Mathematical Sciences

In 1981 the Committee on the Undergraduate Program in Mathematics (CUPM) published a major report entitled RECOMMENDATIONS FOR A GENERAL MATHEMATICAL SCIENCES PROGRAM. *This report comprises six chapters that are reprinted here, with minor editing, as the first six chapters of the present volume. Alan Tucker, Chairman of the CUPM Panel that wrote the 1981 report, has written a new Preface to introduce this reprinting.*

1989 Preface

In the eight years since the CUPM Recommendations on a General Mathematical Science Program appeared, issues in mathematics curriculum, such as calculus reform and discrete mathematics, have become hot topics in the mathematics community and have even received extensive coverage in the popular press. The CUPM Panel on a General Mathematical Sciences Program had the luxury of working in comparative anonymity, although ten panel discussions at national and regional mathematics meetings gave the panel some professional visibility. The Panel's basic goal was to give long-term, general objectives for undergraduate training in mathematics.

The 1960's and 1970's had seen a variety of specialized appeals made to college students interested in mathematics. For example, the discipline of computer science emerged as an exciting career for mathematics students. The earliest CUPM recommendations for the mathematics major were aimed at preparing students for doctoral work in mathematics. By the late 1970's, there was a sense that the mathematics major had lost its way, with upper-division enrollments in traditional core courses like analysis and number theory down by 60% from their levels five years earlier and with industrial employers showing little interest in hiring mathematics majors.

To put these recent events in perspective, the Panel obtained a historical briefing from Bill Duren (the founding chairman of CUPM). He recounted over a century of swings of the pendulum between the theoretical and the practical in American collegiate mathematics education, and between training for careers of the future and training in classical, old-fashioned methods.

The Mathematical Sciences Panel sought to find a common ground for the mathematics major which taught abstraction and application, emerging new problem areas and time-tested old ones. The Panel sought to persuade mathematicians that the curriculum in the mathematics major should be shared among the various intellectual and societal constituencies of mathematics. The challenge was to be diverse without being superficial.

The most concrete consequence of the Panel's work was its name, Panel on a General Mathematical Sciences Program. It asked that the mathematics major be renamed the mathematical sciences major—a change explicitly adopted by hundreds of colleges and universities and implicitly adopted by the vast majority of institutions. The Panel recommended that first courses in most subjects should have a good dose of motivating applications, particularly linear algebra and statistics, and that one advanced course should have a mathematical modeling project. This recommendation also seems to have wide acceptance. There were several panel recommendations that reflected trends already occurring but being resisted by some mathematicians: requiring an introductory course in computer science; not requiring linear algebra as a prerequisite for multivariable calculus; encouraging weaker students to delay core abstract courses until the senior year; and not requiring every mathematics major to take courses in real analysis and abstract algebra (i.e., other mathematics courses at comparable levels of abstraction could be substituted).

Although it was unhappy with calculus, the Mathematical Sciences Panel consciously avoided recommending changes in calculus for fear that the inevitable controversy and the complexity of such an undertaking would undermine acceptance of its basic recommendations about the structure of a mathematics major. The Panel touched only lightly on the issue of discrete versus continuous mathematics, recommending exposure to "more combinatorially-oriented mathematics associated with computer and decision sciences" (Tony Ralston's provocative essays about discrete mathematics had not yet appeared).

It was gratifying to the Mathematical Sciences Panel that its report was well-accepted: all two-thousand copies printed have been sold (another two-thousand copies had been sent gratis to department heads). In reviewing the report for this reprinting, the only changes have been to add a few additional references. On the other hand, there was one panel suggestion that has been ignored thus far and which merits consideration.

It concerns the "modest" version of abstract algebra (in Section III) in which time would be spent sensitizing students to recognize how algebraic systems arise naturally in many situations in other areas of mathematics and outside mathematics (to keep algebra alive in their minds after they leave college).

ALAN TUCKER
SUNY at Stony Brook
March, 1989

1981 Preface

This report of the CUPM Panel on a General Mathematical Sciences Program (MSP) presents recommendations for a mathematical sciences major. The panel has concentrated its efforts on general curricular themes and guiding pedagogical principles for a mathematical sciences major. It has tried to frame its recommendations in general terms that will permit a variety of implementations, tailored to the needs of individual institutions. A prime objective of the original 1960's CUPM curriculum recommendations for upper-level mathematics courses was easing the trauma of a student's first year of graduate study in mathematics. This report refocuses the upper-level courses on the traditional objectives of general training in mathematical reasoning and mastery of mathematical tools needed for a life-long series of different jobs and continuing education.

The MSP panel has tried to avoid highly innovative approaches to the mathematics curriculum. The emphasis, instead, has been on using historically rooted principles to organize and unify the mathematical sciences curriculum. The MSP panel believes that the primary goal of a mathematical sciences major should be to develop rigorous mathematical reasoning. The word 'rigorous' is used here in the sense of 'intellectually demanding' and 'in-depth.' Such reasoning is taught through a combination of problem solving and abstract theory. Most topics should initially be developed with a problem-solving approach. When theory is introduced, it usually should be theory for a purpose, theory to simplify, unify, and explain questions of interest to the students.

CUPM now believes that the undergraduate major offered by a mathematics department at most American colleges and universities should be called a Mathematical Sciences major. Enrollment data show that for several years less than half the courses, after calculus, in a typical mathematics major have been in pure mathematics. Furthermore, applied mathematics, probability and statistics, computer science, and operations research are important subjects which should be incorporated in undergraduate training in the general area of mathematics.

Computer science has become such a large, multifaceted field, with ties to engineering and decision sciences, that it no longer can be categorized as a mathematical science (at the National Science Foundation, computer science and mathematical sciences are different research categories). A mathematical sciences major must involve coursework in computer science because of the usefulness of computing and because of computer science's close ties to mathematics. Undergraduate majors in mathematical sciences and in computer science should complement each other.

The new course recommendations presented in this report do not, in most instances, replace past CUPM syllabi. They describe different approaches to courses; for example, a one-semester combined probability and statistics course, or a multivariate calculus course without a linear algebra prerequisite.

The work of the CUPM Panel on a General Mathematical Sciences Program was supported by a grant from the Sloan Foundation. The chairmen of CUPM during this project, Donald Bushaw and William Lucas, deserve special thanks for their assistance.

For information about other CUPM documents and related MAA mathematics education publications, write to: Director of Publications, The Mathematical Association of America, 1529 Eighteenth Street, N.W., Washington, D.C. 20036.

ALAN TUCKER
SUNY at Stony Brook

Panel Background

The CUPM Panel on a General Mathematical Sciences Program (MSP) was constituted in June, 1977 at a CUPM conference in Berkeley. CUPM members decided that a major re-examination of the mathematics major was needed. The CUPM model for the mathematics major contained in the 1965 CUPM reports on Pregraduate Training in Mathematics and a General Curriculum in Mathematics in Colleges (revised in 1972) was felt to be out of date. Following a six-month study, MSP reported to CUPM that the CUPM mathematics major curriculum should be substantially revised and broadened to define a mathematical sciences major. MSP was charged then with developing mathematical sciences recommendations.

Five subpanels were created to develop course recommendations in:

- The calculus sequence,
- Computer science,
- Modeling and operations research,
- Statistics, and
- Upper-level core mathematics.

The MSP project has had the cooperation of curriculum groups in the American Statistical Association, the Association for Computing Machinery, the Operations Research Society of America, and the Society for Industrial and Applied Mathematics. Graduate programs in the subjects covered by those societies draw heavily on undergraduate mathematics students, and except for computer science, undergraduate courses in these subjects are usually taught by mathematicians. Hence these curriculum groups had a major interest in the design of a mathematical sciences major.

The MSP panel coordinated its work with the National Research Council's Panel on Training in Applied Mathematics (chaired by P. Hilton, a member of MSP). The Hilton panel had a much broader mandate than the MSP panel. Its report addresses the unification of the mathematical sciences, the attitudes of mathematicians, academic-industrial linkages, and society's image of the mathematical sciences, as well as curricula. The Hilton report presented a limited number of general curriculum principles with the expectation that the MSP panel would develop fuller curriculum recommendations. The MSP panel recommendations have incorporated these principles (although the Hilton panel's stress on differential equations has been diminished). The MSP panel strongly endorses the Hilton report's emphasis on the importance within mathematics departments of proper attitudes towards the uses and users of mathematics and of a unified view that respects the content and teaching of pure and applied mathematics equally.

While CUPM and the Hilton panel have been recommending changes in the collegiate mathematics program, the National Council of Teachers of Mathematics has been assessing priorities in school mathematics. The 1980 NCTM booklet, *An Agenda for Action,* recommends "that problem solving be the focus of school mathematics in the 1980s ... that basic skills in mathematics be defined to encompass more than computational facility." Recent nation-wide mathematics tests administered to students in several grades showed uniformly poor performance on questions of a problem solving or application nature. Inevitably these mathematical weaknesses will become more of a problem with college students.

The tentative MSP ideas for curriculum revision were discussed by panel members at sectional and national MAA meetings, at the PRIME 80 Conference, and individually with dozens of mathematics department chairpersons. The helpful criticisms received on these occasions played a vital role in shaping the panel's thinking. It should be noted that several people warned that a mathematical sciences major was unworkable because of the diversity of techniques and modes of reasoning in the mathematical sciences today. Others stated that student course preferences had already "redefined" the mathematics major along the lines being proposed by the MSP panel.

Curriculum Background

Many students today start mathematics in college at a lower level and yet have specific (but uninformed) career goals that require a broad scope of new topics of varying mathematical sophistication. Student changes are reflected in recent upper-level enrollment shifts and the explosion of new theory and applications in all the mathematical sciences. Uncertainties in curriculum produced by these developments have led the MSP panel to look for guidance from past CUPM curriculum development experiences and, farther back, from the traditional goals of the mathematics major before CUPM's creation. No matter how great the advances in the past generation, the traditional intellectual objectives of training in mathematics, defined over scores of years, should be the basis of any mathematical sciences program.

Until the 1950s, mathematics departments were primarily service departments, teaching necessary skills to science and engineering students and teaching mathematics to most students solely for its liberal-arts role as a valuable intellectual training of the mind. The average student majoring in mathematics at a better college in the 1930s took courses in trigonometry, analytic geometry, and college algebra (including calculus preparatory work on series and limits) in the freshman year followed by two years of calculus. While this program may today seem to have unnecessarily delayed calculus, and subsequent courses based on calculus, it did provide students with a background that permitted calculus to be taught in a more rigorous (i.e., more demanding) fashion than it is today.

The mathematics major was filled out with five or six electives in subjects such as differential equations (a second course), projective geometry, theory of equations, vector analysis, mathematics of finance, history of mathematics, probability and statistics, complex analysis, and advanced calculus. Most mathematics majors also took a substantial amount of physics. Training of secondary school mathematics teachers rarely included more than a year of calculus. In the early 1950s, twenty years later, the situation had changed only a little; top schools did now offer modern algebra and abstract analysis.

In 1953, amid reports of widespread dissatisfaction with the undergraduate program, the Mathematical Association of America formed the Committee on Undergraduate Program (CUP, later to be renamed CUPM). CUPM concentrated initially on a unified introductory mathematics sequence Universal Mathematics, consisting of a first semester analysis/college algebra course (finishing with some calculus) followed by a semester of "mathematics of sets" (discrete mathematics). CUPM hoped its Universal Mathematics would "halt the pessimistic retreat to remedial mathematics ...(and) ... modernize and upgrade the curriculum."

The first comprehensive curriculum report of CUPM, entitled Pregraduate Training for Research Mathematicians (1963), outlined a model program designed to prepare outstanding undergraduates for Ph.D. studies in mathematics. Emphasis on Ph.D. preparation represented a major departure from the traditional mathematics program and was the source of continuing controversy. A more standard mathematics major curriculum was published in 1965 (revised in 1972), but many colleges also found it to be too ambitious for their students.

For a fuller history of CUPM, see the article of W. Duren (founder of CUPM), "CUPM, The History of an Idea," *Amer. Math. Monthly* 74 (1967), pp. 22-35.

Current Issues

In 1970, 23,000 mathematics majors were graduated. The numbers of Bachelors, Masters, and Doctoral graduates in mathematics had been doubling about every six years since the late 1950s. The 1970 CBMS estimate for the number of Bachelors graduates in mathematics in 1975 was 50,000, but by the late 1970s only 12,000 were graduating annually. Enrollments in many upper-level pure mathematics courses declined even more dramatically in the 1970s as students turned to applied and computer-related courses.

Yet while the number of mathematics majors is decreasing, the demand for broadly-trained mathematics graduates is increasing in government and industry. Mathematical problems inherent in projects to optimize the use of scarce resources and, more generally, to make industry and government operations more efficient guarantee a strong future demand for mathematicians. These problems require people who, foremost, are trained in disciplined logical reasoning and, secondarily, are versed in basic techniques and models of the mathematical sciences. In Warren Weaver's words, these are problems of "organized complexity" as well as well-structured applied mathematics of the physical sciences. If mathematics departments do not train these quantitative problem-solvers, then departments in engineering and decision sciences will.

The unprecedented growth of computer science as a major new college subject parallels the theoretical growth of the discipline and its ever-expanding impact on business and day-to-day living. The number of computer science majors now substantially exceeds the number of mathematics majors at most schools offering programs in both subjects. However, computer science has not "taken" students from mathematics, any more than science and engineering take students from mathematics. Rather, computers have generated the need for more quantitative problem-solvers, as noted above.

The shortage of secondary school mathematics teachers nation wide has become worse than ever before. This shortage appears to be due in large measure to the greater attractiveness of computing careers to college mathematics students (indeed high-paying computer jobs are currently luring many teachers out of the classroom). Although the training of future teachers should include course work in computing and applications, such course work heightens the probability that these students will switch to careers in computing.

On another front, pre-calculus enrollments have soared as the mathematical skills of incoming freshmen have been declining (a problem that concerned CUP in its first year). The mathematics curriculum may soon need to allow for majors who do not begin calculus until their sophomore year, as was common a generation ago.

At universities, the decline in graduate enrollments has frequently over-shadowed the decline in undergraduate majors. Faced with heavy precalculus workloads, shrinking graduate programs, and competition from other mathematical sciences departments, university mathematics departments appear less able to broaden and restructure the mathematics major than most liberal-arts college mathematics departments. Many university mathematicians prefer to retain their current pure mathematics major for a small number of talented students.

There are also several encouraging developments. A natural evolution in the mathematics major is occurring at many schools. Students and faculty have developed an informal "contract" for a major that includes traditional core courses in algebra and analysis along with electives weighted in computing and applied mathematics (a formal "contract" major at one school is discussed below).

Another important development is the emphasis on systems design, as opposed to mathematical computation, in current computer science curricula. The Association for Computing Machinery Curriculum 78 Report delegates the responsibility for teaching numerical analysis, discrete structures, and computational modeling to mathematics departments. This ACM curriculum implicitly encourages students interested in computer-based mathematical problem solving to be mathematical sciences majors. The MSP panel has been careful to coordinate its work with computer science curriculum groups in order to minimize potential conflicts and maximize compatibility between computer science and mathematical sciences programs.

Curricular Principles

The goal of this panel was to produce a flexible set of recommendations for a mathematical sciences major, a major with a broad, historically rooted foundation for dealing with current and future changes in the mathematical sciences. The panel sought a unifying philosophy for diverse course work in analysis, algebra, computer science, applied mathematics, statistics, and operations research.

Program Philosophy

I. The curriculum should have a primary goal of developing attitudes of mind and analytical skills required for efficient use and understanding of mathematics. The development of rigorous mathematical reasoning and abstraction from the particular to the general are two themes that should unify the curriculum.

II. The mathematical sciences curriculum should be designed around the abilities and academic needs of the average mathematical sciences student (with supplementary work to attract and challenge talented students).

III. A mathematical sciences program should use interactive classroom teaching to involve students actively in the development of new material. Whenever possible, the teacher should guide students to discover new mathematics for themselves rather than present students with concisely sculptured theories.

IV. Applications should be used to illustrate and motivate material in abstract and applied courses. The development of most topics should involve an interplay of applications, mathematical problem-solving, and theory. Theory should be seen as useful and enlightening for all mathematical sciences.

V. First courses in a subject should be designed to appeal to as broad an audience as is academically reasonable. Many mathematics majors do not enter college planning to be mathematics majors, but rather are attracted by beginning mathematics courses. Broad introductory courses are important for a mathematical sciences minor.

Course Work

VI. The first two years of the curriculum should be broadened to cover more than the traditional four semesters of calculus-linear algebra-differential equations. Calculus courses should include more numerical methods and non-physical-sciences applications. Also, other mathematical sciences courses, such as computer science and applied probability and statistics, should be an integral part of the first two years of study.

VII. All mathematical sciences students should take a sequence of two upper-division courses leading to the study of some subject(s) in depth. Rigorous, proof-like arguments are used throughout the mathematical sciences, and so all students should have some proof-oriented course work. Real analysis or algebra are natural choices but need not be the only possibilities. Proofs and abstraction can equally well be developed through other courses such as applied algebra, differential equations, probability, or combinatorics.

VIII. Every mathematical sciences student should have some course work in the less theoretically structured, more combinatorially oriented mathematics associated with computer and decision sciences.

IX. Students should have an opportunity to undertake "real-world" mathematical modeling projects, either as term projects in an operations research or modeling course, as independent study, or as an internship in industry.

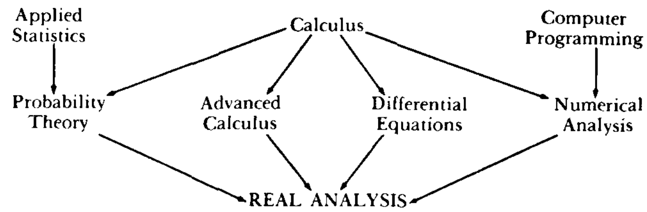
X. Students should have a minor in a discipline using mathematics, such as physics, computer science, or economics. In addition, there should be sensible breadth in physical and social sciences. For example, a student interested in statistics might minor in psychology but also take beginning courses in, say, economics or engineering (heavy users of statistics).

Building Mathematical Maturity

As noted in Principle I, a major in mathematical sciences should emphasize general mathematical reasoning as much as mastery of various subject matter. Implicit in this principle is that less material would be covered in many courses but that students would be expected to demonstrate a better understanding of what is taught, e.g., by solving problems that require careful mathematical analysis.

This mathematical sciences curriculum would model the historical evolution of mathematical subjects: some problems are introduced, formulas and techniques are developed for solving problems (usually with heuristic explanations), then common aspects of the problems are examined and abstracted with the purpose of better understanding "what is really going on." The difference in this scheme between beginning calculus and upper-division probability theory would be primarily a matter of the difficulty of the problems and techniques and the speed with which the material is covered and generalized, i.e., a matter of the mathematical maturity of the audience. In the course of two or three years of such course work, there would be a steady increase in sophistication of the material and more importantly, an increase in the student's ability to learn and organize the ideas of a new mathematical subject. Students should be able to read and learn mathematics on their own from texts. The MSP panel feels that such maturity is a function of how a subject is learned as much as what is learned.

All courses should have some proofs in class and, as the maturity of students increases, occasional proofs as homework exercises. In particular, students should acquire facility with induction arguments, a basic method of proof in the mathematical sciences. After reviewing performances of current students and programs of mathematics students 30 years ago, the MSP panel has concluded that many able students do not now have, nor were they previously expected to have, the mathematical maturity to take theoretical courses before their senior year. On the other hand, by the senior year, all students should be ready for some proof-oriented courses that show the power of mathematical abstraction in analyzing concepts that underlie a variety of concrete problems. For example, part of a flowchart of courses leading to a senior-year real analysis course might be:



Core Requirements

The panel has found the question of whether to require courses in algebra and analysis its most controversial problem. In light of the strongly differing opinions received on this subject, the MSP panel is making only a minimal recommendation (Principle VII) that it feels is reasonable for all students. Possible two course sequences besides a year of analysis or of algebra are: analysis and proof-oriented probability theory, analysis and differential equations, abstract algebra and (proof-oriented) combinatorics, applied algebra and theory of computation, or analysis and a topics-in-analysis seminar. While not a sequence, one course in analysis and one course in algebra also fulfill the spirit of this requirement. Some departments will want to make stronger requirements. The issue of theory requirements is discussed more fully below.

Students should not be required to study a subject with an approach whose rationale depends on material in later courses nor should they be required to memorize (blindly) proofs or formulas. Some upper-level elective courses should always be taught as mathematics-for-its-own-sake, but an instructor should be very careful not to skip the historical motivation and application of a subject in order to delve further into its modern theory.

The recommendation for interactive teaching (Principle III) seeks to encourage student participation in developing new mathematical ideas. It constrains an instructor to teach at a level that students can reasonably follow. Interactive teaching implicitly says that mathematics is learned by actively doing mathematics, not by passively studying lecture notes and mimicking methods in a book. Without needlessly slowing progress in class, an instructor should discuss how one can learn much from wrong approaches suggested by students. New mathematical theories are not divined with textbook-like compact proofs but rather involve a long train of trial-and-error creativity.

