Computer software was developed for use in demonstrating mathematical topics in the classroom. Each program was written to be used as an enhancement for the teacher's usual teaching strategies, and not as a tutorial for the student. The programs were then used for demonstrating topics in college level mathematics courses and the students' reactions were tested with a questionnaire. The use of each program was discussed in detail, intended as a guide for their use. The results of the questionnaire suggest that the majority of the students felt the programs were a benefit and are in favor of the use of a computer in the mathematics classroom.
A STUDY OF THE USE OF MICROCOMPUTERS IN MATHEMATICS EDUCATION

A Thesis
Presented to
the Faculty of the Department of Mathematics
Emporia State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Arts

by
Michael Charles Mosier
December 1985
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Statement of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>Importance of the Study</td>
<td>2</td>
</tr>
<tr>
<td>Procedures Followed</td>
<td>3</td>
</tr>
<tr>
<td>Limitations of the Study</td>
<td>4</td>
</tr>
<tr>
<td>Overview of the Thesis</td>
<td>5</td>
</tr>
<tr>
<td>II. SURVEY OF RELATED MATERIALS</td>
<td>8</td>
</tr>
<tr>
<td>Introduction</td>
<td>8</td>
</tr>
<tr>
<td>Student Motivation</td>
<td>8</td>
</tr>
<tr>
<td>Changes in Teaching</td>
<td>10</td>
</tr>
<tr>
<td>Changes in Learning</td>
<td>12</td>
</tr>
<tr>
<td>Classroom Uses of the Computer</td>
<td>14</td>
</tr>
<tr>
<td>The Lack of Software</td>
<td>15</td>
</tr>
<tr>
<td>III. RESULTS OF STUDENT REACTION QUESTIONNAIRE</td>
<td>17</td>
</tr>
<tr>
<td>Introduction</td>
<td>17</td>
</tr>
<tr>
<td>Question One</td>
<td>17</td>
</tr>
<tr>
<td>Question Two</td>
<td>18</td>
</tr>
<tr>
<td>Question Three</td>
<td>19</td>
</tr>
<tr>
<td>Question Four</td>
<td>21</td>
</tr>
<tr>
<td>Question Five</td>
<td>22</td>
</tr>
<tr>
<td>Question Six</td>
<td>23</td>
</tr>
<tr>
<td>Question Seven</td>
<td>23</td>
</tr>
<tr>
<td>Question Eight</td>
<td>24</td>
</tr>
<tr>
<td>Item Nine</td>
<td>25</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>Summary</td>
<td>26</td>
</tr>
<tr>
<td>IV. OVERVIEW OF USING THE PROGRAMS IN THE CLASSROOM</td>
<td>28</td>
</tr>
<tr>
<td>Introduction</td>
<td>28</td>
</tr>
<tr>
<td>Algebra Programs</td>
<td>29</td>
</tr>
<tr>
<td>Point Plotting</td>
<td>29</td>
</tr>
<tr>
<td>Graphing Linear Equations by the Point Plotting Method</td>
<td>30</td>
</tr>
<tr>
<td>Slope</td>
<td>31</td>
</tr>
<tr>
<td>Slope Intercept Form of a Line</td>
<td>32</td>
</tr>
<tr>
<td>Point Slope Form of a Line</td>
<td>32</td>
</tr>
<tr>
<td>Parabolas</td>
<td>33</td>
</tr>
<tr>
<td>Circles</td>
<td>34</td>
</tr>
<tr>
<td>Ellipses</td>
<td>34</td>
</tr>
<tr>
<td>Hyperbolas</td>
<td>35</td>
</tr>
<tr>
<td>Trigonometry Programs</td>
<td>36</td>
</tr>
<tr>
<td>Law of Sines</td>
<td>36</td>
</tr>
<tr>
<td>Graphing Trigonometric Functions</td>
<td>38</td>
</tr>
<tr>
<td>Polar Coordinates and Equations</td>
<td>40</td>
</tr>
<tr>
<td>Calculus Programs</td>
<td>41</td>
</tr>
<tr>
<td>Limits</td>
<td>41</td>
</tr>
<tr>
<td>Derivatives</td>
<td>43</td>
</tr>
<tr>
<td>Integration</td>
<td>44</td>
</tr>
<tr>
<td>V. SPECIFICS IN THE USE OF EACH PROGRAM</td>
<td>46</td>
</tr>
<tr>
<td>Introduction</td>
<td>46</td>
</tr>
<tr>
<td>Items Common to All Programs</td>
<td>46</td>
</tr>
<tr>
<td>The Program Lines</td>
<td>50</td>
</tr>
</tbody>
</table>
LIST OF TABLES

TABLE                   PAGE
I. Results of the Reaction Questionnaire ........ 27

LIST OF FIGURES

FIGURE                   PAGE
1-A. Slope Portion of the Program Lines ........ 94
1-B. Point-Slope Portion of the Program Lines .... 94
2-A. Ellipse Portion of the Program Conics ....... 94
2-B. Hyperbola Portion of the Program Conics .... 94
3. The Program Function ......................... 94
4-A. Two Angles and Side Portion of the Program Sine-law,
    First Stage .................................. 95
4-B. Two Angles and Side Portion of the Program Sine-law,
    Second Stage ................................. 95
5-A. Two Sides and an Angle Portion of the Program Sine-law,
    First Stage .................................. 95
5-B. Two Sides and an Angle Portion of the Program Sine-law,
    Second Stage ................................. 95
6. Cosecant Portion of the Program Triggraf ....... 95
7. The Program Polar ............................... 95
8-A. Explanation Portion of the Program Limits .... 96
8-B. Epsilon-Delta Graph Portion of the Program Limits .. 96
9-A. Explanation Portion of the Program Derivative .... 96
9-B. Graph Portion of the Program Derivative ........ 96
10-A. Upper Riemann Sums Portion of the Program Integral .. 96
10-B. Trapezoidal Rule Portion of the Program Integral .. 96
Introduction

When the writer of this paper was student teaching, his responsibilities included teaching two courses in computer programming. Most of those students were also enrolled in an algebra class that he taught. Coincidentally, at one point during the semester, the course content of the two courses centered around the graphing of conic sections. For instance, in the algebra class, the students were learning what effect changing certain numbers in the equations of conic sections had on their graphs. Simultaneously, the computer class was learning to graph the same type of equations on the monitor of a microcomputer, using computer graphics techniques.

As one might expect, the students realized quickly the ease with which their algebra problems in graphing might be accomplished using the programs written in their computer class. Very soon, they had not only mastered the graphing of conic sections but had begun to familiarize themselves with the graphs of other equations as well. They experimented with graphing many types of equations and functions, and in a relatively short time, they gained far more insight into the meaning of functions -- for example -- than they could have accomplished from a routine algebra assignment on graphing.

Statement of the Problem

It became apparent that using a computer to help teach students to graph equations could be very beneficial. The
problem which this thesis addresses was to discuss the ramifications of creating and using microcomputer graphics programs in the teaching of mathematics courses. The content of this thesis will provide the secondary and college level classroom teacher with the programs, and the knowledge of how best to use those programs, needed for implementing computer assisted instruction into their normal course curriculum.

Importance of the Study

The use of a computer as a tool to help the classroom teacher present many of the topics of mathematics has many advantages. For example, it is customary for an instructor to teach the graphing of linear equations by drawing examples on the chalkboard. Although there is nothing wrong with that approach, most such drawings are at least a little crude, and, often, not extremely accurate. Using a computer to draw a line insures the line will be perfectly straight and go through exactly the points that it should. Also, the amount of time taken to accurately graph a linear equation on the chalkboard limits the number of examples that are possible for study in one class period. The computer will graph the equation very rapidly and precisely, allowing time to show many more examples and permitting much more practice. Another important advantage a teacher will gain by using a computer is the opportunity for more student input into the lesson. Having students give the equations to be used in the programs helps keep them involved and interested, and allows the teacher to more closely monitor how well they are
learning. When a teacher uses the same teaching strategies every day, students can easily become bored. Thus, any chance an instructor has to vary his/her teaching style and teaching methods can provide a more interesting session. Students find the computer to be very intriguing. Hence, a topic or a course taught to them using the computer certainly serves to focus their interest. Once the usefulness of the computer in teaching is recognized, this study will provide the basis for implementing its use. With the programs contained in this thesis, and the discussions of how to best use them, mathematics teachers will have what is needed in order to bring a microcomputer into their classroom.

**Procedures Followed**

As the author began work on this study, considerable time was spent writing the graphics programs. It was important that each program be simple to use. Thus, each one was formatted to use a command menu. The title of each program was chosen to describe the general topic with which it was concerned. The command menu was designed to separate topics into more specific categories. Some programs might conceivably be used every day for a week, using a different choice of the menu each time.

Most of the graphics techniques used in the programs were fairly simple, allowing for other individuals to modify them to suit their particular needs. The author devised a method to allow for a function or equation to be changed, as many times as desired, during program execution. Each program was designed to
maintain careful balance between versatility and a controlled number of input statements. It was found that, making the programs flexible enough to accept a wide variety of changes forced the user to deal with too many input statements, which slowed the presentation and dulled its effectiveness. The original function in each program, located on the disk, was determined to be the best function with which to begin the program.

After the programs were written, the author spent approximately one year using them in actual classroom situations. A questionnaire was devised, appropriate to the study, which was used to test student reactions to the use of the computer in the classroom. A form of the questionnaire was completed by each student who viewed one of the computer demonstrations. The information gathered through the use of this questionnaire was used to evaluate the usefulness of the graphics programs and is compiled in chapter three.

Limitations of the Study

The writer chose for this study graphing topics primarily related to the fields of Algebra, Trigonometry, and introductory Calculus. Obviously, such topics as three-dimensional graphing could have been considered, but in the spirit of confining discussion to the three mentioned fields, such topics were necessarily excluded.

Most of the programs contained herein, as was alluded to before, were tested before an actual class. Some, however, were not. The untested programs covered topics that the author had
not had the opportunity to teach, either in his own classes or in those of other teachers. However, this is not considered a major limitation, and one should not construe that those programs were inferior, since each one was run many times by the author, using many different test modes.

The questionnaire, designed to test student reactions to the programs, was very brief and somewhat general. When class time must be used to complete questionnaires, there simply cannot be an excessive number of questions, and all questions must be designed so that answers are short. However, since the main thrust of this study was to develop the programs and explain their use, this is not considered a major limitation.

The thesis does not include any instruction as to the basic use of computer systems. It was assumed by the author that any materials needed for this type of instruction would be readily available, and was therefore excluded. Users of the programs are required to know only how to turn the computer on, load the program desired, then how to run it.

As a final limitation, it should be noted that these programs were written specifically for the IBM-PC and IBM-PCjr. As a result, the disk provided will not run on a computer that is not IBM compatible. In order to use the contained programs on any other computer, one must code in the programs from the listings provided, making any necessary changes in syntax.

Overview of the Thesis

Chapter one is a chapter that provides the reader an
introduction to the study. The second chapter is a survey of literature related to the study of computer assisted instruction, and to computer graphics techniques in general. Chapter three contains the results of the questionnaire on program effectiveness, which was completed by every student who observed a demonstration using one of the programs. The actual statistics are listed, then an interpretation, followed by a sampling of comments written by the students pertaining to the use of the programs as a teaching aid.

In chapter four, a detailed discussion of each of the programs is presented. This discussion includes the standard method of presenting each topic, without the use of a computer, and leads into how the computer might be added to enhance that presentation. The chapter offers statements to provide the instructor with some idea as to when the best time is to present a computer demonstration, according to the usual order of the mathematics curriculum.

The fifth chapter provides the technical portion of the thesis. It is here that the actual programs, with step-by-step directions for their use, are discussed. The way in which input statements should be handled, and the best numbers and functions to enter in each program, are important topics in this discussion. The chapter deals with each program stored on the diskette separately. Included in the discussion of each program, are some examples of how text, equations, and input statements appear on the screen of the computer.

The sixth and final chapter is a summary of the thesis.
Conclusions are made, as well as any additional recommendations for instructors using these programs in their classrooms. In addition, some other topics that could be very nicely presented with the use of a computer are suggested for further study.

Following this chapter, is the bibliography, and then two appendixes. The first appendix contains figures that illustrate how the screen of the computer will appear at selected times during the execution of each program. The second appendix contains the program listings for each of the programs designed for this study and found on the provided diskette.
CHAPTER II
A SURVEY OF RELATED MATERIALS

Introduction

Throughout the available reading materials on Computer Assisted Instruction, several themes are most prevalent. Nearly all publications will devote some of the writing to dealing with the added student motivation that is possible with the use of a computer in the classroom. Also two very important topics that occur quite frequently are the changes, or lack of changes, in the teaching style and methods used by instructors who present topics with a microcomputer, and changes in the ways that students learn and retain new material. Any writing about Computer Assisted Instruction lists many new uses of the computer in the classroom; from using it as a gradebook, to using it to illustrate topics in front of a class. Another point of frequent discussion in the literature was the serious lack of available software written for use of the computer as a teaching aid. These are the topics that most often recur in reading on Computer Assisted Instruction, and are the topics of concern here.

Student Motivation

Every teacher, whether it be in mathematics or in any other subject, is very much concerned with student motivation. The very brightest student, who is capable of accomplishing a great deal, will often be satisfied to produce only minimum effort, if not motivated otherwise. The subject of student motivation is an important one when considering the use of Computer Assisted Instruction.
The 1984 Yearbook of the National Council of Teachers of Mathematics, entitled *Computers in Mathematics Education*, dealt with this subject a great deal. In this book, the computer was said to be a "provocative, motivating device" (4,4) that is available for use in today's mathematics classroom. It went on to say, "Students and teachers alike become excited and enthusiastic about the computer as a new tool in the classroom." (4,4)

One author compiled a list of the benefits that a mathematics teacher receives by integrating the use of computer software into classroom curriculum. Among those benefits is, "Computers can stimulate student learning." (4,13) The use of computer graphics, the level of active student participation, and the excitement caused by the challenge of gaming type programs, were given as some of the main contributing factors to the added motivation.

To understand the ways in which computer graphics can increase a student's level of motivation, one needs only to observe a class being presented a lesson with the use of moving graphics. For whatever individual reasons, every student watches the screen of the computer. Alfred Bork, in discussing the use of computer graphics in teaching, wrote, "Visuals have long been recognized as an important factor in motivating people. We cannot afford to neglect such motivation in student environments." (1,29) His point is not just that the computer would be beneficial to use, but that the computer is essential.

Later in his book, Mr. Bork mentions the difference between a passive and an active learning medium. A passive medium, such
as reading, offers no student interaction and no real built in motivational factors. On the other hand, an active medium, such as a computer program, offers moving visuals, and quite often direct student interaction. He writes, "The first advantage, not always realized, is the level of interaction possible." (1,247)

Another factor in the motivation of students, was discovered by the author of this thesis, in the student responses to the questionnaire concerning the computer demonstrations. Many of the comments written by students who had been exposed to Computer Assisted Instruction, mentioned the fact that the computer represents a new and different method for them to learn, which captured their interest. Any time a teacher varies the method of presentation, the possible monotony of class pace can be broken, and the students are motivated by the very act of change. The following chapter includes comments illustrating this point.

Changes in Teaching

The use of a computer in the classroom can cause some changes in the methods used to teach mathematics. One article suggests that the use of software as an in-class teaching aid, will "create an environment for three-way interaction: teacher, students, and microcomputer." According to the article, software can, among other things, "enhance their (teachers) capabilities in presenting new material, and encourage them to use effective teaching strategies." (4,89)

There are some who feel that the increasing use of the computer as a teaching aid is somehow an attempt to replace the
teacher. Of course, this is not the case, since classroom software is intended only to increase the capabilities of the teacher. Warning that the computer is not a panacea, an add-on, or a replacement for the teacher, one article suggests, "Despite the cries of those who fear that education will be dehumanized, the computer will, in fact, only enhance the role of the teacher." (4,13) One author had this to say, "Teaching is still teaching whether done by computer or any other device." (1,12)

As a discussion concerning the potential changes in teaching methods, the following comment from the opening article in the 1984 NCTM Yearbook is offered:

Mathematics teachers stand at the threshold of a new era in education. Imagine how you might have felt if, as a teacher, you had learned of a technological breakthrough called a printing press, which would become an easy, inexpensive, reliable, comprehensive, graphical, dynamic instrument for transmitting information and stimulating learning. Computing technology constitutes such a revolution. (4,9)

Each of the adjectives given above for the printing press, of course, applies equally well to the microcomputer.

The same article suggests that the ways in which mathematics is taught may change, and that the computer will become an instructional tool that will "put the mathematics teacher into deep, personal interactions with the student." (4,5) Robert Burke, the author of CAI Sourcebook, offered this opinion, "I think CAI is one of the most elegant alternatives to traditional methods we have." (2,3)

It is a common theme throughout the reading on CAI, that the modern mathematics teacher needs to make some changes. The use
of the computer as a teaching aid is not presented as a device to be used by those teachers who desire it, but instead as an essential tool that every teacher of mathematics needs to use, for the benefit of the student. "Mathematics teachers need to study state-of-the-art instructional computing -- from the point of view of both the student and the teacher; ... " is how one author worded it. (4,9) Another wrote, "If computers are to have a place in the mathematics classroom, then they should be used as teaching tools when appropriate for what is being taught." (4,135) In the journal for the NCTM, Mathematics Teacher, it was written that "Geometric concepts, including the use of computer graphics, should be integrated into entry-level courses wherever appropriate." (5,373) The same article reports that teachers should begin immediately to deal with the impact of the microcomputer on the content of all mathematics courses.

Changes in Learning

Also in the article just referred to, was a comment concerning the impact of the computer on the way that students learn. The author remarks, "Computers and other elements of information technology are changing in fundamental ways both the mathematics that it is important to learn, and the ways that mathematics, traditional or otherwise, can be learned." (5,373) Being able to see a concept presented with computer graphics has many advantages over traditional methods of teaching the same concept. Learners receive the same information, but in a different manner.
In Learning With Computers, it was written that computer graphics offers an alternate method of conveying information to the learner, "different from the manner information is conveyed through words or numbers." (1,29) The difference between the textbook and computer is very profound. When a topic is presented on a computer screen, the entire text or the complete picture does not need to be displayed at once. For example, in a trigonometry textbook, the diagram used to illustrate solving a triangle by the law of sines, when two triangles are possible, appears as the complete figure. The triangle is on the page, with side of known length that is opposite the known angle already appearing in two places, showing both triangles simultaneously. Customarily the instructor will explain how that side is allowed to "swing" through an arc, touching the base of the triangle whenever a triangle is possible. This arc is usually drawn on the diagram as well, in its entirety. Using a computer, the same diagram is drawn, but the students can actually see it being formed. The side is visually moved through the arc, showing precisely how it is possible to have more than one triangle, and then each triangle may be dealt with separately. The instructor controls when each step of the diagram is constructed, allowing for pauses and discussion. With the textbook drawing, "...there is no sense of information evolving in time to aid the student in understanding the phenomenon." (1,32)

The computer programs written to aid in the instruction of mathematics almost always parallel the conventional teaching strategies and textbook outline, but the information is given in
a method more likely to facilitate student understanding. One researcher of the subject stated, "Students exposed to this type of computer-enhanced approach are likely to witness dynamic illustrations of traditionally developed theory and technique." (3,6)

It is often the case that students form a bias against using graphics techniques for the solution of certain mathematical problems, such as the solution of a system of two linear equations, or the zeroes of a polynomial function. The lack of accuracy in the graphs and the difficulty of creating the graph itself make this method of solution seem unworthy, and the students will often dismiss it before any real understanding of the concept is reached. "Computing technology offers promise of changing this style of operation to a bias in favor of graphics." (3,6)

**Classroom Uses of the Computer**

Programs are continually being created that allow more uses of the microcomputer in the classroom. A teacher can use a program to demonstrate an idea, which will generate examples, questions, and summaries. In the CAI Sourcebook, the statement is made that, "The machine can be made to produce such displays accurately, and flexibly." It goes on to say that, "The teacher needs to integrate such demonstrations into the overall development of instruction by stimulating, guiding, pacing, and directing." (2,3)

In 1980, Robert Taylor classified the instructional roles of
the computer into three main categories -- tutor, tool, and tutee. Most of the software that has been created has been for using the computer as a tutor for the individual student. This means the student needing additional help goes one on one with the computer, using a tutorial program. The computer as a teaching tool, which is the thrust of this thesis, involves the teacher using the computer in front of an entire class at one time, just as one might use the chalkboard. As a tutee, the computer is "taught" by the students. In other words, the students learn mathematics concepts by programming them into a computer.

In discussing the use of the computer for teaching calculus, Kathy Heid points out that graphing functions on a computer screen is the easiest, and most appealing enhancement of calculus teaching. She stated:

With computer relief from the tedium of plotting and connecting \((X,Y)\) points, the teacher in a classroom demonstration, or the student in an individual exploration, can focus on the interpretation of a curve's intercepts, asymptotes, slopes, or concavity. (3,62)

This author continues with a discussion of the opportunity the computer provides for increasing student understanding of the limit concept. The computer allows for a much more vivid demonstration of the epsilon - delta definition of the limit, a typically difficult topic for students to understand. (3,62)

The Lack of Software

As was alluded to before, most of the software that has been created for mathematics instruction, has been for the use of the
computer as a tutor. The shortage of software for use of the computer as an in-class teaching aid is well documented. At a 1975 conference of the AMS/MAA, Stephen Garland asked, "Why, given the great number of experiments using computing to teach calculus, and given the attention devoted to the subject in journals and conferences, hasn't there been more widespread acceptance and use of this approach to calculus?" He cited that limited computing facilities, including the lack of suitable software, was a major factor. (3,61) In the NCTM Yearbook for 1984, it was stated that despite many proposals and much developmental effort, the methods of teaching mathematics in the secondary schools are remarkably similar to those of thirty years ago. (4,20) In another article in the same yearbook, it was written that, "Most of the software currently available for CAI in schools has been developed for use by a single student interacting with a single terminal." It goes on to say that software should be created for the teacher to use in the classroom, since "teachers remain the best source of student learning, irreplacable by any computer." (4,89)
CHAPTER III
RESULTS OF REACTION QUESTIONNAIRE

Introduction

In teaching these courses -- Introduction to Algebra, College Algebra, Trigonometry, and Calculus I -- and appearing as a guest in other Calculus classes, the writer presented a total of 21 computer demonstrations to enhance the topic being studied. For nearly every computer demonstration given, an accompanying questionnaire was distributed among those present. This questionnaire was designed to determine the overall reaction the students had concerning both the value of that particular demonstration, and the use of Computer Aided Instruction in general.

This chapter is a question-by-question treatment of the results of the questionnaire. Each question was to be answered in two ways, first by the participant writing a brief comment, then also by circling a number between one and five, with one being the lowest, and five being the highest. The results are listed by first giving the question, then the total number who responded to that question, the average of all the numerical answers, and a sample of the comments written.

The Questions

Question One. The first question the students were asked to answer dealt with their prior knowledge of the topic demonstrated to them. The answer given to this question had possible effects on the answer to the next three questions, so they are discussed with that concern. For the most part, unless a student was repeating a course, or was in some other way ahead
of the rest of the class, prior knowledge was rated fairly low.

The question read as follows: Rate your prior knowledge of
the subject being demonstrated. A total of 95 people responded to
this question, and the average of the responses was 2.35. The
indication is, as would be expected, that most students were
learning the topic for the first time. As for comments written
on the first question, if any were written, they were usually a
list of background courses taken, or a statement in direct cor-
relation to the number that was circled.

**Question Two.** The second question was as follows: Do you
feel the demonstration gave you an adequate understanding of the
subject, as opposed to the instructor using just the text? As
was mentioned before, the way in which the students rated their
prior knowledge of the subject seemed to have some effect on the
way they answered this question. A total of 78 students gave a
numerical answer to this question, and the overall average answer
was 4.26.

The following is a breakdown of the averages of the answers
for each of the five possible ratings for the first question.
First, for those rating their prior knowledge as a 1, there were
26 who responded to this question, with 3.48 being the average.
There were 18 students who rated themselves at 2, and then re-
responded to the second question. The average of these 18 was
3.83, a value somewhat higher than the last. Of those who rated
themselves at 3, there were 23 who answered question two, with
the value for the average being 3.91. For a rating of 4, there
were 11 students that responded, and their average response was
4.5. There were no students that rated themselves at level 5, who answered the second question. Obviously, there was a definite increase in the average response, as the rating for prior knowledge increased.

Of the comments that were written concerning this question (and the others), there were mixed reactions. The number of positive, negative, and indifferent comments directly relates to the average of the numerical results, but only a few of each are presented here.

There were not any decidedly negative comments on this question, but one student remarked, "I saw no difference between the text and the demonstration." Another student wrote, "I think it helped me understand it, but by using the computer and the text, an even better understanding could be reached."

Others indicated that they preferred the use of the computer, by writing such comments as, "Moving pictures are always better than a textbook." Several students quoted an old saying, and wrote, "A picture is worth of thousand words!" Another positive reaction was, "A few minutes of demonstration can replace a week of wading through textbook examples."

**Question Three.** The third question essentially asked the same information as the second, but from a different viewpoint. The question was: Did the program, along with the instructors help, seem to do a good job of explaining the subject? Instead of rating the understanding they received, the students were rating the performance of the program, and the averages were a little higher. Again, the results are given both as an overall
total, and by the response to the first question.

A total of 94 responses were given to this question, and the average of those responses was 4.35. If students rated their previous knowledge as the lowest, a 1, the average was 4.21, with 28 being in the category. With a rating of 2, there were 23 students, for an average of 4.39. Of those who rated themselves at level 3, there were 26 who responded to this question, giving an average of 4.19. For level four, the total responses was 14, and that average was 4.78. There were only three students who rated their prior knowledge at a 5, and their average answer to question three was 4.67. Unlike the second question, the results seemed to be independent of the students response to question one.

In response to this question, concerning the program on limits, one student had this to say, "Yes, but I was confused about where they got delta." Apparently, the instructor failed to give an explanation concerning how the computer calculated that value, and the student focused on that problem. Another negative comment was, "It was sort of complicated." It is unknown whether this person was referring to the topic, or the way it was presented on the computer, but either interpretation might imply dissatisfaction.

One student, who seemed to have no preference in the method that was used, commented, "Yes, although the text was also helpful to me." A more positive comment was, "Yes, it was more clearly illustrated." Two comments made reference to the input from the teacher, along with the computer program. The first
wrote, "Yes, the chalkboard example before made the CRT (computer) equations very easy to follow." The other concerned
the instructor's help during the demonstration, saying, "Yes, we
needed certain questions answered as it was demonstrated."

Question Four. The fourth question on the questionnaire
was: Try to rate your knowledge after the demonstration, as com-
pared to before it. This question was intended to test the
amount of improvement the students felt they achieved, and for
the most part seemed to accomplish that. There were, however, a
few cases where students were apparently confused, since they
rated their knowledge after the demonstration as lower than
before it. Hopefully the assumption can be made that these
students circled the number that described their increase, and
not their overall knowledge, since to assume otherwise would
imply the demonstration subtracted from their knowledge. There
were four students who answered in this manner, and fourteen that
indicated their knowledge the same before as afterward.

