JONES, CAROLYN THROCKMORTON. An Experimental Course: Calculus with Computing. (1973) Directed by: Dr. William P. Love Pp. 54

It was the purpose of this thesis to report on the development, organization, difficulties, and cost of implementing an experimental mathematics course, Calculus with Computing (Math 133-233). This course was designed to introduce computer programming to first year calculus students. The traditional calculus lectures were supplemented by a computer programming laboratory, in which the students used the computer to solve problems related to the calculus.

When Calculus with Computing was taught (Fall, 1972 - Spring, 1973), an experiment was conducted to compare the calculus achievement of students taking Math 133-233 with students taking traditional calculus (Math 191-292). The results and conclusions of this experiment are included in this report. Statistically it was shown that students in Calculus with Computing were superior in calculus achievement, while at the same time they obtained introductory programming proficiency.

AN EXPERIMENTAL COURSE: CALCULUS WITH COMPUTING

by

Carolyn Throckmorton Jones

A Thesis Submitted to
the Faculty of the Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Arts

Greensboro
1973

Approved by

William P. Love
Thesis Advisor

APPROVAL SHEET

This thesis has been approved by the following committee of the Faculty of the Graduate School at The University of North Carolina at Greensboro.

Thesis Advisor William P. Love

Oral Examination Committee Members Robert L. Miller

E.E. Posey

Michael Willett

Robert B. Muir

August 2, 1973
Date of Examination

ACKNOWLEDGMENT

The author would like to express her appreciation to Dr. William P. Love for his time, patience, and guidance; and to Dr. William A. Powers for his advice and assistance.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER I

BACKGROUND INFORMATION

MOTIVATION

Since 1968 there has been an interest in colleges and universities across the country to introduce computer programming in their freshman or sophomore level mathematics courses. Some experts suggest that computing be introduced in the first year of calculus. Phillip J. Davis, of Brown University, had the following to say concerning computers and mathematics: "The general availability of computer facilities in universities, versatile computer languages, time sharing via remote consoles, and graphical display units provides mathematics departments with a splendid and important opportunity to offer traditional calculus material with a computer slant." [4, p.1]

The Committee on the Undergraduate Program in Mathematics (CUPM) suggested the following reasons for introducing computers into calculus courses:

1. Calculus is the first course in analysis. It would seem that this is a good time to introduce computers into the students' thinking and working habits.
2. Student interest and comprehension is increased as learning becomes a more active than passive experience.
3. Problems in the calculus become more real, more challenging, and less tedious when programmed for a computer. [2]

PROJECTS

Many different attempts to relate computing and calculus (henceforth called computer-calculus) have been tried with varying degrees of success. Below is a discussion of some of the different projects which have been tried.

CRICISAM

In 1966 The Center for Research in College Instruction of Science and Mathematics (CRICISAM), based at Florida State University, began a project, sponsored by National Science Foundation, to integrate calculus and computing. The CRICISAM approach, being used by approximately one hundred schools as of May, 1972, is by far the most extensively used project of its kind. [3] The members of this project, headed by Warren Stenberg, developed a textbook, Calculus: A Computer Oriented Presentation, to be used in the first year calculus course. This textbook is based on the idea of total integration of the calculus and computing and is language free--that is no particular language is presented in the book. The problems and examples are oriented towards solution by using the algorithmic approach and student-written computer programs.

OTHER PROJECTS

1. Mr. Phillip J. Davis and Charles Strauss, of Brown University, directed a computer-calculus course there in 1968-69. An intuitive approach to the material was used throughout the course. The students attended three calculus lectures and two programming sessions each week. A principal complaint of the students was the lack of a

textbook that strongly related to what they were doing. [5]

2. In 1969 Dartmouth College taught computing as an adjunct to second term calculus. The students were expected to learn programming primarily by themselves. [2]
3. The University of Denver taught a computer-calculus course in 1969. Students attended programming sessions for one hour per week for the first eight weeks in the semester. A standard calculus text was used. [2]

As of 1969 the above mentioned schools had plans to modify and expand their particular approach of introducing computing in calculus.

The CUPM Newsletter (August, 1969) stated the following schools were teaching a computer-calculus course: Case Western Reserve University, Claremont Colleges, University of Minnesota, University of New Mexico, Oberlin College, University of Pennsylvania, University of Utah, and Vanderbilt University. [2]

INTEREST AT THE UNIVERSITY OF NORTH CAROLINA AT GREENSBORO (UNC-G)

Because of the need to keep abreast of current trends in curriculum development, Dr. William P. Love and Mr. John R. Martin members of the Mathematics Department at UNC-G formed a committee to investigate the possibility of introducing computer programming to the first year calculus students. Mr. Martin (a graduate student) for his thesis, "Development of Computer Supplements to Calculus" (1972), did extensive research of the various projects being either planned or used at many colleges and universities in the United States. This thesis study was the first step in the development of a computer oriented calculus course at UNC-G.

Dr. Love and Mr. Martin began in the Spring of 1972 a project to make plans and organize a computer-calculus course. This author, who has taught programming and has experience in several programming languages, became interested in the idea of such a calculus course. The purpose of this thesis is to report on this project; its objectives, organization, problems, and the results of an experiment conducted when the experimental course was actually taught in 1972-73.

CHAPTER II

DEVELOPMENT OF THE COMPUTER CALCULUS-COURSE

PRE-1972 CALCULUS CURRICULUM

Elementary calculus at UNC-G is divided into three one semester courses, Mathematics 191, 292, and 293. Each of these courses meets three hours per week and three semester hours credit are given upon completion of each course. Usually about eight sections of 191 are offered in the Fall, six sections of 292 are offered in the Spring, and four sections of 293 are offered in the Fall and Spring. Student enrollment for each of these courses is approximately twenty-five per section. In general, students taking 191 and 292 are primarily mathematics and science majors; a few are in business and other fields. Most mathematics majors continue to 293.

OBJECTIVES FOR EXPERIMENTAL CALCULUS COURSE

In designing a computer-calculus course some of the objectives were formulated and are presented below. A student completing the computer-calculus course should:

1. Have a solid understanding of the theoretical concepts of elementary calculus
2. Understand and be able to use the algorithmic approach (solving a problem using a logical sequence of steps) for analyzing and solving problems
3. Know the capabilities and limitations of the computer in application to the calculus

4. Have an understanding of and an ability to use computer equipment such as the keypunch, card reader, and printer
5. Have sufficient knowledge of a programming language to solve basic problems in the application of calculus.

PROBLEMS AND DIFFICULTIES

During the initial organization of the course there were several questions to be answered regarding how to meet the objectives.

WHAT IS THE BEST APPROACH TO USE IN INTRODUCING COMPUTING WITH CALCULUS?

In reading the literature three basic approaches to the problem of teaching computing and calculus were identified.

Total Separation.-This approach presents the calculus and computing as two separate courses. Any connection between the two subjects must be discovered independently by the student. Traditionally most universities have used this procedure of separate courses.

Total Integration.-This approach, exemplified by CRICISAM, combines calculus and computing in a single textbook. The basic concepts of elementary calculus and techniques of computer programming would be interwoven throughout the text.

Coordinated Laboratory.-The third approach supplements the traditional lecture with a computer programming laboratory. Calculus is taught using a standard textbook and programming is taught from a programming manual. In the laboratory the students would learn programming techniques and solve problems related to the topics being covered in the calculus lecture.