Henry Pollak expressed this need in the Conference Board of Mathematical Sciences book, *The Role of Ax-*

iomatics and Problem Solving in Mathematics (Ginn, 1966):

> A carefully organized course in mathematics is sometimes too much like a hiking trip in the mountains that never leaves the well-constructed trails. The tour manages to visit a steady sequence of the high spots in the natural scenery. It carefully avoids all false starts, dead ends and impossible barriers and arrives by five o'clock every afternoon at a well-stocked cabin. ...However, you miss the excitement of occasionally camping out or helping to find a trail and of making your way cross-country with only a good intuition and a compass as a guide. "Cross country" mathematics is a necessary ingredient of a good education.

Further details about the course work recommendations in Principles VI, VIII, and IX appear in later chapters of this report. Discussion of courses in discrete methods, applied algebra, and numerical analysis appears in the last section of this chapter.

Teaching Mathematical Reasoning

Because a mathematical sciences major must include a broader range of courses than a standard (pure) mathematics major, many mathematicians have expressed concern that it will be harder to teach the average mathematics student rigorous mathematical reasoning in a mathematical sciences major. They believe that the major will develop problem-solving skills but that without more abstract pure mathematics, students will never develop a true sense of rigorous mathematical reasoning. The MSP panel thinks that a mathematical sciences major with a strong emphasis on problem-solving is in keeping with time-tested ways of developing "mathematical reasoning." The question of whether to require "core" pure mathematics courses, such as abstract algebra and real analysis, in any mathematical sciences major is discussed in the next section.

Historically (before 1940), the main thrust of the mathematics major at most colleges was problem-solving. Most courses in the major could be classed as mathematics for the physical sciences: trigonometry, analytic geometry, calculus (first-year and advanced), differential equations, and vector analysis. Proofs in advanced calculus were symbolic computations. Proofs in number theory were, and still are, usually combinatorial problems. The one abstract "pure" course in the curriculum was logic. A "rigorous" course did not mean an abstract course, "mathematics done right." A rigorous course used to mean a demanding, more in-depth treatment that required more skill and ingenuity from the student. The past curriculum surely had some faults, but its problem-solving and close ties to physics came from traditions that go back to the roots of mathematics.

While problem solving may traditionally be the primary way of teaching mathematical reasoning to undergraduates, the complexity and breadth of modern mathematics and mathematical sciences require theory to help organize and simplify learning. Rigorous problem solving should lead students to appreciate theory and formal proofs. In a mathematical sciences major, theory should be primarily theory for a purpose, theory born from necessity (of course, this is also the historical motivation of most theory). Students may find theory difficult, but they should never find it irrelevant.

Most courses in a mathematical sciences major should be case studies in the pedagogical paradigm of real world questions leading to mathematical problem solving of increasing difficulty that forces some abstraction and theory. As mentioned earlier, lower-level courses would concentrate on problem solving to build technical skills with occasional statements of needed theorems, while typical upper-level courses would concentrate on problem solving to build technical skills with occasional statements of needed theorems, while typical upper-level courses would emphasize the transition from harder problem-solving to theory.

Instructors should resist pressures to survey fully fields such as numerical analysis, probability, statistics, combinatorics, or operations research in the one course a department may offer in the field. The instructor of such a course should give students a sense of the problems and modes of reasoning in the field, but after that, should be guided by the pedagogical model given above. All syllabi produced by MSP subpanels should be viewed in this light. Most instructors will cover most of a suggested syllabus, but general pedagogical goals should always take precedence over the demands of individual course syllabi.

The MSP panel believes that for generations mathematics instructors have used the paradigm mentioned above to develop rigorous mathematical reasoning. Implicit in this paradigm is a unity of purpose between students and instructor. Most students like to start with concrete real-world examples as a basis for mathematical problem solving. They expect the problems to get harder and require more skill and insight. And they certainly appreciate theory when it makes their work easier (although understanding formal proofs of useful theory requires maturity). Interactive teaching also becomes natural: students are interested in participating in a class that is developing a subject in a way that they can appreciate.

How Much Theory?

This section summarizes arguments for and against requiring upper-level analysis and algebra courses of all mathematical sciences majors, and why the MSP panel made its "compromise" decision.

Expecting controversy on several issues, the MSP panel organized sessions at national and regional MAA meetings to get input from the mathematics community. The main area of contention was how many courses to require in specific areas. The panel heard complaints that some areas were being neglected or that only one course in a certain area would be so superficial as to be worse than no course. However, most constituencies came to accept the need for compromise recommendations of limited exposure to several areas with students left to choose for themselves an area to study in greater depth. On the other hand, one important issue emerged on which a compromise position seemed to antagonize at least as many people as it pleased. This was the question of whether to require an analysis and/or an abstract algebra course and, more generally, how much proof-oriented course work should be required in a mathematical sciences major.

In the early 1970's, a majority of mathematics programs required at least these two upper-level "core mathematics" courses for all students. Recently, declining enrollments in these courses and student preference for more applied or computing courses have forced many departments either to relax this requirement or to introduce a new applied track which does not require these two courses. People favoring the requirement of analysis and algebra argue that:

- Not requiring them would speed an already dangerous deterioration in the intellectual basis of the mathematics major;
- A major without at least analysis and algebra would be a superficial potpourri of courses—a major of no real value to anyone, e.g., graduate study in statistics requires analysis and (linear) algebra;
- One cannot understand "what mathematics is about" without these two courses—a major without these two courses simply should not be offered by a mathematics department.

People in favor of not requiring analysis and algebra argue that:

- With a more applied emphasis the mathematical sciences major will attract more good students, whereas requiring these courses would mean no change (except for new applied electives) from the 1960s type of mathematics major that today attracts only a marginal number of students;
- Analysis and algebra are fine for some students but demand a mathematical maturity that many other undergraduates lack—these students memorize proofs blindly to pass examinations and never take the follow-on courses needed to appreciate the structure and elegance of these subjects; and
- Proofs and abstraction can equally well be developed through other courses such as applied algebra, probability, differential equations, or combinatorics.

Mathematicians must face the reality of a general change in the attitude of college students towards mathematics. The popularity of science and mathematics in the 1960s drew more of the brightest students to mathematics and also motivated all students to work harder at mathematics in high school. So the average mathematics student was capable of handling a more theoretical mathematics program.

Today, mathematics appears to be getting no more than its traditional (smaller) share of bright students, and high school study habits are less good. However, almost all of today's mathematics students still find a few subjects, pure or applied, particularly interesting and want to study this material in some depth. Also by the senior year, the MSP panel believes that mathematics majors do have the mathematical maturity to appreciate, say, a moderately abstract real analysis course. Examples of new approaches to teaching analysis and other core mathematics courses appear in subsequent chapters.

Since there was agreement on the importance of some theoretical depth, the MSP panel proposed the compromise of Principle VII, recommending "a sequence of two upper-division courses leading to the study of some subject in depth." Because of the lack of consensus on the analysis-algebra question, the MSP panel expects this issue to be debated and modified at individual institutions. The faculty should not require courses that most students strongly dislike, nor should faculty shy away from any theory requirements for fear of losing majors. The faculty rather must motivate students to appreciate the value of some theoretical course work.

Sample Majors

This section presents two 12 semester-course mathematical sciences majors. Many other sample majors could be given. The MSP panel believes that most majors should be a "convex combination" of the two majors given here. Major A contains much of a standard mathematics major, while Major B is a broader program designed for students interested in problem solving. Both majors should be accompanied by a minor in a related subject.

The common core of all majors would be three semesters of calculus, one course in linear algebra, one course in computer science plus either a second computer course or extensive use of computing in several other courses, one course in probability and statistics, the equivalent of a course in discrete methods, modeling experience, and two theoretical courses of continuing depth.

Mathematical Sciences Major A

- Three semesters of calculus
- Linear algebra
- Probability and statistics
- Discrete methods
- Differential equations (with computing)
- Abstract algebra (one-half linear algebra)
- Two semesters of advanced calculus/real analysis
- One course from the following set: abstract algebra (second course), applied algebra, geometry, topology, complex analysis, mathematical methods in physics
- Mathematical modeling
- Plus related course work: two semesters of computer science and two semesters of physics, to be taken in the first two years.

Mathematical Sciences Major B

- Three semesters of calculus
- Linear algebra
- Introduction to computer science
- Numerical analysis *or* second course in computer science
- Probability and statistics
- Advanced calculus *or* abstract algebra
- Discrete methods *or* differential equations
- Mathematical modeling or operations research
- Two electives continuing a subject with theoretical depth.

Subsequent sections in this report contain recommendations for discrete methods, applied algebra, and numerical analysis courses; for calculus, linear algebra, and differential equations courses; for upper-level core mathematics; for computer science; for modeling and operations research; and for probability and statistics.

Major A is meant to be close to the spirit of the major suggested by the NRC Panel on Training in Applied Mathematics. That panel viewed differential equations as a unifying theme in the major. The proper mixture of Majors A and B (with appropriate electives) would also allow students to make statistics or operations research a unifying theme.

The MSP panel feels that a set of courses similar to either of the above two majors, or a mixture thereof, would be reasonable for most mathematical sciences students. Some departments could offer several tracks for the mathematical sciences major. Special areas of faculty strength or student interest should obviously be reflected in the curriculum.

Computing assignments should be used in most courses. When a liberal arts college mathematics department teaches computer science, such computing course work must frequently be counted within the college limit of 12 or 13 courses permitted in one department. This regulation is assumed in Major B. However, the MSP panel believes that counting computer courses this way unfairly restricts a mathematical sciences major. One alternative is to list computer courses through the Computing Center.

The one fundamental new course in these sample majors is discrete methods. As mentioned in Principle VIII, the MSP panel feels that the central role of combinatorial reasoning in computer and decision sciences requires that some combinatorial problem solving should be taught in light of the three semesters devoted to analysis-related problem solving in the calculus sequence. To this end, the modeling course should be heavily combinatorial if students have not taken a formal discrete methods course.

Major A would be good preparation for graduate study in mathematics, applied mathematics, statistics, or operations research as well as many industrial positions as a mathematical analyst or programmer. Major B would be good preparation for most industrial positions and for graduate study in applied mathematics, statistics, or operations research (for such graduate study, both advanced calculus and upper-level linear algebra are usually needed). Representatives from many good mathematics graduate programs have stated that they would accept strong students with Major B-type training.

Many computer science graduate programs would accept Major B if the two electives were in computer science (although some other undergraduate computer science course deficiencies may still have to be made up in the first year of graduate study). In a computer science concentration within a mathematical sciences major, modern algebra might be replaced by applied algebra (see below for more details). Major B with an elective in the theory of interest and a second probability-statistics course would be excellent preparation for actuarial careers. Students interested in physical sciences-related applied mathematics could modify either sample major to get a good program. Both majors provide preparation for secondary school mathematics teaching, when supplemented with teaching methodology and practicum courses (theory courses must include algebra and geometry).

Many smaller schools are being forced to offer a program in the spirit of Major B because almost all of B's courses have the needed enrollment base of students drawn from outside mathematics.

The courses involving numerical analysis, probability and statistics, discrete methods, and modeling all can be designed as lower-level or upper-level courses. A large amount of flexibility is possible in "repackaging" the mathematical sciences material. For example, a Computational Models course (see the 1971 CUPM *Report on Computational Mathematics*) could cover some numerical analysis along with a little applied probability and statistics to be used in simulation modeling. A quarter system institution would have even greater flexibility in implementing this major.

Mathematical Sciences Minor

Just as a mathematical sciences major should be accompanied by a minor in a related subject, so also do many other disciplines encourage their students to have a minor, or double major, in mathematics. At some colleges, as many as half the mathematics majors have another major. Unfortunately, while mathematical methods are playing an increasingly critical role in social and biological sciences and in business administration, students are generally ignorant or misinformed in high school and early college years about the importance of mathematics in these areas.

The result is that many students either do not realize the value of further course work in the mathematical sciences until their junior or senior year, or their poor high school preparation forces them to take a year of remedial mathematics before they can begin to learn any of the college mathematics they need. For such students, a traditional six to eight course minor in mathematics, starting with (at least) three semesters of calculus, is not feasible. When students in the social and biological sciences come to realize the value of mathematics in the junior year, they have frequently had only one semester of calculus, or perhaps a year of calculus with probability.

The MSP panel believes that these students would be well served by a six to eight course mathematical sciences minor consisting of two semesters of calculus, one semester of (calculus-based) probability and statistics, one semester of introductory computer science, plus two to four electives chosen from courses such as numerical analysis, discrete methods, linear algebra, differential equations, linear programming, mathematical modeling, and additional courses in calculus, probability or statistics, and computer science. Such a minor could easily be completed in three semesters. It has little prerequisite structure so that students can immediately pick courses based on personal interests rather than initially "mark time" waiting to complete the calculus sequence.

Such a minor has several important points in its favor. First of all, this minor is a collection of useful mathematical sciences courses which present concepts and techniques that arise frequently in the social and biological sciences. While this minor lacks the mathematical depth of the traditional type of mathematics minor, it nonetheless introduces students to important modes of mathematical reasoning. Second, such a minor will be attractive to students because it enhances employment opportunities and prospects for admission to graduate or professional schools. Third, after the exposure to interesting mathematical sciences topics, some students will want to study these subjects further in graduate school, either in a mathematical sciences graduate program or as electives in other graduate programs. Fourth, this minor will bring more students into mathematical sciences courses, making it possible to offer these courses more frequently. Conversely, offering more mathematical sciences courses each semester will make a mathematical sciences minor, as well as the regular mathematical sciences major, more attractive to students. In addition, when more students are taking mathematical sciences courses and finding out how useful mathematics is, the campus-wide student awareness of the value of mathematics will increase.

Examples of Successful Programs

Proper curriculum is the heart of a mathematical sciences program, but there are many non-academic aspects that also must be considered. A wide variety of course offerings is not as important as the spirit with which the general program is offered. This section discusses salient features of some successful mathematics programs. "Successful" means attracting a large number of students into a program that develops rigorous mathematical thinking and also offers a spectrum of (well taught) courses in pure and applied mathematics. Successful programs typically produce 5% to 8% of their college's graduates, although nation wide, mathematics majors constitute only about 1% of college graduates. Faculty and student morale is uniformly high in these programs. As one would expect, teaching and related student-oriented activities consume most of the faculty's time in such successful programs, and there is little faculty research. The professors' pride in good teaching and in the successes of their students leaves them with few regrets about not publishing. The set of programs mentioned here is only a sampling of successful programs that have come to the attention of this CUPM panel. More detailed information about these mathematics programs is available from individual colleges.

Saint Olaf College, a 2800-student liberal arts college in Northfield, Minnesota, has a contract mathematics major. Each mathematics student presents a proposed contract to the Mathematics Department. The contract consists of at least nine courses (college regulations limit the maximum number of courses that can be taken in one department to 14). The department normally will not accept a contract without at least one upper-level applied and one upper-level pure mathematics course, a computing course or evidence of computing skills, and some sort of independent study (research program, problem-solving proseminar, colloquium participation, or work-study).

Frequently a student and an advisor will negotiate a proposed contract. For example, a faculty member will try to persuade a student interested in scientific computing and statistics that some real analysis and upper-level linear algebra should be included in the contract by showing that this material is needed for graduate study in applied areas, and in any case a liberal arts education entails a more broadly based mathematics major. Conversely, a student proposing a pure mathematics contract would be confronted with arguments about not being able to appreciate theory without knowledge of its uses. In the end, the student and the faculty member understand and respect each other's point of view.

This understanding of each other's interests naturally carries into the classroom. Also, the contract negotiations "break the ice" and make students more at ease in talking to faculty (and encourage constructive criticism). The Mathematics Department offers minors in computing and statistics, but the attractiveness of a contract major in mathematics leads most students interested in these areas eventually to become mathematics majors.

Lebanon Valley College, a small (1000-student) liberal arts college in Pennsylvania, has only five mathematics faculty but its Department of Mathematical Sciences offers majors in Mathematics, Actuarial Science, Computer Science, and Operations Research. The course work in the mathematics graduate preparation track involves a problem seminar, Putnam team sessions, and formal and informal topics courses (because of the limited demand in this area). All mathematical sciences majors must take a rigorous 25 semester-hour core of calculus, differential equations, linear algebra, foundations, and computer science. Most courses are peppered with applications and computing assignments.

The mathematics faculty are heavily involved in recruiting students by attending College Fairs and College Nights and by visiting regional high schools to explain to students and counselors the many diverse and attractive careers in the mathematical sciences, and the importance of mathematics in other professions. As a result of this effort, 10% of the incoming Lebanon Valley freshmen plan majors in the mathematical sciences (the national average is 1%), and 7% of Lebanon Valley graduates are mathematical sciences majors. Many students are initially attracted by the major in actuarial science (an historically established profession) and then move into other areas of applied and pure mathematics, but this pattern may change with the newly established computer science major.

Once the faculty have the "students' attention," they work the students hard. The students respond positively to the demands of the faculty for three reasons. First, known rewards await those who do well in mathematics (besides the obvious long-term rewards, the department awards outstanding students with membership in various professional societies in the mathematical sciences). Second, a personal sense of intellectual achievement is carefully nurtured starting in the freshman year with honors calculus for mathematics majors. Finally, as at St. Olaf, a continuing dialogue between students and faculty allows students to help shape the mathematics program. In fact, students interview candidates for faculty positions and their recommendations carry great weight. The department keeps in close touch with alumni by sending each one a personal letter every other year with news about the department and fellow alumni.

Nearby Gettysburg College has a special vitality in its mathematics program that comes from an interdisciplinary emphasis. The department has held joint departmental faculty meetings with each natural and social science department at Gettysburg to discuss common curriculum and research interests. Several interdisciplinary team-taught courses have been developed, such as a course on symmetry taught jointly by a mathematician and a chemist. An interdepartmental group organized two recent summer workshops in statistics which drew faculty from eight departments. Mathematics faculty have audited a variety of basic and advanced courses in related sciences to learn to talk the language of mathematics users. Mathematics faculty bring this interdisciplinary point of view into every course they teach, giving interesting applications and showing, say, how a physicist would approach a certain problem. Needless to say, a large number of mathematics majors at Gettysburg are double majors.

Frequently a separate computer science department with its own major spells disaster for the mathematics major at a college. But Potsdam State College (in the economically depressed northeast corner of New York) has possibly the greatest percentage of mathematics graduates of any public institution in the country—close to 10%—despite competition from a popular computer science major. The most striking feature to a visitor to the Potsdam State Mathematics Department is the great enthusiasm among the students and the sense of pride students have in their ability to think mathematically. (While it is hard to measure objectively these students' mathematical development, leading technological companies, such as Bell Labs, IBM, and General Dynamics, annually hire several dozen Potsdam mathematics graduates.)

Classes have a limited amount of formal lectures. Most time is spent discussing work of the students. The emphasis on giving students a sense of achievement is due in large part to experiences of the Potsdam chairman when he taught in a Black southern institution. By instilling self confidence, he had helped able but ill-prepared students excel in calculus and even saw some go on to good mathematics graduate programs. The department has various awards for top students, a very active Pi Mu Epsilon chapter, publications about careers in mathematics and successes of former students, and a large student-alumni newsletter. Upper-class mathematics students are used to tutor (and encourage) beginning students. They also communicate their enthusiasm about mathematics to friends and teachers back home. As a result, half the incoming Potsdam freshmen sign up for calculus (although few departments require it).

The computer science major at Potsdam State is viewed by the mathematics faculty as a great asset to the Mathematics Department. The computer science major helps attract good students to Potsdam who often decide to switch to, or double major with, mathematics. Also the computer science program offers career skills and needed mathematical breadth. Numerical analysis, operations research, and modeling are taught in computer science (the Mathematics Department has had to limit severely their upper-level electives in order to keep class size down and preserve small group seminars).

As noted at the start of this section, the preceding mathematics programs represent only a small sampling of the excellent programs in this country. Several women's colleges offer fine programs worth noting. For example, the Goucher College Mathematics Department has integrated computing in almost all courses and has a broad curriculum in pure and applied mathematics; and the Mills College Mathematics Department has successfully promoted the critical role of mathematics for careers in science and engineering. The cornerstone of Ohio Wesleyan's excellent mathematics program is an innovative calculus sequence (with computing, probability, and diverse mathematical modeling). Georgia State University, an urban public institution with a highly vocational orientation, has a Mathematics Department that has broken out of the typical low-level service function mode to offer a fine, well-populated mathematical sciences major. While research and graduate programs often dominate concerns about the undergraduate mathematics major at universities, mathematics faculty at many universities work closely with undergraduate majors in excellent unified mathematical sciences programs. Three such institutions are Clemson University, Lamar University (Texas), and Rensselaer Polytechnical Institute.

Most universities today have separate departments in computing and mathematical sciences. To counter this division, the University of Iowa and Oregon State University have developed unified inter-departmental mathematical sciences majors. The MSP panel strongly endorses such inter-departmental majors. At some universities, most of the mathematical sciences, outside of pure mathematics, have been housed in one department. Although the MSP panel prefers a unified mathematical sciences major (ideally in one department), several of these non-pure mathematical sciences departments have good undergraduate programs that may be of interest to other institutions: the Mathematical Sciences Department at Johns Hopkins University, the Mathematical Sciences Department at Rice University, and the Department of Applied Mathematics and Statistics at the State University of New York at Stony Brook.