For the group that rated their knowledge before as 1, there
were 29 that responded to the fourth question, and the average of
their answers was 3.83. Those that rated themselves as 2 before
averaged a rating of 4.05 after the demonstration, with a total
of 22 responding. Of the students that rated prior knowledge at
level 3, 25 responded for an average post-rating of 3.80. A
rating of 4 on the first question produced a group of 15 that
responded, and the average was 4.27. Of the three people that
rated their prior knowledge at 5, 2 responded to the fourth
question. One of the answers was at level 3, while the other was
at level 5, giving an average of 4. All together, there were 93 responses to the fourth question, and the overall average rating of their knowledge after the demonstration of 3.95, as compared to the overall rating of 2.35 before the demonstration.

Of the comments written, there were none that were completely negative; however, a few expressed concern that the demonstration alone was not enough. Among those comments was, "It helped me, but I need more hands on work, like with the exercises in the book." Another wrote, "Since it was such a new subject to me, I was still a bit confused."

Several students wrote comments concerning how the demonstration helped them. One such reaction, which concerned the Law of Sines program, was, "It helped me to actually watch the triangles form." Writing about a demonstration of parabolas using the Conics program, one student commented, "I learned how different parts of the equation change the thickness and whether the curved line goes (opens) up or down." Another student compared it to the text, saying, "I realize now what the book was trying to explain.

Question Five. From the fifth question on, the questions were designed to get student reactions to the use of Computer Aided Instruction in general, beginning with: Do you feel the use of the computer was interesting? A total of 94 students gave numerical responses to this question, and the average response was 4.67. Of the five possible answers, two students answered with 3, and everyone else responded higher.

Of the comments written pertaining to this question, only
one was negative. That response was, "Yes, but I'd rather see problems written out on the board." Of the other comments, several mentioned the difference in teaching technique. One such comment was, "It seemed to capture everyone's attention, opposed to listening to a teacher lecture." Another student wrote, "It was new and different." Similar to that was, "A little variety never hurts." One student liked seeing the concept presented on the computer, writing, "Yes, one can see the math at work this way." Another mentioned the computer in this way, "Yes, the graphics are very helpful."

Question Six. Question number six asked the students to react to the presence of the computer, asking: Did you feel alienated at all by the computer? A total of 87 people offered a numerical answer, and the average answer was 1.69. There were six students that answered with a 4, but most answered with 1.

Some of the comments received on this question indicate there might have some confusion as to what was being asked. It was intended that they answer pertaining only to their feelings during the demonstration, but many seemed to answer with respect to computers in general. Typical of these comments were, "Yes, since I have no experience with them," and, "No, I've had two programming classes." One student had this remark, "Yes, it takes the instructor away from the class." A more positive comment was, "No, it's just an electric chalkboard." There were many answers such as, "A little bit," or, "No, not at all."

Question Seven. The seventh question was: Would you like to see a computer used in the classroom again, both in this class
and classes you may have in the future? The overall average answer was very positive, at 4.46. Of the 92 people responding, there were none that answered with a 1, and only two that answered with a 2.

Several of the comments received were nondecisive, such as, "It depends on the subject matter." One student indicated having no opinion, with, "It really doesn’t make much difference to me." Others said they would like to see it used again, provided some conditions were met. Of those, one wrote, "Yes, in classes where it would help." Another was, "Yes, as long as there is a teacher."

There were two decidedly negative comments, one of which the meaning is not entirely clear. That comment read, "Not on a regular classroom atmosphere." The other negative reaction was, "No, I found it more distracting than helpful."

The positive comments included, "I can think of a lot of classes I’ve already taken where a little computer demonstration would be practical." Another was, "Yes, since computers are tools of the future." One student remarked that, "All classes should." Concerning the trend in teaching technique, one comment was, "I feel education is going that way."

**Question Eight.** The eighth, and final question, on the questionnaire, concerned using the computer as an out of class tutor. It read this way: How do you feel about being allowed time to use the computer alone, needing no prior computer experience, with a tutorial type program? Again, the average response was above 4, at 4.25, with 87 students giving numerical
answers. Four students answered this question with the lowest value of a 1, while two answered with a 2, and the rest answered higher.

A few of those that indicated they did not want to use the computer as a tutor, wrote comments pertaining to this. One such reaction was, "Computers are not teachers, they can't tell why something happens, just how." Some of the other negative comments indicated that the students involved misunderstood that no computer literacy would be required beforehand, writing, "I have no knowledge of computers," and, "I wouldn't know what to do."

One comment was neither for nor against the idea, saying, "Yes, but it's not really necessary." The rest of the comments were positive, of which a typical reaction was, "I would like that very much." Similar to that was, "Yes, I think it is a great idea and it helps." One student indicated that it would be a good idea for a study aid before tests, and another wanted it as an aid for homework, writing, "For out of class assignments, and possibly even in class."

**Item Nine.** The last item on the questionnaire was not a question, but simply asked for any additional comments and suggestions. Nearly every student who completed a questionnaire wrote some sort of a comment, and quite a few suggestions were made. The most common suggestion for improvement was typical of this comment, "It really needs to be used on a bigger screen." A standard computer monitor was used, in all cases, and for the larger classes, the students in the back apparently could not see well enough. There were no reactions indicating completely
negative attitudes, but one student remarked, "I am not in favor of someone who does not know what they are doing using it as an instructional aid." This student may have been concerned that the computer was going to replace the mathematics teacher, which was also a concern of the student who wrote, "I felt it was a good demonstration, and it could be used as an aid to the instructor, but not as an instructor itself." One student made these observations, "I think the computer is a fantastic way to demonstrate mathematics. It can give examples quicker and with more accuracy (particularly with graphs) than the instructor can on the blackboard." Another very favorable comment was, "It's interesting to watch the computer draw different lines as different variables are substituted for X and Y coordinates and Y intercepts."

Summary

All of the results given above are summarized in the Table I on the following page. The answers to the first question, which was to rate prior knowledge of the subject, are used to categorize the results of the next three items, as was done in the preceding narrative.
## TABLE I

SUMMARY OF NUMERICAL RESPONSES TO QUESTIONNAIRE

<table>
<thead>
<tr>
<th>Questionnaire Item</th>
<th>Number Responding</th>
<th>Average Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td>95</td>
<td>2.35</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rating on item</td>
<td>Number</td>
<td>Average</td>
</tr>
<tr>
<td># 1</td>
<td>26</td>
<td>3.48</td>
</tr>
<tr>
<td># 2</td>
<td>18</td>
<td>3.83</td>
</tr>
<tr>
<td># 3</td>
<td>23</td>
<td>3.91</td>
</tr>
<tr>
<td># 4</td>
<td>11</td>
<td>4.50</td>
</tr>
<tr>
<td># 5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>78</td>
<td>4.26</td>
</tr>
<tr>
<td>Question 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rating on item</td>
<td>Number</td>
<td>Average</td>
</tr>
<tr>
<td># 1</td>
<td>28</td>
<td>4.21</td>
</tr>
<tr>
<td># 2</td>
<td>23</td>
<td>4.39</td>
</tr>
<tr>
<td># 3</td>
<td>26</td>
<td>4.19</td>
</tr>
<tr>
<td># 4</td>
<td>14</td>
<td>4.78</td>
</tr>
<tr>
<td># 5</td>
<td>3</td>
<td>4.67</td>
</tr>
<tr>
<td>Overall</td>
<td>94</td>
<td>4.35</td>
</tr>
<tr>
<td>Question 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rating on item</td>
<td>Number</td>
<td>Average</td>
</tr>
<tr>
<td># 1</td>
<td>29</td>
<td>3.83</td>
</tr>
<tr>
<td># 2</td>
<td>22</td>
<td>4.05</td>
</tr>
<tr>
<td># 3</td>
<td>25</td>
<td>3.80</td>
</tr>
<tr>
<td># 4</td>
<td>15</td>
<td>4.27</td>
</tr>
<tr>
<td># 5</td>
<td>2</td>
<td>4.00</td>
</tr>
<tr>
<td>Overall</td>
<td>93</td>
<td>3.95</td>
</tr>
<tr>
<td>Question 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>94</td>
<td>4.67</td>
</tr>
<tr>
<td>Question 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>87</td>
<td>1.69</td>
</tr>
<tr>
<td>Question 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>92</td>
<td>4.46</td>
</tr>
<tr>
<td>Question 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>87</td>
<td>4.25</td>
</tr>
</tbody>
</table>
CHAPTER IV
OVERVIEW OF USING THE PROGRAMS IN THE CLASSROOM

Introduction

One of the important advantages a teacher gains from using a computer to present a lesson is a little variety in the standard classroom presentation. This can not only be important to the student, but to the instructor as well, since teachers are subject to the problems of repetition also. It should be stressed, however, that using these programs will not induce any radical change in teaching strategies, so an instructor new to them will not be forced to rethink old lesson plans. Instead, the computer will be used as a tool to enhance the standard presentation of a topic, and will allow the teacher to present the material in basically the same method, and at the same time, as would normally be done.

In this chapter, each of the programs will be discussed from the standpoint of the topic involved, and how the program will relate to the usual presentation of that topic. The standard method of presenting a new topic to students is the main thrust of this chapter, showing how the computer might be used to enhance that presentation, as opposed to changing it. The chapter will not be a step-by-step approach to implementing the programs, as that will be done in the next chapter, but instead will describe how the instructor can most easily fit the computer demonstrations into his/her normal curriculum and teaching style.

In order to induce a grouping of these programs, they have been separated into three main categories. The first of these
categories contains all programs pertaining to topics taught in the algebra sequence, such as graphing lines, parabolas, and functions. The second category groups the programs written for trigonometry, and includes such topics as graphing the six trigonometric functions, the law of sines, and polar coordinates and equations. The final category contains the programs that deal with topics taught somewhere in the calculus series, most notable of which are the programs on limits, derivatives, and integration. The remaining part of this chapter, then, is separated as described into these three categories. Each of the three sections will then be subdivided for consideration of each program separately. The sections come in the order that was listed above, and begins with the programs written for algebra.

**Algebra Programs**

Included in the category of programs written for algebra, are three programs. The first of these is the program entitled "Lines," which begins as an introduction to graphing on the Cartesian Coordinate system. The second program is "Conics," which is used to teach graphing each of the five conic sections. The third program, called "Function," will graph more complicated functions.

**Point Plotting.** The first step in teaching algebra students to graph equations in the X-Y plane, is defining this plane for them. They will most likely be familiar with the number line, which is simply a one-dimensional graph. So, it is customary to begin the discussion there. Explaining how the Cartesian Coordi-
nate system is nothing more than two perpendicular number lines, where the second line adds the second dimension, gives them the definition in familiar terms. Of course, if a point is to be plotted in two dimensions, it must have two coordinates, one for the horizontal component and one for the vertical. By this time in the lesson, most teachers will have the axes drawn on the board, and begin giving examples of how to plot points. Letting the students suggest coordinates, or telling where a point with given coordinates will be, is a good way get them involved.

All of the above strategies can be accomplished in a very similar manner, using the program "Lines." The first choice on the menu of this program, is designed to introduce point plotting. It is recommended that the teacher introduce the topic on the chalkboard first, drawing the axes and discussing them. Then, run the program, and use it to show where points with certain coordinates are placed. The computer draws the X and Y axes, labels them, then draws a grid on the screen very similar to graph paper. Next, the user will enter the coordinates of a point, and the point is plotted on the grid. It is very effective to have one student offer the coordinates of a point, and another predict exactly where that point will be, by pointing at the screen. Then, the instructor types the coordinates, presses enter, and the point will appear. This method allows for considerable student involvement, and helps them enjoy the lesson.

**Graphing Linear Equations by the Point Plotting Method.** For the most part, when the students feel comfortable with the X-Y plane, and do well with graphing points, the next step is to
graph basic linear equations by arbitrarily picking values for X, and computing the corresponding Y value, producing an ordered pair. Several of these ordered pairs are graphed, then connected with a straight line. The author's recommendation is to teach this lesson, and the following one on finding the X and Y intercepts, by using the chalkboard and a straight edge. The computer is too fast to effectively illustrate the process of picking values and calculating ordered pairs. Once the students learn to graph by these two methods, they are ready to learn the concept of slope, which will lead to the remaining methods for graphing linear equations.

Slope. When beginning the lesson on slope, the students need to be given the basic concepts of rise over run, positive versus negative slope, and zero versus undefined slope. Once this is done, the program can be used right away. As it runs, the first thing the computer does is to draw the X-Y plane, with the grid. It then pauses and asks the user to input the first of two points. When it is entered the program graphs it, and the second point is requested. After graphing that one, it begins constructing the right triangle formed by the two points and the lines of the grid. Starting with the vertical leg first, it draws a unit at a time, fairly slowly, so that the viewers can count as it goes. When the triangle is complete, it will ask for the correct slope to be entered. If an incorrect slope is entered, the program erases the answer and asks for it again. This continues until it receives the correct slope, at which time it asks if the user wishes to try another two points. It is ben-
eficial to do several examples of each possible type of slope—positive, negative, zero, and undefined—so the students become well acquainted with the differences.

**Slope Intercept Form of a Line.** Once the students have mastered slope and the concept of intercepts, they are of course ready to graph linear equations in the slope–intercept form. The third choice on the menu of the Lines program first allows the user to input an equation. Then, the graph of the line appears on the grid. This program needs no lead in, and can be used to introduce the concepts involved. The linear equation originally in the program is a good one to begin with, and is graphed by defaulting through the input statements. The instructor can then have them discover, by counting, the slope of the line, and its Y-intercept. If the connection between the slope and intercept on the graph, and the numbers in the equation, is not made, change the equation slightly and graph another. Usually, by that time the students will see how the graph and the equation relate, and they are ready to answer questions relating to these concepts. A good way to use the program at this time is to type an equation, and then have the students give the slope and intercept, before it is graphed. Next, after several examples of this kind, the teacher might have individuals give the equation for a line with a certain slope and intercept.

**Point Slope Form of a Line.** The fourth and final choice on the menu of this program will allow the user to graph lines in the two-point form. The students typically enjoy this program a great deal, since it is designed with some of the qualities of a
game. What it does is to randomly select two points, and place them on the graph. Then, the students must give the proper equation that will draw a line connecting those two points. The instructor types in an equation, which has been suggested by the students, and the computer graphs it. It is obvious at that point whether the equation was correct or not, and the program will ask if it was. If "Y" is entered, it assumes the line was correct, and will then graph two more points. If, however, the line was not correct, the program asks for another try at it. The computer requires that the equation is entered as:

\[ Y = m(X - h) + k, \]

and not as:

\[ (Y - k) = m(X - h), \]

so the students need to be aware of how to use both forms.

Parabolas. The next of the Algebra programs will enable one to graph the conic sections, and typically the first of these to be dealt with is the parabola. The usual teaching strategy for parabolas is to begin by using the point plotting method. Before using the computer to graph parabolas, it is best to start at the chalkboard, and graph \( Y = X^2 \). Once it is on the board, tell the students what it is called, and how all parabolas are simply variations of that one. It is helpful to graph, by hand, a few more parabolas that have some of the variations, and even have the students graph one at their desks. When this is done, the conic sections program will offer the best way for them to practice. The first choice on the menu of this program will allow one to graph parabolas, using the general form of the
equation:

\[ Y = a (X - h)^2 + k, \]

and allowing the user to make changes in the three constants \( a, h, \) and \( k \). The use of the program will teach them to graph parabolas in this form, without having to use the point plotting method, since they can see that each time the vertex of the parabola is at \((h, k)\), and also discover how the number \( a \) affects whether it opens up or down and how "thick" it is. Again, it works very well to have individuals give the equation of a parabola that has its vertex at a given point, and has characteristics of being wide or narrow, opening up or down.

**Circles.** After parabolas, the next concept studied is circles. As with most of the programs, it is best to introduce the topic on the board, using the point-plotting method, starting with something simple like:

\[ X^2 + Y^2 = 4. \]

Usually, this is when the teacher will relate the circle to the distance formula, and then derive the center-radius form of the equation. Once the students are somewhat familiar with this type of equation, they are ready for the computer demonstration. The second choice on the menu of the conic sections program is "Circles," and when it is selected, it will allow the user to change the values of \( h, k, \) and \( r \), in the following equation:

\[ (X - h)^2 + (Y - k)^2 = r^2. \]

The resulting circle is graphed, and then the program waits for further changes.

**Ellipses.** The ellipse part of the program is treated in
basically the same method: starting on the board, introducing the ellipse, then moving to the computer. The program uses an equation of the form that follows:

\[ \frac{X^2}{a^2} + \frac{Y^2}{b^2} = 1, \]

with the user entering the values for \( a \) and \( b \). Many times there will be a student who asks if all ellipses have their center at the origin, which of course they do not. The capability to move the center is obtained by making that choice when the ellipse menu selection has been made. Its equation is of the form:

\[ (X - h)^2 / a^2 + (Y - k)^2 / b^2 = 1. \]

The user will be asked which form of the equation is desired, and the program uses whichever is requested. For the above form, values for the four numbers, \( h, k, a, \) and \( b \), must be entered by the user.

**Hyperbolas.** The final choice on the menu is used to graph the one remaining class of conic sections -- hyperbolas. This portion of the conic sections program will have two sub choices, allowing first the \( Y \) term to be the negative one, then the \( X \) term. Both use an equation of the following form:

\[ \frac{X^2}{a^2} - \frac{Y^2}{b^2} = 1, \]

except the second one has a negative sign before the \( X \) term, and a positive sign before the \( Y \) term. Most students, when first faced with this type of equation, will still be thinking in terms of the ellipse. Using the point-plotting method is a good way to illustrate the difference, because the existence of asymptotes becomes apparent, and at the same time the students can see how there cannot exist a \( Y \)-intercept or \( X \)-intercept, depending on the
equation. The standard technique for graphing a hyperbola of this form involves drawing a rectangle, centered at the origin, with the horizontal sides running from \(-a\) to \(a\), and the vertical sides running from \(-b\) to \(b\). The asymptotes of the hyperbola are the extended diagonals of this rectangle, which are dotted in. Then, the hyperbola is drawn in accordingly. Since the students are taught to graph hyperbolas with this method, the program was written so that it graphs the rectangle, draws the asymptotes, and then graphs the curve itself.

**Trigonometry Programs**

The next set of programs is designed for use in front of a trigonometry class, and covers three main topics. The first of these programs illustrates, with moving graphics, the Law of Sines. The second program graphs variations of the six trigonometric functions, and has a menu choice for each. Finally, the third program deals with polar coordinates and the graphing of polar equations.

**Law of Sines.** Trigonometry textbooks vary greatly as to the order in which topics are presented, so the law of sines program is to be used whenever the course deals with solving triangles. Usually, textbook writers discuss the solution of right triangles, and move from there to study triangles that may be solved using the law of sines. Of these triangles, there are two types. The first, which is the simplest, and the first choice on the menu, is a triangle where two angles and the length of a side opposite one of those angles are given. Then, the
missing angle and the length of each of the two missing sides must be found. When first faced with the prospect of "solving" this triangle, many students do not understand that there can only be one triangle possible, and it should be explained to them that the given angles and side force the triangle to have a certain shape, unique from any other.

After the law of sines has been proved, the class should be ready to view the demonstration using the law of sines program. The program will choose one of the unknown sides to be the base of the triangle, and places it horizontally on the bottom of the screen. The other unknown side will be connected, with one of the given angles, to the left end of the base (the other end has yet to be determined). The side with known length is placed at the right edge of the screen, with its end touching the base at the specified angle. When the teacher presses enter, the side on the right moves, keeping the correct angle with the base, until its top end comes in contact with the side extending up from the left. At that point the only possible triangle has been formed, and the program pauses in execution. Across the bottom of the screen is written the law of sines formula, with a missing side left as a variable. The students then compute the correct answer for this side and the teacher enters it. If the answer is within certain tolerances, the value is placed next to the appropriate side on the drawing. Then the above procedure is repeated for the other missing side. If an answer is incorrect, the entry is erased and asked for again. The last of the missing pieces, the third angle, must also be entered, and once it is correctly
given, the triangle is declared solved. Whenever the program
pauses with a question, every student should be encouraged to
calculate the answer, and then when a consensus is reached, the
teacher enters the value.

The second choice on the menu of the Law of Sines program
demonstrates the case of two given sides, and a given angle that
is opposite one of those sides. Before the program is run, it
helps if the students understand that with this information,
there may be none, one, or two possible triangles. The program
is effective at showing why this is true, but students need to be
aware of the different cases before running the program. As with
the previous case of the law of sines, the side of unknown length
is placed across the bottom as the base. To be consistent with
the diagram found in nearly all textbooks, the program then
places one of the given sides on the left endpoint of the base,
with the one given angle between them. The third side, also of
known length, is connected to the top end of that side, at an
arbitrary angle that prevents it from touching the base. On the
instructor's command, this third side swings slowly down until
its free end touches the base. When this occurs, the computer
beeps, leaves a side ( and thus a triangle ), and continues its
arc until it again touches the base. Once the arc has been com-
pleted, a question will appear asking how many triangles were
possible. When the correct answer is given, the class solves
each possible triangle, one at a time. The program ceases execu-
tion when all possible triangles are solved.

Graphing Trigonometric Functions. The trigonometry curricu-
Curriculum includes several topics dealing with the graphs of the six trigonometric functions. At this point in the curriculum, it is customary to spend some time using the point-plotting method. Trigonometry students should be accustomed to graphing, but often they need to review, and the process of determining ordered pairs will provide that review. Once they have seen the graphs of the sine and cosine functions, the next step usually involves changing the amplitudes, the periods, and finally causing phase shifts, of these basic functions. These topics may either be introduced on the board, then demonstrated on the computer, or introduced on the computer first.

The computer program for graphing trigonometric functions does a fine job of presenting these changes, and allows for quick demonstration of many examples. The program, entitled "Triggraf", has a separate choice on the menu for each of the trigonometric functions. It is suggested that they be presented in the order they appear on the menu -- sine, cosine, tangent, cotangent, secant, and cosecant -- beginning with amplitude changes in the sine and cosine waves. After these have been satisfactorily learned, use the sine wave to illustrate period changes. Next, phase shifts are presented. Finally, combinations of all three are used.

Before the tangent function is graphed using the computer, the students should be reviewed as to what an asymptote is, and when one will occur in a graph, so they are prepared for the dotted asymptotes that the program creates on the screen. The secant and cosecant functions are normally studied last, once the
students are already familiar with period changes and phase shifts.

**Polar Coordinates and Equations.** Since the polar coordinate system will be, for the most part, completely new to the students, it is best to begin this topic on the chalkboard. It is also beneficial to distribute copies of polar graph paper, and then explain how a point with coordinates that express a radius and an angle are plotted on it. The program titled "Polar" can be run following just a brief introduction of the coordinate system, using the first of the two menu choices. This choice asks for the coordinates of a point to be given, and plots that point on a polar coordinate graph created on the screen. An effective way of using this part of the program is to ask individuals to give the coordinates for a point that will be graphed at a specific place on the screen, perhaps a spot where the teacher is pointing. This program will also accept negative angles and negative radii, which should not be neglected in the demonstration.

When the students have learned how to plot points in polar form, they are ready to be introduced to polar equations. The usual teaching strategy, as expected, is to graph one or more by hand on the board, using the point-plotting method. By having the students graph an equation on the paper that was distributed, a teacher can be sure they understand what is involved in the process. After that has been done, the second portion of the polar program may be run, which allows one to graph many polar equations. First, the program allows the equation to be entered.
then, it graphs that equation on the same system it used for plotting points. Since graphing polar equations by hand is very time consuming, students learn what general shapes the different equations produce much faster when the computer is used.

Calculus Programs

The last of the three groups of programs consists of those written for the three main concepts of first semester Calculus. The first of these programs is "Limits," and uses various techniques to present that topic. Second is the program titled "Derivatives," which teaches the derivative as the slope of the tangent line to a curve at a given point. The third program, "Integral," teaches upper and lower Riemann sums, and defines integration in those terms. Also, this program has the capability to graph an illustration of the trapezoidal rule, a method used to approximate the definite integral.

Limits. This program is designed to be used as an introduction of the topic, and can be used with very little pre-discussion from the teacher. The menu on the program has three choices, the first of which is an explanation of limits using moving graphics. The user is told that the function in the program has a value of 3 when \( x = 5 \). Then the axes are drawn, and a dotted horizontal line is drawn, at a height of 3, from the Y axis to the line defined by \( x = 5 \), with a dotted vertical line drawn from there to the X axis. The computer begins graphing the function, from both ends of the screen, which causes the curve to close in on the point \((5,3)\). As this is happening, two small
arrows close in, at the same rate, to 5 on the X axis, while two others converge to 3 on the Y axis. When the two arrows along the Y axis get within .1 unit of 3, the program pauses to compute how close the arrows along the X axis had come to the 5. At that time the teacher should introduce the traditional epsilon - delta process of defining the concept of "limit." By explaining that, if no matter how close one wished to bring the arrows in on the Y value, they could do so by bringing the arrows close enough to the given X value, then that particular Y value must be the limit of the function as X approaches the given value, the teacher prepares the students nicely for the formal definition of the limit.

Not until after the students have been given the epsilon-delta definition of the limit, should the second choice on the menu be used. This choice graphs an entered function, asks at what value of X the user wishes to test for a limit, then asks for a value for epsilon. When that has been entered, the program draws the customary two horizontal lines from the Y axis to the curve, at an epsilon's distance on each side of the functional value for the given X. Next, it draws the two vertical lines from the points where these horizontal lines associated with epsilon intersected the curve, down to the X axis. The program computes the distances from the given X value to these two vertical lines, and assigns delta the value of the smaller one. At that point the user may enter another value for epsilon, and the old lines are erased and new ones drawn to fit the new epsilon value. This may be done as many times as is desired.
The last choice on the menu constructs the table often found in textbooks, that lists $X$ values and their corresponding functional values, as $X$ approaches the given value. It allows $X$ to approach from both the right and the left simultaneously, which illustrates whether the function values converge on the limit from both sides.

**Derivatives.** The program entitled "Derivatv" was designed to follow the usual teaching method for presenting the derivative of a function. This method involves establishing a set value for $X$, followed by trying to find the slope of the tangent line to the curve at the point $(X, f(X))$. Since the slope of any secant line through the given points can be determined, a secant line is drawn through the point $(X, f(X))$ and another point on the curve, called $(X+h, f(X+h))$. The slope of this secant line is determined to be $(f(X+h) - f(X))$ divided by $(X+h - X)$. In order to bring the secant line down into the position of a tangent line, the point $(X+h, f(X+h))$ must be moved closer to the set point $(X, f(X))$, which can be accomplished by making the value of $h$ smaller. If $h$ is allowed to approach a value of zero, then the point $(X+h, f(X+h))$ will approach the given point, and the secant line will approach the position of the actual tangent line. Thus, by taking the limit of the slope of the secant line as $h$ approaches 0, the slope of the tangent line is obtained. Then, this is expanded to become the definition of the derivative of the function at $X$. In the spirit of adhering to this strategy for teaching derivatives, the program graphs a function, then draws a secant line through the selected point and an arbitrary
point on the curve, somewhere near the right edge of the screen. This line is erased, as another is drawn through a slightly closer point, and this is continued until the given point is attained. At that point, the secant line becomes the tangent, and is left in place. Its slope is computed and given as the value of the derivative of the function at the given value of \( X \).