In examining these three approaches the following observations were made:

1. Using the method of total separation there is a high possibility that the students would never make any connection between calculus and computing. Also there may be students taking the programming course who do not know calculus; hence class time would have to be used to teach the calculus. For these reasons this approach was rejected.
2. The total integration approach was also rejected. Some comments from CRICISAM users point out the problems related to this method.
   a. Mr. Octavio Diaz, professor of mathematics at North Carolina Agriculture and Technical University, taught their three semester calculus course in 1970-71 using the CRICISAM textbook. Since analytic geometry and multivariable calculus are not included in the text, the course had to be supplemented with material from other sources. The student reaction to the course was favorable. [6]
   b. Mr. M. C. Wicht, Chairman of the Department of Mathematics at North Georgia, called the textbook "impossible." [8]
   c. Mr. William R. Fuller, professor of mathematics at Purdue University, says "Most people who have taught the computer oriented course have expressed the opinion that the students develop a greater appreciation of the infinite processes." [7]

Mr. Fuller found the major weak point of the textbook was the fact that multivariable calculus is not included. Again student reaction to the course was favorable. [7]

d. Mr. Richard Andree, of the University of Oklahoma, related the following problem with CRICISAM. When the textbook is used for the computer-calculus sections, the students cannot switch into the traditional calculus sections without difficulty. The problem is that the order and emphasis of topics in the standard text is different from that in the CRICISAM text. [1]

e. Mr. M. Pownall, of Colgate University, had the following to say about the CRICISAM text: "I really like the text very much as a whole. I do think it tends to be better in the parts of the course involving computing." [1, p. 6].

In conclusion there seem to be two basic problems encountered with the CRICISAM approach. (1) The textbook must be supplemented with other material if multivariable calculus and/or analytic geometry are to be taught in the course. (2) Students who start taking computer-calculus and then decide to change to the traditional calculus have difficulties because of sequencing of topics in the respective textbooks. It is the opinion of this author that the CRICISAM approach is both a failure and a success. It is a failure in respect to the problems mentioned above, which may be alleviated in future revisions of the textbook. It is a success because through CRICISAM the idea of computer related calculus gained attention and interest on a national scale.

3. It was decided that the objectives could best be realized by using the coordinated laboratory approach. By teaching a separate but related programming laboratory, the topics of the calculus and computing should mutually reinforce each other in order that instruction in both subjects be more effective. The same calculus book could be used for both the traditional calculus and computer-calculus. This would enable students who initially register for computer-calculus to transfer to traditional calculus with minimal difficulty.

OF THE SEVERAL PROGRAMMING LANGUAGES AVAILABLE, WHICH ONE IS BEST SUITED AND MOST PRACTICAL FOR USE IN THE COMPUTER-CALCULUS COURSE?

After narrowing the choice of languages to two, PL/C (a student version of PL/1) and WATFIV (a student version of FORTRAN), it was decided that PL/C was better for the needs of the course. There are several reasons for this choice.

1. PL/C is a more versatile language. Mathematics problems as well as business oriented problems may be solved using PL/C. This versatility gives the student a broader understanding of programming languages and is beneficial for future employment opportunities.
2. PL/C is particularly useful in teaching beginners to program. The diagnostic features are better than those of WATFIV,

which makes the task of debugging a program much easier.

3. Theoretically PL/C should be less expensive to use than WATFIV, because of the diagnostic features. Fewer runs are generally needed to produce a correct program, yielding a lower cost per student.

WHAT MODE OF PROCESSING SHOULD BE USED, BATCH OR CONVERSATIONAL (TIME-SHARING)?

Batch.-When batch processing mode is used, students submit their programs (jobs) on punched cards to the computer center, or if a terminal is available the students may run their own programs. After the program is processed, the students receive a printout (listing) of the output via the printer. On the basis of these listings further programming and/or computation may be needed if the program did not run properly, or contained errors. The time between submission of a job and its return is called turn-around-time.

Conversational.-In the conversational or interactive mode the computer can accept simultaneously the programs of several users, each at his individual terminal (teletype). Using this type of processing, the responses from the computer are essentially instantaneous. Since only one student can use the teletype at a time, the use of this mode would require at the minimum one teletype for every five students. This mode does eliminate the need of keypunch machines and cards.

In deciding which of these two modes to adopt, we had to consider the computer equipment available as well as the cost of implementing each one. The following is a list of equipment available at UNC-G.

1. A medium speed UCC 1225 card reader, printer and console keyboard connected via telephone lines to an IBM 370/165 computer at Triangle Universities Computation Center (TUCC), Research Triangle Park, North Carolina, about sixty miles from Greensboro. The academic computing center is operated as "open-shop"--that is the students may run their own programs through the card reader and take their output off the printer when it returns.
2. Three Model 33, keyboard, tapereader teletype terminals connected to TUCC.
3. A Time-Share-Peripherals plotter.
4. Seven IBM Model 29 keypunch machines located in various buildings on campus.

Because of the lack of a sufficient number of teletype terminals, we chose to use the batch processing mode. This type of processing has several advantages over the conversational mode.

1. One of the more obvious advantages is the cost. Processing time (CPU time) using conversational mode cost approximately twice as much as CPU time for batch processing. There would also be the added expense of purchasing more teletypes.
2. More jobs can be processed in an allotted time period using batch processing. This feature is particularly desirable when a large number of students is involved.

One disadvantage of batch is that students do not have the one-to-one interaction with the computer which is achieved in conversational processing.

SHOULD ALL FIRST YEAR CALCULUS STUDENTS BE ALLOWED TO ENROLL IN THE COURSE OR SHOULD IT BE TRIED FIRST ON A SMALL GROUP OF STUDENTS? HOW MANY SECTIONS SHOULD BE SCHEDULED AND AT WHAT TIMES WOULD THESE SECTIONS BE OFFERED?

Since computer-calculus had not been taught at UNC-G, it was decided to schedule two sections of the course with each section limited to thirty students. These two sections were designated Math 133: Calculus with Computing for the first semester, and Math 233: Calculus with Computing for the second semester. Students were enrolled in the course on a voluntary basis until the sections were filled. One section was offered in the morning and one in the afternoon so that students who had science laboratories could take the course. Students were allowed to drop out of Math 133 or 233 into Math 191 or 292, but students who started in Math 191 were not permitted to transfer to 133 or to take 233 instead of 292.

HOW MANY CREDIT HOURS WOULD A STUDENT RECEIVE FOR COMPLETING ONE SEMESTER OF CALCULUS WITH COMPUTING?

Since students were required to do considerably more work outside class than students taking traditional calculus (which gives three hours credit per semester), the committee decided the course should give five semester hours credit. (This was officially approved by the University Curriculum Committee in the Spring of 1973).

WHAT WOULD BE THE APPROXIMATE COST OF IMPLEMENTING SUCH A COURSE?

We estimated that the cost of sixty students per semester taking the course for two semesters would be about \$930. The cost per student was \$8.54 for 133 and \$10.57 for 233. Total cost for both semesters was $768.54. For a complete breakdown of costs by months see Appendix E.

WHAT TYPE OF TEXTBOOKS WOULD WE NEED TO TEACH STUDENTS CALCULUS AND COMPUTING?

For the calculus lectures it was decided to use Calculus with Analytic Geometry by Louis Leithold, the same text used in the regular calculus sections.

For use in the programming laboratory, we wanted a book which not only included the basics of PL/C, but also problems and examples related to the topics in calculus. In the Spring of 1972, the committee was unable to locate such a textbook, so we worked during the Spring and Summer of 1972 preparing a manual, A Computer Manual for Calculus. The book was written by Dr. William P. Love assisted by Mr. J. R. Martin and Ms. Carolyn T. Jones.

This manual consists of forty lessons covering the fundamentals of PL/C programming and topics from the elementary calculus. It was assumed that the students entering the course would have no prior knowledge of computers or programming. The beginning lessons deal with very basic computing techniques. The students are first introduced to the use of the computing equipment; keypunch, card reader and printer. They should then be able to keypunch and run on the computer a program essentially copied from the manual. Next the concept of using the algorithmic approach to solving a problem is presented and the process of drawing a flowchart is introduced.