Departmental Self-Study and Publicity

The MSP panel urges all mathematics departments to engage in serious self-study to identify one or more major themes to emphasize in their mathematical sciences programs: an interdisciplinary focus in cooperation with other departments; an innovative calculus sequence (integrating computing, applications, etc.); a work-study program or other individualized learning experience; special strength in one area of the mathematical sciences (pure or applied); or a track directed towards employment in a regional industry (such as aerospace, automative, insurance). Some colleges have successfully developed a multi-option major, but usually such programs are the outgrowth of successful one-theme programs that slowly added new options (for example, the multiple-major mathematical sciences program at Lebanon Valley College, mentioned in the preceding section, started with just an Actuarial Science option). The MSP panel's advice is first to do one thing well.

A departmental emphasis should be consistent with the general educational purposes of the whole institution and the academic interests of the high school graduates who have historically gone to that institution. It is very risky to design a mathematical sciences program about a theme that the mathematics faculty find attractive and then to try to recruit a new group of high school students to come to the institution for this program. Note that a thematic emphasis does not mean that basic parts of the mathematical sciences program discussed earlier in this chapter can be neglected.

Following a departmental self-study and implementation of its recommendations for new courses or development of industrial work-study contacts, etc., it is next necessary to publicize the mathematics department's program with brochures and visits to regional high schools and College Fairs. Virtually all mathematics departments with large programs (where mathematical sciences majors constitute over 4% of the school's graduates) have extensive publicity programs. Such publicity should emphasize the general usefulness of mathematics in the modern world, whether a student is a prospective mathematical sciences major or minor or an undecided liberal arts student.

High school guidance counselors often do not realize that there are other attractive mathematics-related careers outside straight computing. Counselors tend to be afraid of mathematics because of their own personal difficulties with the subject. Some counselors have been known to discourage students from taking more than the minimum required amount of high school mathematics with the warning that students risk getting poor grades in (hard) mathematics courses and thus hurting their chances of college admission.

College faculty trying to publicize the value of mathematics and its study at their institution should seek the cooperation of local associations of the National Council of Teachers of Mathematics, which have long been working to promote interest in mathematics in the high schools.

New Course Descriptions

Finite structures are used throughout the mathematical sciences today. Two new basic courses about finite structures belong in the mathematical sciences curriculum, one addressing combinatorial aspects and one addressing algebraic aspects. Another topic, numerical analysis, has become more important with the growth of computer science. This section describes a numerical analysis course that is more applied and at a lower level than the previous CUPM numerical analysis recommendations (Course 8 in the CUPM report *A General Curriculum for Mathematics in Colleges.*)

Discrete Methods Course

This course introduces the basic techniques and modes of reasoning of combinatorial problem solving in the same spirit that calculus introduces continuous problem solving. The growing importance of computer science and mathematical sciences such as operations research that depend heavily on combinatorial methods justifies at least one semester of combinatorial problem solving to balance calculus' three semesters of analysis problem solving.

Unlike calculus, combinatorics is not largely reducible to a limited set of formulas and operations. Combinatorial problems are solved primarily through a careful logical analysis of possibilities. Simple ad hoc models, often unique to each different problem, are needed to count or analyze the possible outcomes. This need to constantly invent original solutions, different from class examples, is what makes the discrete methods course so valuable for students.

Like calculus, combinatorics is a subject which has a wide variety of applications. Many of them are related to computers and to operations research, but others relate to such diverse fields as genetics, organic chemistry, electrical engineering, political science, transportation, and health science. The basic discrete methods course should contain a variety of applications and use them both to motivate topics and to illustrate techniques.

The course has an enumeration part and a graph theory part. These parts can be covered in either order. While texts traditionally do enumeration first, the graph material is more intuitive and hence it seems natural to do graph theory first (as suggested below).

With the right point-of-view, many combinatorial problems have quite simple solutions. However, the object of this course is not to show students simple answers. It is to teach students how to discover such simple answers (as well as not so simple answers). The means for achieving solutions are of more concern than the ends. Learning how to solve problems requires an interactive teaching style. It requires extensive discussion of the logical faults in wrong analyses as much as presenting correct analyses.

Since the course should emphasize general combinatorial reasoning rather than techniques, a large degree of flexibility is possible in the choice of topics. The course outline given below contains many optional topics. Some of the core topics, such as the inclusion-exclusion formula, might also be skipped to allow the course to be tailored to the interests of students.

COURSE OUTLINE

I. Graph Theory
- A. *Graphs as models.* Stress many applications.
- B. *Basic properties of graphs and digraphs.* Chains, paths, and connectedness; isomorphism; planarity.
- C. *Trees.* Basic properties; applications in searching; breadth-first and depth-first search; spanning trees and simple algorithms using spanning trees. Optional: branch and bound methods; tree-based analysis of sorting procedures.
- D. *Graph coloring.* Chromatic number; coloring applications; map coloring. Optional: related graphical parameters such as independent numbers.
- E. *Eulerian and Hamiltonian circuits.* Euler circuit theorem and extensions; existence and non-existence of Hamiltonian circuits; applications to scheduling, coding, and genetics.
- F. *Optional topics:*
  - a. Tournaments
  - b. Network flows and matching
  - c. Intersection graphs
  - d. Connectivity
  - e. Coverings
  - f. Graph-based games

II. Combinatorics
- A. *Motivating problems and applications.*
- B. *Elementary counting principles.* Tree diagrams; sum and product role; solving problems that must be decomposed into several subcases. Optional: applications to complexity of computation, coding, genetic codes.
- C. *Permutations and combinations.* Definitions and simple counting; sets and subsets; binomial coefficients; Pascal's triangle; multinomial coefficients; elementary probability notions and applications of counting. Optional: algorithms for enumerating arrangements and combinations; binomial identities; combinations with repetition and distributions; constrained repetition; equivalence of distribution problems, graph applications.
- D. *Inclusion/exclusion principle.* Modeling with inclusion/exclusion; derangements; graph coloring. Optional: rook polynomials.
- E. *Recurrence relations.* Recurrence relation models; solution of homogeneous linear recurrence relations; Fibonacci numbers and their applications.
- F. *Optional topics:*
  - a. Generating functions
  - b. Polya's enumeration formula
  - c. Experimental design
  - d. Coding

The preceding course outline is for either a one-semester or a two-quarter course. A two-quarter course has a natural structure, covering enumerative material in one quarter and graph theory plus designs in another quarter. There are several books available for part or all of the discrete methods course. It is anticipated that as this discrete methods course becomes more widely taught, many more books will become available and the exact nature of the syllabus will evolve.

There are several obvious places where a computer can be used in this course: ways of representing graphs in a computer and performing simple tests (e.g., connectivity); asymptotic calculations in enumeration problems; network flow algorithm; and algorithms for enumerating permutations and combinations. The pedagogical problem is that computer programming takes time away from problem-solving exercises, possibly too much time if a school's computer operation runs in a batch processing mode.

A more advanced second course in combinatorics may also be considered. This course can treat core topics in the discrete methods course in greater depth, and some of the optional topics. Other important topics are Ramsey theory, matroids, and graph algorithms. The course could concentrate on combinatorics or on graph theory, or could be a topics course which varies from year to year. Some of the texts listed below would be suitable for this second combinatorics course.

COMBINATORICS & GRAPH THEORY TEXTS

1. Bogart, Kenneth, *Introductory Combinatorics,* Pitman, Boston, 1983.
2. Brualdi, Richard, *Introductory Combinatorics,* Elsevier-North Holland, New York, 1977.
3. Cohen, Daniel, *Basic Techniques of Combinatorial Theory,* J. Wiley & Sons, New York, 1978.
4. Liu, C.L., *Introduction to Combinatorial Mathematics,* McGraw Hill, New York, 1968.
5. Roberts, Fred, *Applied Combinatorics,* Prentice-Hall, Englewood Cliffs, New Jers., 1984.
6. Tucker, Alan, *Applied Combinatorics,* J. Wiley & Sons, New York, 1980.

GRAPH THEORY TEXTS

1. Bondy, J. and Murty, V.S.R., *Graph Theory with Applications,* American Elsevier, New York, 1976.
2. Chartrand, Gary, *Graphs as Mathematical Models,* Prindle, Weber, and Schmidt, Boston, 1977.
3. Ore, Oystein, *Graphs and Their Uses,* Math. Assoc. of America, Washington, D.C., 1963.
4. Roberts, Fred, *Discrete Mathematical Models,* Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
5. Trudeau, Robert, *Dots and Lines,* Kent State Press, Kent, Ohio, 1976.

COMBINATORICS TEXTS

1. Berman, Gerald and Fryer, Kenneth, *Introduction to Combinatorics,* Academic Press, New York, 1969.
2. Eisen, Martin, *Elementary Combinatorial Analysis,* Gordon-Breach, New York, 1969.
3. Vilenkin, N., *Combinatorics,* Academic Press, New York, 1971.
4. Street, A. and Wallis, W., *Combinatorial Theory: An Introduction,* Charles Babbage, 1975.

Applied Algebra Course

(*Editorial Note in 1989 reprinting*: This course is now called **Discrete Structures** and is usually now taught at the freshman level. The course discussed here is more advanced and intended for the sophomore-junior level.)

A traditional time for an applied algebra course is in the junior year—when students would be ready for a modern algebra course. However, as noted above, many students will not be ready for algebraic abstraction until senior year. The course builds on experiences in beginning computer science courses that have implicitly imparted to students a sense of the underlying algebra of computer science structures, and formally presents topics like Boolean algebra, partial orders, finite-state machines, and formal languages that will be used in later computer science courses. At the same time, this course can also be very rewarding to regular mathematics majors who should appreciate the new algebraic structures such as formal languages and finite state machines that are so different from the structures in the regular abstract algebra course. Substantial class time should be spent on proofs with special emphasis on induction arguments. This course is just as mathematically sophisticated and capable of developing abstract reasoning as abstract algebra, but the topics stress set-relation systems rather than binary-operation systems. Indeed the abstract complexity of the basic structures is much greater in applied algebra, but this complexity precludes the construction of logical pyramids built of simple algebraic inferences common to many areas of abstract algebra.

This course is an advanced version of the lower-division B3 Discrete Structures course in ACM Curriculum 68. The B3 course was the source of much dissatisfaction because it contained a huge amount of material, and it required too great mathematical maturity for a lower-division course. The recent ACM Curriculum 78 recommends that the B3 course be treated as a more advanced course and that it should be taught in mathematics departments rather than computer science departments. The B3 course was the subject of several papers at meetings of the ACM Special Interest Group in Computer Science Education (SIGCSE); see the February issues (Proceedings of SIGCSE annual meeting) of the *SIGCSE Bulletin* in 1973, 1974, 1975, 1976.

The B3 course contained both applied algebra and discrete methods. The MSP panel recommends that a separate full course be devoted to discrete methods (see the discrete methods course description earlier in this Section). Because some computer science courses may devote a substantial amount of time introducing some of the topics in the above applied algebra syllabus, the exact content of this course will vary substantially from college to college. For this reason the syllabus outline was kept brief. At some colleges, applied algebra will still have to be combined with discrete methods in one course (the computer science major may not have the time for two separate courses). The applied algebra part of such a combined course would, in most cases, concentrate on topics 1, 2, 3, 4, 6 in the syllabus. Many of the discrete structures texts listed below cover both applied algebra and discrete methods.

COURSE TOPICS

A. Sets, binary relations, set functions, induction, basic graph terminology.

B. Partially ordered sets, order-preserving maps, weak orders.
C. Boolean algebra, relation to switching circuits.
D. Finite state machines, state diagrams, machine homomorphism.
E. Formal languages, context-free languages, recognition by machine.
F. Groups, semigroups, monoids, permutations and sorting, representations by machines, group codes.
G. Modular arithmetic, Euclidean algorithm.
H. Optional topics: linear machines, Turing machines and related automata; Polya's enumeration theorem; finite fields, Latin squares and block design; computational complexity.

APPLIED ALGEBRA TEXTS

1. Dornhoff, Lawrence and Hohn, Frantz, *Applied Modern Algebra,* Macmillan, New York, 1978.
2. Fisher, James, *Application-Oriented Algebra,* T. Crowell Publishers, New York, 1977.
3. Johnsonbaugh, Richard, *Discrete Mathematics,* Macmillan, New York, 1984.
4. Korfhage, Robert, *Discrete Computational Structures,* Academic Press, New York, 1974.
5. Liu, C.L., *Elements of Discrete Mathematics,* McGraw Hill, New York, 1977.
6. Preparata, Franco and Yeh, Robert, *Introduction to Discrete Structures,* Addison-Wesley, Reading, Mass., 1973.
7. Prather, Robert, *Discrete Mathematical Structures for Computer Sciences,* Houghton Mifflin, Boston, 1976.
8. Stone, Harold, *Discrete Mathematical Structures and Their Applications,* Science Research Associates, Chicago, 1973.
9. Tremblay, J. and Manohar, R., *Discrete Mathematical Structures with Applications in Computer Sciences,* McGraw Hill, New York, 1975.

Numerical Analysis Course

In any elementary numerical analysis course a balance must be maintained between the theoretical and the application portion of the subject. Normally, such a course is designed for sophomore and junior students in engineering, mathematics, science, and computer science. Students should be introduced to a wide selection of numerical procedures. The emphasis should be more on demonstrations than on rigorous proofs (however, this is not meant to slight necessary theoretical aspects of error analysis). At least one or two applied problems from each of the major topics should be included so that students have a good understanding of how the art of numerical analysis comes into play.

The course outline below presents a good selection of topics for a one-semester course. Error analysis should be continuously discussed throughout the duration of the course so as to stress the effectiveness and efficiency of the methods. Alternative methods should be contrasted and compared from the standpoint of the computational effort required to attain desired accuracy.

An optional approach to this course would emphasize a full discussion (with computer usage) of one procedure for each course topic (after the computer arithmetic introduction). A sample of five such procedures is:

1. The Dekker-Brent algorithm (see UMAP module No. 264).
2. A good linear equation solver involving LU-decomposition.
3. Cubic spline interpolation.
4. An adaptive quadrature code.
5. The Runge-Kutta-Fehlberg code RKF4 with adaptive step determination.

Weekly assignments should include some computer usage; in total, four or five computer exercises for each major topic. Students should do computer work for larger applied programs in small groups. However, the concept of utilizing "canned" programs with minor modifications should be stressed. Such an approach nicely brings out the strong interdependence between computers and numerical analysis yet does not overemphasize the efforts necessary to program a problem. An interactive computer system using video terminals is ideal for this course. Microcomputers and even hand-held calculators can also be used effectively. One or two applied homework problems from each of the main topics keep students aware of the balance that is necessary between the art and the science of numerical analysis. Prerequisites for this course should be a year of calculus including some basic elementary differential equations and a computer science course.

For schools on a quarter system, two quarters should be a minimal requirement and the above material would be more than ample. One should spend the first quarter on numerical solutions of algebraic equations and systems of algebraic equations and the last quarter on the other topics.

COURSE OUTLINE

A. *Computer arithmetic.* Discretization and round-off error; nested multiplication.
B. *Solution of a single algebraic equation.* Initial discussion of convergence problems with emphasis on meaning of convergence and order of convergence;

Newton's method, Bairstow's method; interpolation.

C. *Solution systems of equations.* Elementary matrix algebra; Gaussian methods, LU decomposition, iterative methods, matrix inversion; stability of algorithms (examples of unstable algorithms), errors in conditioned numbers.

D. *Interpolating polynomials.* Lagrange interpolation to demonstrate existence and uniqueness of interpolating polynomials and for calculation of truncation error terms; splines, least squares, inverse interpolation; truncation, inherent errors and their propagation.

E. *Numerical integration.* Gaussian quadrature, method of undetermined coefficients, Romberg and Richardson extrapolation (for both integration and differentiation), Newton-Cotes formulas, interpolating polynomials, local and global error analysis.

F. *Numerical solution of ordinary differential equations.* Both initial value and boundary value problems; Euler's method, Taylor series method, Runge-Kutta, predictor-corrector methods, multistep methods; convergence and accuracy criteria; systems of equations and higher order equations.

If this course has an enrollment of under 25 students, non-standard testing can be considered, such as a take-home midterm. At the end of the term, instead of the traditional three hour examination, each student can write an expository paper exploring in greater depth one of the topics introduced in class or investigating a subject not included in the work of the course, either approach to include computational examples with analysis of errors. (Since most of the students will not have had previous experience in writing a paper, topics may be suggested by the instructor or must be approved if student devised; scheduled conferences and preliminary critical reading of papers guard against disastrous attempts or procrastination.) Some examples of final projects are: spline approximations; relaxation methods; method of undetermined coefficients in differentiation and integration; least squares approximations; parabolic (or elliptic or hyperbolic) partial differential equations; numerical methods for multi-dimensional integrals; multi-step predictor-corrector methods.

NUMERICAL ANALYSIS TEXTS

1. Cheney, Ward and Kincaid, David, *Numerical Mathematics and Computing,* Brooks/Cole, Monterey, Calif., 1980.
2. Conte, S. and DeBoor, C., *Elementary Numerical Analysis,* McGraw Hill, New York, 1978.
3. Gerald, Curtis F., *Applied Numerical Analysis, 2nd Edition,* Addison-Wesley, Reading, Mass., 1978.
4. Forsythe, G.E. and Moler, C.B., *Computer Solutions of Linear Algebraic Systems,* Prentice-Hall, Englewood Cliffs, New Jersey, 1967.
5. Hamming, R.W., *Numerical Methods for Scientists and Engineers, 2nd Edition,* McGraw Hill, New York, 1973.
6. James, M.L.; Smith, G.M.; Wolford, J.C., *Applied Numerical Methods for Digital Computation,* Harper & Row, New York, 1985.
7. Ralston, Anthony and Rabinowitz, Philip, *First Course in Numerical Analysis,* McGraw Hill, New York, 1978.

Panel Members

ALAN TUCKER, CHAIR, SUNY, Stony Brook.

RICHARD ALO, Lamar University.

WINIFRED ASPREY, Vassar College.

PETER HILTON, Case Western Reserve/Battelle Institute.

DON KREIDER, Dartmouth College.

WILLIAM LUCAS, Cornell University.

FRED ROBERTS, Rutgers University.

GAIL YOUNG, Case Western Reserve.

Calculus

This chapter contains the report of the Subpanel on Calculus of the CUPM Panel on a General Mathematical Sciences Program, reprinted with minor changes from Chapter II of the 1981 CUPM report entitled RECOMMENDATIONS FOR A GENERAL MATHEMATICAL SCIENCES PROGRAM.

Rationale

The Calculus Subpanel was charged with examining the traditional calculus sequence of the first two years of college mathematics: two semesters of single-variable calculus; one semester of linear algebra; one semester of multivariable calculus. In approaching this task, the subpanel considered syllabi through which this sequence is implemented at various colleges and universities, the syllabus for the Advanced Placement Program in Calculus, and alternatives to calculus as the entry-level course in the mathematical sciences, for example, finite mathematics or discrete methods.

The subpanel eventually came to the conclusion that the rationale for certain parts of the traditional calculus sequence remains valid, although some restructuring and increased flexibility are warranted to reflect the differing mathematical requirements of the social and biological sciences and, increasingly, of computer science. The general recommendations of the subpanel are thus:

1. To make no substantive changes in the first semester of calculus;
2. To restructure the second semester around modeling and computation, although leaving it basically a calculus course;
3. To branch to three independent courses in the second year:
   a. Applied Linear Algebra,
   b. Multivariable Calculus (in dimensions 2 and 3),
   c. Discrete Methods.

Descriptions of the first and second semesters of calculus, applied linear algebra, and multivariable calculus are given below. The discrete methods course is discussed in the first chapter, "Mathematical Sciences."

The subpanel views its recommendations as conservative. Tony Ralston has argued, for example, that calculus need not be the entry-level course in the mathematical sciences and that a course in discrete methods is a reasonable alternative, better serving some areas such as computer science (see "The Twilight of the Calculus," which appeared under the title "Computer Science, Mathematics, and the Undergraduate Curricula in Both" in the *American Mathematical Monthly*, 88:7 (1981) 472-485). In his view, to ignore discrete methods, even in the first two years of college mathematics, would be absurd in this day.

The subpanel does not disagree with the general sense of this position. On the other hand, the subpanel feels that the language, spirit, and methods of traditional calculus still permeate mathematics and the natural and social sciences. To quote Ralston himself, "The calculus is one of man's great intellectual achievements; no educated man or woman should be wholly ignorant of its elements." Perhaps the time is not far off when calculus will be displaced as the entry-level course, but it has not arrived yet.

The place for rigor. The subpanel believes strongly that, in the first two years, theorems should be *used* rather than *proved*. Certainly correct statements of theorems such as the Mean Value Theorem or l'Hôpital's Rule should be given; but motivation, as long as it is recognized as such, and usage are more important than proofs. The place for theoretical rigor is in later upper-level courses. In this regard, the subpanel agrees with the program philosophy outlined in the first chapter, "Mathematical Sciences."