**Integration.** The typical teaching strategy for introducing the integral, is to consider the region bounded by a continuous function, the \( X \) axis, and two vertical lines that intersect the functions graph. The program "Integral" is intended to be a complete introduction, and the first choice on the menu starts at the very beginning. A parabola, opening down, is graphed, and the region between the curve, its \( X \) intercepts, and the \( X \) axis, is considered. First, the curve is enclosed by a rectangle, and the area is given. Then, the interval from the left intercept to the right is partitioned into three equal parts, and three rectangles are drawn, using the maximum height of the curve in each subinterval as the height of the rectangle. The areas of these three rectangles are summed, and the total printed. When this stage is completed, the teacher will explain that this new sum is closer to the actual area under the curve, than was the first single rectangle. Next, the computer repeats the above with ten partitions of the interval, and the sum of the areas is given. As the next step, the computer allows the user to determine the number of partitions, which the students are eager to offer. The program will draw the rectangles for any number of partitions desired, but for a partition of more than about 200, the execution
is very slow. Finally, the computer uses a partition of 200, and graphs the rectangles for that. While the program is running, the instructor can discuss the concept of Upper and Lower Riemann sums, and how they differ from Riemann sums in general. Also, when the program completes the graphing of 200 subintervals, the teacher may consider the concept of the limit as the number of partitions approaches infinity. This is then given as one of the definitions of the integral of a function.

The second choice on the menu of the program is used to change the function under study. Different from the format of most of the programs, this is done because the user will not want to enter a new function for each of the menu choices. The third and fourth choices graph upper and lower Riemann sums, respectively, for the function entered using choice two on the menu. Choice five lists a table of partitions, with their corresponding Riemann sum values and trapezoidal rule value, starting with a partition of 2 and going to 140. The final choice on the menu is used for the approximate integration technique of the Trapezoidal Rule. This choice may not be used until later in the course than the others, since most textbooks present approximate integration as the last section of the integrals chapter. In order to use the trapezoidal rule part of the program, the students need to be reviewed on what trapezoids are, and how the trapezoidal rule formula is derived. The program graphs a given function, over a given interval, accepts the number of partitions, then draws the trapezoids on the graph. Then, the areas are summed and the sum is printed.
CHAPTER V
SPECIFICS IN THE USE OF EACH PROGRAM

Introduction
In order to best implement the programs contained in this thesis, some additional ideas on the most effective way to present each, are needed. For instance, with many of the programs, the user is allowed to enter any number desired as a coefficient in an equation. Although this is a very important characteristic, in extreme cases it could lead to "bad examples" that do a poor job of illustrating the point. If an equation is entered that causes the graph to be beyond the scope of the screen, obviously the students will not benefit. Therefore, the main emphasis of this chapter is to provide information on the most useful numbers and equations to enter when a particular program is in execution. Also, this chapter will give detailed instructions for the use of each program. The chapter is divided into separate sections, one for each of the programs, and each section includes the above mentioned suggestions, as well as any further ideas for class participation not given in the previous chapter.

Items Common to All Programs
The diskette provided is formatted using PC-DOS version 2.1, and if used on a compatible computer, is designed to be self-booting. In other words, all the user is required to do in order to use any of the programs, is to insert the diskette into the computer and turn it on. The autoexec file on the disk automatically runs a program called "prompt," which prints a list of the
programs, with explanations, and asks the user to type the name of the desired program. When this is done, the desired program is automatically run. If, however, the computer being used does not recognize the DOS version, it should be turned on with the appropriate DOS disk, put into BASICA, and then the desired program should be loaded and run.

Every one of the programs has several input statements, which allow the user to enter information into the program. This information must, however, be entered in a very precise method, set by the programmer, and explained each time by the displayed "prompt." A prompt is a message printed on the screen, whenever the program pauses for an input statement, which explains very briefly what is to be entered, and the format that must be used to enter it. In most cases, the prompt will be very short, with the format given as variables. If only one number is to be entered, then no format is given, and the user simply types the number and presses enter. If two or more numbers are needed, the format is displayed, followed by a question mark. For example, if the coordinates of a point are needed, the prompt would be this:

\[ X, Y ? \]

telling the user it expects the X coordinate, a comma, and then the Y coordinate. If information is entered incorrectly, the computer will reject it, displaying the message "Redo From Start." Often, this message will cause the screen to scroll up a line, which in turn causes the coordinate system on the screen to be out of phase with where the program expects it to be. Thus,
anything plotted thereafter will not appear in the correct place. If this happens, the program should be stopped, and returned to the command menu to start over. This is most easily accomplished by performing a Control-Break, done on the IBM-PC by holding down the Control key, marked CTRL, and pressing the BREAK key (which is also marked with Scroll Lock). When the Break message is visible on the screen, RUN the program again, making the same menu choice as before.

All of the programs use one of two methods for inputting a new function or equation. Most common among the programs is the use of the EDIT command within the program, which stops program execution and displays the line containing the function, together with its line number. When this method is used, something similar to the following is printed on the screen:

```
1000  Y = 2*X - 1
GOTO 550
```

The cursor will be positioned under the left digit in the line number, in this case 1000. The user simply moves the cursor past the equals sign, and types in the desired right side of the equation. It is important to note that the symbol * must be used for multiplication, the symbol / for division, and the symbol ^2 for exponentiation (in this case squaring). Since the program is not running while these changes are being made, it must be restarted once the new function has been entered. This is accomplished by sending the program to the next line, in this example 550, through the use of a GOTO statement. The GOTO has been printed in a position where all the user must do is type in the
equation, then press enter, putting the cursor on the GOTO line. By pressing enter a second time, the GOTO is executed, and the program begins execution at the point where it had stopped. The other method used by some of the programs, is to simply use input statements. The general form of the equation is printed on the screen, with letters such as a, b, h, or k for numbers the user is allowed to change. Below the equation is printed an input prompt, asking for the values that are be assigned to the letters. When they are correctly entered, the equation containing the letters is erased, and replaced by the same equation with the numbers that were entered. The following is an example, using the equation of an ellipse:

\[ \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \]

\[ a, b ? \]

After the user enters the values to be assigned to a and b, the equation is rewritten with those values.

For the computer graphics in most of the programs, the computer is put into "high resolution" graphics mode. If the machine on which the programs are run does not have the capability of color graphics in this mode, then some program editing must be done. Somewhere near the start of each of the programs will be two statements appearing as follows:

\[ 30 \text{CLEAR,,32768 : SCREEN 6 : COLOR 3,4} \]
\[ 40 \text{SCREEN 2} \]

For the IBM-PCjr, the first statement clears the graphics memory for use, and turns on Screen 6, the high resolution color screen. On the IBM-PC, the Clear statement is not recognized, and there-
before must be taken out. In line 40 above, there is an apostrophe before the word SCREEN. This symbol signals the computer to ignore that line, so if using the program on an IBM-PC, insert an apostrophe before the word CLEAR in line 30, and delete the one in line 40. Once this is done, the program will run, and all graphics will be in black and white.

The Program LINES

The program entitled "Lines" is designed to aid in the instruction of graphing in the Cartesian Coordinate system, beginning with plotting points, and ending with linear equations in the point-slope form. An example of what the screen looks like during the Slope portion of the program is given as figure 1-A in appendix A, and figure 1-B shows how the screen appears during the Point-Slope portion.

The opening menu of the program is as follows:

    COMMAND MENU

1. Point Plotting
2. Slope
3. Lines in the Slope - Intercept form
4. Lines in the Point - Slope form
5. Terminate Program

ENTER YOUR CHOICE ?

Point Plotting. The first choice, Point Plotting, constructs a grid that extends from -8 to 8 along the X axis, and from -5 to 5 along the Y axis. In the lower left hand corner of the screen is the prompt: X,Y ?, asking for the coordinates of a
point to be entered. Any coordinates within the scope of the grid may be entered, and are subsequently plotted as a small circle at the proper spot. Points graphed on the outer edges of the grid do not show up as complete circles, since part of the point is "off" the screen. If a point entered is off the screen, it will simply not appear on the graph, and the program continues. The program will accept any numerical value as a coordinate, positive or negative, such as 1.5, but not 3/2.

Slope. The second choice on the menu teaches the concept of slope, by having the user enter any two points to be used as the endpoints of a line segment. The prompts for entering these points are:

Point 1: X,Y ?
Point 2: X,Y ?

The coordinates for these points are restricted to integers, so numbers like 4.5 and 3/2 are not allowed. Once the points are entered and graphed on the screen, the line segment joining them is drawn, and then the right triangle, formed by this segment and the grid, is slowly constructed. The vertical leg is drawn first, a unit at a time, then the horizontal leg. When the triangle is complete, the following prompt appears:

Slope ?

The user is expected to enter the correct slope. Since several formats for the slope are possible, the following instructions are pertinent. If the slope is zero, a 0 is entered. If the slope is undefined, the user enters UND. Finally, if the slope is neither of those two, then it must be entered as a fraction.
However, many units are contained in the vertical leg, followed by /, then the number of units in the horizontal leg, is the format used for this case. Very important here is that even if the slope is a reducible fraction, it still must be entered as the exact units of vertical change over the exact units of horizontal change. For instance, if the right triangle constructed has a vertical leg of 4 units, and a horizontal leg of 2 units, the program will consider a slope of 2 as incorrect. Instead, 4/2 must be entered.

**Slope Intercept-Form.** The linear equation that is in the program when it is loaded is $Y = \frac{1}{2} \times X + 1$. It is suggested that this equation is graphed first, by defaulting through the input operation with two presses of the ENTER key. When the line is graphed, the teacher might stop and have the students compute its slope and $Y$-intercept, then see how they relate to the equation. The program then asks if another is desired, which would be answered by typing Y for Yes. This program uses the first method discussed before, which stops the program and allows the user to edit the equation. Any previous lines are not erased before a new one is graphed, allowing for visual comparisons. If the screen becomes cluttered, clear the screen by returning to the command menu and making the same choice again.

Any equation entered is accepted, so care must be taken to insure that the graph appears on the screen. Since the $Y$ axis goes only from -5 to 5, it is best stay within those bounds, although parts of some lines with $Y$-intercepts beyond this will appear, possibly across a corner of the screen. No vertical
answers that is was correct, the computer will not detect this and assume it was correct. In the event that this happens, simply request another problem and go on from there.

It is important that the students recognize that the two equations:

\[ Y = m \cdot (X - h) + k \quad \text{and} \quad Y - k = m \cdot (X - h) \]

are equivalent, since the first of these is required by the program.

**The Program CONICS**

"Conics" is a program written for the purpose of graphing each of the conic sections using the general form of the equation of each. The scale on the axes for each part of the program is from -16 to 16 for X values, and from -10 to 10 for Y values. If, at any time, numbers are entered that exceed this scale, a part or all of the graph will be missing from the screen. Figures 2-A and 2-B illustrate how the screen appears after an ellipse and a hyperbola have been graphed, respectively.

All of the portions of this program use the second method discussed above for changing the equation being graphed, which is by entering the values for certain parameters in the equation. With the exception of hyperbolas, all the conic section equations are graphed without erasing any previously graphed equations. This is to allow for comparison of several graphs, but occasionally the screen will need to be cleared before another is graphed. The screen can be cleared in the same method as mentioned for the program "Lines," by returning to the menu and
making the same choice as before. The menu for this program appears on the screen as follows:

COMMAND MENU
1. Graphing Parabolas
2. Graphing Circles
3. Graphing Ellipses
4. Graphing Hyperbolas
5. Terminate Program

ENTER YOUR CHOICE?

Parabolas. The parabola portion of the program begins by creating a coordinate system very similar to the one used by the Lines program, except for the extended scale of the conics program. Then, in the lower left corner of the screen, the general form of the equation of a parabola is written, with the input prompt below that. The form of parabolic equation used in this program is as follows:

\[ Y = a \cdot (X - h)^2 + k \]

The users are to enter values for the three parameters, \( a \), \( h \), and \( k \), and when they have been given values, the general equation is rewritten with the respective values in place of the \( a \), \( h \), and \( k \). Then, the parabola is graphed. Any values may be entered, but it is best to keep the vertex of the parabola away from the sides of the screen so that a suitable portion of the graph is visible. When the instructor is demonstrating how the value of \( a \) affects the parabola, using a few extreme values is effective. Very large numbers for \( a \) make the parabola close in almost to a
straight line, while very small numbers, such as .01, cause the parabola to be very wide. The teacher should always have the students compare the picture on the screen with the equation, to be sure they realize where the three numbers $a$, $h$, and $k$ are used, and what effect each has on the graph.

**Circles.** The circles portion of the program is used in very much the same method as for graphing parabolas. The equation that is used for graphing circle is the center-radius form, as follows:

$$(X - h)^2 + (Y - k)^2 = r^2.$$

The three values missing from the equation are entered, according to the prompt, and the equation is then written with the numbers instead of the letters. After that has been done, the circle is graphed, and the user is asked if another is desired. As with the parabolas, any values may be entered, but if the center is too close to an edge to accommodate the radius, part of the circle will be missing. This is not a particularly undesirable occurrence, as long as the center does appear on the screen, with enough of the circle to show its shape. The teacher should take care to point out where the three inputted numbers appeared in the equation, and how each affected the graph of the circle.

**Ellipses.** When the third choice on the menu of the conics program is selected, the following will appear on the screen:

Select the form of elliptic equation desired.

1. $X^2 / a^2 + Y^2 / b^2 = 1$
2. $(X - h)^2 / a^2 + (Y - k)^2 / b^2 = 1$

CHOICE ?
The first choice is the form most often taught first, and when it is selected, the axes are drawn, and the ellipse is drawn with its center at the origin, and the characteristics set by a and b. When the graph is complete, the program asks:
Another?
If the user enters an N, the program then asks:
Other form?
If this question is answered with a Y, the other form of equation is offered for graphing. The other form listed above allows the center of the ellipse to be moved from the origin, to the point (h,k).

Hyperbolas. As with ellipses, the hyperbola portion of the program allows for a choice of two types of equations. The first thing appearing on the screen is:
Select the form of hyperbolic equation
1. \( \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \)
2. \( -\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \)

CHOICE?
If the first choice is made, the user may graph hyperbolas that open right and left, while the second choice allows one to graph hyperbolas opening up and down. For whichever choice is made, the corresponding equation is used, and the values of a and b are entered. The values entered should be small enough that at least a small part of the hyperbola will fit on the screen. When the values are entered, they are placed in the equation, and the graph is drawn. First the rectangle and asymptotes (discussed in the last chapter) are drawn as dotted lines, then the curve is
graphed.

For illustrating how the graph of the hyperbola relates to its asymptotes, a good example is to choose the second form of the equation, causing the hyperbola to open up and down, then enter a 3 for the value of a, and 1 for the value of b.

The Program FUNCTION

The program "Function" differs from most of the others in the way it was designed. There is no opening menu for this program, so the first thing done is to enter the function. The method of stopping the program and editing the function line is used in this program, and once it has been entered, some questions are asked of the user about the function. The first of these questions is as follows:

Is the function continuous?

If this question is answered with Y for yes, the program continues. If, however, there are discontinuities, the user should answer with N for no, in which case the following question is asked:

Are there any intervals where the function is undefined?

If the answer is Y, the left and right endpoints of the interval where the function is not defined, are then entered. If, for instance, the left endpoint is negative infinity, then it should be entered as any X coordinate past the left edge of the screen, which the user will determine to be either -16, -10, -5, or -1. The next question appearing on the screen, whether an interval was entered before or not, is as follows:
Number of points where undefined?

If the answer is 0, the program goes to the next stage. An answer of a positive integer, however, will allow the user to enter the X values of all the points where their function is undefined. The reason that the prompts concerning undefined values are needed, is to avoid error messages such as "Division by zero" and "Illegal function call" (which occurs, for instance, if the computer attempts to compute the square root of a negative number).

When all discontinuities have been entered, the user then decides the scale with which the axes will be drawn. That is done in the following manner:

Choose the scale for the X axis.

1. -16 to 16
2. -10 to 10
3. -5 to 5
4. -1 to 1

CHOICE?

These numbers indicate the values on the X axis that will be at the edges of the screen. The values on the Y axis are according to the length of a unit, as determined by the above choice, however, they are not the same as for the X axis. Because of the dimensions of the screen, the number at the top edge will be smaller than the number at the right edge of the screen.

The scale is chosen according to the function entered, by where its intercepts, asymptotes, and other points of interest are located. As an example, the function defined by:
\[ Y = 0.01 \times (X + 9) \times (X + 3) \times (X - 1) \]

would require either the scale -16 to 16, or -10 to 10, in order to illustrate all the X intercepts. The function:
\[ Y = X \times \sin \left( \frac{1}{X} \right) \]
is most interesting near the origin, so the scale -1 to 1 would be chosen.

Figure 3 in the appendix A shows an example of a function graphed using this program. The scale was chosen as -10 to 10, and discontinuities were entered as -4 and 4. The function entered was:
\[ Y = \frac{(X + 4)^2}{(X - 16)^2} \]

After the function has been graphed, the user is asked first if a change of the scale is desired, and if so, the program branches back to that set of choices. If no change is desired, the user is asked if a new function is to be graphed. An answer of Y starts the program over, while an answer of N ends it.

The Program SINE LAW

The program to introduce the law of sines, entitled "Sine law," uses low resolution graphics instead of high. This allows color graphics to be used even on the IBM-PC, and has the effect of making the text on the screen twice as large. When the program is run, the first thing that appears on the screen is the command menu, which is as follows:

**COMMAND MENU**
1. Two Angles and a side opposite one of the angles.
2. Two sides and an angle opposite one of the sides.
3. Terminate program
The first choice demonstrates the case of the law of sines most often taught first. When it is selected, the prompts for entering the two known angles and the known side are placed on the screen. They are as follows:

1st ANGLE?
2nd ANGLE?
SIDE?

There is no labeling of these quantities as Alpha, Beta, side a, and so on, during the prompt, since this can cause confusion for the students when textbook examples use a different label for a particular item. The program always considers the first angle entered as Alpha, the second as Beta, and the entered side as a. Thus, any examples should be entered with that in mind.

For the following description of what appears on the screen, refer to figures 4-A and 4-B. The unknown side c is placed horizontally across the screen, from one edge to the other, and the triangle is constructed above it. Side b, the other unknown side, is connected to the left end of side c, with the given angle Alpha between them. This side will extend to the top of the screen until its length has been determined. The side of known length, a, is connected to the other end of side c, with the angle Beta between them. When the teacher presses enter, the side c is shortened, bringing the known side a along, at the proper angle, until the top end of a touches side b. At that point, side b is shortened to the proper length, and all the pieces of the newly formed triangle are labeled. Known pieces are given their respective values, while unknowns are labeled
with letters. All unknown sides are labeled with small letters, a through c, and unknown angles are given capital letters, A through C. When the triangle has been properly labeled, the following appears below the triangle:

\[ \frac{b}{\sin \beta} = \frac{a}{\sin \alpha} \quad b = ? \]

All variables listed in parentheses are known values, and their numerical value is printed in those places. The students are to compute the value of side b, round off to one decimal place, and tell the teacher what to enter. If the value entered is incorrect, the computer beeps, erases it, redisplay the prompt. When the correct value is given, the letter b on the triangle is replaced by its value, and then the bottom of the screen is cleared. Next, the following is written:

Angle C ?

As soon as the correct angle is given, it is placed on the triangle in place of the letter C. The law of sines is then written again, just as before, only using the letter c and the angle computed for C. When the correct answer is supplied, it is also placed on the triangle, and then the program declares the triangle solved.

In order for all of the triangle to fit on the screen, it is not possible to use any obtuse angles, and sides should be kept less than 12 units in length. If it is calculated that an entered triangle will not fit on the screen, a message indicating so is printed, and the value input prompts are redisplayed. The
permissible lengths of the given side depend on the angles entered, but 12 is a good guideline. If a right triangle is to be displayed, enter 90 degrees for the first angle, then enter the second angle. The entered side, for this case, may be fairly long, but again depends on the value of the second angle.

The second case when the law of sines is used occurs when two side lengths, and the measure of an angle opposite one of those sides, are the known pieces of the triangle. Depending on the values of the known parts, there might be none, one, or two triangles possible. This concept is difficult for the students, so a teacher might elect to introduce it before using the program. It seems to work well, however, to use the program to introduce the concept, choosing first a case where no triangles are possible, then one where one triangle is possible, and finally a case where two are possible. When the second menu choice is made, the following prompts are printed first:

Angle ?
1st Side ?
2nd Side ?

After these values have been entered, the screen is cleared, and the computer begins to construct the triangle. Refer to figure 5-A for an illustration of the following description, and to figure 5-B for the next stage of the construction of the triangle.

As before, the unknown side is placed across the bottom of the screen, from left edge to right edge. This side will be designated as c. The second side entered, which is designated as
b, is connected to the left edge of \( \text{c} \), using the given angle \( \text{A} \). At its top end, the remaining side, \( \text{a} \), is connected with an arbitrary angle. This side will extend to the right, almost horizontally, so that its other end does not touch side \( \text{c} \).

When the teacher presses the enter key, the second stage of the construction begins. Side \( \text{a} \) begins to swing, from its point of contact with side \( \text{b} \), until it gone through an arc sufficient to take it past the vertex at angle \( \text{A} \). If, at any point during its arc, the free end of side \( \text{a} \) touches side \( \text{c} \), the computer beeps and leaves a side in place, drawn in blue. The user is then asked how many triangles were possible, which is determined by how many blue lines are on the screen. If the program is used to develop the concept of how many are possible, the teacher should lead the students to decide why that particular number of triangles occurred. When the correct answer for the number of triangles has been entered, the program redraws the first blue line in white, erases the other (if applicable), and labels the triangle as before. The triangle is then solved exactly as was done with the first case, having the user supply the values of the missing parts. If two triangles were possible, after the first is solved, it is erased and the second one drawn. When all possible triangles are solved, the program returns to the menu.

Again, entered values should be limited in order to keep the triangle on the screen. For illustrating the three cases possible, some recommendations for values are given. The first case, where no triangles are possible, is well demonstrated by entering 30 for the known angle, 4 for the first known side, and
10 for the second known side. These values will cause the side that is swung to never come in contact with the bottom side, showing that it is not possible to construct a triangle of those dimensions. The second case occurs when the side opposite the known angle is exactly equal to the altitude of the triangle to the horizontal base. This produces a right triangle, and the swinging side will touch the bottom side at the very bottom of its arc, and only at that one position. Good values to use for this case are 30 degrees for the known angle, 5 for the first side entered, and 10 for the second side. These values will form a 30-60-90 right triangle. Any values may be used for the third case, two triangles, as long as the triangle fits on the screen, and the first side length entered is smaller than the second, and longer than the height of the triangle. Values such as 55 degrees for the angle, 7 for the first side, and 9 for the second side, work very well. Then, to produce the last case, where the side opposite the entered angle (entered first) is longer than the other entered side. In this case, the arc created by the swinging of the side intersects the bottom side in one point, then again on the line extended from the bottom side, outside the triangle. A good way of illustrating this case is to use the same values as for the previous case, except reverse the values of the two sides that are entered. Enter 55 for the angle, 9 for the first side, and 7 for the second side.

The Program TRIGGRAF

The program entitled "Triggraf" is used to graph the six
trigonometric functions. When it is run, the following command menu appears on the screen.

COMMAND MENU

1. Graph of Sine Function
2. Graph of Cosine Function
3. Graph of Tangent Function
4. Graph of Cotangent Function
5. Graph of Secant Function
6. Graph of Cosecant Function
7. Terminate Program

CHOICE ?

For the first three choices, the program uses the edit command to stop the program and allow the function to be changed. The next three choices, the Cotangent, Secant, and Cosecant, all use input statements for the parameters \( a, b, \) and \( c, \) in the following general equation:

\[
y = a \cdot \text{TRIG}( b \cdot (X - c)).
\]

When any of the six choices are entered, the first thing to appear on the screen are directions on how to enter the function to be graphed. Then, when ready, the instructor presses enter to move into the input procedure. Once the equation has been entered, the axes are drawn and the equation is graphed. An example of the screen output for this program, using the cosecant function, is given as figure 6 in appendix A.

All of the equations will accept the word PI as part of the phase shift parameter. For example ( \( X - \text{PI}/2 \) ), where this was either typed into the equation, or PI/2 was entered for the phase
shift parameter, would be recognized as valid. The scale on the X axis is from negative PI to four times PI, and on the Y axis is from -5 to 5. Entering an amplitude larger than 5 will result in part of the graph being off the screen.

For all of the trigonometric functions that apply, asymptotes are drawn as dotted vertical lines. For graphing the secant and cosecant functions, the value of the parameter \( a \) should not exceed 4, since the scale on the Y axis only goes to 5, and very little of the graph would be visible.

**The Program POLAR**

The program "Polar" is used to plot points, and graph equations, on a polar coordinate system. The command menu for this program appears as follows:

**COMMAND MENU**

1. Plot Points with Polar Coordinates
2. Graph Polar Equations
3. Exit Program

**CHOICE ?**

When the choice is made to plot points, the polar coordinate system is created on the screen, then the user is asked to enter the coordinates, in polar form, of a point to be plotted. The scale on the graph extends to a radius of 11, and lines are drawn at 0, 30, 45, 60, 90, degrees, and their multiples. Either of the coordinates may be entered as a positive or negative value. Once the point has been graphed, the program asks if another is desired. If so, the Point prompt is cleared and redisplayed, but
any previous points are left on the graph.

When the second menu choice, for graphing polar equations, is selected, the first message appearing on the screen is:

Does your equation contain a square root?

If the answer is N, the screen is cleared and the equation is entered by the user. Also, the user is asked to enter the last value to be used for the angle theta, as a multiple of PI. The prompt for this is as follows:

Theta begins as 0. Enter stopping point as PI, 2PI, 3PI, ... ?

If, however, the answer to the square root question is Y, the equation must be entered without the square root, after a variable called PSQR. The user types in what PSQR is to be. Then, when the program is running, the value of PSQR is computed for varying values of Theta. That value is then tested to determine if it is negative. If not, the value is used, and its square root is computed. For example, in order to graph an equation such as:

\[ P = \text{SQR} ( \sin ( \text{Theta} )) \],

the user would be asked to enter the equation without the square root, after PSQR, as follows:

\[ \text{PSQR} = \sin ( \text{THETA} ) \].

Equations such as this look the best if a large coefficient is used before the SIN, such as 20, 30, or 40. This makes the graph large enough to be seen easily by the students. Figure 7 in the appendix A shows the graph of an Archimedes Spiral, using this program. The figure does not appear perfectly round, which is caused by the printer, as the polar coordinate system on the
screen appears circular.

