ED 111 354 IR 002 402

AUTHOR	Deeter, Charles R.
TITLE	A Numerical Calculus Course as an Early Introduction to Problem Solving.
INSTITUTION	Texas Christian Univ., Fort Worth.
PUB DATE	Jun 75
NOTE	6p.; Paper presented at the Conference on Computers in the Undergraduate Curricula (6th, Fort Worth, Texas, June 16 - 18, 1975); Related document is IR 002 399
AVAILABLE FROM	Entire Proceedings; Ted Sjoerdsma, Treasurer, CCUC, 1248 Lindquist Center for Measurement, Iowa City, Iowa 52242 ($10.00, Checks payable to the University of Iowa)
EDRS PRICE	MF-$0.76 HC-$1.58 Plus Postage
DESCRIPTORS	*Calculus; Computer Oriented Programs; *Computer Programs; Computers; *Computer Science; Course Descriptions; Higher Education; Problem Solving; *Programing; Speeches
IDENTIFIERS	FORTRAN IV

ABSTRACT

Briefly described is a numerical calculus course which provides (1) experience and practice in programing and the use of the computer in problem solving, (2) intermediate and/or advanced techniques in FORTRAN IV programing, (3) elementary error analysis, (4) programing efficiency, and (5) documentation of programs. The course, designed to fulfill the needs of several departments at Texas Christian University--background for physics, chemistry, and accounting majors, as well as for students who plan to enter graduate study in mathematics--enables the student to experience, early in his training, some of the ways in which computers can be used, and to see some of the considerations which must be taken in using the computer as a problem solving tool. (Author/KKC)

A NUMERICAL CALCULUS COURSE AS AN EARLY INTRODUCTION TO PROBLEM SOLVING

by

Charles R. Deeter
Department of Mathematics
Texas Christian University
Fort Worth, Texas 76129

Introduction

The Department of Mathematics at Texas Christian University offers degree programs in both mathematics and computer science. Like most other departments of mathematics, it also provides courses to help satisfy the needs of other departments. Among the service courses shared by the mathematics and computer science programs and the programs of other departments are the calculus sequence and introductory computer sicence.

Because TCU is a medium size, private university, all curriculum planning must provide for the highest degree possible of multiple function of the courses offered, in order to insure sufficient broadness to serve adequately the various constituencies. For example, planning of the main-line calculus sequence must try to provide the background necessary for physics, chemistry, and accounting majors when they need it, as well as provide the necessary foundation for students who plan to enter eventually graduate study in mathematics.

The author of this paper has undertaken, in the past three years, the development of a numerical calculus course with the goals of multipurpose service in mind. The course developed mainly from a remark concerning the strong orientation of the computer science program toward study of computer systems to the near exclusion of any study or experience in the use of the computer as a tool for problem solving.

The Role of the Numerical Calculus Course in the Mathematics and Computer Science Programs

The numerical calculus course described here is based primarily upon the author's philosophy concerning the interaction of the mathematics and computer science programs with each other and with the programs of other departments, as well as his perception of the training and educative needs of the students in those various programs. Thus, the course has been developed (with continuing modification) to provide opportunities for the student in the following broad categories.

(1) Experience and practice in programming and the use of the computer in problem solving.

(2) Intermediate and/or advanced techniques in FORTRAN IV programming (e.g., subprograms).

(3) Elementary error analysis.

(4) Programming efficiency (e.g., simplicity of programming vs. accuracy and CPU time).

(5) Documentation of programs.

These categories implicity describe both the principal objectives of the course and the methods by which it proposes to serve the student constituency.

For mathematics majors and for majors from other departments, numerical calculus provides an opportunity, relatively early in their academic careers, to see how the computer may be used to advantage in solving problems which are related to their disciplines and which they have encountered in their calculus course.

The computer science major is provided with an introduction to very elementary numerical analysis, and can gain valuable experience in the facet of his discipline which most closely ties it to the subject areas which use the computer for problem solving. All students, whatever their major, are afforded the chance to practice their programming skills and to gain some basic knowledge concerning the finer points of the efficient use of the computer.

Course Organization

Typically, the numerical calculus course is a 3 semester hour course and is organized into two 1-hour lectures and one 2-hour laboratory per week.

As now constituted, the objectives outlined in the previous section are approached through the development of algorithms and methods of solution of familiar problems which the students have encountered in elementary calculus and differential equations. Current prerequisites for the course include two semesters of calculus and one course in computer science. The computer science prerequisite is usually fulfilled by the introductory computer science course, but the purpose is primarily to ensure that the student has been introduced to elementary programming methods and has some knowledge of at least one programming language. Prerequisites for the course do not include any study of linear algebra or matrix theory, so no numerical methods in this area are considered. (Additionally, the author feels that too little time is available in this course to do justice to the great importance of numerical methods in linear algebra.) The part of the course dealing with differential equations requires only the very basic ideas ordinarily encountered within the elementary calculus courses, so no prerequisite in this area is specified.

A positive attempt is made to use examination procedures as a learning process for the student. Thus, especially for the take-home examinations, problems with a definite applications point of view are used, students being required to describe mathematical methods for their solution and then implement these methods on the computer to arrive at numerical (quantitative) answers.