First Semester Calculus

The first semester of calculus, especially, contains a consensus on essential ideas that are important for modeling dynamic events. This course has evolved through considerable effort in the mathematical community to present a unified treatment of differential and integral calculus, and it serves well both general education and professional needs. It is historically rich, is filled with significant mathematical ideas, is tempered through its demonstrably important applications, and is philosophically complete. Most syllabi for its teaching cover the usual topics:

A. *Limits and continuity.*
B. *Differentiation rules.*
C. *Meaning of the derivative.* Applications to curve sketching, maximum-minimum problems, related rates, position-velocity-acceleration problems.
D. *Antidifferentiation.*

E. *The definite integral and the Fundamental Theorem of Calculus.*
F. *Trigonometric functions.*
G. *Logarithmic and exponential functions.* Including a brief exposure to first-order, separable differential equations (with emphasis on $y' = ky$).

The first (and second) calculus courses should be 4- or 5-credit hour courses. If less time is available, topics will have to be pushed later into the calculus sequence, with some multivariate calculus material left for an analysis/advanced calculus course. Mathematics courses should not rush trying to cover unrealistic syllabi.

It might be desirable to add more non-physical sciences examples to C (e.g., a discussion of the use of the word "marginal" in economics), although serious modeling examples should be postponed to the second semester. Integration as an averaging process can be included in E, but applications and techniques (numerical or algebraic) of integration are better left to the second semester. Exponential growth and decay are important concepts that must be emphasized in G.

Second Semester Calculus

There does not appear to be much slack or fat in the first semester of calculus. It is in the second semester, therefore, when numerical techniques, models, and computer applications can be introduced. Unlike the first semester of calculus, the second semester does not enjoy the same consensus on either its central theme or its content. It tends to be a grab bag of "further calculus topics"—further techniques of integration, more applications of integration, some extension of techniques to the plane (parametric equations), sequences and infinite series, and more differential equations. Each of these topics is, in isolation, important at some stage in the training of scientists and mathematicians. But it is less clear that packaging them in this way and having them occupy this critical spot in the curriculum is justified today, given the pressing needs of computer science and the non-physical sciences.

From time to time it has been urged that multivariable calculus should be started during the second semester. But few institutions have implemented this suggestion. And the subpanel believes that, in the meantime, higher priorities for the second course have materialized in the form of applications and computing.

The subpanel considered recommending branching in the curriculum after the first semester of calculus, with students advised to take courses more directly relevant to their career goals. But it finally concluded that there are still substantial reasons for keeping students in one "track" through the first two courses. In most American colleges, a "choice" in the second course would require most students to be thinking seriously about career goals within a few weeks of arriving on campus as freshmen. This does not strike us as realistic nor in the best interests of liberal education. Moreover, we continue to feel that many of the ideas and technical skills arising in the second calculus course are reasonable to include at this point in the curriculum. Thus, the final conclusion is that a restructuring and change of emphasis in the second semester calculus course is preferable to its replacement.

The Calculus Subpanel recommends the following changes in the second calculus course:

A. *An early introduction of numerical methods.* Implemented through simple computer programs. Solving one (or a system of two) first-order differential equation(s).

B. *Techniques of integration.* General methods such as integration by parts, use of tables, and techniques that extend the use of tables such as substitutions and (simple) partial fraction expansions; less emphasis should be placed on the codification of special substitutions.

C. *Numerical methods of integration.* Examples where numerical and "formal" methods complement each other, e.g., evaluating improper integrals where substitutions or integration-by-parts make the integral amenable to efficient numerical evaluation.

D. *Applications of integration.* Illustrate the "setting up" of integrals as Riemann sums. The emphasis should be on the modeling process rather than on "visiting" all possible applications of the definite integral.

E. *Sequences and series.* These topics should have substantially changed emphasis:
   1. Sequences should be elevated to independent status, defined not only through "closed formulas" but also via recursion formulas and other iterative algorithms. Estimation of error and analysis of the rate of convergence should accompany some of the examples.
   2. Series should appear as a further important example of the idea of a sequence. Power series, as a bridge from polynomials to special functions, should figure prominently. Specialized convergence tests for series of constants can be deemphasized.
   3. Approximation of functions via Taylor series, and estimation of error, accompanied by im-

plementation of such approximations on a computer.

F. *Differential equations.* Should be treated with less (but not zero) emphasis on special methods for solving first-order equations and constant coefficient linear equations (especially the non-homogeneous case). More valuable would be: vector field interpretation for first-order equations, numerical methods of solution, and power series methods for solving certain equations. Applications should arise in mathematical modeling contexts and both "closed form" and "numerical" solutions should be illustrated.

The new second course in calculus does not differ radically in content from the traditional second semester course. It is a conservative restructuring that can be taught from existing textbooks and based on modest modifications of many existing syllabi. But the intended change in "flavor" and emphasis should be more dramatic. About twelve lectures (of the usual 40 lectures) must be modified substantially to achieve the desired computer emphasis. Numerical algorithms will thus figure prominently, along with the formal techniques of calculus. Concepts not usually in a calculus course such as error estimation, truncation error, round-off error, rate of convergence, and bisection algorithms will be included. The theme for the course will be "calculus models." Consideration of even a few UMAP-type models would be enough to change the nature of the course significantly and to provide the intended "tying together" of the traditional calculus topics that are included in the course.

A syllabus for the course could be constructed by starting with the second calculus course described in the CUPM report, *A General Curriculum for Mathematics in Colleges* (revised 1972), or with the Advanced Placement BC Calculus Syllabus. Topics to be diminished or omitted include: emphasis on special substitutions in integrals, l'Hôpital's rule except as it arises naturally in connection with Taylor series, polar coordinates, vector methods, complex numbers, non-homogeneous differential equations and the general treatment of constant-coefficient homogeneous linear differential equations. Many of these topics will appear in examples but will not be emphasized in themselves.

Intermediate Mathematics Courses

Although the Calculus Subpanel recommends retaining a single track for students during their first year, it just as strongly recommends that three different courses be available from which students choose (with advising) their intermediate mathematics courses. Two of these courses, whose descriptions follow, are Applied Linear Algebra and Multivariable Calculus. The third, Discrete Mathematics, is described in the first chapter, "Mathematical Sciences."

Applied Linear Algebra

For a large part of modern applied mathematics, linear algebra is at least as fundamental as calculus. It is the prerequisite for linear programming and operations research, for statistics, for mathematical economics and Leontief theory, for systems theory, for eigenvalue problems and matrix methods in structures, and for all of numerical analysis, including the solution of differential equations. The attractive aspect about these applications is that they make direct use of what can be taught in a semester of linear algebra. The course can have a sense of purpose, and the examples can reinforce this purpose while they illustrate the theory.

A number of major texts have arrived at a reasonable consensus for a course outline. Their outlines are well matched with the needs of both theory and application. Applications can include such topics as systems of linear differential equations, projections and least squares. But the subpanel strongly recommends that more substantial applications to linear models should be a central part of the construction of the course. Many different applications of this kind are accessible and can be found in the texts mentioned. Thus, no rigid outline is required. The development of the subject moves naturally from dimension 2 to 3 to n, and although that is an easy and familiar step, it nevertheless represents mathematics at its best. The combination of importance and simplicity is almost unique to linear algebra. Linear programming is an excellent final topic in the course. It brings the theory and applications together.

The changes in this course are ones of emphasis that recognize that the course must be more than an introduction to abstract algebra. Abstraction remains a valuable purpose, and linearity permits more success with proofs than the epsilon-delta arguments of calculus. However, the main goal is to emphasize applications and computational methods, opening the course to the large group of students who need to *use* linear algebra.

TEXTS

1. Hill, Richard, *Elementary Linear Algebra,* Academic Press, New York, 1986.
2. Kolman, Bernard, *Introductory Linear Algebra with Applications,* Macmillan, New York, 1979.

3. Rorres, Chris and Anton, Howard, *Applications of Linear Algebra,* John Wiley & Sons, New York, 1979 (paperback supplementary text).
4. Strang, Gil, *Linear Algebra and Its Applications, 3rd Edition,* Harcourt Brace Jovanovich, San Diego, 1988.
5. Tucker, Alan, *A Unified Introduction to Linear Algebra,* Macmillan, New York, 1988.
6. Williams, Gareth, *Linear Algebra with Applications,* Allyn and Bacon, Boston, 1984.

Multivariable Calculus

This is the traditional multivariable calculus course at many colleges and universities. It is not a new course, but for many schools it would represent a movement in the direction of "concrete" treatment of multivariable calculus rather than the more recent elegant treatments making heavy use of linear transformations and couched in general (high dimensional) terms. The course begins with an introduction to vectors and matrix algebra. Topics include Euclidean geometry, linear equations, and determinants. The remainder of the course is an introduction to multivariable calculus, including the analytic geometry of functions of several variables, definitions of limits and partial derivatives, multiple and iterated integrals, non-rectangular coordinates, change of variables, line integrals, and Green's theorem in the plane.

Differential Equations

The Calculus Subpanel has considered the place of differential equations in the curriculum. It recommends that the topic be treated at two levels:

1. Through methods and examples involving differential equations, spiraled through the calculus sequence, and
2. Through a substantial course in differential equations, available to students upon completion of the first-year calculus sequence and applied linear algebra.

We note here topics in differential equations that are part of the preceding courses:

- Solutions of $y' = ky$ occur in the first semester of calculus. Exponential growth and decay are discussed.
- Solution of second order linear differential equations are included in the second semester of calculus. Oscillating solutions occur as examples. In addition, geometrical interpretations (direction field), numerical solutions and power series solutions are included.
- Applied Linear Algebra includes the solution of linear constant coefficient systems of differential equations using eigenvalue methods.

Although the Calculus Subpanel has *not* recommended a full course in differential equations in the calculus sequence of the first two years, it has suggestions for a subsequent course. Such a course should not be a compendium of techniques for solving in closed form various kinds of differential equations. Libraries are full of cookbooks; one hardly needs a course to use them. What is important is to develop carefully the models from which differential equations spring. Modeling obviously means more than an application such as:

> According to physics, the displacement $x(t)$ of a weight attached to a spring satisfies $x''-bx'+kx = 0$. Solve for $x(t)$ given that $b = 2, k = 3, x(0) = 1, x'(0) = 0$.

For a more serious approach to applications, we refer to the art forgery problem at the beginning of Braun (see below) or indeed almost any of the models discussed in the suggested texts.

The meaning of the word "solution" must be scrutinized. Different viewpoints must be introduced—numerical, geometric, qualitative, linear algebraic and discrete.

A possible syllabus for a differential equations course is:

A. *First-order equations.* Models; exact equations; existence and uniqueness and Picard iteration; numerical methods.

B. *Higher-order linear equations.* Models; the linear algebra of the solution set; constant coefficient homogeneous and non-homogeneous; initial value problems and the Laplace transform; series solutions.

C. *Systems of equations and qualitative analysis.* Models; the linear algebra of linear systems and their solutions; existence and uniqueness; phase plane; nonlinear systems; stability.

Since some of these topics will have already been introduced in courses from the calculus sequence, there may be time for a brief discussion of partial differential equations and Fourier series. Existence and uniqueness theorems are included here only because of the light they or their proofs might shed on methods of solution (e.g., Picard iteration).

TEXTS

The course can be taught using any of the many reasonable differential equations texts with a modest amount of applications, supplemented by:

Braun, Martin, *Differential Equations and Their Applications, Second Edition,* Springer-Verlag, New

York, 1978.

Braun remains the only text to build extensively on applications, but it has the serious drawback that it is based on single-variable calculus and avoids linear algebra.

A somewhat radical alternative is a theoretical course involving more qualitative or topological analysis emphasizing systems of equations. The subpanel does not suggest a syllabus, but refers instead to V.I. Arnold, *Ordinary Differential Equations,* MIT Press, Cambridge (paperback), 1978.

This course would have applied linear algebra and multivariable calculus as prerequisites and could be taken as early as the second semester of the sophomore year if the two prerequisites were taken concurrently the previous semester.

Subpanel Members

DON KREIDER, CHAIR, Dartmouth College.
ROSS FINNEY, Educational Development Center.
JOHN KENELLY, Clemson University.
GIL STRANG, MIT.
TOM TUCKER, Colgate University.

Core Mathematics

This chapter contains the report of the Subpanel on Core Mathematics of the CUPM Panel on a General Mathematical Sciences Program, reprinted with minor changes from Chapter III of the 1981 CUPM report entitled RECOMMENDATIONS FOR A GENERAL MATHEMATICAL SCIENCES PROGRAM.

New Roles for Core Mathematics

In the 1960's CUPM extensively examined curriculum in core mathematics—upper division subjects that comprise the trunk from which the other specialized branches and applications of mathematics emerge. It reviewed and revised its recommendations in 1972. See the *Compendium of CUPM Recommendations* published by the Mathematical Association of America, especially the 1972 revision of the *General Curriculum for Mathematics in Colleges*. With the current restructuring of the mathematics major into a mathematical sciences major, new questions have been raised about core mathematics curriculum. These questions concern the role of core mathematics in a mathematical sciences major as much as syllabi of individual courses. This chapter focuses on four questions that were addressed to the Core Mathematics Subpanel by the parent Panel on a General Mathematical Sciences Program.

The members of this subpanel represent a variety of institutions, public and private, liberal-arts colleges, and research-oriented universities. All the members have seen at their institutions a divergence of the mathematics major from its form during their own undergraduate training, as career opportunities for mathematics majors have changed. In part, the members lament the passing of the mathematics program that nurtured their love of mathematics. At the same time they acknowledge the challenge of the diversity of the present and future. They realize that it is not now realistic for CUPM to recommend a core set of pure mathematics courses to be taken by all mathematical sciences majors in every institution.

While the mathematics major has generally broadened towards a mathematical sciences major, it is still possible for an institution, large or small, to elect to retain a traditional pure mathematics major, alone or in conjunction with an applied mathematics major. But it is clearly more appropriate to work within current realities to fashion a unified mathematical sciences major with diminished pure content, a major incorporating both breadth and selective depth. (If size warrants, the unified major can have several tracks, one for preparation for graduate study in mathematics.) This subpanel is concerned with the role in a mathematical sciences major of upper-level core mathematics courses, and more generally with appreciation of the depth and power of mathematics.

A prime attribute of a person educated in mathematical sciences is his or her ability to respond when confronted with a mathematical problem, whether in pure mathematics, applied mathematics, or one which uses mathematics that the person has not seen before. Our students should be prepared to function as professionals in areas needing mathematics not by having learned stock routines for stock classes of problems but by having developed their ability in problem solving, modeling and creativity. This general pedagogical theme, that was stressed throughout the first chapter "Mathematical Sciences," guided the thinking of the Core Mathematics Subpanel.

The report of the Core Mathematics Subpanel is meant to be supportive rather than directive. What an individual department does should reflect its constituency of students, their needs, their numbers, and the goals, character and size of the institution.

Four Questions

QUESTION 1: ***Is there a minimal set of upper-level core mathematics (algebra, analysis, topology, geometry) that every mathematical sciences major should study?***

ANSWER: No. There is no longer a common body of pure mathematical information that every student should know. Rather, a department's program must be tailored according to its perception of its role and the needs of its students. Whether pure mathematics is required of all in some substantial way; whether it is used as an introduction to advanced work of applied nature or as a completion to an applied program; or whether pure mathematics is simply one track in a collection of programs in a large department will be an institutional option. Departments must recognize this fact, establish their programs with a clear understanding of objectives that are being met, and be prepared to share and explain these perceptions with their students. The limited resources of smaller departments must be exploited with great efficiency and wisdom. Such departments may face a difficult decision of whether to abandon certain traditional branches of mathematics entirely in order to offer courses and tracks best suited to their students.

The underlying problem is that students enter college with much less mathematics than they used to, but they expect to leave with more. There is a wide span of preparation among entering college students, they want an education that is specific to chosen career goals, and the levels of mathematical and computational skills and sophistication that accompany these goals have risen. Core courses such as abstract algebra and analysis are valuable for continuing study in many fields, but they are not essential for all careers.

The Core Mathematics Subpanel and the parent Mathematical Sciences Panel jointly recommend that all mathematical sciences students take a sequence of two courses leading to the study of some subject in depth (see the first chapter, "Mathematical Sciences").

QUESTION 2: *Should there be major changes in the content or mode of instruction of upper-level core mathematics courses?*

ANSWER: While there will continue to be some students who plan to move toward a doctorate in pure, or applied, mathematics and an academic career, the mathematical sciences major is seen by most students as preparation for immediate employment or for Masters-level graduate training in areas outside of mathematics (but where mathematical tools are needed). Thus mathematics departments can no longer view their upper-division courses as a collection of courses that faculty wish they had had prior to admission to graduate school. Rather, departments must offer pure mathematics courses that are compatible with the overall goals of a mathematical sciences major, courses that are intellectually and pedagogically complete in themselves, courses that are both the beginning and the end of most students' study of the subject. The main objective in such courses now is developing a deeper sense of mathematical analysis and associated abstract problem-solving abilities. In these courses students learn how to learn mathematics.

There is always a continuing need to re-examine the nature and content of any course. Some courses carry baggage that may be there largely for historical reasons. A frequent example of this is the traditional course in differential equations which is populated by isolated discoveries of the Bernoulli clan (and lacking in discussion of numerical methods). Instructors are slow to discard topics that have a strong aesthetic appeal (for the instructors) but are no longer important building blocks in the field. Syllabi and approaches in pure mathematics courses must be adapted to changing constituencies with a careful balance of learning new concepts and modes of reasoning and of using these constructs, a balance of "listening" and "doing." Students should emerge from a course feeling that they have become junior experts in some topics: they should know facts and relationships, know some of the "whys" behind this mathematics.

It would be desirable for courses to be structured with review stages that require reflection by students of what analysis to use to solve a problem. The courses need to contain assignments that ask for short proofs of results and for application of concepts and techniques from one problem to another (apparently unrelated) problem. Proper judgment in the selection of a method of analysis is the key both to constructing mathematical proofs and to problem solving in applied mathematics; nurturing this ability is the critical challenge to instructors. Students should be required to present material both orally and in writing on a regular basis. Since students do not have to know a standard body of theorems for graduate study, the course content in algebra, analysis, topology and geometry can vary according to faculty interests and possible ties with stronger quantitative areas of an institution (e.g., physics or biology).

The density of proofs in an upper-level course is always a controversial issue. It is traditional to feel that one objective of such a course is to teach students how to construct proofs. However, this skill comes slowly and seldom arouses the same pleasure in students as it does in instructors. Some proofs are needed in any upper-level mathematics course to knit together the entire structure that is being presented, but one should probably aim at piecewise rigor rather than a Landauesque totality. Students' mathematical maturity will develop as much, and it will be far less painful.

The preceding pedagogical goals in core mathematics must accommodate the reality that courses such as abstract algebra may only be offered in alternate years and that two-semester sequences or courses with core mathematics prerequisites will be difficult to schedule. With a broad mixture of students in infrequently-offered courses, instructors must be sensitive to the discouragement some students may feel in the presence of more sophisticated seniors.

QUESTION 3: *How can the full scope of mathematics be conveyed to students? Should this be done by one-semester survey courses that cover a range of fields?*

ANSWER: Students pursuing specific career goals in mathematical sciences and those taking upper-level mathematical "service" courses need to be made aware of the depth and breadth of mathematics and the greater mathematical maturity that their subsequent careers may demand. Mathematical survey courses do not appear to be the answer. They will not be able to move beyond vocabulary and notation to give any sense of global structure in any of the fields covered.

Physicists seem to have been remarkably successful in communicating some understanding about the "big picture" to their students and laymen through expository articles that treat highly technical subjects by presenting only a projection or shadow of the true structure, but doing so in a way that does not seem to offend their consciences. Similar approaches should be possible in mathematics using expository *American Mathematical Monthly, Mathematics Magazine,* or *Scientific American* articles. Following the reading of such an article, a (once-a-week) class would discuss concepts, technicalities and applications in the article plus additional examples. Natural topic areas are complex analysis and two-dimensional hydrodynamics; number theory and public key cryptography; calculus of variations and soap films; queueing theory and, say, toll road design.

More traditional ways of projecting the wide-ranging nature of mathematics are by rotation of courses and by providing seminars, extracurricular mathematical activities, summer work opportunities, and by references and linkages to mathematics in courses in other departments. This breadth should also give a sense of the rapidly changing nature of uses of mathematics and of the need of learning how to learn mathematics.

QUESTION 4: *Should pure mathematics courses be postponed for most students until the senior year to follow and abstract from more applied courses earlier in the curriculum?*

ANSWER: Many mathematical sciences students who prefer problem solving to theory appear to have considerable difficulty in their sophomore or junior years with abstract core mathematics. For these students, core mathematics may better wait until a senior year "capstone" course(s) that builds on maturity developed in earlier problem solving courses. This course (preferably year-long if only one such course is required) in a subject such as analysis or abstract algebra would build a student's capacity (and appetite) for abstraction and proof and for solving complex problems involving a combination of analytical techniques. The course would seek depth rather than breadth. The course should link abstract concepts with their concrete uses in previous courses, such as integration concepts used in limiting probability distributions. It should illustrate in several ways the power and usefulness of mathematical abstraction and generalization.

There are two important provisos about senior-year courses. First, when core courses cannot be offered every year, they obviously must be accessible to most juniors. Second, the mathematically gifted student (whether a mathematics major or not) must be able to take such senior core courses in the sophomore year without needing applied prerequisites that other students naturally take before the core course. Such gifted students today are often directed towards popular careers such as engineering or medicine and by their senior year would be too immersed in professional training to take the pure mathematics course that would reveal their mathematical research potential.