For most polar equations, the stopping point for Theta should be 2PI, since the graph will repeat after that. For any of the spirals, however, values of up to BPI might be used, causing the graph to have half that many complete spirals.

**The Program LIMITS**

The program "Limits" is intended to be used for an introduction to limits. The command menu appears as follows:

COMMAND MENU

1. Explanation of Limits
2. Input a new Function
3. Graph Function with Epsilon - Delta lines
4. Table of Function Values as X Approaches some Number
5. Exit Program

CHOICE?

The first choice, an explanation of limits, gives a brief introduction of what is meant by the limit of a function, then uses moving graphics to present an example. Figure 8-A shows how the screen appears after the first example has been graphed. The program writes on the screen that X values will be chosen closer and closer to 5, and that the value of the function at 5 is 3 \( f(5) = 3 \). The function being used is:

\[ Y = \frac{1}{20} \times (X - 1) \times (X - 5) \times (X - 10) + 2 \]

Then, when the teacher presses enter, the screen is cleared, the axes are drawn, and dotted lines are drawn from the point \((0,3)\) to \((5,3)\), and from \((5,0)\) to \((5,3)\). The program then begins
graphing the function, moving from both \( X = 1 \) and \( X = 7 \), in toward the point \((5,3)\). As the different values for \(X\) are used, arrows are moved along just below the \(X\) axis, toward the number 5. When the functional value gets sufficiently close to 5 from both directions, two arrows are moved along the \(Y\) axis, indicating the \(Y\) value, toward the number 3. The process stops when the arrows along the \(Y\) axis are within .1 of the value 3, which has brought the arrows along the \(X\) axis to within .1 of 5. The customary epsilon and delta lines are drawn from the axes to the graph, and the distances between the lines are printed.

Once the first example has been completed, the function is changed (by the program) to:

\[
Y = \begin{cases} 
\frac{1}{20} \cdot (X - 1) \cdot (X - 6) \cdot (X - 10) + 2.5 & \text{if } X \neq 5 \\
3 & \text{if } X = 5 
\end{cases}
\]

This is the same function as before, except it has been shifted up .5 units. The functional value at \(X = 5\) remains the same.

Then, the process of graphing the function is repeated; however this time it is not possible to bring the functional value to within a distance of .1 of 3, so it is determined that 3 cannot be the limit of the function, as \(X\) approaches 5.

The second choice on the menu is used to change the function in the program. The reason for having a separate menu choice for this purpose is so the user does not have to deal with entering a new function every time another menu choice is made. Since the main emphasis of the demonstration is not the graph of the function, but rather to illustrate the epsilon - delta definition of the limit, it is conceivable that one may never wish to change
the original function. The original function, which is:

\[
Y = \sqrt{X} + 1,
\]
is very good for showing the way \( \delta \) depends on the chosen \( \varepsilon \). There are, however, other functions that work well in the program also, to demonstrate some special cases.

Choice two on the menu must be chosen before choices 3 and 4 will be allowed. When it runs, the user is asked to input the desired function. Due to difficulties in constructing the \( \varepsilon \) and \( \delta \) lines, the user is restricted to the first quadrant, so only functions that will appear in that quadrant should be entered.

When the third choice is selected, the user is first asked to enter the value of \( X \) where the limit is to be tested. Then, the equation entered during choice two will be graphed over the interval \([0,14]\), and at the top of the screen will appear this prompt:

\text{EPSILON ?}

The value entered for \( \varepsilon \) must be restricted to less than 1 and greater than 0. If any other values are entered, a message indicating the acceptable values is printed and the prompt is redisplayed. It is recommended that a fairly large value, such as 0.8, be used at first. Then, smaller values may be used later.

When an acceptable value has been given, the function is graphed and the \( \varepsilon \) - \( \delta \) lines are drawn surrounding the specified value of \( X \) and its corresponding value for \( Y \). The value for \( \delta \) is computed, using the smaller of the two distances from \( X \) to the vertical lines drawn. A third \( \delta \) line, in blue, is
drawn at a distance of delta from X on the side where the larger
distance is, so it is shown that values for X must be chosen from
within a delta's distance of the given point, in order to assure
that f(X) is within an epsilon's distance of the limit.

The values of epsilon and delta are printed at the bottom of
the screen, along with the following prompt:

Another value for EPSILON (return to stop)?

If the instructor wishes to choose a different epsilon, it is
entered, and the process is repeated. If another is not wanted,
the user simply presses the enter key. When this is done, the
user is asked if another value for X is desired. If the answer
is Y, the program branches back to the beginning of choice three.
If not, it returns to the command menu.

For an example of a limit demonstration using the third menu
choice, refer to figure 8-8 in appendix A.

The fourth choice on the menu displays a table of function
values, as X approaches a given value. This value is entered
first, using the same prompt as for choice three. When it has
been entered, the following table is constructed:

| From the left | | From the right |
|---------------|---------------|
| X  | f(X) | f(X)  | X |
| 6  | 5   | 6     | 10 |

The values for X begin at two units from the given value, and
each new increment is half of the last one. The last value to be
used is 0.000015 from X on either side. The values for f(X)
are listed on the middle of the table, so that comparisons of the values from the left and right are easily seen.

To demonstrate a function whose limit does not exist at some given value, using something similar to,

\[ Y = \frac{1}{(X - 3)} \]

and testing for a limit at 3, will cause the final values in the table to be very far apart. The function:

\[ Y = \frac{(X^2 - 9)}{(X - 3)} \]

will demonstrate a function that has a limit at \( X = 3 \), but no functional value at \( X = 3 \).

The Program DERIVATV

The program to help teach the second main topic in Introductory Calculus, derivatives, is entitled "Derivatv." The menu for this program will appear on the screen as follows:

COMMAND MENU

1. Explanation of Derivatives
2. Graph a Function and its Derivative Function
3. List a Table of Function Values and First Derivative values over a given interval.
4. Exit Program

CHOICE ?

The first choice, the explanation of derivatives, uses the standard method for introducing the topic. The graph of:

\[ Y = \frac{-1/4}{X - 4} \]

is drawn, and a secant line through the curve is drawn from the point (3,3.75) to the point (6,3). This is illustrated in figure 9-A in the appendix A.
The teacher then explains that the way to find a tangent line to the curve at the point \((3, f(3))\) is to bring the other point where the secant intersects the curve closer and closer to the given point. The slope of the tangent line is found by taking the limit of the slope of the secant, as the distance of the moving point from the given point approaches zero. On the screen it is written this way:

The limit of \(
\frac{f(3+h) - f(3)}{(3+h) - 3}
\) as \(h\) goes to 0, gives the exact slope of the tangent line at \((3, f(3))\).

When ready, the teacher presses enter, and a secant is drawn to a point with an X coordinate .25 units closer to 3, and the old secant is erased. This continues until the X coordinate of the moving point reaches 3, and the tangent line is left in place. For each line drawn, its slope is calculated and printed in the top right corner of the screen. The final value of this slope is the slope of the tangent line, and the value of the derivative of the function.

The second menu choice first allows the user to change the present function, then choose the interval over which that function will be graphed. The scale of the axes is determined by the length of the interval, so any interval may be entered. When the interval has been entered, the screen is cleared, the axes are drawn, and the two curves are graphed, putting the left endpoint of the inputted interval at the left edge of the screen, and the right endpoint at the right edge of the screen. The two curves are graphed separately, starting with the derivative function. After it has been completed, the teacher presses enter.
to start the graphing of the function itself. It works very well for practice to have the students predict the intervals where the function will be decreasing or increasing, and the points where relative extremum and points of inflection will occur, based on the behavior of the derivative function. This gives the students understanding as to how positive, negative, and zero values of the first derivative will later be used to help graph more complicated functions.

Figure 9-8 gives an example of the graph created, using a function similar to the second equation listed below.

The following is a list of functions, together with the appropriate intervals for graphing, that work very well with this portion of the program.

\[
Y = \sin(x) \quad \text{over} \quad [-3,7]
\]

\[
Y = \frac{1}{100} \times (x + 3) \times (x - 1) \times (x - 5) \quad \text{over} \quad [-5,10]
\]

\(\text{(the } \frac{1}{100} \text{ is needed to keep the height of the graph within the limits of the screen)}\)

\[
Y = 2 \times \sin(x) + .5 \times x \quad \text{over} \quad [0,10]
\]

\[
Y = \frac{1}{2} \times x + 3 \quad \text{over} \quad [-5,5]
\]

\(\text{(this shows the constant derivative function)}\)

\[
Y = \text{ABS}(x - 3) \quad \text{over} \quad [0,5]
\]

\(\text{(this shows the discontinuity in the derivative function at a point where there can be no tangent line)}\)

\[
Y = (x - 4)^2 \times .333 \quad \text{over} \quad [0,10]
\]

\(\text{(The powers here are how fractional exponents must be entered. This also has a discontinuous derivative)}\)

There are, of course, many other functions that are very good for illustrating this point, so some experimenting can be done. Higher degree polynomials can be created by adding more binomial factors to the second equation listed above. This will, how-
ever, usually require that the leading coefficient be decreased in size. Otherwise, the graph will be beyond the scope of the screen in several places.

The third choice on the menu creates a table of values for the function and its first derivative for varying values of $X$. The first thing to appear on the screen after this selection has been made is the prompt:

Do you want to change the function?

If the answer is Y, the user may input a new function in the same way as was done with the prior portion of the program. If another function is not desired, the function used is the last used during the second menu choice. After the function has been entered, the following prompt appears on the screen:

Enter the interval for the $X$ values: $A,B$?

Any interval may be entered, and there will always be ten values used from that interval. The values used are computed from the length of the interval, at an increment of one tenth the interval length. The table is intended to be used by first entering a fairly large interval, then continually smaller intervals that contain points where the first derivative has a desired value. When the first interval has been entered and the table is completed, the following prompt is displayed at the bottom of the screen:

Another interval?

If the answer is Y, the program is branched back to the prompt for entering the interval. An answer of N returns the program to the command menu.
Due to the limited capability of the computer to handle large numbers of decimal places, when the interval used becomes less than .02 in length, many of the values for the derivative are slightly inaccurate. For this reason, if an interval smaller than this is entered, the program will not accept it, and another is asked for. Usually an interval that is at least .02 units in length is sufficient to find the values of X where the first derivative is equal to zero, or any other desired value.

The Program INTEGRAL

The program "Integral" is very similar to "Derivatv" in its presentation and the way it is used. The following is the way the command menu will be on the screen:

COMMAND MENU

1. Explanation of Integration, using Riemann Sums
2. Input a Function and Interval of Integration
3. Graph Upper Sums
4. Graph Lower Sums
5. Graph Trapezoidal Rule
7. Exit Program

CHOICE ?

The first choice is a good first time introduction to finding the area under a curve by using Riemann sums, leading to the definition of integration. When the choice is selected, text is written on the screen explaining the function that will be
graphed, the interval over which it will be graphed, and how the area under that curve is to be estimated. A teacher using the program in front of a class will probably not want to read all the text to the students, but it does provide good notes for what should be explained prior to continuing. The equation that is graphed as an example is:

\[ Y = -\frac{1}{4} \times (X - 4)^2 + 4, \]

which has X intercepts at 0 and 8. The interval used is the interval between the intercepts. The text first explains that a way to estimate the area between the curve and the X axis is to enclose the curve in a rectangle, the height of which is equal to the maximum height of the function, and the base of which goes from 0 to 8. The area of this rectangle is computed to be 32, so it is known that the desired area is less than 32. This example is graphed on the screen. Then, when ready, the teacher presses enter to move on to the next explanation. The next set of text explains that if three rectangles were drawn, enclosing the one third of the curve each, a better approximation would result. This example is then drawn, and the sum of the areas of the three rectangles is printed. The process is repeated with ten rectangles, and the concept of a partition, in this case regular, of the interval \([0,8]\) is introduced. Figure 10-A in appendix A shows how the screen appears at this point in the demonstration.

After the sum using a partition of ten subintervals is calculated, the program allows the teacher to input a value for the next partition. The students will benefit from being allowed to suggest this value, and usually will offer a number between 50
and 80. These are very good values to use, at this point, since the graphic display of the Riemann sums will show that the sum computed is a very good approximation. Finally, the computer will graph an example, using a partition of 200, and the printed result is accurate to within 3 decimal places of the actual integral. The concept of taking the limit of the Riemann sums as the number of subintervals in the sequence of regular partitions is allowed to approach infinity is introduced then, and the definition of the integral is developed from that. If, when the instructor asks the students to give the number of subintervals in a partition, a student offers a number larger than 100 to 120, the instructor should probably suggest that they choose a smaller number, so that the computer's use of 200 subintervals is not redundant. Also, using a number larger than about 300 causes the program to run so slowly that much of the interest is lost.

The second choice on the menu is used to input a new function and to define the interval over which the integration will be performed. By using a separate menu selection for this purpose, the teacher does not need to be concerned with entering a function, even when a new one is not desired. For the most part, the best functions to use are those that stay above the X axis for most of the defined interval. Most of the screen is devoted to quadrants I and II. Thus, if the curve goes very far below the X axis, part of it will be absent from view. The scale of the axes is determined from the length of the interval entered, so horizontally, any function will fit. If the curve does go above the top edge of the screen, or below the bottom edge,
the rectangles will be cut off. The values calculated for the areas, however, will still be accurate.

The following is a list of a few functions, and the intervals to use with them, that work well for this program.

\[ Y = \frac{1}{6} \ast (X - 5)^2 + 1 \quad \text{over} \quad [0,10] \]

\[ Y = \frac{1}{6} \ast (X - 5)^2 - 2 \quad \text{over} \quad [0,10] \]

(\text{this puts the vertex of the parabola below the X axis, and demonstrates negative portions of the area sums})

\[ Y = \frac{1}{10} \ast (X + 3) \ast (X) \ast (X - 3) + 4 \quad \text{over} \quad [-5,5] \]

\[ Y = 2 \ast \sin \left( \frac{1}{2} \ast X \right) \quad \text{over} \quad [-5,5] \]

The third and fourth menu choices are used to illustrate Upper and Lower Riemann sums, respectively. When chosen, the first thing that must be entered is the number of subintervals in a regular partition. When the prompt for this entry has been answered, the screen is cleared, and the curve is graphed over the previously defined interval. As soon as the curve is graphed, the appropriate Riemann rectangles are drawn, either using the maximum height in each subinterval, or the minimum height, depending on the choice. The sum of the areas of the rectangles is computed, and printed across the bottom of the screen. The user is then asked if a different number of subintervals is desired. If the answer is N, the program returns to the command menu. An answer of Y starts that part of the program over.

The fifth choice will graph and estimate the integral of the given function using the trapezoidal rule. Any of the above functions will work with this choice, but those with sharp bends
will make the trapezoids more obvious. Also, if a partition of 20 or more subintervals is used, the tops of the trapezoids tend to blend with the curve itself, and are hard to see. Starting with a small partition, such as with 5 subintervals, then moving to the large ones, will show that the trapezoidal rule produces accurate results much faster than Riemann sums. Figure 10-8 illustrates what appears on the screen after using the Trapezoidal Rule portion of the program.

Choice number six on the menu computes upper and lower Riemann sums, and areas by the trapezoidal rule, for varying numbers of subintervals. Beginning with a partition of 2 subintervals, the table prints values for selected numbers of subintervals, and ends when a partition is reached large enough that the values of the upper and lower sums are within two decimal places of each other. The viewers can see that the computed values for the three sums approach each other, as the number of subintervals increases. The last values printed are all equal, since they are rounded off to two decimal places. It should be explained to the students that this is a result of the computer rounding the values off, and that the values would never actually be equal. The table illustrates very well that of these methods of approximate integration, the trapezoidal rule is more accurate than the Riemann Sum approach.
Summary of Thesis

The purpose of this thesis was to study the use, and effect, of a microcomputer in the mathematics classroom. The increasing development of this type of technology, leads to changes in the traditional methods of teaching mathematics. There are, of course, many uses for the computer in the mathematics classroom, but the emphasis of this thesis was on the use of the computer as an in class teaching aid, a tool for the teacher to use, just as one might use a film strip or the chalkboard.

The first chapter was an introduction to the thesis. In it was discussed how the idea for the thesis originated, the importance of such a study, the procedures used to study the problem, and the limitations of the study. Also given in the chapter was an overview of the thesis.

Chapter two was a presentation of what other authors have written concerning the field of Computer Aided Instruction. Intended mainly to prove a case for the use of CAI, quotations were included from notable works on the subject. Used in the study were several works from the National Council of Teachers of Mathematics, including the 1984 Yearbook for that organization, which was entitled, Computers in Mathematics Education. Also used were several books written on the subject, including Learning With Computers, by Alfred Bork, and CAI Sourcebook, by Robert Burke.

The third chapter offered the reactions of the students that
took part in the study. These students had all witnessed the use of a program written by the author, and had completed a questionnaire which asked for their opinion on several items. They were asked their reaction to the particular program they observed, their reaction to CAI in general, and any suggestions they might have. The answers to the questionnaire were both written comments, and a numerical response from 1 to 5. The numerical responses were totalled and averaged, and samples of the written comments were listed.

Chapter four treated the subject of using the programs in the classroom, from the aspect of how a mathematics teacher could most easily incorporate them into the normal curriculum of the course. The usual method of introducing a concept to a class was given. Then, the ways in which each program could be used in conjunction with that were discussed.

The fifth chapter was a detailed look at the use of each program, and could best be described as a user's guide to the programs stored on the diskette. Descriptions of how the teacher should enter values, what values and equations are the best to use, and explanations of what appears on the screen at different times during the execution of each program, were the important parts of the discussion. References were made to appendix A, which contains printouts of the screen during the execution of the programs. These figures were intended to help in interpreting the descriptions given in chapter five.

The sixth and final chapter summarizes the thesis, and gives conclusions and recommendations for those intending to use the
programs offered.

Following the sixth chapter are a bibliography and two appendixes. The first appendix, as was mentioned above, contains printouts from the screen that the reader may wish to refer to. The second appendix contains the listings of the actual programs, as they appear on the diskette. These are included as an aid to those wishing to expand the programs to include other topics, or to modify the programs to more closely fit their individual classroom needs.

Conclusions

During the course of this study, several conclusions became obvious. The following is a list of conclusions made by the author, concerning the study.

1. When introducing algebra students to the Cartesian coordinate system, and particularly to point plotting, it is best to use chalkboard examples. Through conversations with students, and reactions written on the student questionnaires, it became apparent to the author that the point plotting part of the program "Lines" spent more time than necessary teaching a subject that is quickly and easily learned.

2. If the program "Conics" is used to teach the graphing of conic sections, with no other explanations being offered by the instructor, the students will miss a very important point. The fact that these curves result from the intersection of a plane with a cone should be taught, and usually is. But
beyond that, it is very easy for the instructor to neglect the locus of points definition of each. Knowing that a parabola is formed by the locus of points equidistant from a point and a line, is an important way for students to relate geometry and algebra, and adds a point of interest to the lesson. The program does nothing to illustrate this point; therefore, by itself, it is not enough.

3. After demonstrating the concept of limits to a class that had never been exposed to it before, and then again to a Calculus II class, it was shown that the use of moving graphics made possible by the computer, gave the students a better understanding of the concept. Too often the epsilon-delta definition of limits is never quite understood, and computer graphics help to alleviate this problem. The beginning level classes seemed to have satisfactory understanding of the definition following the demonstration, while an alarming number of the Calculus II students seemed to be learning it for the first time. Many of these students commented afterward that they now understood precisely what the definition meant, as opposed to only vague understanding prior to the demonstration.

4. Although the use of CAI designed for use in front of an entire class is quite often favorable, it became apparent from the questionnaires that the small computer terminal that was used was often not adequate. The determining factor was the size of the class involved. In the larger-
classes, the students in the back of the room could not see enough detail on the screen to understand the topic very well. Despite encouragement to move forward in the room, it was difficult for some to see, and they invariably lost interest. For small classes, there was no problem, but if only the standard monitor size of 13 inches is available, then the viewing capabilities of the entire class should be a concern.

5. From the reading of other works on CAI, it can be concluded that there are many mathematics teachers who are pushing to make the use of the computer for teaching mathematics standard practice. Certainly there are those who feel the computer is not a necessity, and is to be used only if the instructor desires it. This is not the recurring theme, however, in the works reviewed. Instead, it is the belief of those authors, and the author of this thesis, that the computer allows the instructor to do a better job of teaching, and that mathematics teachers cannot afford to ignore it.

6. Apparent from the comments supplied by students on the questionnaires, was the fact that students themselves believe CAI is both important and effective. The high average response of 4.67 (out of 5), on the question asking if they thought that the use of the computer was interesting, indicates that a much higher level of student interest is achieved through the use of CAI. It was common for the author of this thesis to be asked by students when the
computer was going to be used again, indicating they were anxious for that, and that they preferred to be taught with the teacher using CAI.

7. Despite efforts to provide a multitude of error traps in the programs, it is still very easy for a user of the programs to enter something incorrectly. To completely error proof the programs, one would be required to spend much time, and add a multitude of lines to each. It seemed plausible to assume that those using the programs could learn easily enough how to avoid these problems, and the extra effort was not necessary.

Recommendations

Initially in this section, recommendations relating to the above conclusions are made, where appropriate. Pertaining to the second conclusion, concerning the use of the program "Conics," it is recommended that the teacher introduce conic sections with solid, three-dimensional visual aids, and with the locus of points definitions, then proceed to the graphing of the algebraic equations. It seems to help the students with the graphing techniques if they have these concepts first.

In relation to the third conclusion, as to the effectiveness of the program limits, it is also recommended that students are taught a strong intuitive definition of limits, before they are ever exposed to the epsilon-delta definition. Too often students develop the idea that the limit definition is beyond their comprehension, when in fact it is a rather straightforward concept.
Concerning the fourth conclusion, that the small size of a standard computer monitor causes viewing problems in large classes, the solutions to this problem are varied. Most desirable, but unfortunately not always an option, is to use a larger monitor. These are expensive, so schools do not typically own them. Microcomputers may be connected to large color television sets, with the use of special adapter cables which are relatively inexpensive, if the television has that capability. Much simpler, and usually effective, is to use a seating arrangement that abandons the desks, and crowds the students together near the computer. This uncomfortable position does not need to be maintained for a long period of time, and is well tolerated by the students.

The seventh conclusion, dealing with the lack of error traps, warrants some further recommendations. First, if a teacher can afford the time to run a program through a few times, in order to become accustomed to the procedures before presenting it to a class, the problem virtually disappears. Also, by carefully following the directions given in chapter five of this thesis, a teacher can avoid most errors. Then, if a problem does occur in front of a class, knowing how to get out of the program, and get it restarted quickly, will minimize any embarrassment. This too should be practiced, and the methods for the procedure are given in the fifth chapter also.

For a teacher wishing to expand the number of topics which may be presented using a microcomputer, the following recommendations are made. First, if one has no real experience with
graphics programs, it is suggested that the program listings in the third appendix be studied. The methods used in those programs are fairly simple graphics commands, and the logic involved should be straightforward. It would be easiest to create a new program by using one of the existing programs as a skeleton, editing where needed. The actual main program for nearly all of them is merely a menu driven series of subroutine calls, so by writing additional subroutines, one could expand the possibilities of any given program.

One recommendation for expansion concerns the program "Lines." The addition of a subroutine that would graph a system of two linear equations, mark the intersection point, and supply the coordinates of that point, would not be a major task. By using the same method of inputting equations, and the same method of graphing those lines, the existing program could serve as a model.

A similar subroutine could be added to the program "Conics," which would graph any two equations, linear or not, and graphically solve the system. The structure of such a subroutine would be nearly identical to the one for a linear system, mentioned above.

**Recommendations For Further Studies**

Any one of the three main program categories studied in this thesis -- algebra, trigonometry, or calculus -- could be considered as a source for providing materials for related studies. As an example, should a person desire to do a study emphasizing only
trigonometry, a program could be written for the solution of triangles using the law of cosines, which would be very similar to the program "Sinelaw." Furthermore, all of the trigonometric function definitions relating to the side lengths of right triangles could be presented nicely through a graphics approach.

It should be noted that the graphics in this study were all two-dimensional. A related study using three-dimensional graphics could certainly be of benefit to students dealing with the solid, analytical geometry concepts encountered in Calculus III.
A. BOOKS


B. PUBLICATIONS OF LEARNED SOCIETIES


APPENDIX A

FIGURES ILLUSTRATING SCREEN OUTPUT
Fig. 8-A

Fig. 8-B

SLOPE = -0.333

Fig. 9-A

Fig. 9-B

The total area is 24.2555

Press enter to continue

Fig. 10-A

Fig. 10-B

The AREA by TRAPEZOIDAL RULE for 5 partitions is 20.48

Do you want to try more partitions?
Press enter to form the triangle.

Fig. 4-A

$$\frac{b}{\sin 50} = \frac{6}{\sin 30} \quad b = ?$$

Fig. 4-B

Press enter to form the triangle(s).