In each of the first fourteen lessons a new PL/C topic is presented along with sample calculus problems which are worked completely (flowchart, PL/C coding, computer output). The remaining lessons concentrate on writing programs to solve problems from the calculus. As the lessons progressively become more difficult, the students are eventually led to a point where they must write algorithms and programs entirely on their own.

Below is an example of one of the student exercises in the lesson dealing with limits.

Write a program which will determine a possible value for D (delta) when given a positive epsilon, E = 0.00001, for $\lim_{x \to a}$ (SIN(X) + TAN(X))/X = L for various values of a and L.

try $a = 0$, $L = 2$;
$a = 0$, $L = 1.9$;
$a = 1$, $L = (2 + \sqrt{2})/\pi$;
$a = \frac{\pi}{4}$ $L = (2 + \sqrt{2})/\pi$.

For further reference, the table of contents and a sample lesson are included in Appendices A and B respectively.

The manual was revised in the Summer of 1973 as a result of class testing and suggestions from the faculty and students involved.

CHAPTER III

EXPERIMENTAL COURSE MATH 133, 233

ORGANIZATION

Mathematics 133 was first taught in the Fall of 1973. Two sections were offered, one meeting at 10:00 a.m. Monday through Friday, and another meeting at 1:00 p.m. Monday through Friday.

The Monday, Wednesday, Friday sessions were devoted to traditional calculus lectures. These classes were taught by Dr. Love, an experienced calculus instructor, using the Leithold text. Traditional homework problems and examinations were required of the students. The Tuesday, Thursday computer laboratory sessions were taught by Ms. Jones, an experienced programming instructor. The two instructors worked as a team; both attending all the classes to insure proper coordination between the lectures and laboratory. Frequently the calculus instructor indicated in his lectures the need or adaptability of the computer to a particular type problem. On the other hand the computing instructor pointed out certain calculus problems for which a particular programming technique might be useful. In this manner inter-relation of calculus and computing was established.

Actual "hands-on" experience with the computing equipment came outside these class periods. The students were required to keypunch their programs onto IBM cards, submit their card decks to TUCC for processing (via the card reader), and hand in a correct program to the instructor.

MOOD OF THE CLASS

For the first several weeks of the semester the students were completely fascinated with the idea of being able to write programs and run them on the computer. They were excited about the speed with which the computer could "crunch numbers" and return answers. As the assignments became more difficult and the novelty of programming diminished, many of the students became frustrated. The prospects of sitting at the now very familiar computer center running programs, correcting them and re-running them did not seem very exciting, but rather boring and tedious. One reason for this low in enthusiasm, aside from familiarity, may be that the semester was nearly over and the students had demands and pressures from other courses they were taking.

About three weeks into second semester (Math 233) the mood of the class began to change again. As the students became more proficient in programming, their former interest and enthusiasm was renewed. In fact some students were so eager that they were writing and running programs that had not been assigned. This kind of "boiling over" enthusiasm prevailed throughout the remainder of second semester.

Because the students were writing programs, which entailed spending a considerable amount of time at the computer center and working together to solve particular programming problems, they grew to know each other and like each other very much. This had the effect of unifying the class and created a relaxed atmosphere, which contributed greatly to the learning process and in general to the success of the course.

At the end of 133, the students were given a course evaluation questionaire. A copy of the questionaire and a frequency distribution for each question is included in Appendix D.

GRADES AND ENROLLMENT

The grades for Math 133 were very good. There were twenty-four A's, fourteen B's, five C's, and one D. Of the fourty-four students who took 133, thirty-seven continued to Mathematics 233 in the Spring semester, 1973. The seven students who dropped out did so for various reasons. Two dropped out of school completely, one changed to Calculus 292, and the other four did not want to take further mathematics courses. The grades for 233 were also very good; twenty-seven A's, six B's, and four C's.

PUBLICITY

Because of the nature of Calculus with Computing and the fact that it was an experimental course, the UNC-G News Bureau ran newspaper and radio publicity concerning the course. Dr. Love and Ms. Jones were interviewed by Ms. Sherry Johnson, a reporter from the University News Bureau. She wrote a feature article concerning Calculus with Computing which appeared in about fifteen regional North Carolina newspapers. A taped interview with Dr. Love was the topic of "Accent on Education", a five minute radio show, which was broadcast in thirty North Carolina cities and towns during the week of March 12-16, 1973.

EQUIPMENT USE

Since students from many other courses were using the computer equipment on campus, we had feared that the number of keypunch machines would not be sufficient for our needs. As it happens the seven keypunches seemed to be adequate for the number of students enrolled, but as more sections are added, additional keypunch machines may be required.

CHAPTER IV

A COMPARISON WITH TRADITIONAL CALCULUS: RESULTS AND CONCLUSIONS

INTRODUCTION

In most of the literature reviewed by the committee, there was very little, if any, evidence to support opinions as to the effectiveness of introducing computer-calculus. Most schools simply stated their experience was "successful", without any measure of this "success."

In an attempt to quantify the effectiveness of Math 133-233 (Calculus with Computing), an experiment was conducted to compare the calculus achievement of students who took traditional calculus (Math 191-292) and students who took Calculus with Computing. The students who took Math 191-292 were denoted the control group and the students who took Math 133-233 were designated the experimental group.

The null hypothesis for this experiment was:

> H_0: There is no difference in calculus achievement between students taking traditional and students taking Calculus with Computing.

The students who took Math 133-233 had more exposure to calculus, and had the added advantage of being able to solve calculus problems using the computer. For this reason, it was anticipated that if the null hypothesis was rejected, the calculus achievement for these students would be greater. A significance level of $\alpha = .05$ was used throughout in determining the rejection of the null hypothesis, although results significant at $\alpha = .10$ are mentioned.

TESTING INSTRUMENT

The testing instrument was a standardized mathematics test -- Calculus: Form A, Parts I and II -- acquired from the Educational Testing Service (ETS), Princeton, New Jersey. Each part consists of thirty questions and requires forty minutes to complete. Part I contains questions concerning analytic geometry, functions, limits, derivatives, continuity, and some integration. Part II contains questions concerning the exponential function, the logarithmic function, trigonometric functions, and integration.

TESTING PROCEDURE

All students in the control group and the experimental group were given Part I of the test at the beginning of the Fall semester (August, 1972). The scores on this test represent the knowledge of calculus prior to the first semester.

The same students were retested using Part I of the test at the end of Fall semester (December, 1972). The scores on this test represent the calculus achievement after one semester of calculus.

Part II was given to the students in the control group and the experimental group at the end of the Spring semester (May, 1973). The scores on this test represent the calculus achievement after the students had completed two semesters of calculus.

The above three tests will henceforth be referred to as pre-test, mid-test, and post-test respectively. Each time the test was given, it was administered to both groups during regular class meetings within a three day period. To insure experimental objectivity, a standard explanation was devised for each test and was read to each class prior to the testing period.

MEASURES USED

The following measures were used to compare the experimental and control groups

1. Pre-test score: The score on Part I of the test (August, 1972).
2. Mid-test score: The score on Part I of the test (December, 1972).
3. Difference score: The difference score is the student's mid-test score minus his pre-test score, representing the increase in calculus achievement from August, 1972, to December, 1972.
4. Limit-difference score: It was expected that the students in the experimental group might have an increased understanding of the limit concept due to a programming lesson concerning limits. For this reason the two groups were also compared according to the number of correct answers (limit-score) to two selected limit problems in Part I. The limit-difference score is the increase in the limit-score from pre-test to mid-test.
5. Post-test score: The score on Part II (May, 1973).