It is worthwhile recalling that before 1950 few colleges offered regular courses in abstract algebra, topology, or up-to-date advanced calculus. The 1950's and 1960's were memorable in mathematics education, but today's students must be viewed as in the historical mainstream rather than as slow in learning to handle abstractions.

Individual institutions will differ greatly in the design of such senior courses. As noted in the discussion of Question 2, these courses should require oral and written student presentations. The spirit of this recommendation could be achieved with a year-long course in a subject such as differential equations or combinatorics that begins with applications and leads to abstraction or a course that begins with abstraction and leads to applications.

Sample Course Outlines

In this section we discuss two approaches to the fundamental upper-level core subjects of abstract algebra and analysis. We suggest an ideal treatment and then a more modest version that is appropriate for most current mathematical sciences students. The descriptions are stated in terms of student objectives.

The philosophy behind each of the course descriptions is that the student needs a working understanding of the subject far more than a detailed intensive and critical knowledge. The instructor's central goal is to teach the student how to learn mathematics, expecting that students will correctly retain only a tiny portion of what was taught, but that when they need to refresh their knowledge, they will be far better able to do so than if they had never taken the course. Proofs are not of major importance, but in both approaches students should be able to understand what the hypotheses of a theorem mean and how to check them. They should also be able to detect when seemingly plausible statements are false (and should be shown counterexamples to such statements; e.g., integrals that should converge but do not).

Abstract Algebra I (Ideal)

A. Give the student a guided tour through the algebraic "zoo," so that he or she knows what it means to be a group, a ring, a field, an associative algebra, etc. Include associated concepts such as category, morphism, isomorphism, coset, ideal, etc.
B. Show the student useful ways for generating one algebraic structure out of another, such as automorphism groups, quotient groups, algebras of transformations, etc.
C. Give the student an understanding of the basic structure theorems for each of the algebraic systems discussed, as well as an understanding of their proofs.
D. Give the student experience in using the preceding ideas and constructions and seeing how these ideas arise in other branches of mathematics (analysis, number theory, geometry, etc.).
E. Show the student how algebra is used in fields outside of mathematics, such as physics, genetics, information theory, etc.

Abstract Algebra II (Modest)

A. Combine parts of A and B of Course I by showing students at least two different types of algebraic structures and several instances in which such an algebraic structure evolved or is constructed out of another mathematical structure. The goal is for a student to be able to recognize when a situation has aspects that lend themselves to an algebraic formulation; e.g., rings out of polynomials.
B. Describe part of the theory for one of the structures introduced in A and illustrate several of the deductive steps in the theory. Students should see the nature of tight logical reasoning and the usefulness of algebraic concepts, as well as come to appreciate the cleverness of the theory's discoverers.
C. Discuss at least one application of algebra outside of mathematics.
D. Assign students a variety of problems which require recognition of algebraic structures in unfamiliar forms, proof of small deductive steps, and use of theory in B.

Analysis I (Ideal)

A. Give the student a working knowledge of point set topology in R^n and analogous concepts for a metric space.
B. Study the class of continuous maps from a region in R^n into R^m, and the special properties of maps in class C' and C''.
C. Study integration of continuous and piecewise continuous functions over appropriately chosen sets, bounded and unbounded, and then extend this to integration with respect to set functions.
D. Extend to the theory of differential forms and develop a relationship between differentiation of forms and the boundary operator, via Stokes' theorem.

Analysis II (Modest)

A. Give the student a glossary of terms in point set topology, appropriate also to a metric space and applied to R^n, and practice in their meanings. (Do not prove inter-relations, but state them clearly.)
B. Introduce the class of C'' maps from R^n into R^m, and discuss a few problems involving such functions, each motivated by a concrete "real" situation. Solve each of the problems by stating and illustrating the appropriate general theorems, and in a few cases, sketching part of the proofs.
C. Discuss integration in terms of measurement and averaging, extend this to R^n, and explain briefly techniques of numerical integration. At all stages give attention to improper integrals.
D. Extend the notion of function to differential forms, illustrated with physical and geometric examples. Motivate Stokes' theorem as the analogue of the fundamental theorem of calculus, and arrive at a correct formulation of it without proof. Illustrate the theorem with examples, including some involving the geometric topology of surfaces; if students' background is appropriate, examples in physics (hydrodynamics or electromagnetism) should be given.

An analysis course can also be given an "advanced calculus" emphasis including topics such as Fourier series and transforms, special functions, and fixed-point theorems, with applications of these topics to differential equations. For further discussion of this approach, see versions one and three of Mathematics 5 in the CUPM recommendations for a *General Curriculum in Mathematics for Colleges* (revised 1972).

Subpanel Members

PAUL CAMPBELL, CHAIR, Beloit College.
LIDA BARRETT, Northern Illinois University.
R. CREIGHTON BUCK, University of Wisconsin.
MARGARET HUTCHINSON, University of St. Thomas.

Computer Science

This chapter contains the report of the Subpanel on Computer Science of the CUPM Panel on a General Mathematical Sciences Program, reprinted with minor changes from Chapter IV of the 1981 CUPM report entitled RECOMMENDATIONS FOR A GENERAL MATHEMATICAL SCIENCES PROGRAM.

A Growing Discipline

Computer Science is a new and rapidly growing scientific discipline. It is distinct from Mathematics and Electrical Engineering. The subject was once closely identified in mathematicians' minds with writing computer programs. In the beginning, however, computer scientists concentrated on the discipline's mathematical theories of numerical analysis, automata, and recursive functions, as well as on programming. In the past decade, theories developed to understand problems in software design (compilers, operation systems, structured programs, etc.) have blossomed. These theories involve the analysis of complex finite structures, and in this sense have a strong mathematical bond with the finite structures common in operations research and diverse areas of applied mathematics.

More importantly, these computer science theories are needed by analysts who design algorithms for complex problems in the mathematical sciences. For this reason, all mathematical sciences students must be given an introduction to the basic concepts of computer science. Further, facility in computer programming is required of all mathematical sciences students so that they can perform practical computations in mathematical sciences courses and in subsequent mathematical sciences careers.

Although only one-third of the country's colleges and universities now have computer science departments, the number of students currently majoring in computer science taught in a computer science department (approximately 50,000 students) is greater than the number of all majors in mathematics, mathematical sciences, and applied mathematics. The computer science recommendations in this chapter are designed for institutions where computer science is taught in a mathematical sciences department or in a mathematics department. When a separate computer science department exists, that department's diversity of computer science offerings will enhance a mathematical sciences major. A mathematical sciences undergraduate program and a computer science undergraduate program should complement one another to the advantage of both departments and their students (for example, see the description of the interaction at Potsdam State in Chapter I, "A General Mathematical Science Program").

Introductory Courses

The foundation for a computer science component in a mathematics department is a one-year introductory sequence. Courses CS1 and CS2, proposed in the Association of Computing Machinery Curriculum 78 (see last section of this chapter), are excellent models for this year sequence. The Subpanel on Computer Science endorses the objectives of these two courses, and recommends that all mathematical sciences majors should be required to take the first course and strongly encouraged to take the second course in this sequence. If the second course is not required, substantial use of computers should be an integral part of other mathematical sciences courses.

The primary emphasis in the first course should be on:

- Problem solving methods and algorithmic design and analysis,
- Implementing problem solutions in a widely used higher-level programming language,
- Techniques of good programming style, and
- Proper documentation.

Lectures should include brief surveys of the history of computing, hardware and architecture, and operating systems.

The second course should include at least one major project. The course should cover topics such as recursive programming, pointers, stacks, queues, linked lists, string processing, searching and sorting techniques. The concepts of data abstraction and algorithmic complexity should be introduced. Proofs of correctness may also be discussed.

Good design and style in programming should be emphasized throughout both courses: the use of identifiers to indicate scope, modularity, appropriate choice of identifiers, good error recovery procedures, checks for integrity of input, and appropriate commentary and documentation. Of course, efficient algorithms and coding should also be stressed. There is a strong tendency among students to worry only about whether their programs run correctly. Through class lectures and careful grading of programming assignments, the instructor must teach the students the importance of good design, style, and efficiency in programming.

A source of useful commentary about introductory computer science courses is the *SIGCSE (Special Interest Group on Computer Science Education) Bulletin.* The bulletin is published quarterly, and issue #1 each year, which contains papers presented at the SIGCSE annual meeting, is especially valuable.

Most introductory texts have many sample projects. In addition, the following three texts are good general sources of computer projects.

1. Bennett, William R., *Scientific and Engineering Problem-Solving with the Computer,* Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
2. Gruenberger, Fred and Jaffray, G., *Problems for Computer Solution,* John Wiley & Sons, New York, 1965.
3. Wetherall, Charles, *Etudes for Programmers,* Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

Mathematicians teaching introductory computer science often emphasize numerical computation in programming assignments. At the introductory level, the computer science issues involved in numerical computation are quite simple. Assignments requiring symbolic manipulation and data organization present more substantive programming problems and, in general, require more thought. The following is a sample assignment that could be given late in the first course:

> Write a program which obtains a five-card poker hand from some source (terminal, input deck, or file), prints the hand in a reasonably well-formatted style, and determines whether or not the hand contains a pair, three of a kind, a straight, a full house, etc.

Intermediate Courses

Intermediate-level computer science courses building on CS1 and CS2 should address basic underlying issues in computer science. In describing computer science in the first two years, the ACM Curriculum 78 report states that the student should be given "a thorough grounding in the implementation of algorithms in programming languages which operate on data structures in the environment of hardware." Thus these courses should develop general topics about algorithms, concepts in programming languages, data structures, and computer hardware.

The intermediate-level courses should be taught by a computer scientist, that is, by an individual who has significant graduate-level training in computer science (see below).

The Subpanel on Computer Science, in concurrence with ACM curriculum groups, strongly rejects the idea of a set of courses that each address a specific programming language, e.g., a sequence of advanced FORTRAN, COBOL, RPG, and APL. The argument for such a sequence is usually based on the employability of students completing it. If indeed this argument is valid, and there is some question about that, it is a short range benefit. Students completing such a sequence will soon find that the lack of underlying concepts will put them at a severe disadvantage. However, it may be acceptable, resources permitting, to have one "vocational" elective course that studies a second higher-level language such as COBOL. Of course, it is also natural to discuss new programming languages in several intermediate (and advanced) computer science courses. However, the new language would not be the focus of the course, but rather a tool used in learning and illustrating fundamental concepts.

The role of numerical and computational mathematics in computer science has diminished in recent years. While the ACM Curriculum 68 treated numerical analysis as part of core computer science, today numerical mathematics is considered by most computer scientists to be simply another mathematical sciences field that has overlap with computer science. Numerical mathematics is very important in a mathematical sciences major, but it is not a part of the computer science component.

Following the CS1 and CS2 courses, the ACM Curriculum 78 specifies six additional courses in core computer science.

CS3 Introduction to Computer Systems
CS4 Introduction to Computer Organization
CS5 Introduction to File Processing
CS6 Operating Systems and Computer Architecture
CS7 Data Structures and Algorithm Analysis
CS8 Organization of Programming Languages

The syllabi of these courses are given at the end of this chapter. Ideally, all six of these courses would be offered. A concentration or a minor in computer science would commonly consist of CS1 and CS2, followed by two of CS3, CS4, and CS5, and two of CS6, CS7, and CS8. For the purposes of a mathematical sciences program, it may be justified to place more emphasis on the software oriented areas. This would imply, if there was difficulty in offering all six courses, that CS3, CS5, CS7, and CS8 would be most useful. Then CS3, CS5, CS7, and CS8 would be offered once a year, and CS4 and CS6 offered as topics courses every other year.

At many schools, it may not be feasible to offer at least four of these intermediate courses in computer science on a regular basis. Then one can combine parts of these intermediate courses to provide a significant offering in two courses above CS1 and CS2. In this case, only two computer science courses, one elementary and one intermediate, would be offered each semester. One approach would be to combine topics from CS5 and CS7 into one course, and topics from CS3, CS4, and CS6 into the other. This would yield two courses with the following sort of syllabi (for more details about these topics, see the ACM Curriculum 78 syllabi at the end of this chapter):

A1. Algorithms for Data Manipulation
1. Algorithm design and development illustrated in areas of sorting and research (25%)
2. Data structure implementation (30%)
3. Access methods (25%)
4. Systems design (15%)
5. Exams (5%)

A2. Computer Structures
1. Basic logic design (15%)
2. Number representation and arithmetic (10%)
3. Assembly systems (35%)
4. Program segmentation and linkage (15%)
5. Memory management (10%)
6. Computer systems structure (10%)
7. Exams (5%)

This approach focuses on data structures and software issues that relate to operating systems. An alternative approach could concentrate on programming languages and algorithms involved in computer systems performance. This theme could be realized by combining topics in CS3, CS5, and CS8 into one course, and topics in CS4, CS6, and CS7 into the other course. This would yield two courses with the following syllabi:

B1. Language Types and Structures
1. Assembly systems (25%)
2. Program segmentation and linkage (15%)
3. Language definition structure (10%)
4. Data types and structures (15%)
5. Control structures and data flow (20%)
6. Access methods (10%)
7. Exams (5%)

B2. Algorithms for Computer Systems
1. Basic logic design (20%)
2. Algorithm design and analysis (20%)
3. Procedure activation algorithms (15%)
4. Memory management (15%)
5. Process management (15%)
6. Systems design (10%)
7. Exams (5%)

It is important to note that an individual wishing to go on from these courses to advanced work in computer science may have to make up, as deficiencies, areas in core computer science that are not represented in these condensed pairs of courses.

Concentrations and Minors

A computer science concentration in a college mathematics department can be defined as an option within a mathematical sciences major or as a "stand-alone" minor. A computer science minor should consist of about six courses, ACM Curriculum 78 courses CS1 and CS2 plus four intermediate courses.

A computer science concentration within a mathematical sciences major has three components:

A. Mathematics: 5-plus courses;
B. Computer Science: 4-6 courses;
C. Applied Mathematics: 3-plus courses.

A. The mathematics component would include the three semester freshman-sophomore "calculus sequence" plus linear algebra. As recommended in Chapter I, "A General Mathematical Sciences Program," any mathematical sciences major should contain upper-level course work of a theoretical nature, typically algebra or advanced calculus. In a major with a computer science concentration, algebra is the natural area. Specifically, the applied algebra course given in Chaper I would be excellent for the computer science concentration. The course's syllabus incorporates most of the topics of the ACM 78 discrete mathematics course (required of computer science majors). A small department could offer applied algebra and standard abstract algebra courses in alternate years. Logic and automata theory are attractive electives in the mathematics component if a mathematics department wishes to focus on more theoretical aspects of computer science.

It should be noted that several computer science educators have questioned the reliance on calculus as the basic mathematics for future computer scientists; ACM Curriculum 78, for instance, requires a (freshman) year of calculus. They advocate a mathematics component based on discrete mathematics with only one semester of calculus (taught, say, in the junior year). See A. Ralston and M. Shaw, "Curriculum 78—Is Computer Science Really that Unmathematical?", *Communications ACM* 23 (1980), pp. 67-70.

B. The computer science component would include ACM Curriculum 78 courses CS1 and CS2 plus two to four intermediate courses, as described in the preceding section. The syllabi of ACM Curriculum 78 core courses are given at the end of this chapter.

C. The applied mathematics component should include a course in numerical analysis and a course in probability and statistics. The third applied mathematics course would be discrete methods, which would cover the combinatorial material in the ACM Curriculum 78 discrete mathematics course in greater depth, including operations-research-related graph modeling (see Chapter I for a full description of this course). The CUPM Mathematical Sciences Program panel recommends that all mathematics departments should offer a discrete methods course. Other good courses for the applied mathematics component are ordinarily differential equations, mathematical modeling, and operations research. The 1971 CUPM *Report on Computational Mathematics* describes courses in computational models, in combinatorial computation, and in differential equations with numerical methods; these courses combine topics from a variety of mathematical sciences and computer science courses and hence are particularly attractive to small departments.

In either the computer science concentration or minor, all six computer science courses are needed for future graduate study in computer science. Incoming graduate students with less preparation are commonly required to make up undergraduate course deficiencies.

Faculty Training

For the foreseeable future, the dominant factor affecting computer science instruction at all institutions, but particularly at smaller colleges and universities, will be the extreme shortage of qualified computer scientists in academe. At smaller colleges and universities it may therefore be effectively impossible to hire a computer scientist to teach core computer science courses. Among the possible solutions to this problem are:

1. Using adjunct faculty to teach computer science courses.
2. Using existing (non-computer science) faculty to teach computer science courses.

The first solution is acceptable for some courses. Although one cannot build a program with adjunct faculty and although staffing courses with adjunct faculty is never as desirable as using full-time faculty (e.g., student advising is a particular problem), this is a feasible way to get computer science courses taught when such faculty exist in the local community. However, since so many smaller colleges are located away from the metropolitan areas where most technical and scientific employers of such adjunct faculty are found, this solution will not be useful to most smaller institutions.

A crucial point that must be emphasized when using existing non-computer science faculty (i.e., mathematicians) to teach computer science courses is that computer science cannot be treated like most other new mathematics course topics which mathematicians will (quickly) learn as they teach it. Mathematicians untrained in computer science are very likely to teach computer science badly, hurting both the students and the mathematics department's reputation. Therefore, if a current mathematics faculty member is to be used to teach computer science, especially beyond the first course, he or she must first acquire some formal education in computer science.

The most plausible approach to such computer science training is through some program of released time. The pertinent questions about the training are: how long? where? and how financed?

Assuming that the mathematician who is to be trained is, at most, familiar with programming in a high-level language, then full-time study for one year is the minimum period needed to acquire the background, knowledge, and experience necessary to teach several of the intermediate-level core computer science courses. Since one year is also the maximum period which would be administratively or financially feasible, this should be viewed as the canonical period for faculty training in computer science. Part-time study over a longer period or a succession of summers can also be considered. However, both because the needs to train faculty in computer science are pressing and because intermittent study is almost always less effective than continuous study, at least one faculty member in a mathematics department should have completed a one-year program of full-time study in computer science.

The most logical place at which to study computer science for the purpose of becoming able to teach it is at a university with undergraduate and graduate (preferably Ph.D.) programs in computer science. Although there are exceptions, the current level of computer science instruction in American colleges and universities is so uneven that only at such institutions can one be reasonably assured of an atmosphere in which there will be the necessary broad understanding of the principles of computer science. Such an atmosphere is particularly important for an academic mathematician preparing to teach the subject.

Another possibility which should be mentioned is for the faculty member to spend one year at one of those (relatively few) major industrial firms with good in-house training programs in computer science. An additional attraction to this idea is that it might be possible to arrange an exchange in which a member of the firm taught at the college for a year.

Methods of financing such a program of faculty training in computer science are fairly obvious:

1. Through released time at full pay from the mathematician's home institution.
2. Through grants from current, and hopefully new, federal programs; officials of both the MAA and ACM are currently pressing NSF to provide more funds for this purpose.
3. Through grants from private foundations; individual institutions and departments may be more effective than professional associations in obtaining such private funds.
4. Through corporate sponsorship of participation in in-house training programs or academic-corporate exchanges.

Computer Facilities

Facilities to support computing in mathematical sciences instruction can be provided in a variety of ways, ranging from one large centrally administered system to many small personal computing devices. The suitability of a particular means depends not only upon its intended applications, but also upon factors such as cost, ease of use, and local politics. At present, computing services in most colleges and universities are provided by a large centralized facility, the Computing Center. Growing numbers of institutions, however, are beginning to decentralize computing on campus. Three current modes of providing service are discussed below:

- Centralized facilities
- Departmental computers
- Personal computers.

There is a fourth mode that is primarily a form of access to centralized or departmental computers:

- Terminals

The second half of this section discusses the cost and ease of implementation of various applications with different types of computing facilities.

It should be noted that it is possible for an institution to form a consortium with nearby schools to operate a common central computing facility or to buy time (and services) from commercial computing centers. This option allows an institution to have a mix of computing, using large computers for problems requiring great speed or memory size, such as "number crunching," and smaller computers for student programs and other instructional purposes.

CENTRALIZED FACILITIES

Historically, so-called "economies of scale" encouraged the development of increasingly larger computers; and of increasingly larger organizations to administer them. Such computer systems are capable of providing a great variety of services with a low cost for each service. In addition, the organizations which administer these systems can play an important role in developing and supporting instructional uses of computing on campus.

On the other hand, the very size of such facilities and the organizations that administer them create certain problems. First, large systems have a high unit cost, in the range of half a million to several million dollars; replacing or enhancing such a system involves a major administrative decision. Second, instructional users of such systems must often compete with other powerful and better-financed constituencies; either separate facilities are needed to reduce competition among instructional, research, and administrative uses of the computer, or policies are needed to allocate the services provided by a single facility. And third, large organizations can be bureaucratic and inflexible.

DEPARTMENTAL COMPUTERS

For the last ten years minicomputers have provided an alternative to a large centralized facility. Lower unit costs (around $100,000 or less) and the possibility of local control have made it attractive for academic and administrative departments to acquire facilities of their own. Such facilities can be tailored to a department's needs and can provide almost as many services as a large centralized system.