Fig. 5-A

$$\frac{\sin B}{9} = \frac{\sin 30}{8} \quad B = ?$$

Fig. 5-B

Fig. 6

Fig. 7
Fig. 1-A

Fig. 1-B

Fig. 2-A

Fig. 2-B

Fig. 3
APPENDIX B

PROGRAM LISTINGS
GRAPHING LINES

STORED UNDER: LINES
JUNE 15, 1985

CLEAR,$32768! : SCREEN 6 : COLOR 3,4
SCREEN 2
CLS
LOCATE 3,15:PRINT "PROGRAM LINES"
LOCATE 5,15:PRINT "COMMAND MENU"
LOCATE 8,10:PRINT "1. Point Plotting"
LOCATE 10,10:PRINT "2. Slope"
LOCATE 12,10:PRINT "3. Lines in the Slope - Intercept form"
LOCATE 14,10:PRINT "4. Lines in the Two Point form"
LOCATE 16,10:PRINT "5. Terminate Program"

LOCATE 22,10:INPUT "ENTER YOUR CHOICE", CMD
this is an error trap for their choice!
WHILE CMD <> FIX(CMD) OR CMD < 1 OR CMD > 5
LOCATE 24,10:PRINT "Please enter a 1, 2, 3, 4, or 5"
GOTO 170

LOCATE 22,10:INPUT "ENTER YOUR CHOICE", CMD
this is an error trap for their choice!
WHILE CMD <> 5
ON CMD GOSUB 330, 470, 920, 1220
LOCATE 23,1 :INPUT "Press enter to return to -the COlland l'Ienu. ",B$
GOTO 80

SCREEN 0
END

----------- SUBROUTINE TO PLOT POINTS -----------
CLS
GOSUB 1820 'to draw the Cartesian Plane
LOCATE 23,1 : INPUT "X,Y ";X,Y
PX = 320 + 40 * X : PY = 100 - 20 * Y 'figure pixels and plot
CIRCLE (PX,PY),2 : CIRCLE (PX,PY),7 ' the point
LOCATE 23,1 : PRINT " "
LOCATE 23,1 : INPUT "Another ";A$
LOCATE 23,1 : PRINT " "
IF A$ = "Y" OR A$ = "y" THEN 360
RETURN

----------- SUBROUTINE TO EXPLAIN SLOPE -----------
CLS
GOSUB 1820 'to draw the graph
LOCATE 23,1 : INPUT "Point 1 ";X1,Y1 'enter the two points
LOCATE 23,9 : PRINT " 
LOCATE 23,1 : INPUT "Point 2 ";X2,Y2
LOCATE 23,9 : PRINT " 
IF X1 > X2 THEN SWAP X1,X2 : SWAP Y1,Y2 'leftmost point should be X1,Y1
PX1 = 320 + X1 * 40 : PY1 = 100 - Y1 * 20 'figure the pixels
PX2 = 320 + X2 * 40 : PY2 = 100 - Y2 * 20 'and plot the two points
CIRCLE (PX1,PY1),2 : CIRCLE (PX1,PY1),7
CIRCLE (PX2,PY2),2 : CIRCLE (PX2,PY2),7
LINE (PX1,PY1)-(PX2,PY2) : LINE (PX1+1,PY1)-(PX2+1,PY2) 'connect them
IF Y2 < Y1 THEN SIGN = 1 ELSE SIGN = -1 'determines pos. or neg. slope
FOR J = 1 TO 500 : NEXT J 'all J loops are time eaters
FOR I = PY1 TO (PY2-SIGN*20) STEP SIGN*20 'loop to draw vertical leg
LINE (PX1-2,I)-(PX1+2,I+SIGN*20) 'of the triangle
LINE (PX1-4,I)-(PX1+4,I+SIGN*20)
FOR J = 1 TO 500 : NEXT J
NEXT I
FOR I = PX1 TO PX2-40 STEP 40 'loop to draw horizontal leg of triangle
LINE (I,PY1+1)-(I+40,PY1+1)
LINE (I,PY2+2)-(I+40,PY2+2)
FOR J = 1 TO 500 : NEXT J
NEXT I
LOCATE 23,1 : PRINT "Slope ",M$ IF DELTAY AND DELTAX THEN 900 ELSE LOCATE 23,1 : PRINT " " : GOTO 830
LOCATE 23,1 : INPUT "Slope ";M$ IF VAL(LEFT$(M$,2)) = DELTAY AND VAL(RIGHT$(M$,NUMLEN)) = DELTAX THEN 900 ELSE LOCATE 23,1 : PRINT " " : GOTO 830
LOCATE 23,1 : INPUT "Slope ";M$ IF M$ = "0" THEN 900 ELSE LOCATE 23,1 : PRINT " " : GOTO 850
LOCATE 23,1 : INPUT "Slope ";M$ IF M$ = "UNO" OR M$ = "und" THEN 900 ELSE LOCATE 23,1 : PRINT " " : GOTO 870 'to let them try again
LOCATE 23,1 : INPUT "Another ";A$ IF A$ = "Y" OR A$ = "y" THEN 470 ELSE RETURN
*************** SUBROUTINE TO GRAPH SLOPE - INTERCEPT FORMS ***************
CLS
GOSUB 1820 to draw graph
GOSUB 1110 to input new equation
X = -7 'initial value for X
GOSUB 1170 'figure Y for that X
PX1 = 320 + X * 40 : PY1 = 100 - Y * 20 'convert point to pixels
FOR X = -6 TO 7 STEP .5 'loop to draw the line
GOSUB 1170 'to figure Y
PX2 = 320 + X * 40 : PY2 = 100 - Y * 20 'convert point to pixels
IF PY2 > 199 OR PY2 < 1 THEN 1060 'since it's off the screen
LINE (PX1,PY1)-(PX2,PY2) 'draw the line through the points
LINE (PX1,PY1+1)-(PX2,PY2+1)
PX1 = PX2 : PY1 = PY2
NEXT X
LOCATE 23,1 : INPUT "Another ";A$ IF A$ = "Y" OR A$ = "y" THEN GOTO 850
GOTO 260
1110 ' SUBROUTINE TO INPUT EQUATION
LOCATE 23,1: PRINT "GOTO 1150"
LOCATE 22,1: EDIT 1190
GOSUB 950

"SUBROUTINE TO FIGURE FUNCTIONAL VALUES"

Y = 1/2*X
RETURN

"SUBROUTINE TO GRAPH TWO-POINT FORMS"

CLS
GOSUB 1820 'to draw graph

the following statement turns on the random number generator and uses
RANDOMIZE VAL(RIGHT$(TIME$,2)) + VAL(MID$(TIME$,4,2)) 'the clock for seed

XVAL1 = INT( RND * 10 ) 'pull out a value, decide if pos. or neg., then
IF XVAL1 / Z = INT (XVAL1 / Z) THEN SIGN = 1 ELSE SIGN = -1 'make sure
IF XVAL1 > 7 THEN 1280 'it fits on screen
X1 = XVAL1 * SIGN

the next three blocks repeat procedure for other coordinates
YVAL1 = INT( RND * 10 )
IF YVAL1 / Z = INT (YVAL1 / Z) THEN SIGN = -1 ELSE SIGN = 1
IF YVAL1 > 4 THEN 1340
Y1 = YVAL1 * SIGN

XVAL2 = INT( RND * 10 )
IF XVAL2 / Z = INT (XVAL2 / Z) THEN SIGN = -1 ELSE SIGN = 1
IF XVAL2 > 7 THEN 1390
X2 = XVAL2 * SIGN

YVAL2 = INT( RND * 10 )
IF YVAL2 / Z = INT (YVAL2 / Z) THEN SIGN = 1 ELSE SIGN = -1
IF YVAL2 > 4 THEN 1430
Y2 = YVAL2 * SIGN

PX1 = 320 + 40 * X1: PY1 = 100 - 20 * Y1 'these plot the two
CIRCLE (PX1,PY1),2: CIRCLE (PX1,PY1),7 'random points
PXZ = 320 + 40 * XZ: PYZ = 100 - 20 * YZ
CIRCLE (PXZ,PYZ),2: CIRCLE (PXZ,PYZ),7

LOCATE 23,1: PRINT "GOTO 1550" 'let user input equation
LOCATE 22,1: EDIT 1790

X = -8 'draw the line that was entered
GOSUB 1770
PX1 = 320 + 40 * X : PY1 = 100 - 20 * Y
IF PY1 > 199 OR PY1 < 1 THEN X = X + .1: GOTO 1570 'make sure it's on the
screen
FOR X = -7 TO 7 STEP .5
GOSUB 1770
PX2 = 320 + 40 * X : PY2 = 100 - 20 * Y
IF PY2 > 199 OR PY2 < 1 THEN 1670
LINE (PX1,PY1) - (PX2,PY2)
LINE (PX1,PY1+1)-(PX2,PY2+1)
PX1 = PXZ : PY1 = PYZ 'move the points up
NEXT X
LOCATE 23,1 : INPUT "Correct ": A$
LOCATE 23,11: PRINT " "
IF A$ <> "Y" AND A$ <> "y" THEN GOTO 1530
LOCATE 23,11: PRINT " "
LOCATE 23,1 : INPUT "Another ": A$
IF A$ = "Y" OR A$ = "y" THEN GOTO 1230
GOTO 260
SUBROUTINE WITH THE EQUATION
Y = M * (X - H) + K
RETURN
SUBROUTINE TO DRAW GRAPH
SCREEN 2
first draw the axes
LINE (0,100)-(640,100) : LINE (0,101)-(640,101)
LINE (320,0)-(320,200) : LINE (321,0)-(321,200)
LOCATE 14,77 : PRINT "X" : LOCATE 2,39 : PRINT "Y"
loop to draw the vertical grid
FOR I = 0 TO 640 STEP 40
LINE (I,0)-(I,200),Z
NEXT I
loop to draw the horizontal grid
FOR I = 0 TO 200 STEP 20
LINE (0,I)-(640,I),Z
NEXT I
RETURN
**CONIC SECTIONS**

MIKE MOSIER

---

STORED UNDER: CONICS

JUNE 30, 1985

CLEAR,,32768! : SCREEN 6 : COLOR 3,4
SCREEN 2

CLS

LOCATE 3,12:PRINT "THE PROGRAM CONICS"

LOCATE 5,15:PRINT "COMMAND MENU"

LOCATE 9,10:PRINT "1. Graphing Parabolas"

LOCATE 10,10:PRINT "2. Graphing Circles"

LOCATE 12,10:PRINT "3. Graphing Ellipses"

LOCATE 14,10:PRINT "4. Graphing Hyperbolas"

LOCATE 16,10:PRINT "5. Terminate Program"

LOCATE 22,10:INPUT "ENTER YOUR CHOICE ", CMD

WHILE CMD <> FIX(CMD) OR CMD < 1 OR CMD > 5

LOCATE 24,10:PRINT "Please enter a 1, 2, 3, 4, or 5"

GOTO 180

WEND

WHILE CMD > 5

ON CMD GOSUB 320, 630, 1030, 1570

END

---

**SUBROUTINE TO GRAPH PARABOLAS**

GOSUB 2270 ' to draw the X - Y plane

LOCATE 22,1 : PRINT "Y = a ( X - h )^2 + k"

LOCATE 23,1 : INPUT "a,h,k ";A,H,K

IF H < 0 THEN SIGN1$ = "( X + " ELSE SIGN1$ = "( X - "

IF K < 0 THEN SIGNZ$ = " - " ELSE SIGNZ$ = " + "

LOCATE 22,1 : PRINT "Y = ";A;SIGN1$;ABS(H);")^2 ";SIGNZ$;ABS(K)

X = -15 : Y = A*(X-H)^2 + K ' set the initial value for X, and figure Y

PX1 = 320 + 24*X : PY1 = 100 - 10*Y 'figure the first pixels

loop to graph the parabola

FOR X = -15 TO 15 STEP .3

Y = A * ( X - H )^2 + K
PX2 = 320 + 24*X : PY2 = 100 - 10*Y

IF PY2 > 199 OR PY2 < 0 THEN 530

LINE (PX1,PY1)-(PX2,PY2) : LINE (PX1+1,PY1)-(PX2+1,PY2)

PX1 = PX2 : PY1 = PY2 ' moves old point up to new point position

NEXT X
See if they want to graph another
LOCATE 23,1 : PRINT "Another ":A$ 
IF A$ = "y" OR A$ = "Y" THEN LOCATE 23,1 : PRINT " ": GOTO 360
RETURN

************** SUBROUTINE TO GRAPH CIRCLES **************
GOSUB 2270 'to draw the X - Y plane
LOCATE 23,1 : PRINT " ": GOTO 660

get the equation of the circle
LOCATE 23,1 : PRINT "(X - h)^2 + (Y - k)^2 = r^2"
LOCATE 23,1 : INPUT "h,k,r "; H,K,R
LOCATE 23,1 : PRINT " ": GOTO 740

IF R < 0 THEN LOCATE 23,1 : PRINT "NO CIRCLE !! ": GOTO 970

IF H < 0 THEN FIRST$ = "(X + " ELSE FIRST$ = "(X - 

IF K < 0 THEN SECND$ = "(Y + " ELSE SECND$ = "(Y - 

PRINT FIRST$;ABS(H)j")^2 + ";SECND$;ABS(K)j")^2 = ";R,"j")^2"
LOCATE 23,1 : PRINT FIRST$;ABS(H)j")^2 + ";SECND$;ABS(K)j")^2 = ";R,"j")^2"

XCENTER = 320 + Z4*H 
YCENTER = 100 - 10*K 'finds the pixels of the center 
mark the center
PX1 = XCENTER - R *Z4 : PY1 = YCENTER : PY2 = YCENTER

begin graphing the circle
FOR X = -R TO R STEP .15
SQ = R^2 - X^2
Y = SQR ( SQ )
PX2 = XCENTER + 24*X : PY3 = YCENTER - 10 * Y : PY4 = YCENTER +10 * Y
IF PY3 > 0 THEN LINE (PX1,PY1)-(PX2,PY3)
IF PY4 < 200 THEN LINE (PX1,PY2)-(PX2,PY4)
PX1 = PXZ : PY1 = PY3 : PY2 = PY4
NEXT X

X = R
SQ = R^2 - X^2
Y = SQR ( SQ )
PX2 = XCENTER + 24*X : PY3 = YCENTER - 10 * Y : PY4 = YCENTER +10 * Y
IF PY3 > 0 THEN LINE (PX1,PY1)-(PX2,PY3)
IF PY4 < 200 THEN LINE (PX1,PY2)-(PX2,PY4)

see if they want to graph another
LOCATE 23,1 : INPUT "Another ":A$ 
LOCATE 23,1 : PRINT " ": GOTO 660
RETURN

************** SUBROUTINE TO GRAPH ELLIPSES **************
CLS : XCENTER = 320 : YCENTER = 100
LOCATE 10,15 : PRINT "Choose which equation you want. 
LOCATE 12,15 : PRINT "1. X^2 / a^2 + Y^2 / b^2 = 1 
LOCATE 14,15 : PRINT "2. (X - h)^2 / a^2 + (Y - k)^2 / b^2 = 1 
LOCATE 17,15 : INPUT "CHOICE ";CHC
IF CHC = 2 THEN 1210
GOSUB 2270
XCENTER = 320 : YCENTER = 100

LOCATE 22,1 : PRINT "X^2 / a^2 + Y^2 / b^2 = 1"

LOCATE 23,1 : INPUT "a,b "; A,B

LOCATE 21,1 : PRINT "X = A^2 / ; + Y^2 / ; = 1"

IF A < 0 OR B < 0 THEN LOCATE 21,1 : PRINT "Positives only." : GOTO 1150

LOCATE 22,1 : PRINT "X - h)^2 / a^2 + (Y - k)^2 / b^2 = 1"

LOCATE 23,1 : INPUT "h,k "; H,K

LOCATE 21,1 : PRINT "X = A^2 / ; + Y^2 / ; = 1"

LOCATE 21,1 : PRINT "X = A^2 / ; + Y^2 / ; = 1"

IF A < 0 OR B < 0 THEN LOCATE 21,1 : PRINT "Positives only." : GOTO 1270

IF H < 0 THEN FRST$ = "(X +" ELSE FRST$ = "(X -" IF K < 0 THEN SCND$ = "(Y +" ELSE SCND$ = "(Y -"

LOCATE 22,1 : PRINT FRST$; ABS(H);^2 / A^2 + ; SCND$; ABS(K);^2 / B

XCENTER = 320 + 24*H : YCENTER = 100 - 10*K

PX1 = XCENTER - 24*A : PY1 = YCENTER : PY2 = PY1

begin graphing the ellipse

FOR X = -15 TO 15 STEP .2

SQ = B^2 - X^2 * B^2/A^2

IF SQ < 0 THEN 1470

Y = SQR ( SQ )

PX2 = XCENTER + Z4*X : PY3 = YCENTER - 10*Y : PY4 = YCENTER + 10*Y

IF PY4 > 199 OR PY3 < 0 THEN 1460

LINE (PX1,PY1)-(PX2,PY3) : LINE (PX1,PY2)-(PX2,PY4)

PX1 = PX2 : PY1 = PY3 : PY2 = PY4

NEXT X

see if they want to graph another

LOCATE 23,8 : PRINT "Another "; A$

IF A$ = "Y" OR A$ = "y" THEN IF CHC = 1 THEN 1120 ELSE 1220

LOCATE 21,1 : INPUT "Other form "; A$

IF A$ = "Y" OR A$ = "y" THEN IF CHC = 2 THEN 1110 ELSE 1210

RETURN

'*************** SUBROUTINE TO GRAPH HYPERBOLAS ***************

CLS

YSIGN = -1

LOCATE 10,15 : PRINT "Choose which equation you want. "

LOCATE 12,15 : PRINT "1. X^2 / a^2 - Y^2 / b^2 = 1"

LOCATE 14,15 : PRINT "2. -X^2 / a^2 + Y^2 / b^2 = 1"

LOCATE 17,15 : INPUT "CHOICE "; CHC

IF CHC = 2 THEN YSIGN = 1 : GOTO 1750

GOSUB 2270 'to draw the X - Y plane
1650 LOCATE 22,1: PRINT "X^2 / a^2 - Y^2 / b^2 = 1"
1700 LOCATE 23,1: PRINT "If a <= 0 OR b <= 0 THEN LOCATE 21,1: PRINT "Positives only." : GOTO 1700
1740 LOCATE 22,1: PRINT "X^2 / "A;"^2 - Y^2 / "B;"^2 = 1"
1750 GOTO 1850
1760 ' get the equation of the HYPERBOLA
1770 60SUB 2270
1780 LOCATE 22,1: PRINT "-
1790 LOCATE 22,1: PRINT "X^2 / a^2 + Y^2 / b^2 = 1"
1800 LOCATE 23,1: PRINT "If a <= 0 OR b <= 0 THEN LOCATE 21,1: PRINT "Positives only." : GOTO 1800
1840 LOCATE 22,1: PRINT "- X^2 /"A;"^2 + Y^2 /"B;"^2 = 1"
1850 RBOX = 320 + 24*A : LBOX = 640 - RBOX : TBOX = 100 - 10*B : BBOX = 100 + 10*B
1860 FOR I = LBOX TO RBOX-10 STEP 20
1870 LINE (I,TBOX+1)-(I+10,TBOX+1) LINE (I,8BOX-1)-(I+10,8BOX-1)
1880 NEXT I
1890 FOR J = TBOX TO BBOX-5 STEP 10
1900 LINE (RBOX-1,J)-(RBOX-1,J+5) LINE (LBOX+1,J)-(LBOX+1,J+5)
1910 NEXT J
1920 M = (Z * B)/(Z * A) 'slope of the asymptote
1930 FOR X = -15 TO 15 'following loops draw asymptotes
1940 Y = M*X
1950 PX1 = 320 + 24*X : PY1 = 100 - 10*Y
1960 XZ = X+.5 : YZ = M*XZ
1970 PXZ = 320 + 24*XZ : PYZ = 100 - 10*YZ
1980 LINE (PX1,PY1)-(PXZ,PYZ)
1990 NEXT X
2000 FOR X = -15 TO 15
2010 Y = -M*X
2020 PX1 = 320 + 24*X : PY1 = 100 - 10*Y
2030 XZ = X+.5 : YZ = -M*XZ
2040 PXZ = 320 + 24*XZ : PYZ = 100 - 10*YZ
2050 LINE (PX1,PY1)-(PXZ,PYZ)
2060 NEXT X
2070 PX1 = 320 - 24*15 : BPY1 = 210 : TPY1 = -10
2080 'begin graphing the hyperbola
2090 FOR X = -15 TO 15 STEP .2
2100 IF YSIGN = -1 THEN IF X > -A AND X < A THEN TPY2 = 100 : BPY2 = 100 : PX2 = 320 + 24*A : GOTO 2180
2110 SQ = B^2*X/A^2 + YSIGN * B^2
2120 IF SQ < 0 THEN 2180
2130 Y = SQRT(SQ)
2140 PX2 = 320 + 24*X : BPY2 = 100 + 10*Y : TPY2 = 100 - 10*Y
2160 IF BPY2 > 199 OR TPY2 < 0 THEN 2180
2170 LINE (PX1,TPY1)-(PX2,TPY2) : LINE (PX1,BPY1)-(PX2,BPY2)
2180 PX1 = PX2 : TPY1 = TPY2 : BPY1 = BPY2
2190 NEXT X
2200 'see if they want to graph another
LOCATE 2,3 : PRINT "Another "; A$ 
IF A$ = "Y" OR A$ = "y" THEN 1590 
RETURN

'* SUBROUTINE TO DRAW GRAPH PAPER ************
CLS

' draw the axes first
LINE (0,100)-(640,100) : LINE (0,101)-(640,101)
LINE (320,0)-(320,200) : LINE (321,0)-(321,200)
LOCATE 14,77 ; PRINT "X" : LOCATE 2,39 : PRINT "Y"

loop to draw the vertical grid
FOR I = 8 TO 640 STEP 24 
LINE (I,0)-(I,200),2
NEXT I

loop to draw the horizontal grid
FOR I = 0 TO 200 STEP 10 
LINE (0,I)-(640,I),2
NEXT I

LOCATE 14,77 ; PRINT "X" : LOCATE 2,39 : PRINT "Y"

RETURN
FUNCTIONS

STORED UNDER: FUNCTION
MAY 15, 1984

CLS
CLEAR, 32768! : SCREEN 6 : COLOR 3,4
SCREEN 2

A = -20 : B = -20 : P(I) = -20 'flag values for points of discont

LOCATE 10,1 : PRINT "Move the cursor to the right of the equals and make changes.

LOCATE 11,1 : PRINT "Then, press enter twice."

LOCATE 15,1 : PRINT "GOTO 150"

LOCATE 14,1 : EDIT 650

LOCATE 17,1 : INPUT "Is f(X) continuous ";A$ IF A$ = "Y" OR A$ = "y" THEN 270

LOCATE 17,1 : INPUT "Any intervals where f(X) is undefined ";A$

LOCATE 17,1 : PRINT ""

LOCATE 11,1 : INPUT "a,b ";A,B
LOCATE 17,1 : PRINT "Number of points where undefined ";N

FOR I = 1 TO N
LOCATE 16,1 : PRINT "Point ";I : LOCATE 16,9 INPUT P(I)
LOCATE 16,9 : PRINT ""
NEXT I

CLS
LOCATE 10,20 : PRINT "Choose the scale for the X axis."

LOCATE 12,25 :PRINT "1. -16 to 16
LOCATE 14,25 :PRINT "2. -10 to 10
LOCATE 16,25 :PRINT "3. -5 to 5
LOCATE 18,25 :PRINT "4. -1 to 1
LOCATE 20,25 :INPUT "CHOICE ";CHOICE