METHOD OF COMPARISON

For the comparison, a one-way analysis of variance was performed on the scores for the pre-test, mid-test, difference, limit-difference, and post-test. A packaged program, ANOVA, included in Tele-Storage and Retrieval (TSAR), was used for the analysis of variance.

RESULTS

Because there were some students who took all three tests, and others who took only one or two of the tests, the results are divided into the following categories:

1. Table 1 contains the results for students who took both the pre and mid-tests. The scores compared are pre-test, mid-test, difference, and limit-difference.
2. Table 2 contains the results for students who took all three tests. The scores compared are pre-test, mid-test, difference, limit-difference, and post-test.
3. Table 3 contains the results for students who took the post-test regardless of whether they had taken the first two tests. The scores for the post-test are compared.

The number of students in the control and experimental groups applicable to each category is given in the appropriate table.

TABLE 1

RESULTS FOR STUDENTS WHO TOOK PRE-TEST AND MID-TEST

	Experimental Group N=44		Control Group N=106		F value with (1,148) d.f.	Significance Level
	Mean	S.D.	Mean	S.D.		
Pre-test score	3.93	3.19	4.58	3.23	1.138	not significant at $\alpha = .10$
Mid-test score	10.45	4.23	9.28	3.98	2.599	not significant at $\alpha = .10$
Difference score	6.52	3.09	4.74	3.72	7.884	significant at $\alpha < .01$
Limit-difference score	0.52	0.63	0.16	0.63	10.197	significant at $\alpha < .01$

TABLE 2

RESULTS FOR STUDENTS WHO TOOK PRE-TEST, MID-TEST, AND POST-TEST

	Experimental Group N=37		Control Group N=54		F value with (1,89) d.f.	Significance Level
	Mean	S.D.	Mean	S.D.		
Pre-test score	3.92	3.17	4.35	3.38	0.379	not significant at $\alpha = .10$
Mid-test score	10.92	4.15	10.66	4.19	0.940	not significant at $\alpha = .10$
Difference score	7.00	3.10	5.70	3.47	3.340	significant at $\alpha < .10$
Limit-difference score	0.54	0.650	0.22	0.604	5.735	significant at $\alpha < .05$
Post-test score	11.11	3.41	8.77	3.07	11.505	significant at $\alpha < .001$

TABLE 3

RESULTS FOR STUDENTS WHO TOOK THE POST-TEST, REGARDLESS OF WHETHER THEY TOOK THE FIRST TWO TESTS

	Experimental Group N=37		Control Group N=67		F value with (1,102) d.f.	Significance Level
	Mean	S.D.	Mean	S.D.		
Post-test score	11.11	3.41	8.56	3.06	15.123	significant at $\alpha < .0005$

CONCLUSIONS AND DISCUSSIONS

There was no significant difference in calculus achievement between the two groups on pre-test scores (see Tables 1 and 2), indicating that (at least on initial knowledge of calculus) the control and experimental groups were comparable. Additionally, when the mid-test scores for the two groups were analyzed without considering the students initial achievement (as measured by pre-test), no difference was found (see Tables 1 and 2). However, making use of difference scores for each student enabled the detection of a significant difference between the two groups of students who took both the pre-test and mid-test (see Table 1). When the same analysis was made for the groups in Table 2, the results were not significant at $\alpha = .05$ (see Table 2).

As previously mentioned, special attention was given to a comparison of achievement on the concept of a limit. As was anticipated, students in the experimental group had significantly higher scores for the limit-difference than students in the control group (see Tables 1 and 2).

Most importantly a significant difference was observed in the post-test scores for the two groups (see Tables 2 and 3). It was expected that the experimental group would score significantly higher on the post-test than the control group. The students in the experimental group had obtained introductory programming proficiency by the end of Math 133. Hence more concentration was given to calculus problems than programming problems in Math 233. Since the basic programming techniques had been taught in 133, the lessons in the laboratory manual (taught in 233) were more closely related to topics being covered in the calculus sessions. This enabled the students to have a better understanding of certain calculus concepts (for example, limits, integration, and infinite processes). This expectation for the experimental group was realized.

Because the experimental group had significantly higher scores on the post-test and the difference score than the students in the control group, it appears that computing contributed to their understanding of elementary calculus.

Other factors which may have influenced the results are:

1. Although students were enrolled for the course on a voluntary basis, it is possible that the more motivated or better students were naturally attracted to the experimental course. Also there was a difference in the population characteristics for the two groups (see Appendix C). Among the characteristics which may have influenced the results were the respective ages, number of mathematics majors, classification, and semesters of previous mathematics.
2. The teachers in the class naturally influenced the results somewhat. Of the six teachers for Math 191-292, two were graduate students who had taught calculus at least twice prior to the experiment, two were instructors who had taught calculus at least four times prior to the experiment, one was an assistant professor (a "seasoned" calculus teacher),

and one was a full professor (a "seasoned" calculus teacher). The teacher for the calculus sessions of Math 133-233 was an assistant professor (experienced in calculus instruction), and the laboratory sessions were taught by a graduate student (experienced in programming instruction). Since Math 133-233 was an experimental course, the teachers were highly motivated and were greatly interested in the outcome of the course. Hence these instructors probably spent more time working with the individual students.

Calculus with Computing proved to be a highly successful endeavor from both an educational and an experimental standpoint. Based on student opinions, it fulfilled their expectations and requirements. From a faculty viewpoint the course was considered successful; this was supported by the UNC-G Curriculum Committee when the course was approved as a permanent departmental offering (Spring, 1973). Statistically it was shown that students in Calculus with Computing were superior in calculus achievement, while at the same time they obtained introductory programming proficiency.

APPENDICES

APPENDIX A

TABLE OF CONTENTS FOR A COMPUTER MANUAL FOR CALCULUS

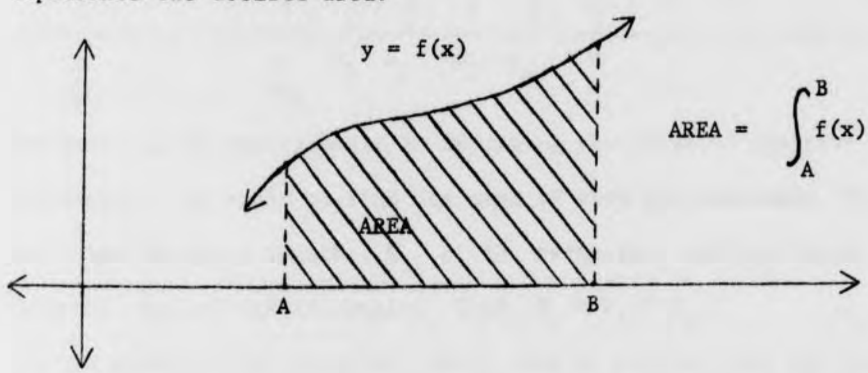
APPENDIX B

SAMPLE LESSON FROM A COMPUTER MANUAL FOR CALCULUS

Lesson 24 THE DEFINITE INTEGRAL

1. PURPOSE: The objective of this lesson is to understand the definition of Riemann Sums and how they are used to approximate the value of a definite integral of a function.

2. THEORY: Essentially, the problem is to determine the value of the definite integral of a function; $\int_A^B f(x)dx$. One may visualize this value as the area under the curve $y=f(x)$ to the x-axis between the vertical lines $x=A$ and $x=B$. The shaded area in the figure below represents the desired area.