Some Conclusions

Enrollment in the numerical calculus course has largely been mathematics and computer science majors (the course is required for the Bachelor of Science in Computer Science degree), but several physics and chemistry majors have either taken or audited the course.

Student reaction, while taking the course, has been (typically) a feeling of overwork (much of which is due, in the author's opinion, to their own tendencies toward procrastination). Drop rates seem no higher than normal, however, averaging approximately 2 students per 15 enrollments per semester. Post-course reaction has been almost unanimously favorable (at least from those successfully completing the course), with the most frequent comments in connection with the opportunity for concentrated and extensive programming experience and/or the opportunity to work "pseudo-real-world" problems. (These reactions usually do not begin to surface until midway through the semester following the course, or later.) The most frequent negative evaluations concern what the students feel to be an undue emphasis on error analysis, although time is not available to pursue this topic in anything but superficial depth.

Three principal ingredients seem to be necessary for the success of the course: (1) continual revision of the course and the laboratory exercises to maintain "freshness" and progress toward the objectives of the program; (2) reasonably fast "turn-around" for student programs to enable the students to accomplish the laboratory assignments; (3) a capable laboratory assistant, both for helping students with their programs and for checking completed assignments.

A large scale revision of the course currently being taught at TCU may become necessary if current experiments in integrating computer use with the introductory calculus course are successful. Institutions where computer calculus courses already exist should, by all means, exploit the additional experience of the students to up-grade the topics included in the numerical calculus course.

Summary

The use of a numerical calculus course at a fairly early stage in mathematics and computer science curricula in order to give the student early experience in the use of the computer in a problem solving environment, as well as to provide extensive experience and practice in the art of programming, appears to be both feasible and desirable. This conclusion is based upon the author's experience in the design, teaching, and continuing development of such a course which embodies some elements of intermediate programming, an introduction to error analysis, and much work in programming elementary mathematical methods and algorithms for the solution of problems ordinarily encountered in an elementary calculus course.

Clearly, the course described here embodies no new, radical, or highly innovative changes in the usual numerical calculus course. Rather, the emphasis has been changed to enable the student to experience, early in his training, some of the ways in which computers can be used, and to see some of the considerations which must be taken in using the computer as a problem solving tool. It is within this change of philosophic viewpoint that the author feels one can find most of the innovative features of the course. Specifically, the complete freedom of the student to choose his or her own programming language (except for an occasional request for a FORTRAN subprogram) clearly places the emphasis of the course on methods and useful results rather than tying it into a particular language. This emphasis on methods and usable results is also stressed by the number of programs required in the course (about 70 to 80 during the semester usually). The series of problems which require the student to evaluate the same five definite integrals by a large variety of methods, and then to compare programming efficiency, CPU time, accuracy, etc., of all methods, is a feature not usually encountered in a course at this level. Similarly, the "Ghost Program" is used to impress the student with the importance of good documentation, but the problem itself increases his knowledge of the problem solving capabilities of the computer.

While it is true that the course described here could be, and probably has been, designed and offered by many departments of mathematics and/or computer science, the author feels that, based upon fairly wide ranging informal discussions, many departments, especially smaller ones, have not attempted such a course. It has been his intention therefore, to use this opportunity as much as a means of encouragement in those places, as for a description of a highly innovational program which might not be attractive to smaller institutions with limited capacities. Perhaps seeing that this approach can work will encourage at least some serious thought for those involved in the design of computer science and mathematics curricula, as well as stimulate the generation of ideas for similar courses at other schools.

APPENDIX I

Lecture Topics and Timetable

Topic*	Lecture Periods**	Related Course Components
Error analysis, evaluation of functions	3	Error analysis, programming efficiency, use of subroutines
Solution of equations	6	Convergence of iterative processes, error analysis, programming for general use
Limits	1	Programming logic applied to elementary processes, quantizing error, overflow and underflow errors, documentation
Differentiation	2	Round-off error, computer verification of known processes
Integration	7	Algorithms for quadrature, truncation error, round-off error, error control, efficient CPU usage ("time economics"), documentation
Differential equations	6	Propagation of error, control of propagation of error, stability, algorithms for solution, multipurpose subroutines

*Other topics touched upon when time allows have included data fitting (interpolation and least-squares) and elementary internal analysis.

**Three periods are used for examinations and review. The usual examination treatment includes one in-class exam and one take-home exam, plus the final examination.

APPENDIX II

Laboratcry Topics and Timetable

Laboratory Session	Exercise Number	Topics
1	1	FORTRAN subprograms, efficiency of Horner's nesting procedure for evaluation of polynomials
2	——	Review of subprograms, distribution of take-home exam covering subprograms
3	2	Evaluation of functions, critique of examination
4	3	Solution of equations by general iterative processes and Newton-Raphson iteration
5	4	Limits, "epsilon-delta", limits at infinity, documentation
6	5	Numerical differentiation
7,8,9,10	6,7,8,9	Numerical integration
(11)	10	"Ghost" program: undocumented program whose purpose the student is asked to try to discover
11,12,13,14	——	Cooperative in-class program for solution of differential equations by 4th order Runge-Kutta