IF CHOICE = 1 THEN SCALE = 20 : X = -16 : INC = .2 'INC is the increment

IF CHOICE = 2 THEN SCALE = 32 : X = -10 : INC = .1 'for the graphing loop

IF CHOICE = 3 THEN SCALE = 58 : X = -5 : INC = .05 'SCALE is the pixels

IF CHOICE = 4 THEN SCALE = 198: X = -1 : INC = .005 'per unit X

CLS
GOSUB 690 'to create the coordinate system

begin graphing the function

IF X >= A AND X <= B THEN PX1 = 320+SCALE*B : X = B 'since discont.
GOSUB 640
PX1 = 320 + SCALE * X : PY1 = 100 - SCALE/2 * Y 'first set of pixels
XX = X 'store initial X
K = 1
FOR X = XX TO -XX STEP INC
IF X >= A AND X <= B THEN PX1 = 320+SCALE*B : GOTO 580 'check for discont.
IF P(K) = X THEN : K = K + 1 : GOTO 580 'continuities
GOSUB 640
PX2 = 320 + SCALE * X : PY2 = 100 - SCALE/2 * Y
IF PY2 > 199 OR PY2 < 0 THEN 570
IF PY1 > 199 OR PY1 < 0 THEN 570
560  LINE (PX1, PY1) - (PX2, PY2) : LINE (PX1+1, PY1) - (PX2+1, PY2)
570  PX1 = PX2 : PY1 = PY2
580  NEXT X
590  ' NEXT X
600  LOCATE 23, 1 : INPUT "Another ": A$
610  IF A$ = "Y" OR A$ = "y" THEN GOTO 10
620  CLS
630  END
640  '****** SUBROUTINE CONTAINING THE FUNCTION *******
650  Y = (X^2 + 4) / (X^2 - 16)
660  RETURN
670  ' 670
680  '*********** SUBROUTINE TO DRAW THE AXES ***********
690  ' 700
700  ' 710  LINE (0, 100) - (640, 100) : LINE (0, 101) - (640, 101)
720  LINE (320, 0) - (320, 200) : LINE (321, 0) - (321, 200)
730  LOCATE 2, 39 : PRINT "Y": LOCATE 13, 77 : PRINT "X"
740  FOR I = 0 TO 320 STEP SCALE
750  LINE (320+I, 0) - (320+I, 200) : LINE (320-I, 0) - (320-I, 200)
760  NEXT I
770  ' 780
790  FOR I = 0 TO 100 STEP SCALE/2
800  LINE (0, 100+I) - (640, 100+I) : LINE (0, 100-I) - (640, 100-I)
820  RETURN
10 '********** TRIGONOMETRY, LAW OF SINES **********
20   
30 ' 
40 SCREEN 1 : CLS 
50 COLOR 4,7 
60 CLS 
70 LOCATE 6,10 : PRINT "PROGRAM SINELAW"
80 LOCATE 10,15 : PRINT "CHOICES"
90 LOCATE 12,5 : PRINT "1. TWO ANGLES AND A SIDE"
100 LOCATE 13,5 : PRINT "2. TWO SIDES AND AN OPPOSITE ANGLE"
110 LOCATE 14,5 : PRINT "3. TERMINATE PROGRAM"
120 LOCATE 18,5 : INPUT "ENTER CHOICE", CHOICE
130 ' 
140 WHILE CHOICE <> 3
150   
160 GOTO 60
170 WEND
180 SCREEN 0 : WIDTH 80
190 END
200 ' 
210 '********** SUBROUTINE TO DRAW 2 ANGLES & A SIDE **********
220 ' 
230 PI = 3.1415927#
240 CLS 
250 LOCATE 5,15: PRINT "input user's data "
260 LOCATE 5,5 : INPUT "1st ANGLE ", ALPHA
270 IF ALPHA > 90 THEN LOCATE 6,5 : PRINT "Must be <= 90" : GOTO 250
280 LOCATE 7,5 : INPUT "2nd ANGLE ", BETA
290 LOCATE 9,5 : INPUT "SIDE ", SIDEA
300 ' 
310 CLS 
320 LINE(20,125)-(300,125) 'draw side c 
330 ' 
340 ALPH2 = 90 - ALPHA 'complement of ALPHA for finding XB below
350 AL2RAD = ALPH2 * (PI / 100) : ALRAD = ALPHA * (PI / 100) 'find all angles
360 BET2 = 90 - BETA 'and convert to radians
370 BETRAD = BETA * (PI / 100) : BETRAD = BET2 * (PI / 100)
380 GAMMA = 180 - (ALPHA + BETA) : GAMRAD = GAMMA * (PI / 100)
390 ' 
400 SIDEB = SIN(BETRAD) * (SIDEA / SIN(ALRAD)) 'XB,YB are coordinates of
410 SDEC = SIN(GAMRAD) * (SIDEA / SIN(ALRAD))
420 YB = SIDEB*SIN(ALRAD) : PB = 125 - 14*YB 'top vertex
430 XB = SIDEB*SIN(AL2RAD) : PXB = 20 + 18*XB 'end of side b
440 IF PB < 5 AND PB < 310 AND PB > 10 THEN 460 ELSE LOCATE 10,15 : PRINT "Triangle won't fit on the screen." 
450 LOCATE 12,15 : INPUT "Please press enter." , A$ : GOTO 240
460 TXB = (125/14) * TAN(AL2RAD) : PTXB = 18*TXB +20 'current pixels for top
470 LINE (PTXB,0)-(XB,YP) : LINE (PXB,YP)-(20,125) 'end of side b
480 LOCATE 17,2 :PRINT ALPH
490 YA = SIDEC*SIN(BETRAD) : XA = SIDEA*SIN(BETRAD)
500 PXA = 300 - XA*18 : PYA = 125 - 14*YA 'pixels for top of side A
510 IF PXA > PXB THEN 530 ELSE LOCATE 10,15 : PRINT "Triangle won't fit on the screen." 
520 LOCATE 12,15 : INPUT "Please press enter." , A$ : GOTO 240
530 LINE (300,125)-(PX,A) 'draw side a
540 ' 
550 CTEN = CINT(10*(SIDEA / SIN(ALRAD) + SIN(GAMRAD)))
550 SIDE = CTEN / 10  'this avoids internal round off errors that exist
570 PC = INT(SIDE * 18 + 20)  'PC is pixel for right end of SIDE
580 LOCATE 17, 36 : PRINT BETA
590 LOCATE 13, 36 : PRINT SIDE
600 LOCATE 21, 5 : INPUT "Press enter to form the triangle. ", A$  
610 LOCATE 21, 5 : PRINT "  
620 LOCATE 17, 36 : PRINT "  
630 LOCATE 13, 36 : PRINT "  
640 ' the following loop shortens bottom side until it equals SIDE
650 FOR I = 0 TO 200 STEP 1
660 IF 300 - I = PC THEN GOTO 710
670 LINE (300 - I, 125) - (PXA, PYA), 0  
680 LINE (300 - I - 1, 125) - (PXA - 1, PYA)
690 PXA = PXA - 1  'moves top of side a into new position
700 NEXT I
710  
720 LINE (PX, PY) - (PTX, 0), 0  'erase unneeded portion of side b
730  
740 MIDX = (PX + 300 - I) / 2 : MIDY = (125 + PYA) / 2  'begin labelling
750 LOCATE 12*MIDY + 12*MIDX + 3 : PRINT SIDE 'the triangle
760  
770 PC = .12*((PC + 20) / 2)
780 IF PC < 1 THEN PC = 1 ELSE IF PC > 36 THEN PC = 36
790 LOCATE 17, PC : PRINT "c"
800 LOCATE 17, 12*PC + 2 : PRINT BETA
810 LOCATE 22, 20
820  
830 LOCATE .12*PY - 1, .12*PX + 1 : PRINT "c"
840 LCBX = .12*((PB + 20) / 2) - 2 : LCBY = .12*((125 + PY) / 2)
850 IF LCBX < 1 THEN LCBX = 1 : IF LCBY < 1 THEN LCBY = 1
860 LOCATE LCBY, LCBX : PRINT "b"
870  
880 'following block prints the law of sines for side b
890 LOCATE 20, 6 : PRINT "b" ; SIDE
900 LOCATE 21, 1 : PRINT "  
910 LOCATE 23, 1 : PRINT "sin" ; BETA ; "sin" ; ALPHA
920  
930 LOCATE 21, 27 : PRINT "  
940 LOCATE 21, 28 : INPUT "b = "; B
950 IF 10 * B <> CINT(10 * SIDE * SIN(BETR) / SIN(ALRAD)) THEN GOTO 930  
960  'if answer for B is correct
970 IF LCBX - 2 < 0 THEN LCBX = 3
980 LOCATE LCBY, LCBX : PRINT B 'place value on the triangle
990 LOCATE 20, 6 : PRINT "  
1000 LOCATE 21, 1 : PRINT "  
1010 LOCATE 23, 1 : PRINT "  
1020 LOCATE 21, 28 : PRINT "  
1030  
1040 LOCATE 21, 24 : PRINT "  
1050 LOCATE 21, 10 : INPUT "Angle C = "; G
1060 IF G <> GAMMA THEN GOTO 1040
1070 LOCATE .12*PY - 1, .12*PX + 1 : PRINT G
1080 LOCATE 21, 10 : PRINT "  
1090 LOCATE 20, 6 : PRINT "c" ; SIDE
1100 LOCATE 21, 1 : PRINT "  
1110 LOCATE 23, 1 : PRINT "sin" ; GAMMA ; "sin" ; ALPHA
1120 '  
1130 LOCATE 21,30 : PRINT "  
1140 LOCATE 21,28 : INPUT "c = "; c  
1150 IF c <> SIDEc THEN GOTO 1130  
1160 '  
1170 LOCATE 17,pxc : PRINT USING " ##.#"; sidec  
1180 LOCATE 20,6 : PRINT "  
1190 LOCATE 21,1 : PRINT "  
1200 LOCATE 23,1 : PRINT "  
1210 LOCATE 21,28 : PRINT "  
1220 LOCATE 21,5 : PRINT "The triangle is solved!"  
1230 LOCATE 23,10 : INPUT "Press enter for menu ", a$  
1240 RETURN  
1250 '************ SUBROUTINE FOR 2 SIDES AND AN ANGLE ************  
1260 '  
1270 CLS : pi = 3.1415927# : sub = 0  
1280 LOCATE 10,10 : INPUT "ALPHA = "; alpha  
1290 LOCATE 12,10 : INPUT "SIDE a = "; sidea  
1300 LOCATE 14,10 : INPUT "SIDE b = "; sideb  
1310 CLS  
1320 LINE (20,125) - (300,125)  
1330 ALPH2 = 90 - alpha  
1340 AL2RAD = alph2 * (pi / 180) : alrad = alpha * (pi / 180)  
1350 SINFTRD = SIN(alrad) * sideb / sidea : gosub 2480 'to compute beta  
1360 GAMMA = 180 - (alpha + beta) : gamrad = gamma * pi/180  
1370 SIDEc = SIN(gamrad) * sidea / SIN(alrad)  
1380 HEIGHT = SIN(alrad) * sidea  
1390 PYB = 125 - (14 * height)  
1400 PXB = 20 + (18 * SIN(al2Rad) * sideb)  
1410 line(20,125)-(pxb,pyb)  
1420 LOCATE 17,1 : PRINT ALPHA  
1430 LCBX = .12*(pxb+20)/2 - 1 : LCBy = .12*(pyb+125) / 2 - 1  
1440 LOCATE LCBy,LCBX : PRINT SIDEb  
1450 LCAX = .12*(pxb)+5 : LCAY = .12*(pyb)  
1460 LOCATE LCAY,LCAX : PRINT SIDEA  
1470 LCCY = .12*pyb-1 : IF LCCY < 1 THEN LCCY = 1  
1480 LCCX = .12*pxb  
1490 LOCATE LCCY,LCCX : PRINT "C"  
1500 '  
1510 PYA1 = PYb + (14 * SIN(.00) * SIDEa)  
1520 PxA1 = PXb + (18 * SIN PI/2 - .00 ) * SIDEa)  
1530 LINE (pxb,pyb)-(pxa1,pya1)  
1540 LOCATE 21,3 : INPUT "Press enter to form the triangle(s)", a$  
1550 LOCATE 21,3 : PRINT "  
1560 LOCATE LCAY,LCAX : PRINT "  
1570 FOR g = .1 TO 2.6 STEP .02 'loop increments the angle of side a off hor.  
1580 PYAZ = PYb + (14 * SIN(g) * SIDEa) 'figure new endpoints for side a  
1590 PxAZ = PXb + (18 * SIN PI/2 - g ) * SIDEa)  
1600 LINE (pxb,pyb)-(pxa1,pya1) ,0 'erase old side and draw new one  
1610 LINE (pxb,pyb) -(pxa2,pya2)  
1620 LINE (20,125) - (300,125) 'fill in accidentally erased points  
1630 IF ABS(BETRD - g) >= .01 AND ABS((PI-BETRD) - g ) >= .01 THEN 1700  
1640 ' if not, form a triangle  
1650 SUB = SUB + 1 'sub counts the number of triangles  
1660 IF SIDEA = SIDEb AND SUB > 1 THEN SUB = 1 : GOTO 1700 'isosceles tri.  
1670 IF SIDEA > SIDEb AND SUB > 1 THEN SUB = 1 : GOTO 1700 'side A > side B
1680 BEEP 'store the pixels for end of side a in arrays
1690 LINE (PX8,PYB)-(PXAI,PYAI),Z : PYA(SUB) = PYAI : PXA(SUB) = PXAI
1700 PXAI = PXAZ : PYAI = PYAZ 'delete old value with new
1710 IF PYA(SUB) <> 0 THEN LINE (PX8,PYB)-(PXAI,PYAI),Z
1720 NEXT G
1730
1740 LOCATE 20,1 : INPUT "How many triangles are possible ":NUM
1750 IF NUM <> SUB THEN BEEP : GOTO 1740
1760 IF NUM=0 THEN LOCATE 22,10:PRINT "CORRECT !!":FOR I=1 TO 1000:NEXT :RETURN
1770 LOCATE 20,1 : PRINT ".
1780 LOCATE 20,1 : INPUT "Press enter to look at first triangle. ":A$nnLOCATE 20,1 : PRINT ""n
1790 LOCATE 20,1 : PRINT ""nLOCATE 20,1 : INPUT "How many triangles are possible ":NUM
1800 IF PYA(SUB) <> 0 THEN LINE (PX8,PYB)-(PXAI,PYAI),Z
1810 NOE LOCATE 22,10:PRINT "CORRECT !!":FOR I=1 TO 1000:NEXT :RETURN
1820 LINE (PX8,PYB)-(PXAI,PYAI),0 : LINE (PX8,PYB)-(PXAI(PYAI),Z)
1830 LINE(Z0,12S)-(PX8,PYB) : redraw side B
1840 LINE (PX8,PYB)-(PXAI(PYAI))
1850 LINE (PX8(PY8),12S)-300,12S),0 'erases extra part of bottom side
1860 FOR I = PYB TO 120 STEP 10 : LINE (PX8,I)-(PX8,I+5),Z : NEXT I'draws height
1870
1880 LOCATE 17,1 : PRINT ALPHA 'label the triangle
1890 LOCATE 17,.12*PXAK(I)+1 : PRINT "B"
1900 LCAK = .12*(PX8+PXAK(I))/2 + 2 ; LCAK = .12*(PYB+125)/2 -1
1910 LOCATE LCAK,LCAK : PRINT SIDEA
1920 PX8 = .12*(PXAK(I)+20)/2 : LOCATE 17,PX8 : PRINT "c"
1930 LOCATE 20,4 : PRINT "sin B ; sin ";ALPHA
1940 begin solving the triangle
1950 LOCATE 21,1 : PRINT ".nnLOCATE 23,1 : PRINT " " ;SIDEA ; "nnSIDEA
1960 LOCATE 21,1 : PRINT ""
1970
1980 LOCATE 21,30 : PRINT ""
1990 LOCATE 21,28 : INPUT "B = ";B
2000 IF B <> BETA THEN BEEP: GOTO 1980
2010 LOCATE 17,.12*PXAK(I)+1 : PRINT BETA
2020
2030 LOCATE 20,4 : PRINT ""
2040 LOCATE 21,1 : PRINT ""
2050 LOCATE 23,1 : PRINT ""
2060 LOCATE 21,28 : PRINT ""
2070
2080 LOCATE 21,14 : PRINT ""
2090 LOCATE 21,5 : INPUT "Angle C = ":C
2100 IF CINT(10*C) <> CINT(10*GAMMA) THEN BEEP: GOTO 2080
2110 LOCATE LCCY,LCCX : PRINT C
2120
2130 LOCATE 21,5 : PRINT ""
2140
2150 LOCATE 20,4 : PRINT "c " ;SIDEA
2160 LOCATE 21,1 : PRINT "sin " ;C ; " SIN ";ALPHA
2170
2180
2190 LOCATE 21,30 : PRINT ""
2200 LOCATE 21,28 : INPUT "c = ";SDCEE
2210 SIDE = CINT( 10 * SIN(GAMMARAD) * SIDEA / SIN(ALRAD)) / 10
2220 IF CINT(10*SDCEE) <> CINT(10*SIDE) THEN BEEP: GOTO 2180
2230 LOCATE 17,PX8 : PRINT USING "##.##";SIDEA
240 LOCATE 20,4 : PRINT " This triangle is solved!"
250 LOCATE 21,1 : PRINT " 
260 LOCATE 23,1 : PRINT " 
270 LOCATE 21,28 : PRINT " 
280
t 290 LOCATE 21,5 : PRINT "This triangle is solved!"
300 IF SUB = 1 THEN LOCATE 23,5 : INPUT "Press enter for command menu.",A$ : RETURN
310 IF K = 2 THEN LOCATE 23,5 : INPUT "Press enter for command menu.",A$ : RETURN
320 LOCATE 23,5 : INPUT "Press enter for the second one.",A$
330
t
334 K = 2 signals that second triangle is solved
350 LOCATE 17,.12*PXA(1)+1: PRINT " 
360 LOCATE LCCY,LCCX : PRINT " 
370 LOCATE 17,PXC : PRINT " 
380 LOCATE LCA,LCAX : PRINT " 
390 LINE (PXB,PYB)-(PXA(1),PYA(1)),0 
400 LINE (PXB,PYB)-(PXA(2),PYA(2)) 
410 LOCATE 21,5 : PRINT " 
420 LOCATE 23,5 : PRINT " 
430
t
440 BETA = 180 - BETA : GAMMA = 180 - (ALPHA + BETA) 'compute data for 2nd tri.
450 BTR = BETA * PI / 180 : GAMMA = GAMMA * PI / 180
460 GOTO 1890
470 RETURN
480 '*************** SUBROUTINE TO FIGURE INVERSE SINE ******************
490 INC = .01 'the closer the SINE is to 1, the more accurate the increment
500 LOCATE 10,5 : PRINT "Computing, please stand by ...
510 IF SINBTR >= .7 THEN B = .77 : INC = .007 'must be
520 IF SINBTR > 1 THEN BETR = 1 : GOTO 2880
530 IF SINBTR >= .8 THEN B = .92 : INC = .006
540 IF SINBTR >= .95 THEN B = 1.25 : INC = .003
550 IF SINBTR >= .99 THEN B = 1.41 : INC = .0008
560 IF SINBTR >= .999 THEN B = 1.54 : INC = .0001
570
t
580 'each successive loop narrows in the angle whose sine is equal to SINBTR
590 FOR I = B TO 1.58 STEP .01
600 IF ABS( SIN(I) - SINBTR ) < 5*INC THEN GOTO 2620
610 NEXT I
620 IF ABS(SIN(I) - SINBTR) < 1E-08 THEN BETR = I : GOTO 2860
630 FOR J = I-.05 TO I+.1 STEP .005
640 IF ABS( SIN(J) - SINBTR ) < INC THEN GOTO 2660
650 NEXT J
660 IF ABS(SIN(J) - SINBTR) < 1E-08 THEN BETR = J : GOTO 2860
670 FOR K = J-.01 TO J+.05 STEP .001
680 IF ABS( SIN(K) - SINBTR ) < INC/5 THEN 2700
690 NEXT K
700 IF ABS(SIN(K) - SINBTR) < 1E-08 THEN BETR = K : GOTO 2860
710 FOR L = K-.005 TO K+.01 STEP .0005
720 IF ABS( SIN(L) - SINBTR ) < INC/10 THEN 2740
730 NEXT L
740 IF ABS(SIN(L) - SINBTR) < 1E-08 THEN BETR = L : GOTO 2860
750 FOR M = L-.001 TO L+.0001 STEP .0001
760 IF ABS( SIN(M) - SINBTR ) < INC/50 THEN 2780
770 NEXT M
780 IF ABS(SIN(M) - SINBTR) < 1E-08 THEN BETR = M : GOTO 2880
790 FOR N = M-.0005 TO M+.0005 STEP .00005
2800 IF ABS(SIN(N) - SINBTRD) < INC/500 THEN 2820
2810 NEXT N
2820 IF ABS(SIN(N) - SINBTRD) < 1E-08 THEN BETRD = N : GOTO 2880
2830 FOR P = N-.0001 TO N+.002 STEP .00001
2840 IF ABS(SIN(P) - SINBTRD) < INC/1000 THEN 2860
2850 NEXT P
2860 '  
2870 BETRD = P
2880 BETA = BETRD * (180/3.1415927#)
2890 BETA = CINT(BETA*10) / 10
2900 '  
2910 LOCATE 10,5 : PRINT "  
2920 RETURN
GRAPHING TRIG FUNCTIONS

STORED UNDER: TRIGRAF
JUNE 15, 1985

CLEAR,.,32768! : SCREEN 6 : COLOR 3,4
SCREEN 2
CLS
LOCATE 15,10:INPUT "Please press CAPS LOCK if not previously done, then ENTE R. " ,A$
CLS
LOCATE 5,15:PRINT "COMMAND MENU"
LOCATE 8,10:PRINT "1. Graph of Sine Function"
LOCATE 10,10:PRINT "2. Graph of Cosine Function"
LOCATE 12,10:PRINT "3. Graph of Tangent Function"
LOCATE 14,10:PRINT "4. Graph of Cotangent Function"
LOCATE 16,10:PRINT "5. Graph of Secant Function"
LOCATE 18,10:PRINT "6. Graph of Cosecant Function"
LOCATE 20,10:PRINT "7. Terminate Program"
LOCATE 22,10:INPUT "ENTER YOUR CHOICE " , CMD
this is an error trap for their choice!
WHILE CMD <> FIX(CMD) OR CMD < 1 OR CMD > 7
LOCATE 24,10:PRINT "Please enter a 1, 2, 3, 4, 5, 6, or 7"
GOTO 210
WEND
WEND
WHILE CMD <> 7
ON CMD GOSUB 370,710,1040,1460,2050,2630
LOCATE 23,1:INPUT "Press enter to return to the command menu. ",B$
GOTO 110
WEND
SCREEN 0
END

********** SUBROUTINE TO GRAPH SINE WAVES **********
LOCATE 10,10 : PRINT "In general, we'll be graphing equations of the
LOCATE 11,10 : PRINT "form  Y = a SIN b ( X - c )"
LOCATE 13,10 : PRINT "When you see the number 690, followed by an equation,
LOCATE 14,10 : PRINT "please EDIT that EQUATION as desired, then press enter
twice."
LOCATE 17,10 : INPUT "Press enter to continue. ",A$
CLS : GOSUB 3200
LOCATE 23,1 : PRINT "GOTO 490"
LOCATE 22,1 : EDIT 690
LOCATE 23,1 : PRINT ""
PI = 3.1415927#
begin graphing their equation
X = -PI : GOSUB 690 'to figure the Y value for the initial X of -PI
PX1 = 150 + 40 * X : PY1 = 100 - 20 * Y 'compute first PIXELS
FOR X = -PI TO 3*PI STEP .05
560  GOSUB 690
570  PX2 = 150 + 40*X : PY2 = 100 - 20 * Y
580  IF PY2 > 199 OR PY2 < 0 THEN GOTO 600
590  LINE (PX1,PY1)-(PX2,PY2)
600  PX1 = PX2 : PY1 = PY2
610  NEXT X
620  ' subroutine containing the equation
630  Y = 1 + SIN ( 1 * ( X + 0 ))
640  RETURN
650  ' SUBROUTINE TO GRAPH COSINE WAVES
660  CLS
670  LOCATE 10,10 : PRINT "In general, we'll be graphing equations of the"
680  LOCATE 11,10 : PRINT "form Y = a COS b ( X - c )"
690  LOCATE 13,10 : PRINT "When you see the number 1010, followed by an equation,"
700  LOCATE 14,10 : PRINT "please EDIT that EQUATION as desired, then press enter twice."
710  LOCATE 17,10 : INPUT "Press enter to continue. ":A$
720  ' subroutine containing the function
730  Y = 1 + COS ( 1 * ( X + 0 ))
740  RETURN
750  ' SUBROUTINE TO GRAPH TAN FUNCTION
760  CLS
770  LOCATE 10,10 : PRINT "In general, we'll be graphing equations of the"
780  LOCATE 11,10 : PRINT "form Y = a TAN b ( X - c )"
790  LOCATE 13,10 : PRINT "When you see the number 1340, followed by an equation,"
800  LOCATE 14,10 : PRINT "please EDIT that EQUATION as desired, then press enter twice."
810  LOCATE 17,10 : INPUT "Press enter to continue. ":A$
820  ' subroutine containing the function
830  Y = 1 + TAN ( 1 * ( X + 0 ))
840  RETURN
CLS : GOSUB 3200
LOCATE 23,1 : PRINT "GOTO 1160"
LOCATE 22,1 : EDIT 1360
LOCATE 23,1 : PRINT ""
PI = 3.1415927#
begin graphing their equation
X = -PI : GOSUB 1360 'to figure the Y value
PX1 = 150 + 40 * X : PY1 = 100 - 20 * Y
FOR X = -PI TO 4 * PI STEP .05
GOSUB 1360
Y = 1 * TAN( 3 * ( X - PI/2 ) )
RETURN
Asymptote Subroutine
COUNT = COUNT + 1 : IF COUNT > 1 THEN RETURN
FOR I = 0 TO 200 STEP 10
PX = 150 + 40 * X
LINE (PX, I) - (PX, I+5)
NEXT I
RETURN
---------- SUBROUTINE TO GRAPH COTANGENT FUNCTION ----------
CLS
LOCATE 10,10 : PRINT "In general, we'll be graphing equations of the"
LOCATE 11,10 : PRINT "form Y = a COT b( X - c )"
LOCATE 13,10 : PRINT "When you see a,b ? enter those two numbers, with the"
"comma."
LOCATE 14,10 : PRINT "When you see c ? enter the value of c, then press ent er."
LOCATE 15,10 : PRINT "When entering c, use PI anyplace that value is desire d."
LOCATE 16,10 : PRINT "For example; c ? -PI/2 would be accepted."
LOCATE 18,10 : INPUT "Press enter to continue. " , A$
CLS
SIGN = 1 : PI = 3.1415927#
LOCATE 22,1 : PRINT "Y = a COT b( X - c )"
LOCATE 23,1 : INPUT "a,b "; A,B
LOCATE 23,1 : PRINT ""
LOCATE 23,1 : INPUT "c "; C$
LOCATE 23,1 : PRINT ""
IF C$ = "0" THEN C = 0 : GOTO 1700 'the following block determines
IF LEFT$(C$, 1) = "-" THEN SIGN = -1 'the shift from the string C$
IF C$ = "PI" OR C$ = "-PI" THEN GOTO 1690
1680 C = SIGN * PI / (VAL(RIGHT$(C$,1))) : GOTO 1700
1690 C = PI * SIGN
1700 IF SIGN = -1 THEN SIGN$ = "( X - " ELSE SIGN$ = "( X + "
1710 LENG = LEN(C$) : IF SIGN = -1 THEN LENG = LENG - 1
1720 LOCATE 22,1 : PRINT "Y =";A$;"COT";B$;SIGN$;RIGHT$(C$,LENG)$
1730 .
1740 ' begin graphing their equation
1750 X = - PI ' initialize X, then get first pixel point
1760 COT = TAN ( B * ( X - C ) ) : IF COT = 0 THEN X = X + .1 : GOTO 1760
1770 Y = A$ * ( 1 / TAN ( B * ( X - C ) ) )
1780 PX1 = 150 + 40 * X : PY1 = 100 - 20 * Y
1790 FOR X = -PI TO 4*PI STEP .08
1800 IF TAN ( B*(X-C) ) = 0 THEN 1890
1810 Y = A$ * ( 1 / TAN ( B * ( X - C ) ) )
1820 IF ABS(Y) > 15 THEN COUNT = COUNT + 1 : GOSUB 1980 ' to draw asymptote
1830 PX2 = 150 + 40*X : PY2 = 100 - 20*Y
1840 IF PY2 > 199 OR PY2 < 0 THEN GOTO 1880
1850 IF PY1 > 199 OR PY1 < 0 THEN 1880
1860 LINE (PX1,PY1)-(PX2,PY2)
1870 COUNT = Q
1880 PX1 = PXZ : PY1 = PYZ
1890 NEXT X
1900 .
1910 LOCATE 23,1 : PRINT " 
1920 LOCATE 23,1 : INPUT "Another ";A$
1930 IF A$ = "Y" OR A$ = "y" THEN 1570
1940 GOTO 300
1950 .
1960 LOCATE 3,10 : PRINT "Press enter to continue .... A$
1970 .
1980 .
1990 .
2000 FOR I = 0 TO 200 STEP 10
2010 PX = 150 + 40*X
2020 LINE (PX,I)-(PX,I+5)
2030 NEXT I
2040 RETURN
2050 '*********** SUBROUTINE TO GRAPH SECANT FUNCTION ***********
2060 .
2070 CLS
2080 LOCATE 10,10 : PRINT "In general, we'll be graphing equations of the
2090 LOCATE 11,10 : PRINT "form \ Y = a SEC b ( X - c )
2100 LOCATE 13,10 : PRINT "When you see a,b ? enter those two numbers, with the
2110 LOCATE 14,10 : PRINT "When you see c ? enter the value of c, then press ent- er.
2120 LOCATE 15,10 : PRINT "When entering c, use PI anyplace that value is desire-
d."
2130 LOCATE 16,10 : PRINT "For example; c < PI/2 would be accepted."
2140 LOCATE 18,10 : INPUT "Press enter to continue. ",A$
2150 .
2160 CLS : GOSUB 3200
2170 SIGN = 1 : PI = 3.1415927#  
2180 LOCATE 22,1 : PRINT "Y = a SEC b ( X - c )
2190 LOCATE 23,1 : INPUT " a,b ";A$;B$
2200 LOCATE 23,1 : PRINT " 
2210 LOCATE 23,1 : INPUT "c ";C$
2220 LOCATE 23,1 : PRINT " 
2230 .
2230 ' the following block determines
2240 IF C$ = "0" THEN C$ = 0 : GOTO 2290 'the shift from the string C$
2250 IF LEFT$(C$,1) = "-" THEN SIGN = -1
2260 IF C$ = "PI" OR C$ = "-PI" THEN GOTO 2280
2270 C$ = SIGN * PI / (VAL(RIGHT$(C$,1))) : GOTO 2290
2280 C$ = PI * SIGN
2290 IF SIGN = -1 THEN SIGN$ = "( X + " ELSE SIGN$ = "( X - "
Z300 LNGTH = LEN(C$) : IF SIGN = -1 THEN LNGTH = LNGTH - 1
Z310 LOCATE 22,1 : PRINT "Y =";A;"SEC";D;SIGN$;RIGHT$(C$,LNGTH)"
2320 ' begin graphing their equation
2330 X = - PI 'initialize X, then get first pixel point
2340 CSE = COS ( B * ( X - c )) : IF CSE = 0 THEN X = X + .1 : GOTO 2350
2350 Y = A * CSE
2360 Y = A* 1 / CSE
2370 PX1 = 150 + 40 * X : PY1 = 100 - 20 * Y
2380 FOR X = -PI TO 4*PI STEP .05
2390 IF COS ( B*(X-C)) = 0 THEN 2480
2400 Y = A * ( 1 / COS ( B * ( X - C )))
2410 IF ABS(Y) > 18 THEN COUNT = COUNT + 1 : GOSUB 2500 ' to draw asymptote
2420 PX2 = 150 + 40*X : PY2 = 100 - 20 * Y
2430 IF PY2 > 199 OR PY2 < 0 THEN GOTO 2470
2440 IF PY1 > 199 OR PY1 < 0 THEN 2470
2450 LINE (PX1,PY1)-(PX2,PY2)
2460 COUNT = 0
2470 PX1 = PX2 : PY1 = PY2
2480 NEXT X
2490 ' begin graphing their equation
2500 LOCATE 23,1 : PRINT ""
2510 LOCATE 23,1 : INPUT "Another ";A$
2520 IF A$ = "Y" OR A!C = "y" THEN 2160
2530 GOTO 300
2540 ' Asymptote Subroutine
2550 ' Subroutine to graph cosecant function
2560 CLS
2570 LOCATE 10,10 : PRINT "In general, we'll be graphing equations of the
2580 LOCATE 11,10 : PRINT "form Y = a CSC b ( X - c )"
2590 LOCATE 13,10 : PRINT "When you see a,b ? enter those two numbers, with the
2600 LOCATE 14,10 : PRINT "When you see c ? enter the value of c, then press enter.
2610 LOCATE 15,10 : PRINT "When entering c, use PI anyplace that value is desire
2620 LOCATE 16,10 : PRINT "For example: c ? -PI/2 would be accepted."
2630 LOCATE 18,10 : INPUT "Press enter to continue. ",A$
2640 ' Subroutine to graph cosecant function
2650 CLS
2660 LOCATE 10,10 : PRINT "In general, we'll be graphing equations of the
2670 LOCATE 11,10 : PRINT "form Y = a CSC b ( X - c )"
2680 LOCATE 13,10 : PRINT "When you see a,b ? enter those two numbers, with the
2690 LOCATE 14,10 : PRINT "When you see c ? enter the value of c, then press ent
2700 LOCATE 15,10 : PRINT "When entering c, use PI anyplace that value is desire
2710 LOCATE 16,10 : PRINT "For example: c ? -PI/2 would be accepted."
2720 LOCATE 18,10 : INPUT "Press enter to continue. ",A
2730 ' Subroutine to graph cosecant function
2740 CLS : GOSUB 3200
2750 SIGN = 1 : PI = 3.1415927#
2760 LOCATE 22,1 : PRINT "Y = a CSC b ( X - c )"
2770 LOCATE 23,1 : INPUT "a,b ";A,B
2780 LOCATE Z3, 1: PRINT " "
2790 LOCATE Z3, 1: INPUT "c "; C$
2800 LOCATE Z3, 1: PRINT " 
2810 ' 
2820 IF C$ = "0" THEN C = 0: GOTO 2870 ' the following block determines
2830 IF LEFT$(C$, 1) = "-" THEN SIGN = -1 ' the shift from the string C$
2840 IF C$ = "PI" OR C$ = "-PI" THEN GOTO 2860
2850 C = SIGN * PI / (VAL(RIGHT$(C$, 1))) : GOTO 2870
2860 C = PI * SIGN
2870 IF SIGN = -1 THEN SIGNS := (X - 1)
2880 ELSE SIGN$ := (X - )
2890 LENGTH = LEN(C$) : IF SIGN = -1 THEN LENGTH = LENGTH - 1
2900 LOCATE Z2, 1 : PRINT "Y := A+(CSC)B;SIGN;RIGHT$(C$,LENGTH);"
2910 ' begin graphing their equation
2920 X = - PI ' initialize X, then get first pixel point
2930 Y = A* 1 / SNE
2940 PX1 = 150 + 40 * X : PY1 = 100 - 20 * Y
2950 FOR X = - PI TO 4*PI STEP .05
2960 IF SIN (B*(X-C)) = 0 THEN 3050
2970 Y = A * (1 / SIN (B*(X - C)))
2980 IF ABS(Y) > 18 THEN COUNT = COUNT + 1: GOSUB 3130 ' to draw asymptote
2990 PX2 = 150 + 40*X : PY2 = 100 - 20 * Y
3000 IF PY2 > 199 OR PY2 < 0 THEN GOTO 3040
3010 IF PY1 > 199 OR PY1 < 0 THEN 3040
3020 LINE (PX1,PY1)-(PX2,PY2)
3030 COUNT := 0
3040 PX1 = PX2 : PY1 = PY2
3050 NEXT X
3060 ' 
3070 LOCATE Z3, 1: PRINT " 
3080 LOCATE Z3, 1: INPUT "Another "; A$
3090 IF A$ = "Y" OR A$ = "y" THEN 2740
3100 GOTO 300
3110 ' 
3120 RETURN
3130 ' Asymptote Subroutine
3140 IF COUNT > 1 THEN RETURN
3150 FOR I = 0 TO 200 STEP 10
3160 PX = 150 + 40*I
3170 LINE (PX,97)-(PX,103)
3180 NEXT I
3190 RETURN
3200 '******************** SUDROUTINE TO DRAW GRAPH ********************
3210 ' 
3220 'SCREEN 2
3230 ' first draw the axes
3240 LINE (0,100)-(640,100) : LINE (0,101)-(640,101)
3250 LINE (150,0)-(150,200) : LINE (151,0)-(151,200)
3260 LOCATE 14,77 : PRINT "X" : LOCATE 2,18 : PRINT "Y"
3270 '
3280 ' mark off units of PI / 2 on the X axis
3290 PI = 3.1415927#
3300 FOR X = -(PI) TO (4*PI) STEP (PI/2)
3310 I = 150 + 40*X
3320 LINE (I,97)-(I,103) : LINE (I+1,97)-(I+1,103)
3330 NEXT X
3340 ' mark units of 1 on the Y axis
3350 FOR Y = -5 TO 5
3360   I = 100 - 20*Y
3370 LINE (145,I)-(155,I) : LINE (145,I+1)-(155,I+1)
3380 NEXT Y
3390 ' 3400 RETURN
3410 '
**POLAR COORDINATES PROGRAM**