Minicomputers, however, are not necessarily the answer to every department's computing needs. First, there is the question of which services they will provide. Second, there are hidden costs associated with administering any computer facility: personnel are needed to operate and maintain the facility and to provide technical assistance to users. Small departments run the risk of diverting attention from their primary task of teaching mathematics to the subsidiary task of managing such an enterprise. One way to deal with such hidden costs is for departments to contract with a central campus organization to manage their facilities. Third, there are inconveniences for students faced with using, and first learning to use, several different departmental systems. Of course, this difficulty can be overcome by requiring departments to purchase compatible systems and by interconnecting all systems.

Many academic computing specialists expect interconnected departmental computers to become the dominant means of academic computing in the next decade.

PERSONAL COMPUTERS

The recent development of personal microcomputers provides another alternative for instructional computing. Very low unit costs (one or two thousand dollars) make computing possible for departments otherwise unable to afford or gain authorization for large facilities. Microcomputer facilities suffer from many of the same problems as minicomputer facilities. In addition, microcomputers are limited in the services they provide, are slower than their large competitors, and may not be designed for rugged use by large groups of students. Still they can prove quite adequate for elementary applications. Further, by being less intimidating and more exciting than larger computers, they can play a role in overcoming a student's "computer anxiety."

TERMINALS

Terminals are used for remote, interactive access to large computers. Some have small memories and primitive editing capabilities. Departments often have a greater choice in selecting terminals to connect to computer systems than they do in selecting the systems themselves. Cost, speed, and durability are primary factors influencing the selection of a terminal. By these criteria, video terminals are preferable. The availability of graphical output and local editing features are other factors to consider when choosing terminals. Hard-copy (printing) terminals are more expensive and tend to be slower than video terminals, but they do provide users with a permanent record of their work, and so some printing terminals are necessary (medium or high speed printers can be used in conjunction with video terminals to provide this record). Video terminals may also be used in conjunction with television monitors to provide classroom displays of computer output. For such output to be visible in a large classroom, either many monitors must be provided or the video terminals employed must use larger, and hence fewer, characters in their display.

Applications

The suitability of a particular computing facility depends most upon its intended applications. The rest of this section discusses the most common academic uses of computers and how well different types of computing facilities serve these uses.

INTRODUCTORY PROGRAMMING

Any of the three types of facilities can serve as a vehicle for teaching beginners to program and for introducing computational examples into elementary mathematics courses. Such uses typically involve large numbers of students writing relatively simple programs. Larger facilities tend to provide a greater choice of programming languages, although modern languages such as PASCAL and PL/I are becoming increasingly available even on microcomputers. Larger machines tend to be faster also; even though use of such machines is shared, students will find that they process simple programs much faster than microcomputers. Costs, however, tend to be roughly equal for simple interactive computing on the three types of facilities—around $2.00 per hour. These costs can be reduced significantly by using larger machines in a noninteractive, batch-processing mode. This mode of use, while predominant in the past, is becoming less popular as minicomputers and microcomputers make a more responsive computing environment available and affordable.

ADVANCED PROGRAMMING

Advanced programming is more distinguished from introductory programming in its requirements for more sophisticated languages and for facilities to handle large programs. Microcomputers at present do not meet these requirements; the languages they provide are quite restrictive, and large programs exceed their capacity. Execution times and costs for large programs tend to be lowest on large machines under batch processing, but minicomputers are becoming competitive both in price and speed.

PROGRAM DEVELOPMENT AND MAINTENANCE

Program development is influenced heavily by the computing environment in which it occurs. Convenient interactive editing capabilities accelerate the task of writing and correcting a program; microcomputers, with almost instantaneous response, do a particularly good job of editing. Facilities for file storage enable program development to be spread over several sessions. Large machines provide less expensive storage and much faster retrieval of information; they also facilitate sharing programs among users and provide centralized backup. Microcomputer facilities can distribute the costs of file storage by requiring users to purchase individual floppy disks, but unless a centralized store is provided through a network, sharing information can be difficult.

GRAPHICS

One of the primary attractions of personal microcomputers is their ability to generate graphic displays and to enable users to interact with these displays. Larger systems, unless specifically tailored to graphic applications, tend to have primitive graphic facilities at best.

APPLICATION PACKAGES

Application packages available for various machines provide aids for numerical and symbolic computations. Typical areas of application include statistics, linear programming, numerical solution of differential equations, and algebraic formula manipulation. Such packages are more widely available on larger machines. Large computations often require an unacceptably long time on microcomputers (several hours) and may exceed the memory size of small computers.

MISCELLANEOUS APPLICATIONS

Word processing systems facilitate production of course notes, research papers, and term papers. If good word processing facilities are available, they are likely to quickly generate heavy faculty use. Simple word processing software is available for personal computers, but a minicomputer (or powerful $5,000-plus microcomputer) is needed for good mathematically-oriented word processing software, such as the UNIX system. Large computers often have poor word processing capabilities.

Data base systems are of more use in the social sciences than in the mathematical sciences, but can be used to provide real data for analysis in statistics courses. Such systems require a centralized file store on a larger computer.

Real-time data acquisition is of interest in the natural sciences. They can also be used to provide real data for mathematical analysis. Dedicated microcomputers are better suited to laboratory instrumentation than are shared machines.

ACM Curriculum 78

The following computer science course syllabi are reproduced from the ACM Curriculum 78 Report in *Communications of ACM,* March 1979, pp. 147-166. (Copyright 1979, Association for Computing Machinery, Inc.) They provide eight core courses for a computer science major.

CS1. Computer Programming I

OBJECTIVES:

- To introduce problem solving methods and algorithm development;
- To teach a high-level programming language that is widely used; and
- To teach how to design, code, debug, and document programs using techniques of good programming style.

COURSE OUTLINE:

The material on a high-level programming language and on algorithm development can be taught best as an integrated whole. Thus the topics should not be covered sequentially. The emphasis of the course is on the techniques of algorithm development and programming with style. Neither esoteric features of a programming language nor other aspects of computers should be allowed to interfere with that purpose.

TOPICS:

A. *Computer Organization.* An overview identifying components and their functions, machine and assembly languages. (5%)

B. *Programming Language and Programming.* Representation of integers, real, characters, instructions. Data types, constants, variables. Arithmetic expression. Assignment statement. Logical expression. Sequencing, alternation, and iteration. Arrays. Subprograms and parameters. Simple I/O. Programming projects utilizing concepts and emphasizing good programming style. (45%)

C. *Algorithm Development.* Techniques of problem solving. Flowcharting. Stepwise refinement. Simple numerical examples. Algorithms for searching (e.g., linear, binary), sorting (e.g., exchange, insertion), merging of ordered lists. Examples taken from such areas as business applications involving data manipulation, and simulations involving games. (45%)

D. *Examinations.* (5%)

CS2. Computer Programming II

OBJECTIVES:

- To continue the development of discipline in program design, in style and expression, in debugging and testing, especially for larger programs;
- To introduce algorithmic analysis; and
- To introduce basic aspects of string processing, recursion, internal search/sort methods and simple data structures.

PREREQUISITE: CS 1.

COURSE OUTLINE:

The topics in this outline should be introduced as needed in the context of one or more projects involving larger programs. The instructor may choose to begin with the statement of a sizable project, then utilize structured programming techniques to develop a number of small projects each of which involves string processing, recursion, searching and sorting, or data structures. The emphasis on good programming style, expression, and documentation, begun in CS1, should be continued. In order to do this effectively, it may be necessary to introduce a second language (especially if a language like Fortran is used in CS1). In that case, details of the language should be included in the outline. Analysis of algorithms should be introduced, but at this level such analysis should be given by the instructor to the student.

Consideration should be given to the implementation of programming projects by organizing students into programming teams. This technique is essential in advanced level courses and should be attempted as early as possible in the curriculum. If large class size makes such an approach impractical, every effort should be made to have each student's programs read and critiqued by another student.

TOPICS:

A. *Review.* Principles of good programming style, expression, and documentation. Details of a second language if appropriate. (15%)
B. *Structured Programming Concepts.* Control flow. Invariant relation of a loop. Stepwise refinement of both statements and data structures, or top-down programming. (40%)
C. *Debugging and Testing.* (10%)
D. *String Processing.* Concatenation. Substrings. Matching. (5%)
E. *Internal Searching and Sorting.* Methods such as binary, radix, Shell, quicksort, merge sort. Hash coding. (10%)
F. *Data Structures.* Linear allocation (e.g., stacks, queues, deques) and linked allocation (e.g., simple linked lists). (10%)
G. *Recursion.* (5%)
H. *Examinations.* (5%)

CS3. Introduction to Computer Systems

OBJECTIVES:

- To provide basic concepts of computer systems;
- To introduce computer architecture; and
- To teach an assembly language.

PREREQUISITE: CS 2.

COURSE OUTLINE:

The extent to which each topic is discussed and the ordering of topics depends on the facilities available and the nature and orientation of CS4 described below. Enough assembly language details should be covered and projects assigned so that the student gains experience in programming a specific computer. However, concepts and techniques that apply to a broad range of computers should be emphasized. Programming methods that are developed in CS1 and CS2 should also be utilized in this course.

TOPICS:

A. *Computer Structure and Machine Language.* Memory, control, processing and I/O units. Registers, principal machine instruction types and their formats. Character representation. Program control. Fetch-execute cycle. Timing. I/O Operations. (15%)
B. *Assembly Language.* Mnemonic operations. Symbolic addresses. Assembler concepts and instruction format. Data-word definition. Literals. Location counter. Error flags and messages. Implementation of high-level language constructs. (30%)
C. *Addressing Techniques.* Indexing. Indirect Addressing. Absolute and relative addressing. (5%)
D. *Macros.* Definition. Call. Parameters. Expansion. Nesting. Conditional assembly. (10%)
E. *File I/O.* Basic physical characteristics of I/O and auxiliary storage devices. File control system. I/O specification statements and device handlers. Data handling, including buffering and blocking. (5%)
F. *Program Segmentation and Linkage.* Subroutines. Coroutines. Recursive and re-entrant routines. (20%)
G. *Assembler Construction.* One-pass and two-pass assemblers. Relocation. Relocatable loaders. (5%)
H. *Interpretive Routines.* Simulators. Trace. (5%)
I. *Examinations.* (5%)

CS4. Introduction to Computer Organization

OBJECTIVES:

- To introduce the organization and structuring of the major hardware components of computers;
- To understand the mechanics of information transfer and control within a digital computer system; and
- To provide the fundamentals of logic design.

PREREQUISITE: CS 2.

COURSE OUTLINE:

The three main categories in the outline, namely computer architecture, arithmetic, and basic logic design, should be interwoven throughout the course rather than taught sequentially. The first two of these areas may be covered, at least in part, in CS3 and the amount of material included in this course will depend on how the topics are divided between the two courses. The logic design part of the outline is specific and essential to this course. The functional, logic design level is emphasized rather than circuit details which are more appropriate in engineering curricula. The functional level provides the student with an understanding of the mechanics of information transfer and control within the computer system. Although much of the course material can and should be presented in a form that is independent of any particular technology, it is recommended that an actual simple minicomputer or microcomputer system be studied. A supplemental laboratory is appropriate for that purpose.

TOPICS:

A. *Basic Logic Design.* Representation of both data and control information by digital (binary) signals. Logic properties of elemental devices for processing (gates) and storing (flipflops) information. Description by truth tables, Boolean functions and timing diagrams. Analysis and synthesis of combinatorial networks of commonly used gate types. Parallel and serial registers. Analysis and synthesis of simple synchronous control mechanisms; data and address buses; addressing and accessing methods; memory segmentation. Practical methods of timing pulse generation. (25%)
B. *Coding.* Commonly used codes (e.g., BCD, ASCII). Parity generation and detection. Encoders, decoders, code converters. (5%)
C. *Number Representation and Arithmetic.* Binary number representation, unsigned addition and subtraction. One's and two's complement, signed magnitude and excess radix number representations and their pros and cons for implementing elementary arithmetic for BCD and excess-3 representations. (10%)
D. *Computer Architecture.* Functions of, and communication between, large-scale components of a computer system. Hardware implementation and sequencing of instruction fetch, address construction, and instruction execution. Data flow and control block diagrams of a simple processor. Concept of microprogram and analogy with software. Properties of simple I/O devices and their controllers, synchronous control, interrupts. Modes of communications with processors. (35%)
E. *Example.* Study of an actual, simple minicomputer or microcomputer system. (20%)
F. *Examinations.* (5%)

CS5. Introduction to File Processing

OBJECTIVES:

- To introduce concepts and techniques of structuring data on bulk storage devices;
- To provide experience in the use of bulk storage devices; and
- To provide the foundation for applications of data structures and file processing techniques.

PREREQUISITE: CS 2.

COURSE OUTLINE:

The emphasis given to topics in this outline will vary depending on the computer facilities available to students. Programming projects should be assigned to give students experience in file processing. Characteristics and utilization of a variety of storage devices should be covered even though some of the devices are not part of the computer system that is used. Algorithmic analysis and programming techniques developed in CS2 should be utilized.

TOPICS:

A. *File Processing Environment.* Definitions of record, file, blocking, compaction, database. Overview of database management system. (5%)
B. *Sequential Access.* Physical characteristics of sequential media (tape, cards, etc.). External sort/merge algorithms. File manipulation techniques for updating, deleting and inserting records in sequential files. (30%)
C. *Data Structures.* Algorithms for manipulating linked lists. Binary, B-trees, B^*-trees, and AVL trees. Algorithms for transversing and balancing trees. Basic concepts of networks (plex structures). (20%)
D. *Random Access.* Physical characteristics of disk, drum, and other bulk storage devices. Algorithms and techniques for implementing inverted lists, multilist, indexed sequential, and hierarchical structures. (35%)
E. *File I/O.* File control systems and utility routines, I/O specification statements for allocating space and cataloging files. (5%)
F. *Examinations.* (5%)

CS6. Operating Systems & Comp. Architecture

OBJECTIVES:

- To develop an understanding of the organization and architecture of computer systems at the register-transfer and programming levels of system description;

- To introduce the major concept areas of operating systems principles;
- To teach the inter-relationships between the operating system and the architecture of computer systems.

PREREQUISITES: CS3 AND CS4.

COURSE OUTLINE:

This course should emphasize concepts rather than case studies. Subtleties do exist, however, in operating systems that do not readily follow from concepts alone. It is recommended that a laboratory requiring hands-on experience be included with this course.

The laboratory for the course would ideally use a small computer where students could actually implement sections of operating systems and have them fail without serious consequences to other users. This system should have, at a minimum, a CPU, memory, disk or tape, and some terminal device such as a teletype of CRT. The second best choice for the laboratory experience would be a simulated system running on a larger machine.

The course material should be liberally sprinkled with examples of operating system segments implemented on particular computer system architectures. The interdependence of operating systems and architecture should be clearly delineated. Integrating these subjects at an early stage in the curriculum is particularly important because the effects of computer architecture on systems software has long been recognized. Also, modern systems combine the design of operating systems and the architecture.

TOPICS:

A. *Review.* Instruction sets. I/O and interrupt structure. Addressing schemes. Microprogramming. (10%)
B. *Dynamic Procedure Activation.* Procedure activation and deactivation on a stack, including dynamic storage allocation, passing value and reference parameters, establishing new local environments, addressing mechanics for accessing parameters (e.g., displays, relative addressing in the stack). Implementing non-local references. Re-entrant programs. Implementation on register machines. (15%)
C. *System Structure.* Design methodologies such as level, abstract data types, monitors, kernels, nuclei, networks of operating system modules. Proving correctness. (10%)
D. *Evaluation.* Elementary queueing, network models of systems, bottlenecks, program behavior, and statistical analysis. (15%)
E. *Memory Management.* Characteristics of the hierarchy of storage media, virtual memory, paging, segmentation. Policies and mechanisms for efficiency of mapping operations and storage utilization. Memory protection. Multiprogramming. Problems of auxiliary memory. (20%)
F. *Process Management.* Asynchronous processes. Using interrupt hardware to trigger software procedure calls. Process stateword and automatic SWITCH instructions. Semaphores. Ready lists. Implementing a simple scheduler. Examples of process control problems such as deadlock, product/consumers, readers/writers. (20%)
G. *Recovery Procedures.* Techniques of automatic and manual recovery in the event of system failures. (5%)
H. *Examinations.* (5%)

CS7. Data Structures and Algorithm Analysis

OBJECTIVES:

- To apply analysis and design techniques to non-numeric algorithms which act on data structures;
- To utilize algorithmic analysis and design criteria in the selection of methods for data manipulation in the environment of a database management system.

PREREQUSITES: CS5.

COURSE OUTLINE:

The material in this outline could be covered sequentially in a course. It is designed to build on the foundation established in the elementary material, particularly on that material which involves algorithm development and data structures and file processing. The practical approach in the earlier material should be made more rigorous in this course through the use of techniques for the analysis and design of efficient algorithms. The results of this more formal study should then be incorporated into data management system design decisions. This involves differentiating between theoretical or experimental results for individual methods and the results which might actually be achieved in systems which integrate a variety of methods and data structures. Thus, database management systems provide the applications environment for topics discussed in the course.

Projects and assignments should involve implementation of theoretical results. This suggests an alternative way of covering the material in the course; namely, to treat concepts, algorithms, and analysis in class and deal with their impact on system design in assignments. Of course, some in-class discussions of this impact would occur, but at various times throughout the course rather than concentrated at the end.

TOPICS:

A. *Review.* Basic data structures such as stacks, queues, lists, trees. Algorithms for their implementation. (10%)
B. *Graphs.* Definition, terminology, and property (e.g., connectivity). Algorithms for finding paths and spanning trees. (15%)
C. *Algorithms Design and Analysis.* Basic techniques of design and analysis of efficient algorithms for internal and external sorting/merging/searching. Intuitive notions of complexity (e.g., NP-hard problems). (30%)
D. *Memory Management.* Hashing. Algorithms for dynamic storage allocation (e.g., buddy system, boundary-tag), garbage collection and compaction. (15%)
E. *System Design.* Integration of data structures, sort/merge/search methods (internal and external) and memory media into a simple database management system. Accessing methods. Effects on run time, costs, efficiency. (25%)
F. *Examinations.* (5%)

CS8. Organization of Programming Languages

OBJECTIVES:

- To develop an understanding of the organization of programming languages, especially the run-time behavior of programs;
- To introduce the formal study of programming language specification and analysis;
- To continue the development of problem solution and programming skills introduced in the elementary level material.

PREREQUSITES: CS2; RECOMMENDED: CS3, CS5.

COURSE OUTLINE:

This is an applied course in programming language constructs emphasizing the run-time behavior of programs. It should provide appropriate background for advanced level courses involving formal and theoretical aspects of programming languages and/or the compilation process.

The material in this outline is not intended to be covered sequentially. Instead, programming languages could be specified and analyzed one at a time in terms of their features and limitations based on their run-time environments. Alternatively, desirable specification of programming languages could be discussed and then exemplified by citing their implementations in various languages. In either case, programming exercises in each language should be assigned to emphasize the implementations of language features.

TOPICS:

A. *Language Definition Structure.* Formal language concepts including syntax and basic characteristics of grammars, especially finite state, context-free, and ambiguous. Backus-Naur Form. A language such as Algol as an example. (15%)
B. *Data Types and Structures.* Review of basic data types, including lists and trees. Constructs for specifying and manipulating data types. Language features affecting static and dynamic data storage management. (10%)
C. *Control Structures and Data Flow.* Programming language constructs for specifying program control and data transfer, including DO ...FOR, DO ...WHILE, REPEAT ...UNTIL, BREAK, subroutines, procedures, block structures, and interrupts. Decision tables, recursion. Relationship with good programming style should be emphasized. (15%)
D. *Run-time Consideration.* The effects of run-time environment and binding time on various features of programming languages. (25%)
E. *Interpretative Languages.* Compilation vs. interpretation. String processing with language features such as those available in SNOBOL 4. Vector processing with language features such as those available in SPL. (20%)
F. *Lexical Analysis and Parsing.* An introduction to lexical analysis including scanning, finite state acceptors and symbol tables. An introduction to parsing and compilers including push-down acceptors, top-down and bottom-up parsing. (10%)
G. *Examinations.* (5%)

Subpanel Members

ALAN TUCKER, CHAIR, SUNY-Stony Brook.
GERALD ENGEL, Christopher Newport College.
STEPHEN GARLAND, Dartmouth College.
BERT MENDELSON, Smith College.
ANTHONY RALSTON, SUNY-Buffalo.

Modeling and Operations Research

This chapter contains the report of the Subpanel on Modeling and Operations Research of the CUPM Panel on a General Mathematical Sciences Program, reprinted with minor changes from Chapter V of the 1981 CUPM report entitled RECOMMENDATIONS FOR A GENERAL MATHEMATICAL SCIENCES PROGRAM.

Experience in Applications

This chapter is concerned with mathematical modeling and associated interactive and experience-oriented approaches to teaching mathematical sciences. Mathematical modeling attempts to involve students in the more creative and early design aspects of problem formulation, as well as provide them with a more complete exposure to how mathematics interfaces with other activities in solving problems arising outside of mathematics itself. Model building is a major ingredient of operations research and the contemporary uses of mathematics in the social, life and decision sciences. In addition to being important in their own right, these newer uses of mathematics provide a rich source of suitable materials for interaction and modeling which complement the many modern and classical applications of mathematics in the physical sciences and engineering.