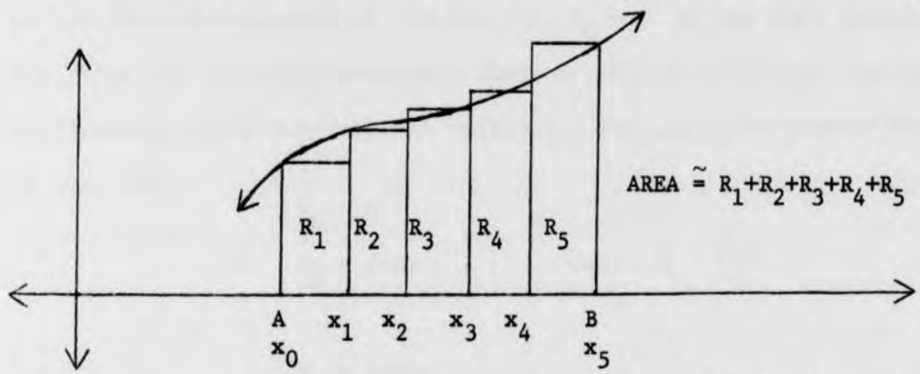


$$\text{AREA} = \int_A^B f(x)\, dx$$

One is tempted to think that there is always SOME area under every function, but this is not true. There are some functions defined at every point in the interval [A,B] such that area is not even defined under them.

Let us therefore restrict the functions under consideration to be functions which are defined over [A,B], positive over this interval, and continuous over this interval. One can evaluate the definite integral when these conditions are not satisfied, but for instructional purposes these restrictions are helpful.

In order to find the area under the curve, $y = f(x)$, one actually approximates the area by cutting the region into small rectangles, perpendicular to the x-axis, and sums up their areas. In the figure below, the area, A, is divided into 5 sub-rectangles, $R_1, R_2, \ldots, R_5$.



The Area, A, is approximated by adding up the areas of the five sub-rectangles. In order to find the area of each sub-rectangle, R_i, one must know the base (width), W_i, of the rectangle, and the height (length) H_i, of the rectangle. Thus $R_i = W_i \cdot H_i$.

In general, the interval [A,B] may be divided into sub-intervals which serve as the bases for the rectangles. These sub-intervals may be made various lengths, if necessary, but it is often easier from the computation standpoint to make them all of equal length. Also, there may be as many sub-intervals as one desires. In the figure above, there are five sub-intervals, but one could have selected five thousand just as easily. In the above figure, [A,B] is divided into five sub-intervals, each of length $(B-A)/5 = W_i$. If one should divide the interval into N sub-intervals, then the width of each rectangle (the length of each sub-interval) is $W_i = (B-A)/N$.

Lesson 24 (cont)

Frequently the width of each sub-interval, W_i, is denoted by DX, hence $DX = (B-A)/N$ where N is the number of sub-intervals.

It is important to determine the coordinates of the points which define the sub-intervals of [A,B]. If $X_0 = A$ is the left endpoint and $X_N=B$ is the right endpoint, then $X_1, X_2, X_3, \ldots, X_{N-1}$ are the coordinates of the sub-interval endpoints, and their coordinates may be found by:

$$X_0 = A$$

$$X_1 = A+DX \qquad \text{where } DX = \frac{B-A}{N}$$

$$\vdots$$

$$X_i = A+iDX$$

$$\vdots$$

$$X_N = B$$

Next, one must determine the appropriate height, H_i, for each sub-rectangle R_i. Suppose, for example, one wanted to approximate the height, H_1, of the first rectangle, R_1, in the figure above. There are many possible ways this may be done, all of which are different but all of which are acceptable. Here are some possibilities:

$H_1 = f(A)$ (left endpoint of sub-interval)

$H_1 = f(x_1)$ (right endpoint of sub-interval)

$H_1 = f\left(\dfrac{A+x_1}{2}\right)$ (midpoint of sub-interval)

$H_1 = f(P_1)$ (P_1 is any point in sub-interval)

Hence one approximation for the area of R_1 is $(H_1)(DX)$ or

$$R_1 = f(x_1)\cdot(B-A)/5$$

Lesson 24 (cont)

For the sake of convenience, we arbitrarily pick x_1, the right-hand end point of the first sub-interval.

Hence, to find the approximate area of the ith sub-rectangle, R_i, the width is $W_i = (B-A)/N$ and the height, $H_i = f(x_i)$, where N is the number of sub-rectangles and x_i is the right-hand end point of the ith sub-interval.

Thus to obtain an approximation for the entire area, one must sum up all the areas of the sub-rectangles:

$$\text{Area approximation} = \sum_{i=1}^{5} R_i$$

$$= \sum_{i=1}^{5} f(P_i) \cdot DX$$

This sum is called a RIEMANN SUM for the function. Obviously there may be many possible Riemann Sums since this value depends upon how many sub-rectangles are being used and the selection of the points, P_i. Notice as the number of sub-rectangles is increased (N → INF) then the error between the true area and the Riemann Sum approximation becomes smaller and smaller. As N becomes extremely large, the error factor reduces to zero.

In general, the exact area under the curve $f(x)$ is defined to be the limit as N tends to infinity of the Riemann Sum approximation.

Lesson 24 (cont)

$$\text{Area} = \lim_{N\to\infty} \sum_{i=1}^{N} f(P_i)\ DX$$

$$= \lim_{N\to\infty} \sum_{i=1}^{N} f(x_i)(B-A)/N$$

One must be careful, for this limit may not exist and hence the true area is not even defined. If the limit does exist, however, then the value of this limit is exactly the value of the definite integral of the function.

$$\int_A^B f(x) = \lim_{N\to INF} \sum_{i=1}^{N} f(P_i)\ \frac{(B-A)}{N}$$

provided this limit exists.

Remember that since the computer can not determine exactly if a limit exists or not, but can only provide some evidence, then it is impossible to precisely evaluate a definite integral using Riemann Sums, and hence to find the area under the curve.

3. EXAMPLE 1: Write a program which will calculate a Riemann Sum for a functions over an interval [L,R]. For the height of each sub-rectangle, H_i, choose the value of the function evaluated at the <u>left endpoint</u> of the sub-interval. Use this program to compute a Riemann Sum for $f(x) = \sqrt{x+2}$ over the interval [0,2] for N=1, 2,4,16 and 100. Note: This program includes an option. This option allows one to evaluate a Riemann Sum for the <u>built-in</u> function ($f(x)= \sqrt{x+2}$) or

Lesson 24 (cont)