**BY MIKE MOSIER**

**STORED UNDER: POLAR**

**JUNE 1985**

```
10 ** ********** POLAR COORDINATES PROGRAM **********
20 **
30 **
40 **
50 **
60 **
70 CLEAR,H,32768: SCREEN 6: COLOR 3,4
80 SCREEN 2
90 **
100 CLS
110 **
120 LOCATE 10,30: PRINT "COMMAND MENU"
130 LOCATE 12,20: PRINT "1. PLOT INDIVIDUAL POINTS IN POLAR COORDINATES"
140 LOCATE 14,20: PRINT "2. GRAPH A FUNCTION IN POLAR COORDINATES"
150 LOCATE 16,20: PRINT "3. EXIT PROGRAM"
160 **
170 LOCATE 20,20: INPUT "Enter your choice. ", CHOICE
180 WHILE CHOICE <> 3
190 ON CHOICE GOSUB 250, 450
200 GOTO 60
210 WEND
220 **
230 SCREEN 0: CLS
240 END
250 ** SUBROUTINE TO PLOT POINTS **
260 **
270 CLS
280 GOSUB 890
290 LOCATE 21,1: PRINT "ENTER POINT"
300 LOCATE 22,3: PRINT "( , )"
310 LOCATE 22,4: INPUT "", P, THETA
320 LOCATE 21,1: PRINT " "
330 LOCATE 22,3: PRINT " "
340 LOCATE 22,4: PRINT " 
350 THETA = THETA * PI/180
360 X = P * COS (THETA): Y = P * SIN (THETA)
370 PX1 = 300 + 20 * X: PY1 = 100 - 8.399999 * Y
380 CIRCLE(PX1,PY1),2: CIRCLE(PX1,PY1),3: CIRCLE(PX1,PY1),5
390 **
400 LOCATE 22,1: INPUT "Another point ": A$
410 LOCATE 22,1: PRINT " 
420 IF A$ = "Y" OR A$ = "y" THEN 290
430 **
440 RETURN
450 ** SUBROUTINE TO GRAPH POLAR FUNCTIONS **
460 **
470 CLS
480 LOCATE 10,1: INPUT "Will your equation include a square root ": ROOT$
490 IF ROOT$ = "Y" OR ROOT$ = "y" THEN 540
500 LOCATE 10,1: PRINT "Make any desired change in the following equation"
510 LOCATE 11,1: PRINT "then hit ENTER TWICE. ( or default through )"
520 LOCATE 16,1: PRINT "GOTO 590"
530 LOCATE 15,1: EDIT 830
540 CLS: LOCATE 10,1: PRINT "Let PSQR be equal to P squared, and type in"
550 LOCATE 12,1: PRINT "your equation. "
```
560 LOCATE 15,1 : PRINT "GOTO 500"
570 LOCATE 14,1 : EDIT 850
580 ROO$ = "Y"
590 CLS:LOCATE 15,1 : INPUT "Theta will begin at 0. Enter stopping point as PI, 2PI, 3PI, ...": DONE$ 
600 IF DONE$ = "PI" OR DONE$ = "pi" THEN DONE = 3.1415927# : GOTO 620
610 DONE = VAL(LEFT$(DONE$,1)) * 3.1415927#
620 GOSUB 890
630 GOSUB 800
640 PX1 = 300 + 20*P : PY1 = 100
650 FOR THETA = 0 TO DONE STEP .05
660 GOSUB 800
670 X = P * COS (THETA) : Y = P * SIN (THETA)
680 PXZ = 300 + 20 * X : PY2 = 100 - 8.399999 * Y
690 LINE (PX1,PY1) - (PXZ, PY2 )
700 LINE (PX1,PY1+1) - (PXZ, PY2+1)
710 LINE (PX1+1,PY1+1) - (PXZ+1, PY2+1)
720 PX1 = PXZ : PY1 = PY2
730 NEXT THETA
740
750 LOCATE 23,1 : INPUT "Another equation ": A$
760 LOCATE 23,1 : PRINT ""
770 IF A$ = "Y" OR A$ = "y" THEN GOSUB 450
780 ELSE GOTO 60
790
800 "************ SUBROUTINE TO GIVE FUNCTIONAL VALUES"
810
820 IF ROO$ = "Y" OR ROO$ = "y" THEN 850
830 P = 10*SIN (3*THETA)
840 RETURN
850 PSQR = -40* COS (3*THETA)
860 IF PSQR < 0 THEN P = 0 : RETURN
870 P = SQR( PSQR )
880 RETURN
890 "************ POLAR COORDINATES GRAPH PAPER ************"
900
910 PI = 3.1415927#
920 CLS
930
940 FOR I = 0 TO 220 STEP 20
950 CIRCLE(300,100),I,2
960 NEXT I
970
980 LINE (55,100)-(545,100),2
990 LINE (300,0)-(300,200),2
1000
1010 FOR ANGLE = 0 TO 360 STEP 30
1020 ANG = ANGLE * PI/180
1030 X = 12 * COS(ANG) : Y = 12 * SIN(ANG)
1040 PX1 = 300 + 20*X : PY1 = 100 - 8.399999 * Y
1050 LINE (PX1,PY1) - (300,100),2
1060 NEXT ANGLE
1070
1080 FOR ANGLE = 45 TO 315 STEP 45
1090 ANG = ANGLE * PI/180
1100 X = 12 * COS(ANG) : Y = 12 * SIN(ANG)
1110 PX1 = 300 + 20*X : PY1 = 100 - 8.399999 * Y
1120 LINE (PX1,PY1) - (300,100),2
1130 NEXT ANGLE
1140
1150 RETURN
LIMITS INSTRUCTION

STORED UNDER: LIMITS
JULY 30, 1984

CLEAR,,32768!: SCREEN 6: COLOR 3,4
SCREEN 2
CLS
LOCATE 3,15:PRINT "PROGRAM LIMITS"
LOCATE 5,15:PRINT "COMMAND MENU (Note that choice 2. must be made before 3
or 4)
LOCATE 8,10:PRINT "1. Explanation of Limits"
LOCATE 10,10:PRINT "2. Input a Function"
LOCATE 12,10:PRINT "3. Draw and Figure an Epsilon for a given Delta"
LOCATE 14,10:PRINT "4. List a table of values as X approaches given number"
LOCATE 16,10:PRINT "5. Terminate Program"

LOCATE 22,10:INPUT "ENTER YOUR CHOICE ", CMD
WHILE CMD > 2 AND CMD < 5
IF TRAP = 2 THEN 230
LOCATE 24,10:PRINT "You must choose 2. before 3 or 4"
GOTO 170
WEND
WHILE CMD <> 5
ON CMD GOSUB 370, 1180, 1420, 2330
LOCATE 25,10:INPUT "Press enter to return to the command menu. ",B$ GOTO 80
WEND
CLS
LOCATE 5,10: PRINT "When talking about the limit of a function,"
LOCATE 6,10: PRINT "we mean that as we get closer to some value of X,"
LOCATE 7,10: PRINT "the function gets closer to some specific value."
LOCATE 9,10: PRINT "This specific value, if it exists, is called the"
LOCATE 10,10: PRINT "LIMIT of the Function, or f(x), as x approaches"
LOCATE 11,10: PRINT "a given value. In the following example, we'll let"
LOCATE 12,10: PRINT "X get closer and closer to 5, knowing that f(5) = 3."
LOCATE 14,10: PRINT "If the graph closes in on 3, as x closes in on 5, then"
LOCATE 15,10: PRINT "3 must be the limit of f(x) as x approaches 5."
LOCATE 20,10: INPUT "Press enter for an example. ",A$

CLS
LOCATE 5,10: PRINT "LIMIT of the Function, or f(x), as x approaches"
LOCATE 6,10: PRINT "we mean that as we get closer to some value of X,"
LOCATE 7,10: PRINT "the function gets closer to some specific value."
LOCATE 9,10: PRINT "This specific value, if it exists, is called the"
LOCATE 10,10: PRINT "LIMIT of the Function, or f(x), as x approaches"
LOCATE 11,10: PRINT "a given value. In the following example, we'll let"
LOCATE 12,10: PRINT "X get closer and closer to 5, knowing that f(5) = 3."
LOCATE 14,10: PRINT "If the graph closes in on 3, as x closes in on 5, then"
LOCATE 15,10: PRINT "3 must be the limit of f(x) as x approaches 5."
LOCATE 20,10: INPUT "Press enter for an example. ",A$

CLS
COUNT = 0: VRTSHFT = 2
LOCATE 5,10: PRINT "When talking about the limit of a function,"
LOCATE 6,10: PRINT "we mean that as we get closer to some value of X,"
LOCATE 7,10: PRINT "the function gets closer to some specific value."
LOCATE 9,10: PRINT "This specific value, if it exists, is called the"
LOCATE 10,10: PRINT "LIMIT of the Function, or f(x), as x approaches"
LOCATE 11,10: PRINT "a given value. In the following example, we'll let"
LOCATE 12,10: PRINT "X get closer and closer to 5, knowing that f(5) = 3."
LOCATE 14,10: PRINT "If the graph closes in on 3, as x closes in on 5, then"
LOCATE 15,10: PRINT "3 must be the limit of f(x) as x approaches 5."
LOCATE 20,10: INPUT "Press enter for an example. ",A$

CLS
COUNT = COUNT + 1
LINE (1,150) - (630,150)
LINE (75,1) - (75,165)
LINE (76,1) - (76,165)
560  PX3 = .12 * 75  : LOCATE 2, PX3: PRINT "Y"
570  LOCATE 20, 77: PRINT "X"
580  
590  X = 1  : GOSUB 1160  ' to figure Y
600  PX1 = 75 + 48 * X  : PY1 = 150 - 24 * Y
610  X = 9  : GOSUB 1160
620  PX4 = 75 + 48 * X  : PY4 = 150 - 24 * Y
630  HASH = 75 + 48 * 5  ' pixel for the dotted line to X
640  PX3 = .125 * HASH: LOCATE 21, PX3: PRINT 5
650  LIM = 150 - 24 * 3  ' pixel for the dotted line to y
660  FOR I = 150 TO LIM  STEP -10  : LINE(HASH, I) - (HASH, I + 3), I = NEXT I
670  FOR I = 75 TO HASH  STEP 20  : LINE(I - 5, LIM) - (I, LIM), I = NEXT I
680  PSET(HASH, LIM): PSET(HASH + 1, LIM): PSET(HASH + 1, LIM + 1): PSET(HASH, LIM + 1)
700  
710  LOCATE 23, 15: INPUT "Press enter to continue." , A$
720  LOCATE 23, 15: PRINT ""
730  FOR I = 0 TO 3.9  STEP .05
740  X = I + I  : GOSUB 1160  ' approach XL from the left
750  PX2 = 75 + 48 * X  : PY2 = 150 - 24 * Y
760  IF I < 2 THEN 810
770  LINE(65, PY1) - (65, PY2)  ' these draw the y-arrows
780  LINE(65, PY1 - 10) - (65, PY1 - 5), 0  '
790  PRESET(63, PY1 - 1) : PRESET(67, PY1 - I)
800  PSET(63, PY2 - 1) : PSET(67, PY2 - 1)
810  PRESET(PX1 - 3, 154) : PRESET(PX1 - 3, 155)
820  PSET(PX2 - 3, 154) : PSET(PX2 - 3, 155)
830  LINE(PX1 - 15, 155) - (PX1 - 10, 155), 0
840  LINE(PX1, 155) - (PX2, 155)
850
860  LINE(PX1, PY1) - (PX2, PY2)
870  LINE(PX1, PY1 + 1) - (PX2, PY2 + 1)
880  PX1 = PX2  : PY1 = PY2
890
900  X = 9 - I  : GOSUB 1160
910  PX3 = 75 + 48 * X  : PY3 = 150 - 24 * Y
920  IF I < 2 THEN 970
930  LINE(65, PY4) - (65, PY3)
940  LINE(65, PY4 + 10) - (65, PY4 + 5), 0
950  PRESET(63, PY4 + 1) : PRESET(67, PY4 + 1)
960  PSET(63, PY3 + 1) : PSET(67, PY3 + 1)
970  PRESET(PX4 + 3, 154) : PRESET(PX4 + 3, 155)
980  PSET(PX3 + 3, 154) : PSET(PX3 + 3, 155)
990  LINE(PX4 + 15, 155) - (PX4 + 10, 155), 0
1000  LINE(PX4, PY4) - (PX3, PY3)
1010  LINE(PX4, PY4 + 1) - (PX3, PY3 + 1)
1020  LINE(PX4, 155) - (PX3, 155)
1030  PX4 = PX3  : PY4 = PY3
1040
1050  NEXT I
1060  LINE(75, PY1) - (PX1, PY1), 1
1070  LINE(75, PY3) - (PX3, PY3), 1
1080  LINE(PX1, 150) - (PX1, PY1), 1
1090  LINE(PX3, 150) - (PX3, PY3), 1
1100  E = ABS( Y - 3 )
1110  LOCATE 22, 10: PRINT "Bringing X within .1 of 5, brought f(X) within "
1120 LOCATE 22,57:PRINT USING"#.##":E;PRINT "distance of 3."
1130 VRTSHFT = 2.5:IF COUNT = 2 THEN GOTO 1150
1140 LOCATE 23,10:INPUT "Press enter for a different function.",A$:GOTO 510
1150 RETURN
1160 Y = 1/20 * (X-1)^2 * (X-6) * (X-10) + VRTSHFT
1170 RETURN
1180 '************** SUBROUTINE TO INPUT FUNCTION AND DECIDE AXES ********
1190 .
1200 CLS
1210 LOCATE 5,10:PRINT "I have written on the screen below, the number
1220 LOCATE 6,10:PRINT "2790, followed by Y = ( some function of X)
1230 LOCATE 8,10:PRINT "The cursor is at the start of that line, so if
1240 LOCATE 9,10:PRINT "that's not the function you want, move the cursor
1250 LOCATE 10,10:PRINT "to the right of the equals sign, and type your
1260 LOCATE 11,10:PRINT "function, then hit ENTER twice."
1270 LOCATE 16,10:PRINT "GOTO 1300 "
1280 .
1290 LOCATE 15,10:EDIT 2930
1300 CLS:LOCATE 10,10:PRINT "Do you want to enter an interval for the graph.
1310 LOCATE 11,10:INPUT "Enter Y for yes, or press ENTER to default to [0,10 ] ",A$
1320 IF A$ = "" THEN A = 0 : B = 10 : GOTO 1360
1330 CLS
1340 LOCATE 11,10:PRINT "Enter the Left and Right endpoints or your interval
1350 LOCATE 12,10:INPUT "as A,B ",A,B
1360 A = 0 : B = 10 : SCALE = 600 / ( B - A )
1370 YAXE = 640 - B*SCALE ' pixel the Y-Axis is at
1380 .
1390 TRAP = 2 ' this is an error trap making sure the user picks 2 before 3-4
1400 .
1410 GOTO 300
1420 '*************** SUBROUTINE TO DRAW EPSILON - DELTA STUFF ************
1430 CLS
1440 LOCATE 8,10:INPUT "At what value of X do you want to test for a limit":XL
1450 CLS
1460 GOSUB 2570 'to draw the graph
1470 LOCATE 2,15:PRINT "" ' to clear the line first
1480 LOCATE 2,10:INPUT "EPSILON = "; EPSLN
1490 LOCATE 2,10:PRINT ""
1500 .
1510 WHILE EPSLN < 0 OR EPSLN > 1 ' error trap
1520 LOCATE 2,29:PRINT "PLEASE KEEP IT BETWEEN 0 AND 1 !!"
1530 GOTO 1470
1540 WEND
1550 .
1560 .
1570 X = XL : GOSUB 2920 : YL = YE ' find YL for the following statements
1580 PEPUP = 150 - SCALE/2 * (YL + EPSLN) ' pixel for upper epsilon line
1590 PEPDO = 150 - SCALE/2 * (YL - EPSLN) ' pixel for lower epsilon line
1600 PLIM = 150 - SCALE/2 * YL ' pixel for limit epsilon line
1610 .
1620 X = X + .5 : GOSUB 2920
1630 IF Y > YL THEN INC = -1
1640 PY1 = PLIM : X = XL
1650 WHILE ABS(PY1 - PEPUP) > 1
1660 IF INC THEN X = X + .04 ELSE X = X - .04
1670 GOSUB 2920 'to figure Y for this X
1680  PY1 = 150 - SCALE/Z * Y
1680  IF PY1 < 0 OR PY1 > 150 THEN X = 0 : PY1 = PEPUP
1700  WEND
1710  DELT1 = (X - XL)  "this will be the delta to the right of X
1720  X = XL  "set X back to the middle
1730  .
1740  PY1 = PPLIM
1750  WHILE ABS(PY1 - PEPDU) > 1
1760   IF INC THEN X = X - .04 ELSE X = X + .04
1770   IF PY1 < PEPDU THEN PY1 = PEPDU : GOTO 1820
1780   GOSUB 2920  'to figure Y for this X
1790   IF INC THEN IF Y < PREDY THEN DELT2 = 0 : GOTO 1830
1800   PY1 = 150 - SCALE/Z * Y
1810   IF PY1 > 150 OR PY1 < 0 THEN X = 0 : PY1 = PEPDU
1820  WEND
1830  DELT2 = (XL - X)  "this will be the delta to the left of X
1840  IF ABS(DELT1) <= ABS(DELT2) THEN DELTA = ABS(DELT1)
1850     ELSE DELTA = ABS(DELT2)  'take the small
1860  .
1870  X = XL - DELT2
1880  GOSUB 2920
1890  PX1 = YAXE + SCALE * X : PY1 = 150 - SCALE/Z * Y
1900  X = XL + DELT1
1910  GOSUB 2920
1920  PX2 = YAXE + SCALE * X : PY2 = 150 - SCALE/Z * Y
1930  PXL = YAXE + SCALE * XL : PLY = 150 - SCALE/Z * YL
1940  LINE (PXL,150) - (PXL,PYL)  'drawing the LIMIT lines
1950  LINE (PXL-B,PYL) - (YAXE,PYL)
1960  LX3 = .125 * PXL
1970  LOCATE 20,LX3: PRINT XL
1980  .
1990  LINE (PX1-B,PY1) - (YAXE,PY1)  'drawing the EPSILON lines
2000  LINE (PX2-B,PY2) - (YAXE,PY2)
2010  LX3 = .125 * PYL : LX4 = .125 * YAXE - 1
2020  LOCATE LX3,LX4-2 : PRINT "f(x)
2030  .
2040  X = XL + DELTA : GOSUB 2920
2050  PX3 = YAXE + SCALE * X : PY3 = 150 - SCALE/Z * Y
2060  X = XL - DELTA : GOSUB 2920
2070  PX4 = YAXE + SCALE * X : PY4 = 150 - SCALE/Z * Y
2080  FOR I = PY3 TO 145 STEP 10 : LINE (PX3,I) - (PX3,I+5) : NEXT I
2090  FOR J = PY4 TO 145 STEP 10 : LINE (PX4,J) - (PX4,J+5) : NEXT J
2100  LINE (PX1,150) - (PX1,PEPDU)
2110  LINE (PX2,150) - (PX2,PEPUP)
2120  FOR I = YAXE TO PX3 STEP 10 : LINE(I,PY3)-(I+5,PY3) : NEXT I
2130  FOR J = YAXE TO PX4 STEP 10 : LINE(J,PY4)-(J+5,PY4) : NEXT J
2140  .
2150  LOCATE 22,10:PRINT "For e = ";EPSSLN: PRINT USING "d = ";"###": DELTA
2160  LOCATE 23,40: PRINT ""
2170  LOCATE 23,40:INPUT "NEW EPSILON ( return to stop )": EPSLN
2180  .
2190  IF EPSLN = 0 THEN 2300
2200  LINE (PX1,150) - (PX1,PY1),0
2210  LINE (PX4,150) - (PX4,PY4),0
2220  LINE (PX3,150) - (PX3,PY3),0  'erasing delta lines
2230  LINE (PX2,150) - (PX2,PY2),0  'erasing delta lines
2240  LINE (PX3-8,PY3) - (YAXE,PY3),0
2250  LINE (PX1-8,PY1) - (YAXE,PY1),0
2260  LINE (PX4-8,PY4) - (YAXE,PY4),0  'erasing epsilon lines
2270  LINE (PX2-8,PY2) - (YAXE,PY2),0  'erasing epsilon lines
2280  GOTO 1500
2290  
2300  LOCATE 23,10:INPUT "Do you want to pick another value for X "; A$
2310  IF MID$(A$,1,1) = "Y" OR MID$(A$,1,1) = "y" THEN 1420
2320  RETURN
2330  *************** SUBROUTINE TO LIST VALUES AS X IS APPROACHED *******
2340  
2350  CLS
2360  LOCATE 10,10:PRINT "At what value of X do you want to test for a limit ?"
2370  LOCATE 12,20:INPUT "X = ",XL
2380  CLS
2390  PRINT " FROM THE LEFT"; TAB(35) " FROM THE RIGHT"
2400  PRINT
2410  PRINT " X   f(X)";TAB(25)"1";TAB(35)"1"; " f(X)"          X"
2420  PRINT " ";TAB(35)"1"; " ;TAB(35)"1"; " ;
2430  
2440  I = 1
2450  WHILE (XL + I) >= (XL + 9.99999E-06)
2460    LEFT = XL - I
2470    RIGHT = XL + I
2480    X = LEFT : GOSUB 2920 'to find Y for this X
2490    PRINT LEFT,Y;TAB(25)"1";TAB(35)"1";
2500    X = RIGHT : GOSUB 2920
2510    PRINT Y; TAB(50) RIGHT
2520    I = I/2
2530  WEND
2540  
2550  RETURN
2560  LOCATE 22,10:INPUT "Do you want to try another EPSILON "; A$
2570  *************** SUBROUTINE TO DRAW THE GRAPH ***************
2580  
2590  CLS
2600  LINE (1,150) - (630,150)  'FIRST DRAW THE AXES AND LABEL THEM
2610  LINE (YAXE,1) - (YAXE,165)
2620  LINE (YAXE+1,1) - (YAXE+1,165)
2630  PX3 = .12 * YAXE ;LOCATE 2,PX3:PRINT "Y"
2640  LOCATE 20,77:PRINT "X"
2650  
2660  'Find the first set of pixels and draw that boundary
2670  X = A  : GOSUB 2920
2680  PX1 = YAXE + SCALE * X  : PY1 = 150 - SCALE/2 * Y
2690  LINE (PX1,PY1) - (PX1,150)
2700  LINE (PX1+1,PY1) - (PX1+1,150)
2710  PX3 = .12*PX1 + 2
2720  LOCATE 20,PX3: PRINT A
2730  
2740  'Graph the function
2750  FOR I = A TO B STEP .15
2760    X = I  : IF X = 0 THEN 2840
2770    GOSUB 2920
2780    PX2 = YAXE + SCALE * X  : PY2 = 150 - SCALE/2 * Y
2790    IF PY2 <= -19 THEN 2830
IF PY1 < 0 OR PY1 > 199 THEN 2830
2810 LINE (PX1,Py1) - (PX2,Py2)
2820 LINE (PX1+1,Py1) - (PX2+1,Py2)
2830 PX1 = PX2 : PY1 = PY2
2840 NEXT I
2850
2860 LINE (PX1,Py1) - (PX1,150)
2870 LINE (PX1+1,Py1)-(PX1+1,150)
2880 PX3 = .12*PX1 + 1 : LOCATE 20,PX3:PRINT B
2890 RETURN
2900 '
2910 '********** FUNCTION SUBROUTINE **********
2920 Y = SQR( X ) + 1
2930 RETURN
10 '********** DERIVATIVES **********
20 ' STORED UNDER: DERIV
30 ' JUNE 15, 1985
40 ' 50 CLEAR,,327681 : SCREEN G : COLOR 3,4
70 SCREEN 2
80 CLS
90 100 CLS
110 LOCATE 3,13:PRINT "PROGRAM DERIVATV
120 LOCATE 5,15:PRINT "COMMAND MENU
130 LOCATE 8,10:PRINT "1. Derivative Demonstration
140 LOCATE 10,10:PRINT "2. Graph of Function with its Derivative
150 LOCATE 12,10:PRINT "3. Table of Function and Derivative Values
160 LOCATE 14,10:PRINT "4. Terminate Program
170 '
180 LOCATE 20,10:INPUT "ENTER YOUR CHOICE ", CMD
190 200 WHILE CMD <> FIX(CMD) OR CMD < 1 OR CMD > 4
210 220 GOTO 180
230 WEND
240 '
250 WHILE CMD <> 4
260 270 GOTO 100
280 WEND
290 '
300 SCREEN 0
310 END
320 '********** SUBROUTINE TO GIVE EXPLANATION **********
330 340 CLS : YAXE = 100 : XAXE = 150
350 LOCATE 10,10: PRINT "The graph of f(X) = -1/6 (X - 4)^2 + 4 will be drawn, and we"
360 LOCATE 12,10: PRINT "will construct the tangent line to the curve at X = 3"
370 LOCATE 14,10: PRINT "by first drawing a secant line to another point on the"
380 LOCATE 16,10: PRINT "curve, then bring that point over to meet X = 3."
390 LOCATE 18,10: PRINT "The slope of any of these secant lines is given by"
400 LOCATE 20,10: PRINT "the value of ( f(3+h) - f(3) ) / ( 3+h - 3 ), where"
410 LOCATE 22,10: PRINT "h is how far the other end of the secant is from 3."
420 LOCATE 24,20: INPUT "Press enter to continue.",A$
430 '
440 CLS : GOSUB 1770
450 BEGINX = 0 : STPX = 8 : GOSUB 790 ' to graph the curve
460 '
470 X = 3 : FX = -1/6 * (X - 4)^2 +4
480 PNT = 7 : FPNT = -1/6 * ( PNT - 4 )^2 + 4
490 SLOPE = (FPNT - FX) / (PNT - 3)
500 LOCATE 5,60 : PRINT "SLOPE = ";PRINT USING "##.####";SLOPE
510 REND = SLOPE*(7) + FX : RPX = 500 : RPY = 150 - 20*REND
520 LEND = SLOPE*(-4) + FX : LPX = 60 : LPY = 150 - 20*LEND
530 LINE (RPX,RPY)-(LPX,LPY)
540 LOCATE 22,1 : INPUT "Press enter to move secant.",A$
550 LOCATE 22,1 : PRINT ""
following loop moves the secant line toward a tangent