This chapter is intended to assist mathematics faculty in implementing the main panel's recommendation that mathematical sciences majors should have substantial experience with mathematical modeling. Subsequent sections discuss the modeling process in some detail; provide specific suggestions for conducting student projects, applications-experience-related courses and other such programs, along with general recommendations concerning modeling courses at different levels; explain the field of operations research and the requirements for graduate study. The final two sections present outlines for four courses in operations research and modeling, and a compendium of resources and references for modeling courses.

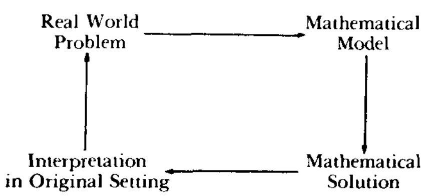
Learning and doing mathematics is a rather individualized and personal activity. The typical classroom lecture in which students are passive spectators has obvious limitations. Students need supervised hands-on experience in problem solving and constructing rigorous proofs. A large variety of alternate teaching techniques and special programs have been developed in attempts to meet this need. These include problem solving approaches using materials from pure and applied mathematics, such as the methods of G. Pólya and R.L. Moore. Problem solving teams for competitions such as the Putnam contest and special departmental practica exist in many colleges. Special courses or seminars on modeling, case studies, and project-oriented activity are becoming more common, as are mathematics clinics and consulting bureaus. Co-op and work-study programs, summer internships, and various other student exchanges have been successfully implemented at some institutions.

The Modeling Subpanel believes that applications and modeling should be included in a nontrivial way in most college-level mathematical sciences courses. Concern with applications has been an important historical force and a major cultural ingredient in the development of all mathematics. Further, the Modeling Subpanel strongly recommends that all mathematical sciences students should obtain first-hand experience with realistic applications of mathematics from the initial stage of model formulation through interpretation of solutions. This can be done in a project-oriented modeling course in one of the alternate out-of-class modes mentioned above. Such an experience yields insight into the place of mathematics in the larger realm of science. It provides an appreciation for the need for interdisciplinary interaction and the limits of specialization. It offers a chance for individuals to make use of their own intuition and creative abilities, to sense the great joy of personal accomplishment, and to develop the confidence to confront similar problems after graduation. Finally, such experience may assist students in choosing careers and fields for future study.

Mathematical Modeling

Modeling is a fundamental part of the general scientific method and is of primary importance in applied mathematics. A model is a simpler realization or an idealization of some more complex reality created for the purpose of gaining new knowledge about a real situation by investigating properties and implications of the model. Models may take many different forms, from physical miniatures to pure intellectual substitutes. Study of a model will hopefully provide understanding and new information about real phenomena which are too complex, excessively expensive, or impossible to analyze in their original setting.

We tend to take the amazing effectiveness of models for granted today. The reader should give a moment's thought to the following examples. One can learn a great deal about a proposed aircraft from wind tunnel experiments before building a costly prototype, and one can learn much about flying an existing airplane from a computer-aided cockpit simulator. Simple computer simulations can provide insights into the complex flow or queueing behavior of traffic in a transportation system. Theoretical studies about elementary particles have provided new insight into fundamental physical laws and have guided subatomic experimentation.



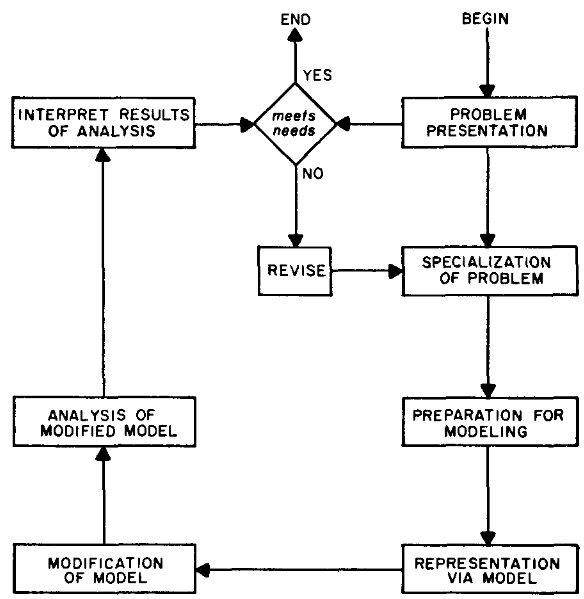
A Simple Model of Mathematical Modeling

Figure 1

The process of mathematical modeling can be simply represented with the diagram in Figure 1. One begins with a problem which arises more or less directly out of the "real world." One builds an abstract model for purposes of analysis, and this frequently takes a mathematical form. The model is solved in this abstract setting. The solution is then interpreted back into its original context. Finally, the analytical conclusions are compared with reality. If they fall short of matching the real situation, then modifications of the model may be called for, and one proceeds around this cycle again. One often proceeds back and forth within a cycle and makes successive iterations about this figure many times before arriving at a satisfactory representation of the real world.

The creation of new knowledge via this modeling route is at the heart of theoretical science and applied mathematics. We will use the word "modeling" to describe the complete progress illustrated in Figure 1. Frequently this term is used only for the model formulation step (the top arrow in the figure). A full discussion of the four steps in this modeling paradigm follow. Additional steps refining the modeling process are sometimes inserted; for example, see Figure 2.

First consider the downward pointing arrow on the right side from "mathematical model" to "mathematical solution." This is the deductive activity of finding solutions to well-formulated mathematical problems. It is usually the most logical, well-defined and straightforward part of modeling, although not necessarily the easiest. It is often the most immediately pleasing, elegant, and intellectual part. This "side" of the "modeling square" is the one covered best in standard applied mathematics courses. Unfortunately, most teaching of applied mathematics is confined to discussing just model-solving mathematical techniques, with superficial treatment of the other three sides of the square, whereas these other sides often involve much more creativity, interaction with other disciplines, and communication skills.



A Refined Model of Mathematical Modeling

Figure 2

The bottom arrow in Figure 1 is concerned with translating or explaining a purely mathematical result in terms of the original real world setting. This involves the need to communicate in a precise and lucid manner. (Inexperience in this skill, according to many employers, is a serious shortcoming in mathematics graduates). This aspect of a mathematical scientist's training should not be left to courses in other sciences or to on-the-job learning after graduation.

In describing the meaning of a mathematical solution one must take great care to be complete and honest. It is dangerous to discard quickly some mathematical solutions to a physical problem as extraneous or having no physical meaning; there have been too many historical incidences where "extraneous" solutions were of fundamental importance. Likewise, one should not select out just the one preconceived answer which the "boss" is looking for to support his or her position. A decision maker frequently does not want just one optimal solution, but desires to know a variety of "good" solutions and the range of reasonable options available from which to select.

There is an old adage to the effect that bosses do not act on quantitative recommendations unless they are communicated in a manner which makes them understandable to such decision makers. This communication can often be a difficult task because of the technical nature of the formulation and solution, and also because large quantities of data and extensive computation may need to be compressed to a manageable size for the layman to understand in a relatively limited time. If mathematical education gave more attention to this aspect of mathematical modeling, there might be wider recognition and visibility of mathematicians in society beyond the academic world!

A major step in real world modeling is to validate models critically and to check out solutions against the original phenomena and known results. This step, represented by the left upward arrow in Figure 1, may involve experimentation, verifying, and evaluating. Two major criteria for evaluating a model are simplicity and accuracy of prediction. Questions about the range of validity, sensitivity of parameters errors resulting from approximations, and such should be investigated. In many cases, a modeling project will simply confirm from another perspective properties that are already believed to be true. The real gain from modeling activity occurs when the modeling leads to discovery of new knowledge (which subsequently is confirmed by other methods).

Modern mathematics education rarely involves itself with this left hand side of the modeling process, except perhaps for an occasional "eyeballing" of an answer or in projects undertaken by a mathematics or statistics consulting clinic. By omitting this activity, mathematical education misses an opportunity to become involved with real-world decision making, judgmental inputs, the limitations of its mathematical tools, and other more human aspects of science, as well as the reward of witnessing the acceptance of a new theory.

Finally, consider the top arrow in Figure 1 which represents the heart of the modeling activity. The construction of an abstract model from a real situation is the really creative activity and an important component of all theoretical science. Building models involves translating into mathematics, maintaining the essential ingredients while filtering out a great amount of excess baggage, and arriving at realistic and manageable intellectual limitations. The three basic elements of a model are:

1. A logical mathematical structure such as calculus, probability, or game theory;
2. An appropriate interpretation of the variables in that structure in terms of the given problem; and
3. A characterization with the structure of all laws and constraints pertinent to the problem.

To build such a mental construct, one must conceptualize, idealize and identify properties precisely. A model builder must carefully balance the tradeoffs between coarse simplifications and unnecessary details—often the effects of such tradeoffs are not apparent until subsequent validation (three steps later in the modeling process).

This initial part of modeling is clearly the most essential and valuable part of the whole process. It is usually the most difficult part. Eddington said "I regard the introductory part of a theory as the most difficult, because we have to use our brains all of the time. Afterwards we can use mathematics." Model building is an art, and must be taught as such.

An Undergraduate Modeling Course

This section discusses various approaches to designing mathematical sciences courses concentrating on the modeling process. The resources listed at the end of this chapter contain a wealth of additional information on models, the modeling process and specific modeling courses as well as references to supplementary materials which the reader may find useful in course design.

Practitioners in the physical or social sciences or engineering have an instinctive feeling of what the modeling process is all about, even if they are not able to articulate it well. Modeling is an important part of their work-a-day activity. For the most part, however, they prefer to leave the analysis and structure of the modeling process itself to workers in other disciplines, like mathematics, or to philosophers of science who are trying to understand the abstract theories underlying these results and how scientists get their results.

How does one go about acquiring experience in real-world modeling? The wrong place to start is looking at big models in the scientific literature which are broad in scope and the epitome of their kind. Indeed, one could probably learn more about sculpting by looking at the pieces that Michelangelo discarded than by looking at the Pieta. The mathematical techniques with which one is familiar will be a primary limiting factor in understanding models. Another factor is that real-world problem areas have their own peculiar "empirical laws" and "principles" which are commonly known to specialists in an area but are not easily accessible to the casual reader.

Apprentice modelers need some help and guidance in selecting model areas for study which will build their modeling skill without discouraging progress. The ideal way to do this within the college curriculum is to begin the modeling process as early as possible in the student's career and reinforce modeling over the entire period of study. That is, the modeling process should be an integral part of the curriculum. Most mathematics departments, for a variety of reasons, are not prepared to give modeling such a major emphasis. For them, a more reasonable approach is to design a course specifically around the modeling process.

Efforts to emphasize the modeling process in undergraduate courses on a broad scale began in the 1960's and were promoted mainly by engineers, operations researchers and social scientists. Extensive discussions of modeling in mathematics courses developed later. The modeling process has been brought into the classroom in many ways but two particular approaches are worth describing in some detail.

First there is the case study approach in which the modeling process is described in a series of examples that are more-or-less self-contained. The examples selected by the instructor are designed to bring out the basic features of the modeling process as well as to inform the students about basic models within a discipline. An excellent early example is *You and Technology: A High School Case Study Text* developed by the engineering departments of the PCM Colleges (Chester, PA), edited by N. Damaskos and M. Smyth.

The second approach applies "hands-on" experience to problems that may only be vaguely described. This approach is sometimes called "open-ended" or "experiential," because it is not clear at the outset what kind of a model will be successful in analyzing a problem, or indeed whether a particular problem is well-posed in any sense. An interesting sidelight on this approach to teaching the modeling process is that the models proposed by students for a particular problem depend not only on the students' breadth of knowledge but, as much as anything else, on time constraints and computer (and other) resources available. Engineers popularized the experiential approach in the early sixties with the high school program *Man Made World,* mostly as a means of exposing students at an early stage to engineering as a profession (a text of the same name was written for this program by J. Truxal, et al., McGraw-Hill publisher).

A range of courses emphasizing the modeling process is clearly possible between the case study approach and the experiential approach.

It is important to note that the scope of the engineering approach to modeling is much broader than just the technical aspects of the problem at hand. In designing a solution to a problem, engineers must take into account time constraints and build into their models prescribed economic and other technical constraints as well as consideration of the impact of their design on society. Engineers do not build elaborate models to explain the fundamental workings of nature nor do they seek the best possible solution to a problem in the absence of the proposed application of that solution. In spite of these differences, there is obviously a large overlap between the engineering and mathematical approaches to modeling.

We now characterize the components of a modeling course in a way that readers should find useful in designing a course to fit their own local needs. The Table on pp. 46-47 organizes much of this information for easy reference. There are six basic aspects of teaching modeling that must be considered:

1. Prerequisites. For whom is the course intended?
2. Effort level. How long—a few weeks, a semester, a year?
3. Course format. Experiential or case study approach? Team or individual work? Instructor's role. Communication skills used.
4. Resources available. Computer system, remote access, good software packages (students should become familiar with using some major software package). Access to expertise in fields considered. Appropriate handouts to keep students progressing.
5. Source of problems. Real-world or contrived? Open-ended or can student answer all questions by looking them up in the literature?
6. Technical thrust. What technical areas should the course emphasize, or avoid? Continuous or discrete models? Deterministic or stochastic? Role of computer programming.

We now expand a little on two of these components, effort level and course format. The level of effort devoted to a modeling course can range from "mini-projects," using a team approach to short projects within an established course, to major projects which last an entire year. The mini-project format requires a great deal of organization and preparation to make it work. See Borrelli and Busenburg "Undergraduate Classroom Experiences in Applied Mathematics" (*UMAP Journal,* Volume 1, 1980) for one approach to structuring a mini-project program, together with its pro's and con's. The one-semester case study course, judging from its popularity, is the best understood and trusted of modeling courses. There are good textbooks and a great many modules written for use in such a course (see list at end of chapter).

While most case studies texts on mathematical modeling are designed for upper-level courses, the text *You and Technology* (mentioned above), supplemented with modules, can easily be adapted for use in a freshman case studies course. Such a course might also present an opportunity for students to see the fundamental differences between engineering and mathematical approaches to modeling (this issue is treated nicely in *You and Technology*). An extensive outline is provided below for a special custom-made, lower-level modeling course.

Experiential modeling courses are not used as often as case study courses. Since the experiential approach is typically used on open-ended problems where the outcome is difficult to predict in advance, this approach is especially risky for a mathematics instructor who is teaching a modeling course for the first time. Nevertheless, experiences of various colleges over the last several years show that the experiential approach is feasible and that, whatever happens, students and instructors find it a rewarding experience. Several successful formats for experiential modeling courses have emerged. All seem to use the team approach with occasional guidance by consultants, as needed. It should be noted that many industrial employers treat such experiential modeling as job-related experience in assessing a student's job qualifications. References at the end of this chapter contain descriptions of the well-known Mathematics Clinics in Claremont and other experiential modeling courses (interested readers can write directly to Harvey Mudd College for first-hand advice).

We close this Section with some important general points to keep in mind when designing any modeling course.

- To encourage initiative and independent work, students should have access to, and be responsible for using, support resources such as documentation of software and previous student projects.
- If high standards are imposed on writing of reports, then these reports deserve some exposure; they should not just be shoved in filing cabinets and forgotten. Instructors should encourage students to seek publication of a paper based on their reports, if warranted, or an article in the campus newspaper. Abstracts of recent reports should be made available to students early in a modeling course. When students know their work will get exposure, they are motivated to write good reports.
- It is valuable to integrate the modeling process into the curriculum as widely as possible and not just as an add-on special course with no connection to any other mathematical sciences course.
- A problem with most modeling courses is that the material in them quickly becomes dated. When students discover that they are working on the same projects or models as their classmates did last year, they lose enthusiasm. What is needed is a format for automatically updating the material. A constant flow of real-world problems, as come into a mathematics consulting clinic, is a great advantage.

Operations Research

Operations research is a mathematical science closely connected to mathematical modeling. Although some notable contributions were made prior to 1940, operations research grew out of World War II. The analysis of military logistics, supply and operational problems by scientists from many different disciplines generated the techniques and approaches that evolved into modern operations research. This subject studies complex systems, structures and institutions with a view towards operating such multiparameter systems more efficiently within various constraints, such as scarce resources. Operations research analyses are used to optimize current activities and predict future feasibility. The complexity of its problems has made operations research heavily dependent on high-speed digital computers. It is now used in fields in which decisions were traditionally made on the basis of less quantitative approaches, such as "experience" or mere hunches. There is frequently a major concern with "people" as well as "things," and the man-system interface in a complex social activity. Major national concerns such as productivity, environmental impact and energy supply have a large operations research component.

The approach in operations research is multidisciplinary in nature, and uses common sense, data, and substantial empiricism (heuristics) combined with new, as well as repackaged traditional, mathematical methodologies. The principal mathematical theories of operations research are mathematical programming and stochastic processes. Major topics in these theories are mentioned in the operations research course contents in the next section. Operations research has major overlap with the fields of industrial engineering, management science, mathematical economics, econometrics and decision theory.

Anatomy of a Modeling Course

Ingredients	Background and Source Material	Remarks
PREREQUISITES:		
Lower Division. Single variable calculus, a science course with lab, some computing.	Case study approach most likely. See, e.g., "You and Technology" or suitable UMAP modules.	If the team approach is selected then there can be some flexibility in these prerequisites.
Upper Division. Multivariable calculus, linear algebra, computation and some computer programming, basic prob/stat., some diff. eqns., a science course with lab.	For experiential approach and case study approach consult appropriately noted reference.	If modeling course is not required, then some thought must be given as to how students can be attracted to such a course: descriptions in registration packets, posters, note to advisor, etc.
EFFORT LEVEL:		
Partial Course. Recommended minimum of 2 weeks out of a 3 hour course preceded by a tooling up period.	Mini-projects are a possibility here. See Borelli and Busenberg. Format of mini-projects can be effectively structured. See Becker, *et al.,* "Handbook for Projects."	Important that mini-project work *not* be simply added to standard load of the host course—it should replace some required work; e.g., an exam.
Full Course. May be designed to fit into special options, either to give job-related training or introduction to modeling process with important models in a discipline.	Many possibilities exist for modeling courses for a full semester—see items below. For a discussion of pros and cons, see Borelli and Busenberg.	Format of instruction can seriously affect the student's interest as well as his capacity for effective work—see "Format" section below for possibilities.
COURSE FORMAT:		
Case Study. The modeling process presented via examples that are more-or-less self-contained.	Material selected from modules, textbooks, conference proceedings, or journals.	Advanced students can be asked to lecture on material that is well enough organized.
Experiential. Hands-on approach to open-ended projects incorporating the modeling process. Some possibilities are:		Internships, work-study programs not appropriate for inclusion here.
1. Problem-centered Course. Class divided into teams to work on a sequence of projects and share experience.	Needs highly experienced instructor to select and present the projects and watch over progress of the teams. Class size limited by instructor's energy. See Borelli and Busenberg for more details.	Oral presentation and written reports are emphasized. Most demanding of instructor's time.
2. Mathematics Clinic, Consulting Group. Intensive, industry-supported team effort on a single project, usually for one year.	Composition of team is critical. See Claremont Clinic Articles for details. Because of time constraints, able support staff must be readily available.	Team communication skills highly emphasized in Clinic program and is crucial to success. Team has main responsibility for work, instructor advises. Student handbook at Claremont Clinic (by Handa) available on request.
3. Research Assistance. Students aid faculty in research work.	MIT has a highly organized program which does this. Mostly, however, it's catch-as-catch-can. The Institute of Decision Science, Claremont Men's College, has developed a classroom approach to such work.	A danger here is that the success of the faculty member's research may take precedence over the impact on the students' education. Students' needs could get lost in the shuffle.
4. Mini-projects. Team approach on short projects within an established course.	See Borelli and Busenberg.	Emphasizes writing skills, highly structured activity; see "Handbook for Projects" by Becker, *et al.*

Anatomy of a Modeling Course

Ingredients	Background and Source Material	Remarks
RESOURCES AVAILABLE:		
Computer. Good access to a high level computer (preferably with time-sharing capability) having good software packages is very important for the success of most modeling courses.		Computer graphics capabilities and knowledgeable (and accessible) consultants at the computer center add not only a professional touch but also help teams live within their time constraints.
Experienced Consultants. Access to knowledgeable colleagues, experts in local industrial firms, and talented computer center personnel are all helpful in keeping a team's progress from faltering.	A successful, long-term program depends to a large extent on the Director's ability to secure *willing* assistance from able consultants.	Be sure that consultants help is acknowledged by the students in all written reports, even if it is only of a casual nature.
Supplemental Materials. Handouts on how to work in a team on projects, or where to go for help, etc., lessen the student's feeling of abandonment when working on projects.	For project work, see the Handbooks by Becker, *et al.*, Handa, Seven and Zagar, and for computer graphics, Saunders, *et al.* (all were developed at Harvey Mudd College and are available on request).	
SOURCE OF PROBLEMS:		
Real World. Open-ended problems submitted by local industrial firms or government agencies which are of current interest to them, or problems from current research of colleagues.	See Borelli and Spanier for a description of one effective method of recruiting sponsored projects from industry. MIT has a highly organized way of advertising current research of its faculty and laboratories and whether undergraduates can play a role or not.	Used only in experiential type modeling courses.
Contrived. Open-ended problems pulled from a variety of sources: from technical journals, suggestions from colleagues, books, etc.	The modeling books in the references are good sources of problems.	Used mostly in experiential type modeling course.
Case Studies. Reasonably well self-contained descriptions of completed projects or problems.	Good sources in modules, proceedings of conferences on case studies and books.	Used only for case study type of modeling course.
TECHNICAL THRUST:		
Discrete-OR. Problems whose models involve discrete structures, programming, or optimization within discrete settings. Also interpolation with finite structures in continuous settings.		Deterministic and stochastic methods are both possibilities here.
Continuous. Problems whose models involve differential or integral equations, continuous probabilities, or optimization within continuous setting.		
Computer. Problems with main goal the production of software either at the systems level or solvers for a class of equations in continuous settings, along with error analysis of same. For DEC users, the IMSL package is a good all-around one to have available on the system.		