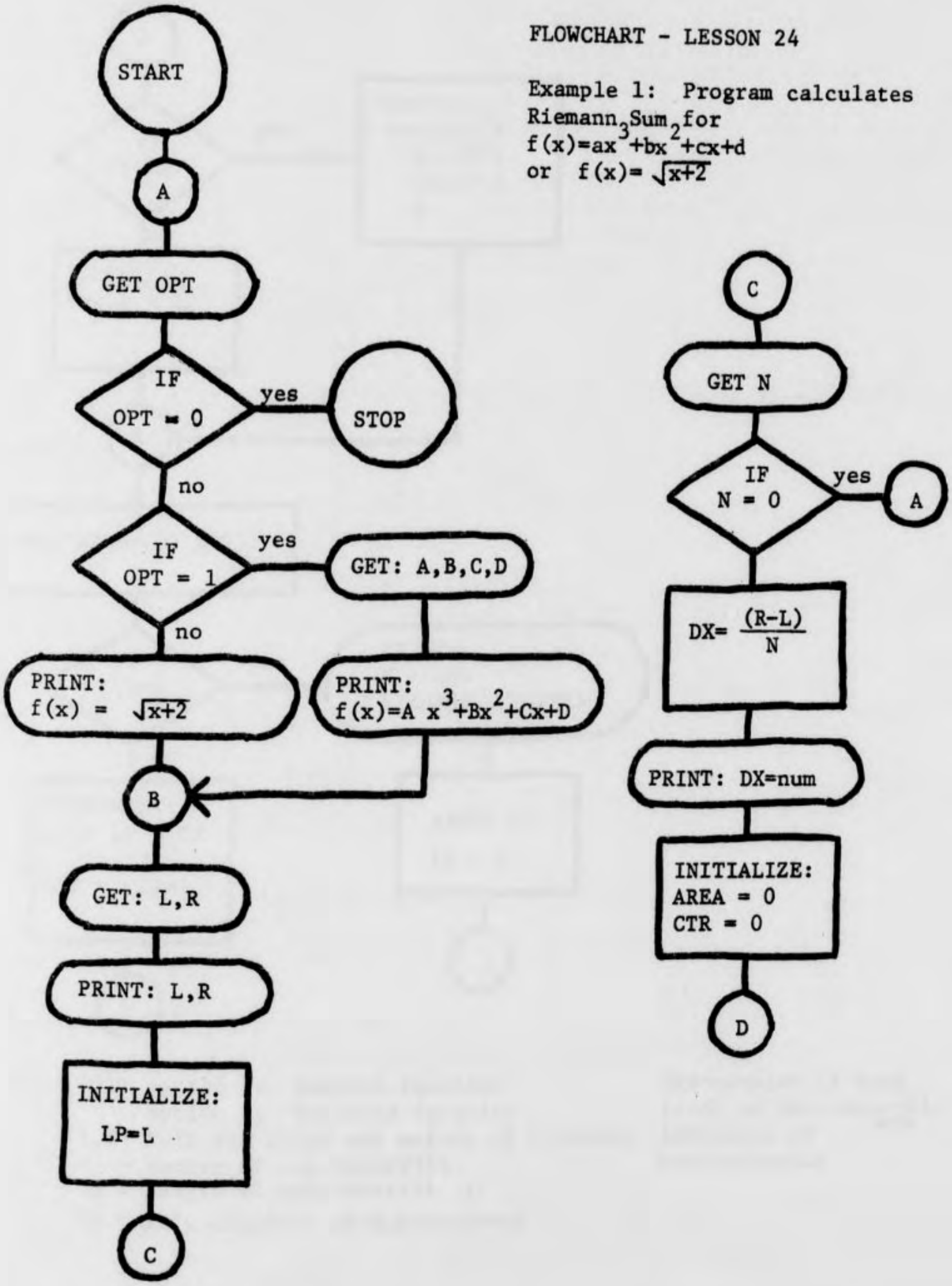
to evaluate a Riemann Sum for a read-in function of the form $f(x)=Ax^3+Bx^2+Cx+D$. Hence for polynomials of degree 3 or less, one need only input coefficient data, while other functions require card changes in the main program. Using the option, this program also computes a Riemann Sum for the functions $f(x) = x$ over [0,2] and $f(x) = x^2$ over [0,2].

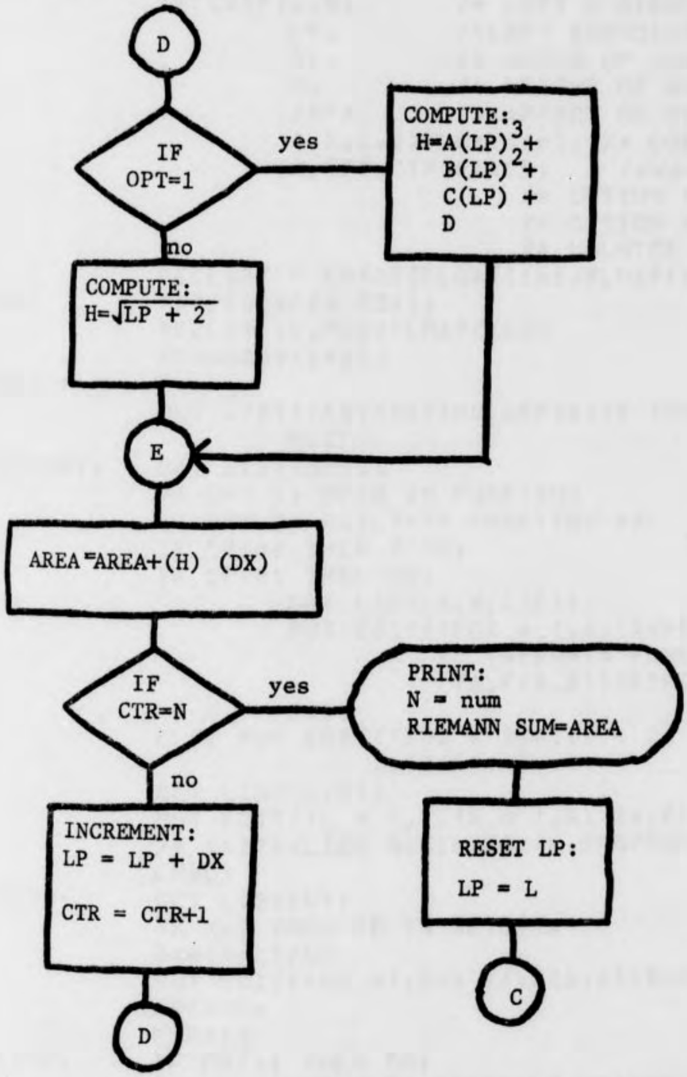
A. One flowchart for this problem is shown on the following page:

Lesson 24 (cont)

FLOWCHART - LESSON 24

Example 1: Program calculates Riemann Sum for $f(x)=ax^3+bx^2+cx+d$ or $f(x)=\sqrt{x+2}$





Key: OPT= option 1: Read-in function
option 2; Built-in function
L,R left and right end points of interval
N = number of sub-intervals
DX = length of sub-interval
LP = left endpoint of sub-interval

CTR=counter to keep track of sub-interval
AREA=area of sub-rectangle

B. The program based on this flowchart is shown below:


```
          /*PROGRAM CALCULATES A RIEMANN SUM */
RIEMANN:  PROCEDURE OPTIONS(MAIN);
          DECLARE(L,R,        /* LEFT & RIGHT END POINTS*/
                LP,           /*LEFT ENDPOINT OF SUBINTERVAL*/
                DX,           /* WIDTH OF SUBINTERVAL */
                H,            /* HEIGHT OF SUBRECTANGLE */
                AREA,         /* APPROX OF RIEMANN SUM */
                A,B,C,D)FLOAT(16), /* COEFFICIENTS OF EQUATION*/
               (N,OPT,CTR)FIXED;      /*NUMBER OF SUBINTERVALS*/
                              /* OPTION 1: READ IN EQUATION */
                              /* OPTION 2:BUILT IN EQUATION */
                              /* COUNTER */
          DECLARE F ENTRY(FLOAT(16),FLOAT(16));
F:        PROCEDURE(X,FOX);
          DECLARE(X,FOX)FLOAT(16);
          FOX=SQRT(X+2);
END F;
          PUT LIST('ESTIMATING DEFINITE INTEGRAL BY RIEMANN SUMS')
                PAGE;
GETOPT:   GET LIST(OPT);
          /* OPT 1: READ IN FUNCTION
             OPT 2: BUILT-IN FUNCTION */
          IF OPT=0 THEN STOP;
          IF OPT=1 THEN DO;
                GET LIST(A,B,C,D);
                PUT EDIT('FOX = ',A,'X**3+',B,'X**2+',C,'X+',D,
                      '   (READ-IN FUNCTION)')
                         (A,F(6,2))SKIP(4);
                END;
          ELSE PUT EDIT('FOX = SQRT(X + 2)    (BUILT-IN FUNCTION)')
                    (A)SKIP(4);
          GET LIST(L,R);
          PUT EDIT('L = ',L,'R = ',R) (A,F(6,2),X(4))SKIP(2);
          /* INITIALIZE SUBINTERVAL ENDPOINT TO LEFT ENDPOINT*/
          LP=L;
GETN:     GET LIST(N);
          IF N=0 THEN GO TO GETOPT;
          DX=(R-L)/N;
          PUT EDIT('DX =',DX) (A,F(6,2))SKIP(4);
          AREA=0;
          CTR=1;
LOOP:     IF OPT=1 THEN DO;
                H=A*(LP**3)+B*(LP**2)+C*LP+D;
                GO TO NEXT;
                END;
          CALL F(LP,H);
NEXT:     AREA=AREA+(DX*H);
          IF CTR=N THEN DO;
                PUT EDIT('N=',N,'RIEMANN SUM =',AREA)
                         (A,F(4),X(5),A,F(10,7))SKIP(2);
                LP=L;   /* SET SUBINTERVAL BACK TO L */
                GO TO GETN;
                END;
          LP=LP+DX;   /* MOVE TO NEXT SUBINTERVAL*/
          CTR=CTR+1;
          GO TO LOOP;
END RIEMANN;
```