FOR PNT = 6.7505 TO 3.0005 STEP -.25
  PPNT = 100 + 40*PNT : LINE (PPNT,155)-(PPNT,151)
  LINE (RPX,RPY)-(LPX,LPY),0
  FPNT = -1/6 * ( PNT - 4 )^2 + 4
  SLOPE = (FPNT - FX) / (PNT - 3)
  LOCATE 5,60 : PRINT "SLOPE = "; PRINT USING "###.#####";SLOPE
  R = 10 ; L = -1
  REND = SLOPE*(R-3) + FX : RPX = 100+40*R : RPY = 150 - 20*REND
  LEND = SLOPE*(L-3) + FX : LPX = 100+40*L : LPY = 150 - 20*LEND
  IF RPY < 0 THEN R = R - 1 : GOTO 660
  LINE (RPX,RPY)-(LPX,LPY)
  BEGINX = PNT-.5 : STPX = PNT + .5
  GOSUB 790 \to redraw anything erased from the curve
  IF PNT > 3.005 THEN LINE (PPNT,155)-(PPNT,151),0
  NEXT PNT

LOCATE 23,10 : INPUT "Press ENTER to return to the command menu. ",A$  
RETURN

**SUBROUTINE TO GRAPH THE DERIVATIVE FUNCTION **

CLS : LOCATE 10,20 : PRINT "Type in any function

LOCATE 12,20 : PRINT "after the Y# is seen below.

LOCATE 14,20 : PRINT "Then, press enter twice."

LOCATE 18,1 : PRINT "Goto 950"

LOCATE 17,1 : EDIT

CLS : LOCATE 12,20 : PRINT "Enter the left and right endpoints of the "},A,B

LOCATE 13,20 : INPUT "the interval you want to graph over, as A,B ";A,B

X = A : GOSUB 1420 \to compute Y

FX# = Y#

SCALE is number of pixels per unit

SCALE = 600 / ( B - A ) : YAXE = G20 - SCALE * B : XAXE = 100

CLS : GOSUB 1770 \to draw axes

PY1 = YAXE + X*SCALE : PY1 = 100 - Y#*SCALE/2

LCY = .12*PY1 \IF LCY < 1 THEN LCY = 1 \LCY is for locate command below

IF LCY > 23 THEN LCY = 23

LOCATE LCY,2 : PRINT "f(X)"

H = .00005 \H is given a value very near 0

X = X + H : GOSUB 1420 \to compute f(X+h)

FXHH = Y#

DY# = (FXHH - FX#) / H

PX1 = 100 - DY#*SCALE/2

LCDY = .12*PDY1 \IF LCDY < 1 THEN LCDY = 1
1120 IF LCDY > 23 THEN LCDY = 23
1130 LOCATE LCDY,2 : PRINT "f'(X)"
1140 FOR X = A TO B STEP .1
1150  GOSUB 1420
1160  FXII = YII
1170  PX2 = YAXE + X*SCALE : PY2 = 100 - Y#*SCALE/Z
1180  X = X + H : GOSUB 1420
1190  FXII = YII
1200  DY# = (FXH# - FXII) / H
1210  PDY2 = 100 - SCALE/Z = DY#
1220  IF ABS(PDY2-PDY1) > 150 THEN PDY1 = PDY2 : GOTO 1240
1230  IF PDY2 > 0 AND PDY2 < 199 THEN LINE (PX1,PDY1)-(PX2,PDY2)
1240  PX1 = PX2 : PDY1 = PDY2
1250 NEXT X
1260 LOCATE 23,1 : INPUT "Press enter for f(X)",A$
1270 LOCATE 23,1 : PRINT ""
1280 X = A : GOSUB 1420
1290 PX1 = YAXE + A*SCALE : PY1 = 100 - Y#*SCALE/Z
1300 FOR X = A TO B STEP .1
1310  GOSUB 1420
1320  FX = YII
1330  PX2 = YAXE + X*SCALE : PY2 = 100 - Y#*SCALE/Z
1340  IF PY2 > 0 AND PY2 < 199 THEN LINE (PX1,PY1)-(PX2,PY2)
1350  PX1 = PX2 : PY1 = PY2
1360 NEXT X
1370 .
1380 LOCATE 23,1 : INPUT "Another ";A$
1390 IF A$ = "Y" OR A$ = "y" THEN 880
1400 GOTO 270
1410 .
1420  SUBROUTINE TO CONTAIN THE FUNCTION
1430  Y# = .1 * (X + 3)*(X - 1)*(X - 4)
1440 RETURN
1450 .
1460 .
1470 ******************** SUBROUTINE TO GIVE TABLE OF VALUES ********************
1480 .
1490  CLS : LOCATE 10,20 : INPUT "Do you want to change the function ";A$
1500 IF A$ = "N" OR A$ = "n" THEN 1550
1510  CLS : LOCATE 10,20 : PRINT "Make desired changes, then press enter twice.
1520 LOCATE 13,1 : PRINT "GOTO 1540"
1530 LOCATE 12,1 : EDIT 1430
1540  CLS
1550 LOCATE 10,50 : PRINT ""
1560 LOCATE 10,10 : INPUT "Over what interval do you the table: a,b ";A,B
1570 IF (B-A) < .0199 THEN LOCATE 12,15 : PRINT "Interval must be at least .02 in length." : GOTO 1550
1580  CLS : INC = (B - A) / 10
1590  loop to increment X over the interval, and print results
1600  LOCATE 2,2 : PRINT "X" f(X)  1st DER at X
1610 PRINT
1620 HII = .00005
1630 FOR I = A TO B STEP INC
1640 X = I : GOSUB 1420 : FXII = YII
1650 X = I + HII/Z : GOSUB 1420 : FXHII# = YII
1660 X = I - HII/Z : GOSUB 1420 : FXH2# = YII
1670  DY# = (FXH# - FXH2#) / HII
1680  CLS : LOCATE 10,50 : PRINT "f'(X)"
1690  CLS
1700  CLEAR
1680 PRINT USING "###.###"+SPACE$(10)+"###.###"+SPACE$(10)+"###.###"
";X,FN$,DY/$
1690 NEXT I
1700 '...
1710 LOCATE 23,10: INPUT "Another interval ";A$
1720 IF A$ = "y" OR A$ = "Y" THEN 1540
1730 LOCATE 23,28: PRINT " "
1740 LOCATE 23,10: INPUT "Another function ";A$
1750 IF A$ = "y" OR A$ = "Y" THEN 1510 ELSE 270
1760 '........................ SUBROUTINE TO DRAW GRAPH ...........................
1770 '...
1780 '          first draw the axes
1790 LINE (0,XAXE)-(640,XAXE): LINE (0,XAXE+1)-(640,XAXE+1)
1800 LINE (YAXE,0)-(YAXE,200): LINE (YAXE+1,0)-(YAXE+1,200)
1810 LOCATE 20,77: PRINT "X": LOCATE 2,12: PRINT "Y"
1820 '...
1830 '...
1840 RETURN
********** INTEGRATION  **********

STORFD UNDER: INTEGRAL
JULY 30, 1984

10  ** INTEGRATION **
20 :
30  STORFD UNDER: INTEGRAL
40 :
50 :
60  CLS
70  CLEAR,,32768! : SCREEN 6 : COLOR 3,4
80  'SCREEN 2
90  CLS
100 LOCATE 2,13:PRINT "PROGRAM INTEGRAL"
110 LOCATE 4,15:PRINT "COMMAND MENU (Note that choice 2. must be made before 3 - 6)"
120 LOCATE 7,10:PRINT "1. Explanation of Riemann Sums and Integrals"
130 LOCATE 9,10:PRINT "2. Input a Function and Define an Interval"
140 LOCATE 11,10:PRINT "3. Draw and Figure an Upper Riemann Sum"
150 LOCATE 13,10:PRINT "4. Draw and Figure a Lower Riemann Sum"
160 LOCATE 15,10:PRINT "5. Draw and Use the Trapezoidal rule"
170 LOCATE 17,10:PRINT "6. List a Table of Values Using Each"
180 LOCATE 19,10:PRINT "7. Terminate Program"
190 :
200 LOCATE 21,10:PRINT "ENTER YOUR CHOICE ", CMD
210 WHILE CMD > 2 AND CMD < 7
220   IF TRAP = 2 THEN 260
230   LOCATE 23,10:PRINT "You must choose 2. before 3 - 6"
240   GOTO 200
250   WEND
260 WHILE CMD <> FIX(CMD) OR CMD < 1 OR CMD > 7
270   LOCATE 23,10:PRINT "Please enter a 1, 2, 3, 4, 5, 6, or 7"
280   GOTO 200
290   WEND
300 :
310 WHILE CMD <> 7
320   ON CMD GOSUB 400, 1490, 1740, 2140, 2510, 2790
330   LOCATE 23,1 :PRINT "Press enter to return to the command menu. ",B$
340   GOTO 90
350   WEND
360 SCREEN 0
370 END
380 :
390 :
400  ** SUBROUTINE TO GIVE EXPLANATION **

410 :
420  CLS
430 PRINT TAB(20)"Riemann Sums are a method of approximating the area ",
440 PRINT TAB(15)"between a function and the X-axis. Suppose we have the func-
450 PRINT TAB(15)"tion defined f(x) = -.25* (X - 4)^2 + 4 , which is a 
460 PRINT TAB(15)"parabola opening down, with its vertex at (4,4), and X-int-
470 PRINT TAB(15)"ercepts at (0,0) and (0,8). If we want to find the area 
480 PRINT TAB(15)"under the parabola for the interval [0,8], then we're look-
490 PRINT TAB(15)"ing for the area of the region enclosed by the curve and 
500 PRINT TAB(15)"the X-axis.
510 PRINT
520 PRINT TAB(20)"To get some idea of what the area would be, let's 
530 PRINT TAB(15)"look at the rectangle with a base on the X-axis, and sides 
540 PRINT TAB(15)"at X=0 and X=8 . The top of the rectangle will be through 
550 PRINT TAB(15)"the vertex, so its height is the same as the maximum height
of the function. Since our parabola is completely inside this rectangle, we know the actual area is less than the area of the rectangle, which is $8 \times 4 = 32$.

Let's take our interval $[0, 8]$, and partition it into 3 equal subintervals, $[0, 2.61200 PRINT TAB(15)"of the function. Since our parabola is completely inside this rectangle, we know the actual area is less than the area of the rectangle, which is $8 \times 4 = 32$.

Let's take our interval $[0, 8]$, and partition it into 3 equal subintervals, $[0, 2.6]$, $[2.6, 5.2]$ and $[5.2, 8]$. Now, just like before, we'll draw rectangles, only three this time. Each will have a base equal to the length of the subintervals, which is $8/3 = 2.66$. For the height of each rectangle, we'll again use the maximum height of the function, only this time the height in each subinterval.

Now, let's find the area of the first one, find the maximum height of the first subinterval (which is $f(2.66)$, or 3.55), then multiply by the interval length, or 2.66. This gives an area of about 9.45.

The middle rectangle has a maximum height of 4, times the length of the subintervals (2.66), gives about 10.64. Notice that the area of the third rectangle will be the same as the area of the first. Add the three up and we have 29.54, which is closer to the real thing.

As you can see, by dividing the interval into three rectangles, we've cut off some of the excess area. At this point you might guess that if we divided it into more subintervals yet, we'd cut off more of the excess.

Press enter, and I'll divide the interval into 10 subintervals (giving 10 rectangles to add up).

The total area is going down, as the size of the partition goes up. This is because with more and more rectangles, there is less and less excess area above the function being added.

Now you tell me how many subintervals to use this time.

Press enter and I'll divide the interval into 10 subintervals (giving 10 rectangles to add up).

The total area is going down, as the size of the partition goes up. This is because with more and more rectangles, there is less and less excess area above the function being added.
1120 CLS
1130 PRINT TAB(20)"By now, you can guess that the more times we partition the
1140 PRINT TAB(15)"interval, the closer our area is to the real thing, so the
1150 PRINT TAB(15)"better our approximation.
1160 PRINT
1170 PRINT TAB(20)"The process of partitioning the interval into subintervals,
1180 PRINT TAB(15)"then finding rectangles that go up to some point on the
1190 PRINT TAB(15)"function inside each subinterval, is called a Riemann
1200 PRINT TAB(15)"sum. Any point inside would do, but so far we've always
1210 PRINT TAB(15)"used the maximum height in each subinterval for the height of
1220 PRINT TAB(15)"the corresponding rectangle. This is called the upper Riemann
1230 PRINT TAB(15)"sum.
1240 PRINT TAB(20)"If, instead, we used the minimum height in each subinterval,
1250 PRINT TAB(15)"they would be lower Riemann sums.
1260 PRINT
1270 PRINT TAB(20)"If we took the limit as the number of subintervals goes to
1280 PRINT TAB(15)"infinity, of either the upper or the lower sums, we should
1290 PRINT TAB(15)"have the exact area under the curve.
1300 PRINT
1310 PRINT TAB(20)"That value is called the Riemann integral of the function
1320 PRINT TAB(15)"over that interval.
1330 LOCATE 20,15:PRINT "Press enter and I'll draw it with a very large number
1340 LOCATE 21,15:INPUT "of subintervals, and thus get a good approximation.",A$
1350 N = 200 : GOSUB 1830
1360 CLS
1370 PRINT TAB(20)"So, to rehash this a little, the upper Riemann sum for
1380 PRINT TAB(15)"N subintervals, over any interval \([a,b]\) , is the sum of the
1390 PRINT TAB(15)"area of N rectangles, each with a base length of \((b-a)\)
1400 PRINT TAB(15)"divided by N. The maximum height of the function in each
1410 PRINT TAB(15)"subinterval is used for the height of the corresponding rectangle.
1420 PRINT
1430 PRINT TAB(20)"The lower Riemann sum is the same, except you use the minimum
1440 PRINT TAB(15)"sum in each subinterval.
1450 PRINT
1460 PRINT TAB(20)"The limit of these as N approaches infinity, is the Riemann
1470 PRINT TAB(10)"integral over the interval.
1480 RETURN
1490 *************** SUBROUTINE TO INPUT FUNCTION AND DECIDE AXES ********
1500 .
1510 CLS
1520 LOCATE 5,10:PRINT "Written on the screen below, is the number
1530 LOCATE 6,10:PRINT "9100, followed by Y = (some function of X)
1540 LOCATE 8,10:PRINT "The cursor is at the start of that line, so if
1550 LOCATE 9,10:PRINT "that's not the function you want, move the cursor
1560 LOCATE 10,10:PRINT "to the right of the equals sign, and type your
1570 LOCATE 11,10:PRINT "function, then hit enter two times.
1580 LOCATE 14,1 : PRINT "GOTO 1600" : PRINT
1590 LOCATE 13,1 : EDIT 3620
1600 .
1610 CLS
1620 LOCATE 10,10:PRINT "You will now want to choose an interval to figure
1630 LOCATE 11,10:PRINT "the sums on.
1640 LOCATE 15,10:INPUT "Enter the left and right endpoints as A,B. ",LE,RE
1650 .
1660 ILNTH = RE - LE \(\text{ILNTH is the length of the interval}\)
1670 UNIT2 = 600 / ILNTH : UNIT = UNIT2/2 \(\text{pixels in a unit}
1680 YAXE = 620 - UNIT2 * RE
1680 pixel the Y-Axis is at
1700
1710 TRAP = 2 'this is an error trap making sure the user picks 2 before 3-5
1720
1730 CLS : GOTO 330
1740 ************** SUBROUTINE TO DRAW THE UPPER SUMS **************
1750
1760 CLS
1770 LOCATE 10,30:PRINT "    "  'to clear the line first
1780 LOCATE 10,10:INPUT "How many subintervals do you want "; N
1790 WHILE N < 1 OR N <> FIX(N) 'error trap
1800 LOCATE 12,10:PRINT "POSITIVE WHOLE NUMBERS ONLY, PLEASE!!"
1810 GOTO 1770
1820 WEND
1830 GOSUB 3260 'to draw the graph
1840
1850 DELTA = ILNTH / N 'figures the width of each subinterval
1860 STP = RE - DELTA + .0001 'stopping point for loop
1870 SUM = 0 'SUM is the Riemann Sum
1880 FOR I = LE TO STP STEP DELTA
1890 X = I : GOSUB 3610 'to figure the functional value for that X
1900 MAX = Y
1910 FOR J = I TO I+DELTA STEP .05 'goes thru the Ith subinterval
1920 X = J
1930 GOSUB 3610
1940 IF Y = MAX THEN MAX = Y
1950 NEXT J
1960
1970 SUM = SUM + MAX * DELTA 'figure the Riemann Sum
1980 PX1 = YAXE + UNIT2*I : PY1 = 150 - UNIT * MAX 'now draw the rectangles
1990 LINE (PX1,PY1) - (PX1,150)
2000 PX2 = PX1 + UNIT2 * DELTA
2010 LINE (PX2,PY1) - (PX2,150)
2020 LINE (PX1,PY1) - (PX2,PY1)
2030
2040 NEXT I
2050 IF FROMI <> 1 THEN 2080 'this tells it if we came from sub 1
2060 LOCATE 23,10:PRINT "The total area is ";SUM
2070 LOCATE 24,10:INPUT "Press enter to continue ",A$
2080 RETURN
2090 LOCATE 1,10:PRINT "The UPPER SUM for ";N ;"Subintervals is "; SUM
2100 LOCATE 2,10:INPUT "Do you want another Upper Sum "; A$
2110 IF MID$(A$,1,1) = "Y" OR MID$(A$,1,1) = "y" THEN GOTO 1760
2120
2130 RETURN
2140 ************** SUBROUTINE TO DRAW THE LOWER SUMS **************
2150
2160 CLS
2170 LOCATE 10,30:PRINT "    "  'to clear the line first
2180 LOCATE 10,10:INPUT "How many subintervals do you want "; N
2190 WHILE N < 1 OR N <> FIX(N)
2200 LOCATE 12,10:PRINT "POSITIVE WHOLE NUMBERS ONLY, PLEASE!!"
2210 GOTO 2170
2220 WEND
2230 GOSUB 3260 'to draw the graph
2800 CLS
2810 PRINT TAB(10)"SUBINTERVALS";TAB(27)"UPPER SUM";TAB(45)"LOWER SUM";TAB(63)"TRAP RULE"
2820 N = 1  'N is number of subintervals, start with 1
2830 FOR K = 1 TO 3  'the following loops all increase N
2840    N = N + 2
2850    GOSUB 3010
2860 NEXT K
2870
2880 FOR L = 20 TO 40 STEP 20
2890    N = L
2900    GOSUB 3010
2910 NEXT L
2920
2930 FOR M = 60 TO 200 STEP 60
2940    N = M
2950    GOSUB 3010
2960 IF DONE = 1 THEN 3000  'upper = lower, so stop
2970 NEXT M
2980
2990
3000 RETURN  'Nested subroutine to figure sums for N
3010
3020
3030 DELTA = ILNTH / N  'figures the width of each subinterval
3040 STP = RE - DELTA + .0001  'stopping point for loop
3050 USUM = 0  : LSUM = 0  : TRAP = 0
3060 FOR I = LE TO STP STEP DELTA
3070    X = I  : GOSUB 3610  : Y1 = Y
3080    MAX = Y  : MIN = Y
3090    X = I + DELTA  : GOSUB 3610  : Y2 = Y
3100 FOR J = I TO I + DELTA STEP .1  'goes thru the Ith subinterval
3110    X = J
3120    GOSUB 3610
3130    IF Y >= MAX THEN MAX = Y  'Maximum and minimum height in Ith
3140    IF Y <= MIN THEN MIN = Y
3150 NEXT J
3160 TRAP = TRAP + .5 * DELTA * (Y1 + Y2)
3170 USUM = USUM + MAX * DELTA  'Upper sum
3180 LSUM = LSUM + MIN * DELTA  'Lower sum
3190 IF USUM = LSUM THEN DONE = 1  'no need to continue
3200 NEXT I
3210
3220 PRINT
3230 PRINT USING SPACE$(10)"######";N;USUM;LSUM;TRAP
3240
3250 RETURN
3260 ************ SUBROUTINE TO DRAW THE GRAPH ************
3270
3280 CLS
3290 LINE (1,150) - (630,150)  'FIRST DRAW THE AXES AND LABEL THEM
3300 IF LE > 0 THEN 3340
3310 LINE (YAXE,1) - (YAXE,165)
3320 LINE (YAXE+1,1) - (YAXE+1,165)
3330 PX3 = .12 * YAXE  : LOCATE 2,PX3:PRINT "Y"
3340 LOCATE 20,77:PRINT "X"
3350
3360 ' Find the first set of pixels and draw that boundary
3370 X = LE : GOSUB 3610
3380 PX1 = YAXE + UNITZ * X : PY1 = 150 - UNIT * Y
3390 LINE (PX1, PY1) - (PX1, 150)
3400 LINE (PX1+1, PY1) - (PX1+1, 150)
3410 PX3 = .12*PX1 + 2
3420 LOCATE 20, PX3: PRINT LE
3430 ' 3440 ' Graph the function
3450 FOR I = LE TO RE STEP .15
3460 X = I
3470 GOSUB 3610
3480 PXZ = YAXE + UNITZ * X : PYZ = 150 - UNIT * Y
3490 IF PYZ < 0 OR PYZ > 199 THEN 3520
3500 LINE (PX1, PY1) - (PXZ, PYZ)
3510 LINE (PX1+1, PY1) - (PXZ+1, PYZ)
3520 PX1 = PXZ : PY1 = PYZ
3530 NEXT I
3540 ' 3550 LINE (PX1, PY1) - (PX1, 150)  ' draw the last boundary and label
3560 LINE (PX1+1, PY1) - (PX1+1, 150)
3570 PX3 = .12*PX1 + 2 : LOCATE 20, PX3: PRINT RE
3580 ' 3590 RETURN
3600 ' 3610 '********* FUNCTION SURROUTINE *********
3620 Y = -.25*(X-4)^2 + 4
3630 RETURN