is about 42.5°. The CH11 subdirectory now contains the menu items A and B where the vectors A=(7,-2) and B=(6,3) are stored.



In a sense ANG2 is better than a key since it reminds you that it requires two vectors. You might want to modify ANG1 or ANG2 slightly so that the program really behaves like a key. If either A or B is the zero vector both programs (properly) display the message "Undefined Result."▮

The calculation of the distance between a point $(x_1,y_1)$ and a line ax+by+c=0 from the formula

$$d = | ax_1 + by_1 + c | / \sqrt{a^2 + b^2}$$

is another multi-step procedure that is easily automated by combining all the steps into a single program. The program DLP below shows one way to do this.

DLP asks for the five parameters in the formula

$$d=|ax_1+by_1+c|/\sqrt{a^2 + b^2}$$

for the distance between a point and a line and uses them to calculate the distance.

```
DPL
« "ax+by+c=0"
"(x1,y1)"
"a,b,c,x1,y1=?" HALT
→ a b c x1 y1
  « a x1 * b y1 * c
+ + ABS a SQ b SQ +
√ /
  »
»
```

Store DLP to use on the next example.

**EXAMPLE 11.4 DISTANCE FROM A POINT TO A LINE** ***G482**

Use DLP to find the distance from (4,-7) to y = 2x+3.

Write the equation for the line as

$$-2x + y -3 = 0$$

in order to use DLP (as you would do to use the formula directly). Then the stroke sequence

DLP 2 CHS,1,3 CHS,4,7 CHS CONT

will yield the distance 8.049844719 between the line and the point.▮

As a final example of a useful program combining keystrokes we present the program D2 on the next page that finds the distance between two points.

D2 asks for two points and uses ABS to find the length of the vector they determine.

```
D2
«
"(x1,y1),(x2,y2)=?"
HALT - ABS SWAP DROP
»
```

D2 is the "better" program promised in Chapter 1 (p. 6) where the program D1 was given for finding the distance between two points. We close by using D2 to do a calculation.

**EXAMPLE 11.5 DISTANCE BETWEEN TWO POINTS**

Calculate the distance between the points (4,-2) and (7,2) using D2. To do this press the stroke sequence

D2 (4,2 CHS ENTER (7,2 CONT

and the distance 5 appears on the stack.█

**11.4 Summary**

In Chapter 11 you have seen how to represent vectors as arrays on the HP28S. You learned how to do the basic operations of vector algebra using calculator keys. You entered and used the following programs that perform calculations associated with vector operations:

* ANG1 and ANG2 for finding the angle between two vectors.

* DLP for calculating the distance between a point and a line.

* D2 for calculating the distance between two points.

You can be proud of your success in learning to use your graphing-programmable calculator-computer at the same time that you studied precalculus. The calculator skills that you developed will serve you well in future mathematics and mathematics based courses. There are many keyboard features that we did not go into in **The Calculator Edge** that you may find useful in the future. Furthermore there is no limit to what you can do with the programming features of your calculator.

# APPENDIX

## A.1 Glossary

Terms introduced in the HP28S **Owner's Manual** and **Reference Manual** are defined in the Glossary of the **Reference Manual**. **Bold underlined** terms in **The Calculator Edge** are defined below.

**algebraic notation** The common mathematical notation (as opposed to RPN) of high school algebra that uses parentheses and hierarchy of operations.

**angular mode** (or **angle mode**) The calculator mode that determines whether angles are to be understood as measured in degrees or radians.

**argument** The number (or other object) on which a function operates.

**command line** The input line that ENTER sends to the stack.

**complete graph** A graph that shows all the significant features of a function.

**compound interest formula** The exponential function relationship between time, interest, present value, and future value that defines compound interest.

**cross cursor** The cursor activated when a graph is displayed.

**cursor keys** The same as the six *menu item* keys. When the calculator is in cursor mode, these keys can be used for editing programs and locating points on a graph.

**default plot parameters** The default information for graphs that includes the lower left and upper right viewing screen corners of (-6.8, -1.5) and (6.8, 1.6) respectively.

**direction keys** See **cursor keys**.

**editing mode** The command line mode that allows editing and does not allow immediate evaluation.

**error message** A message sent to the screen when normal execution is impossible (such as division by zero or applying + to an empty stack).

**function** In **The Calculator Edge** *function* has the same meaning as it does in your precalculus text.

**menu items** Programs, variables, constants, functions, and directories that are activated by the six keys in the top row of the right hand keyboard.

**numerical iteration** A process that yields numbers that get closer and closer to a solution to a problem.

**operation key** A key that performs an arithmetic operation (for example, the + key).

**page** Menu items that appear together on the screen. For example press STAT NEXT NEXT to see three pages of the STAT menu.

**pixel** A single indivisible picture element. The calculator approximation to a point.

**pixel grid** The 137 by 32 grid of pixels that make up the screen of the HP28S.

**plot parameters** The information stored under PPAR determining the position, scaling, and center of a graph as well as the name of the independent variable.

**program** A menu item combining a sequence of calculator operations in a single unit.

**rescale** To change the scale of the axes.

**round to pixel accuracy** To replace a "point" by the pixel (really a collection of points) that covers it.

**RPN** or **Reverse Polish Notation** The mathematical notational convention of having a function follow its arguments, for example "x SIN 3 +" to mean "sin(x) + 3". The Stack Logic of the HP28S is designed to follow RPN, as opposed to *algebraic notation*.

**shift key** The red key that changes the action of a key to the operation printed above the key in red.

**size the screen** To adjust the plot parameters in order to get a more suitable representation of a graph.

**stack** The series of objects "stacked" on level 1, level 2, etc. that follower the "first in, last out" logic of the calculator operations.

**stack logic** The convention of operators taking arguments from the stack on a "last in, first out" basis and returning the result to the stack. For example ÷ divides the object in level 2 by the object in level 1 and puts the result in level 1.

**subdirectory** A menu item which is itself a menu.

**variable** A name that is allowed to take on different values.

**viewing rectangle** The x and y values that cover the screen used for a graph. For example the default plot parameters correspond to the viewing rectangle [-6.8, 6.8] by [-1.5, 1.6].

**zoom in** To change the plot parameters to give a smaller viewing rectangle.

**zoom out** To change the plot parameters to give a bigger viewing rectangle.

**A.2 Index of Programs**

## A.3 Index of Examples

# INDEX

## Q



## R



## S



## T



## U



## V



## W, X, Y, Z