C. The output for $f(x) = \sqrt{x+2}$ is shown below:

```
ESTIMATING DEFINITE INTEGRAL BY RIEMANN SUMS

FOX = SQRT(X + 2)    (BUILT-IN FUNCTION)
L =    0.00     R =   2.00

DX =  2.00
N=    1      RIEMANN SUM = 2.8284271

DX =  1.00
N=    2      RIEMANN SUM = 3.1462643

DX =  0.50
N=    4      RIEMANN SUM = 3.2991159

DX =  0.12
N=   16      RIEMANN SUM = 3.4109687

DX =  0.02
N=  100      RIEMANN SUM = 3.4418539
```

D. The output for $f(x) = x$ is shown below:

```
FOX =   0.00X**3+  0.00X**2+  1.00X+  0.00  (READ-IN FUNCTION)
L =   0.00     R =   2.00


DX =  1.00
N=   2      RIEMANN SUM = 1.0000000


DX =  0.50
N=   4      RIEMANN SUM = 1.5000000


DX =  0.12
N=  16      RIEMANN SUM = 1.8750000


DX =  0.02
N= 100      RIEMANN SUM = 1.9799999


DX =  0.01
N= 200      RIEMANN SUM = 1.9899999
```

E. The output for $f(x)=x^2$ is shown below:

```
FOX =  0.00X**3+ 1.00X**2+ 0.00X+ 0.00 (READ-IN FUNCTION)
L =  0.00    R =  2.00

DX =  1.00
N=   2     RIEMANN SUM = 1.0000000

DX =  0.50
N=   4     RIEMANN SUM = 1.7500000

DX =  0.12
N=  16     RIEMANN SUM = 2.4218750

DX =  0.02
N= 100     RIEMANN SUM = 2.6267999
```

Lesson 24 (cont)

F. Comments on program:

The data input for this program was of two forms depending on the option used:

for option 1: OPT,A,B,C,D,L,R,N1,N2,N3,...,0

for option 2: OPT,L,R,N1,N2,N3,...,0

For this program the data was:

2,0,2,1,2,4,16,100,0

1,0,0,1,0,0,2,2,4,16,100,200,0

1,0,1,0,0,0,2,2,4,16,100,0,0

The program reads OPT=2, gets the interval [0,2], then reads N=1,2,4,16,100. When it reads N=0 this causes it to get a new option. The new option is OPT=1 in the next card. It then reads A,B,C,D as 0,0,1,0 for the interval [0,2] for N=2,4,16,100,200. When it reads N=0, this causes it to read the new option, which is again OPT=1 for the function $0X^3+1X^2+0X+0$ over [0,2] for N=2,4,16,100. When it reads N=0, this causes it to read the new option, OPT=0 which terminates the program.

LP is used to be the left endpoint for each sub-interval. Hence for the first sub-interval LP=L, then for the second sub-interval, LP=L+DX, etc.

Lesson 24 (cont)

The exact values for the definite integrals are:

$$\int_0^2 \sqrt{x+2}\, dx = 4/3(4-\sqrt{2}) \approx 3.446138$$

$$\int_0^2 x\, dx = 2$$

$$\int_0^2 x^2\, dx = 8/3 \approx 2.666667$$

Compare these values with the Riemann Sums when N=100. Observe how the Riemann Sums provide better approximations as N increases from N=1,2,4,16, to 100. As N becomes very large the difference between the Riemann Sum and the exact value reduces to zero.

To illustrate how the program computes Riemann Sums graphically, consider $f(x) = x$ over [0,2].

(a) when N=4

$DX=(2-0)/4=.5$

$A_i=H_i \cdot DX$

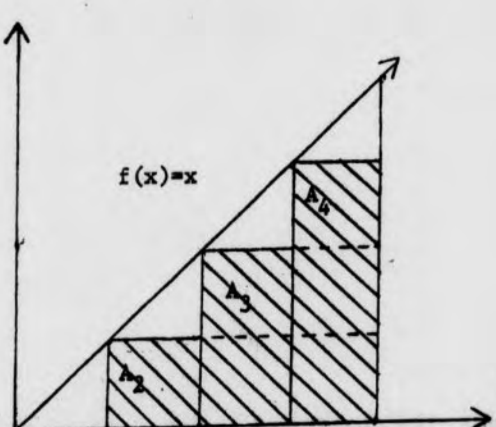
$A_1=(0)(.5)=0$

$A_2=(.5)(.5)=.25$

$A_3=(1.0)(.5)=.50$

$A_4=(1.5)(.5)=.75$

RIEMANN SUM = $A_1+A_2+A_3+A_4 = 1.50$



4. STUDENT EXERCISES:

(a) Modify the program in Example 1 to compute a Riemann Sum by using the height of the sub-rectangle to be the value of the function evaluated at RP, the right endpoint of each sub-interval.

Lesson 24 (cont)

Use this program on the functions $\sqrt{x+2}$, x, and x^2 over [0,2] and compare your results with those from Example 1.

(b) Modify the program in Example 1 as in part (a) above, using MP, the midpoint of each sub-interval. Compare your results with those from Example 1.

(c) Using either program above, determine by means of Riemann Sums the approximate values for these definite integrals. At this point you should not have the techniques for finding the integrals, but you will learn them later in your class lectures.

$$\int_1^2 1/x \, dx$$

$$\int_0^{\pi/2} \sin x \, dx$$

$$\int_0^1 1/(x^2+1)dx$$

$$\int_1^2 \mathrm{LOG}(x)dx$$

$$\int_0^1 4\sqrt{1-x^2}\, dx$$

(d) Write a program which will compute the definite integral of any polynomial of the form $Ax^4+Bx^3+Cx^2+Dx+E$ over an interval [L,R]. Use this program to evaluate some polynomials.

Lesson 24 (cont)

(e) Write a program which will compute the definite integral of any function of the form $f(x) = (f(x))^P(f'(x))$ wher $P \neq -1$ over an interval [L,R]. Use this program to evaluate some integrals.

APPENDIX C

POPULATION CHARACTERISTICS

	COURSE	MATH 191								MATH 133		
	SECTION	01	02	03	04	05	06	07	TOTAL	01	02	TOTAL
	NUMBER	16	14	7	14	17	25	13	106	28	16	44
CLASS	FRESHMAN	11	9	3	6	8	9	8	54	26	16	42
	SOPHOMORE	1	2	4	4	6	12	3	32	1	0	1
	JUNIOR	2	3	0	3	3	2	2	15	0	0	0
	SENIOR	0	0	0	0	0	1	0	1	0	0	0
	GRADUATE	2	0	0	1	0	1	0	4	1	0	1
MAJOR	MATH	0	3	2	1	0	0	4	10	20	13	33
	SCIENCE	5	2	3	6	2	6	3	27	1	3	4
	OTHER	5	8	1	4	10	12	5	45	3	0	3
	UNDECIDED	6	1	1	3	5	7	1	24	4	0	4
YEARS OF HIGH SCHOOL MATH	TWO	1	0	0	1	0	0	0	2	0	0	0
	THREE	3	0	1	3	3	3	1	14	2	1	3
	FOUR	10	12	4	8	12	22	10	78	19	11	30
	FIVE	2	2	2	2	2	0	2	12	7	4	11
SEMESTERS OF HIGH SCHOOL CALCULUS	ZERO	11	13	5	12	10	23	12	86	21	12	33
	ONE	1	1	1	1	5	1	0	10	6	2	8
	TWO	4	0	1	1	2	1	1	10	1	1	2
	THREE	0	0	0	0	0	0	0	0	0	1	1
SEMESTERS OF COLLEGE MATH	ZERO	10	9	5	7	9	11	7	58	25	15	40
	ONE	3	0	0	3	0	4	3	13	2	0	2
	TWO	3	5	2	3	8	10	3	34	1	1	2
	THREE	0	0	0	1	0	0	0	1	0	0	0

APPENDIX D

QUESTIONNAIRE CONCERNING MATH 133

(The numbers below each item represent the number of students who selected that item).

Please complete this form. Do NOT sign you name. Be honest.

1. Current major:

a. Math	b. Science	c. Undecided
23	2	1

2. Status:

a. Freshman	b. Sophomore	c. Junior	d. Senior	e. other
26				

3. In general, you feel that your progress and understanding in calculus has been:

a. excellent	b. pretty good	c. satisfactory	d. poor
8	12	5	1

4. In general, you feel that your progress and understanding in computing has been:

a. excellent	b. pretty good	c. satisfactory	c. poor
5	12	8	1

5. Did you decide to change your major as a result of being in 133?

a. no	b. yes
26	

6. How do you feel about the quality of teaching (explanations) in calculus?

a. excellent	b. satisfactory	c. poor	d. horrible
24	2		

7. How do you feel about the quality of teaching (explanations) in computing?

a. excellent	b. satisfactory	c. poor	d. horrible
11	10	4	

8. What best describes the calculus homework?

a. too much	b. right amount	c. too little	d. too dificult
	24	1	1

9. What best describes the computing assignments?

a. too hard	b. about right	c. too easy	d. too many	e. too few
	25	1	1	

10. How do you feel about the rate of instruction in calculus?

a. too fast	b. about right	c. too slow
7	18	1

11. How do you feel about the rate of instruction in computing?
    a. too fast b. about right c. too slow
    3 23

12. Did the computer exercises help your understanding of calculus?
    a. frequently b. occasionally c. rarely d. never
    11 11 4

13. 133 was taught with lecture and laboratory separate, but related. Which of these ways of teaching calculus and computing would you prefer?
    a. continue as-is b. teach computing and calculus all together
    21 4
    c. teach them as totally separate subjects

14. Which do you prefer:
    a. having two teachers b. having a single teacher
    20 6

15. Do you think 133 should be:
    a. made a permanent course b. eliminated c. changed drastically
    26

16. After completing 133, you plan to:
    a. continue to 233 b. transfer to 292 c. never take math again
    24 1
    d. none
    1

17. If you do not plan to continue in 233, please explain why:

18. How many hours-credit should be given for Math 133?
    a. 6 b. 5 c. 4 d. 3
    10 15 1

19. How do you feel about the computing facilities at UNC-G?
    a. The facilities are completely satisfactory.
    3
    b. Need more keypunch machines.
    16
    c. Need more help from assistants at computer center.
    10
    d. Need faster turn-around-time.
    10
    e. Need computer center open more frequently, longer hours, etc.
    3

20. About what percentage of time did you spend on calculus vs. computing.
(first percentage = calculus, second = computing) (both outside class)
a. 90%-10% b. 75%-25% c. 50%-50% d. 25%-75% e. 10%-90%
4 15 8 1

21. Which best describes the explanations and examples in the Manual?
a. easy to follow and generally helpful.
15
b. theory helpful, but examples not too helpful.

c. theory not too helpful, examples very helpful.
8
d. very difficult to understand and follow.
3

22. Which best describes the student exercises in the Manual?
a. generally pretty good.
23
b. too much busy work, and not too helpful.
3
c. programs much too easy.

d. programs much too dificult and time consuming.
1

23. When doing a computer assignment, which was the most difficult part?
a. solving the problem b. making a flowchart
9 3
c. writing program in PL/C d. keypunching program
4 2
e. debugging the errors
12

24. In computing, which concept did you find most difficult?
a. declare statements b. put edit statements c. if-then-else
7 7
d. do-loops and do-groups e. subroutines
10 6

25. Generally, how many times did you have to run your program to produce a finished copy?
a. 1 b. 3 c. 5 d. 7 e. more than 9
11 11 2 1

26. Roughly how many hours per week did you spend in the Computer Center?
a. 2 b. 4 c. 6 d. 10 e. more than 12
3 11 7 5 1

27. In calculus, which concepts did you find most difficult?
    a. functions b. limits c. continuity d. derivitives e. max-min.
    2 11 1 1 9

28. If you had to do everything over again, would you:
    a. take 133 b. take 191 c. take 121 d. have your head examined
    25 1

29. What grade do you expect to make in the course?
    a. A b. B c. C d. D e. F
    12 11 2

30. Can you suggest any weak points of the 133 course?

31. Can you tell us what you consider the strong points?

32. Can you make any suggestions to improve 133?

APPENDIX E

COMPUTING COST FOR MATHEMATICS 133-233

	Math 133 Number of Students = 44				Math 233 Number of Students = 37				
Month	Sept.	Oct.	Nov.	Dec.	Jan.	Feb.	March	April	May
No. of runs	815	965	644	254	236	508	732	250	1
No. of assignments	5	3	2	1	2	2	2	1	....
Total cost	$74.50	127.38	117.12	58.45	$48.06	111.47	157.47	73.09	1.00
Cost per run	$ 0.09	0.13	0.18	0.23	0.20	0.22	0.22	0.29	1.00
Cost per assignment per student	0.33	0.95	1.35	1.33	0.64	1.52	2.18	1.97	....
Runs per assignment per student	3.7	7.3	7.5	5.8	3.2	6.9	9.9	6.8	....

Cost per student (133) = $ 8.58
Cost per student (233) = $ 10.57
Average cost per student (133-233) = $ 9.49

Computing cost (133) = $377.45
Computing cost (233) = $391.09
Total $768.54

REFERENCES

1. Center for Research in College Instruction of Science and Mathematics. Computer-Calculus. Tallahassee, Florida: Center for Research in College Instruction of Science and Mathematics, Newsletter Number 2, (February, 1971).

2. Committee on the Undergraduate Program in Mathematics. Calculus with Computers. Berkley, California: CUPM Newsletter Number 4, (August, 1969).

3. Committee on the Undergraduate Program in Mathematics. Recommendations for an Undergraduate Program in Computational Mathematics. Berkley, California: CUPM, (May, 1971).

4. Davis, Phillip J. "Computer Calculus." Brown University, (December, 1967). (Mimeographed.)

5. Davis, Phillip J. "Computer Calculus-First Year Report." Brown University, (May, 1969). (Mimeographed.)

6. Diaz, Octavio. Information in a letter to Mr. J. Randolph Martin from Octavio Diaz of North Carolina A & T State University, (February, 1972).

7. Fuller, William R. Information in a letter to Mr. J. Randolph Martin from William R. Fuller of Purdue University, (March, 1972).

8. Wicht, M. C. Information in a letter to Dr. William P. Love from M. C. Wicht, Chairman of Mathematics Department, of North Georgia College, (April, 1972).