There are many opportunities for mathematical sciences majors to pursue graduate studies or find employment in operations research and related fields. Industrial mathematicians in all fields find themselves faced with operations research problems from time-to-time. Thus it is important for mathematical sciences students to have some exposure to operations research and its applications, and also knowledge of its career possibilities. This classroom exposure to operations research can occur in conjunction with undergraduate modeling experience or in a specific course on operations research. The current relevance and naturalness of this subject are immediately clear to students, and realistic projects at various levels of difficulty are readily available. An interesting article by D. Wagner about operations research appeared in the *American Mathematical Monthly* (82, p. 895). Students should also be referred to the booklet *Careers in Operations Research,* available from the Operations Research Society of America, 428 Preston Street, Baltimore, MD 21202.

A student interested in graduate work in operations research should have a solid preparation in undergraduate core mathematics: calculus, linear algebra, real analysis, plus courses in probability, introductory computer science and modeling. A course in operations research itself is more important as a way to learn if one likes the field than as a prerequisite for graduate study. A substantial minor in a relevant area outside mathematics (as recommended for all mathematical sciences majors in the first chapter, "Mathematical Sciences") is important. This outside work should include a sampling of quantitative courses in the social sciences, business, or engineering (if available). Experience solving some problems involving substantial computer computation and an exposure to nontrivial algorithms are also desirable.

At some institutions, mathematics departments are now preparing to offer an operations research course for the first time, while other institutions may have many operations research courses offered in mathematics, economics, business, industrial engineering and computer science. In either extreme and situations in between, mathematical sciences students are best served by some form of interdepartmental cooperation, or at least coordination of offerings. If a mathematics department is planning to offer an operations research course when none previously existed at the institution, mathematics should work closely with other interested departments.

In planning this first course, mathematicians could seek contacts with local industry to obtain practitioners as visiting lecturers. On the other hand, an introductory operations research course can be taught by most college mathematics professors with appropriate attitudes if they are willing to undertake some self study. Indeed, faculty without formal operations research training who are going to teach such a course should be strongly encouraged to learn about the field by attendance at short courses, participation in a department seminar on the subject, or by sabbatical leave (or other released time) at universities or industrial laboratories with operations research activities.

Course Descriptions

Four sample courses on operations research and modeling are described below. Only more general remarks are given for the courses in operations research and stochastic processes since these have become fairly standardized in recent years. More specific details are provided for an elementary-level modeling course using discrete mathematics and for a more advanced modeling course using continuous methods. These are merely illustrations of the wide variety of different sorts of modeling courses which can be taught. The 1972 CUPM *Recommendations on Applied Mathematics* contain a detailed description of a physical-sciences oriented modeling course. Such a modeling course continues to be very valuable and in no way should be considered dated. Many basic intermediate-level courses in the physical sciences are also excellent modeling courses, from the point of view of a mathematical sciences major.

Introductory Operations Research

Much of the material in an introductory operations research course for undergraduates has become fairly standard. The course covers primarily deterministic methods. Most publishing companies have good introductory operations research texts (the text title may be Linear Programming, the course's main topic). The level of this course can vary depending on the prerequisites and student maturity. It is normally an upper-level offering with a prerequisite or corequisite of linear algebra. Calculus and probability should be required if stochastic models are also included.

An operations research course can be a "pure mathematics" course which stresses the fundamental properties of systems of linear inequalities, basic geometry of polyhedra and cones, discrete optimization and complexity of algorithms. Most operations research courses, however, emphasize the many applications which can be solved by linear programming and related techniques of combinatorial optimization. Such courses usually devote some time to efficient algorithms and practical numerical methods (to avoid roundoff errors), as well as basic notions of computational complexity. While problem solving and modeling are important, a first operations research course should cover some topic in reasonable depth and not be merely a collection of simple techniques and routine applications.

COURSE CONTENT

The course should start with a brief discussion of the general nature, history and philosophy of operations research. Some of the older texts such as *Introduction to Operations Research* by C. Churchman, R. Ackoff and E. Arnoff, Wiley, 1957, and *Methods of Operations Research* by P. Morse and G. Kimball, Wiley, 1951, devote extensive space to history. The instructor should not spend much time on history at the beginning of a course but instead should weave it into discussions throughout a course.

The first half of the course in usually devoted to linear programming: its theory, the simplex algorithm, and applications. The course then continues on to a series of special linear programming problems, such as optimal assignment, transportation, trans-shipment, network flow, minimal spanning tree, shortest path, PERT methods and traveling salesperson, each with its own algorithms and associated theory. Basic concepts of graph theory are normally introduced in conjunction with some of the preceding problems. If time permits, elementary aspects from decomposition theory, dynamic programming, integer programming, or nonlinear programming may be included.

It is difficult to find space in an introductory operations research course for even a small sampling of probability or stochastic models. If possible, it is better to include this material in a second course. Similarly, there is usually little time available to discuss game theory, except possibly for showing that two-person, zero-sum games are equivalent to a dual pair of linear programs. Game theory is probably best treated in a separate "topics" course.

Elementary Modeling Course

The following course on mathematical modeling and problem solving is intended for freshmen and sophomores with a solid preparation in high school mathematics. The primary objective is to provide lower-level students with a first-hand experience in forming their own mathematical models and discovering their own solution techniques. A secondary goal is to introduce some of the concepts from modern finite mathematics and illustrate their applications in the social sciences. The instructor might supplement these main themes with brief discussions of some important recent mathematical developments and indicate the current relevance of mathematics to contemporary science and policy making.

The course should maintain an open-minded and questioning approach to problem solving. Much of the class time should be devoted to student discussions of their models and how to improve them. Students should be asked to make formal oral and written expositions. Many of the topics covered are also suitable, with proper adjustments, for upper-level courses or for lower-level "mathematics appreciation" courses. (Readers interested in the latter courses should consult the 1981 Report of the CUPM Panel on Mathematics Appreciation, reprinted later in this volume.) Not all of the topics mentioned below can be covered in any one course, and frequent changes in course content are necessary to maintain the originality of problems.

No one current textbook appears appropriate for this course, although a simpler "prepackaged" version of this course could use the high-school-oriented text *You and Technology* with supplementary modules. The course described below is an example of how various sources can be assembled (as handouts or on library reserve) to form a modeling course, in this instance emphasizing modeling in the social sciences.

COURSE CONTENT

Overview and Patterns of Problem Solving. Introduction to the nature of modeling and problem solving. The role of science, engineering and social sciences in making and implementing new discoveries. The nature of applied mathematics and the interdisciplinary approach to problems. Illustrations of problems solved by quick insight rather than by involved analysis. Many books have chapters on modeling and problem solving; also see *Patterns of Problem Solving* by M. Rubinstein, Prentice-Hall, 1975, or "Foresight-Insight-Hindsight" by J. Frauenthal and T. Saaty, in *Modules in Applied Mathematics,* vol. 3 (W. Lucas, editor), Springer-Verlag.

Graph and Network Problems. A large variety of problems related to undirected and directed graphs and network flows can be assigned and discussed at the outset with no hint of any theory or technical terms. At a later stage, a lecture can be devoted to theory to develop a common vocabulary. The language and general approach of systems analysis can be developed. The four-color theorem can be discussed. References are *Applied Combinatorics,* by F. Roberts, Prentice-Hall, 1984, *Graphs as Mathematical Models* by G. Chartrand, Prindle, Weber and Schmidt, 1977, and *Applied Combinatorics* by A. Tucker, Wiley, 1980.

Some lecture time can be spent illustrating how graphs are applied: to simplify a complex problem, such as Instant Insanity (Chartrand, p. 125 or Tucker, p. 355), or the more difficult Rubik's Cube (*Scientific American,* March, 1981); for purely mathematical purposes, such as to prove Euler's formula $V - E + F = 2$ and use it in turn to prove the existence of exactly five regular polyhedra; or to examine R. Connelly's flexing (nonconvex) polyhedra (*Mathematical Intelligencer,* Vol. 1, No. 3, 1979). The analogy between transportation, fluid flow, electric and hydraulic networks can be illustrated (see G. Minty's article in *Discrete Mathematics and Its Applications* Proceedings of a Conference at Indiana University, ed. M. Thompson, 1977).

Enumeration Problems. (Tucker, 2nd ed., Chapter 5 or Roberts, Chapter 2.) Some practical uses can be covered briefly, e.g., to probability problems or the Pigeonhole Principle. Computational complexity and its application to hard-to-break codes can be discussed.

Value and Utility Theory. Expected utility versus expected value; St. Petersburg paradox; construction of a money versus utility curve: axioms for utility; assessing Coalitional Values (see module by W. Lucas and L. Billera in *Modules in Applied Mathematics,* vol. 2, W. Lucas, editor, Springer-Verlag).

Conflict Resolution. Some three-person cooperative game experiments and analysis; the Prisoner's Dilemma for two or more persons (H. Hamburger in *Journal of Math. Sociology* 3, 1973); illustrations of equilibrium concepts; two-person zero-sum games, e.g., batter versus pitcher (*Economics and the Competitive Process* by J. Case, NYU Press, 1979, p. 3; also see *The Game of Business* by John McDonald, Doubleday, 1975, Anchor paperback, 1977, and *Game Theory: A Nontechnical Introduction* by M. Davis, Basic Books, 1970).

A Discrete Optimization Problem and an Algorithm. Possible topics are the complete and optimal assignment problems (UMAP module 317 by D. Gale), or the marriage problem (D. Gale and L. Shapley, *American Mathematical Monthly* 69, 1962, p. 9).

Simulation. See chapters on simulation in many books and "Four-Way Stop or Traffic Light? An Illustration of the Modeling Process" by E. Packel (in *Modules in Applied Mathematics,* vol. 3, W. Lucas, editor, Springer-Verlag). Additional ideas from Inventory Theory, Scheduling Theory, Dynamic Programming, and Control Theory, e.g., lunar landing, can be included.

Projects and Mini-projects. At least one significant project type activity should be pursued over several weeks by the whole class by means of a sequence of graded exercises and class discussions. Some of the topics listed above can be treated in this mode. Other suitable topics are: the Apportionment Problem (Fair Representation by M. Balinski and H. Young, Yale Press); measuring power in Weighted Voting situations (W. Lucas in *Case Studies in Applied Mathematics* MAA, 1976); Cost Analysis (C. Clark in same *Case Studies* on harvesting fish or forests); some simple topics from statistics such as Asking Sensitive Questions, module by J. Maceli (in *Modules in Applied Mathematics,* vol. 2, W. Lucas, editor, Springer-Verlag); and Social Choice Theory and Voting (*Theory of Voting* by R. Farquharson, Yale, 1960).

In addition to the class project, teams of two or three students can spend a few weeks on a mini-project. Many of the topics above can be applied to a local practical problem. Scheduling, inventory and optimal allocations are good topics, as are gaming experiments, simulations and elementary statistical studies. More theoretical topics, ranging from walking versus running in the rain to designing the inside mechanism of the Rubik's Cube are also possible. Some attempt at discussing possible implementation of a mini-project result, e.g., with a campus administrator, is encouraged in order to show the practical difficulties of implementing mathematically optimal procedures.

Introductory Stochastic Processes

The purpose of this course is to introduce the student to the basic mathematical aspects of the theory of stochastic processes and its applications. This course can equally well be offered under such alternate titles as Applied Probability or Operations Research: Stochastic Models. Stochastic processes is a large and growing field. This course will lay background for further learning on the job or in graduate school.

The prerequisite for this course is at least the equivalent of a full course of post-calculus probability including the following topics: random variables, common univariate and multivariate distributions, moments, conditional probability, stochastic independence, conditional distributions and means, generating functions, and limit theorems. Such a course is fairly traditional now, but if most students have had just the integrated statistics and probability course suggested by the Statistics Subpanel, then the beginning of the stochastic processes course would have to be devoted to completing the needed probability background. It is also desirable for students to have some experience with basic matrix algebra and with using computer terminals.

The course should slight neither mathematical theory nor its applications. It is better to cover few topics with a full discussion of both theory and applications to survey theory alone or to cover only applications. The course emphasizes ***problem solving*** and develops an acquaintance with a variety of models that are widely used. Stochastic modeling and ***problem formulation*** are different activities that should be treated in a modeling course. If many students do not subsequently take a modeling course, then the instructor should consider allocating some time (assuming course time did not also have to be devoted to probability) to a module on stochastic modeling in business or government (see list of modules below) or to a real problem at the local college, e.g., modeling the demand for textbooks in the bookstore or utilization of campus parking spaces.

Computers should be used in this course in two ways:

- As a computational aid to perform, for example, matrix calculations needed in Markov chain theory; and
- As a simulation device to exhibit the behavior of random processes.

Understanding randomness is difficult for undergraduates and discussion of data accumulated in simulation studies can help overcome students' deterministic biases.

COURSE CONTENT

Bernoulli process; Markov chains (random walks, classification of states, limiting distributions); Poisson process (as limit of binomial process and as derived via axioms); Markov processes (transition functions and state probabilities, Kolmogorov equations, limiting probabilities, birth-death processes).

These basic topics have numerous applications that should be an essential feature of the course. In addition, some applied topics can be covered such as quality control, social and occupational mobility, Markovian decision processes, road traffic, reliability, queueing problems, population dynamics or inventory models. Instructors can find these and other applications in the many good texts on stochastic processes. Also see the modules and modeling texts listed at the end of this chapter.

Continuous Modeling

A primary goal of a continuous modeling course is to present the mathematical analysis involved in scientific modeling, as for example, the derivation of the heat equation. The course should also give an introduction to important applied mathematics topics, such as Fourier series, regular and singular perturbations, stability theory and tensor analysis. A few advanced topics can be chosen from boundary layer theory, nonlinear waves and calculus of variations. The course should give a solid motivation for more advanced courses in these topics. A (non-original) paper on a topic of interest to the students serves the dual purpose of developing communication skills and introducing pedagogical flexibility.

A course on continuous modeling usually has as a prerequisite a course in differential equations, although the modeling can be taught concurrently or integrated in one course, using a book such as Martin Braun's ***Differential Equations and Their Applications*** (second edition), Springer-Verlag, 1978. Continuous modeling problems frequently involve concepts from natural sciences. In this case, it is important that either an appropriate background is required of students or the technical essentials are adequately introduced in the course.

The texts by Lin and Segal and by Haberman (see below) are well suited for this course. Selections from the two-volume Lin and Segal text can be used to provide a solid basis for physics and engineering modeling using both classical subjects, such as fluids, solids and heat transfer, and modern subjects, such as fields of biology. The text's broad coverage probably includes an introduction to an area of expertise of the instructor to which he or she can bring personal research insights.

A course which requires a little less sophistication can be designed around Haberman's book. This text's topics in population dynamics, oscillations, and traffic theory require less scientific background than topics in mechanics and mathematical biology, but still provide an excellent basis for modeling discussions. For example, population dynamics provide a good introduction to dynamical systems. Topics in regular and singular perturbation theory can be presented in the context of oscillations. Traffic theory provides a vehicle for introducing continuum mechanical modeling in which the processes are readily appreciated by students. Here the "microscopic" processes involve cars and drivers, and interesting models are obtained by car-following theory. Traffic flows also involve partial differential equations and shock waves.

References on Modeling

Modules

A. MODULE WRITING PROJECTS

Claremont Graduate School (Department of Mathematics):

- A Fractional Calculus Approach to a Simplified Air Pollution Model for Perturbation Analysis.

- Continuous-system Simulation Languages for DEC-10.
- Free Vibrations in the Inner Ear.
- Modeling of Stellar Interiors.
- Subsurface Areal Flow Through Porous Media.
- Variance Reduction for Monto Carlo Applications Involving Deep Penetration.
- Voting Games and Power Indices.

Mathematical Association of America's Committee on the Undergraduate Program in Mathematics Project, Case Studies in Applied Mathematics (designed especially for open-ended experiential teaching).

- Measuring Power in Weighted Voting Systems.
- A Model for Municipal Street Sweeping Operations.
- A Mathematical Model of Renewable Resource Conservation.
- Dynamics of Several-species Ecosystems.
- Population Mathematics.
- MacDonald's Work on Helminth Infections.
- Modeling Linear Systems by Frequency Response Methods.
- Network Analysis of Steam Generator Flow.
- Heat Transfer in Frozen Soil.

Mathematical Association of America Summer 1976 Module-writing Conference (at Cornell University Department of Operations Research):

- About sixty modules covering virtually all areas of application, such as biology, ecology, economics, energy, population dynamics, traffic flow, vibrating strings, and voting.
- Selected modules from this conference along with MAA applied mathematics case studies (ii) above were published by Springer-Verlag (New York, 1983) in four volumes, edited by William Lucas.

Rensselaer Polytechnic Institute (Department of Mathematical Sciences), published in *Case Studies in Mathematical Modeling,* by W. Boyce, Pitman, Boston, 1981:

- Herbicide Resistance.
- Elevator Systems.
- Traffic Flow.
- Shortest Paths in Networks.
- Computer Data Communication and Security.
- Semiconductor Crystal Growth.

State University of New York at Stony Brook (Department of Applied Mathematics and Statistics):

- A Model for Land Development.
- A Model for Waste Water Disposal, I and II.
- A Water Resource Planning Model.
- Man in Competition with the Spruce Budworm.
- Smallpox: When Should Routine Vaccination be Discontinued.
- Stochastic Models for the Allocation of Fire Companies.

B. MODULES DEVELOPED BY INDIVIDUALS

- Undergraduate Mathematics Application Project (UMAP): UMAP has several hundred modules covering all areas of application. Selected modules appear in the *UMAP Journal* (four issues a year), published by Birkhauser-Boston. UMAP catalogue available by writing to: UMAP, Educational Development Center, 55 Chapel Street, Newton, MA 02160.

C. PROCEEDINGS OF MODELING CONFERENCES

1. Discrete Mathematics and Its Applications, Proceedings of a Conference at Indiana University, ed. M. Thompson, 1976.
2. Mathematical Models in the Undergraduate Curriculum, Proceedings of Conference at Indiana University, ed. D. Maki and M. Thompson, 1975.
3. Proceedings of Summer Seminar on Applied Mathematics, ed. M. Thompson, Indiana University, 1978.
4. Mathematical Models for Environmental Problems, Proceedings of the International Conference at the University of Southampton, 1976.
5. Proceedings of Conference on Environmental Modeling and Simulation, Environmental Protection Agency, 1976.
6. Proceedings of a Conference on the Application of Undergraduate Mathematics in the Engineering, Life, Managerial and Social Sciences, ed. P. Knopp and G. Meyer, Georgia Institute of Technology, 1973.
7. Proceedings of the Pittsburgh Conferences on Modeling and Simulations, Vols. 1-9 (1969-78), Instrument Society of America.
8. Proceedings of the Summer Conference for College Teachers on Applied Mathematics, University of Missouri-Rolla, 1971.
9. Information Linkage Between Applied Mathematics and Industry, ed. P. Wang, Academic Press, 1976.

Articles on Teaching Modeling

1. J. Agnew and M. Keener, A Case-study Course in Applied Mathematics Using Regional Industries, *American Mathematical Monthly* 87 (1980).
2. R. Barnes, Applied Mathematics: An Introduction Via Models, *American Mathematical Monthly* 84 (1977).
3. C. Beaumont and R. Wieser, Co-operative Programmes in Mathematical Sciences at the University of Waterloo, *Journal of Co-operative Education* 11 (1975).

4. J. Becker, R. Borrelli, and C. Coleman, *Models for Applied Analysis,* Harvey Mudd College, 1976 and revised annually.
5. R. Borrelli and J. Spanier, The Mathematics Clinic: A Review of Its First Seven Years, *UMAP Journal* 2 (1981).
6. R. Borton, *Mathematical Clinic Handbook,* Claremont Graduate School, 1979.
7. J. Brookshear, A Modeling Problem for the Classroom, *American Mathematical Monthly* 85 (1978).
8. E. Clark, *How To Select a Clinic Project,* Harvey Mudd College, 1975.
9. C. Hall, Industrial Mathematics: A Course in Realism, *American Mathematical Monthly* 82 (1975).
10. L. Handa, *Mathematics Clinic Student Handbook: A Primer for Project Work,* Harvey Mudd College, 1979.
11. J. Hachigian, Applied Mathematics in a Liberal Arts Context, *American Mathematical Monthly* 85 (1978).
12. E. Rodin, Modular Applied Mathematics for Beginning Students, *American Mathematical Monthly* 84 (1977).
13. R. Rubin, Model Formulation Using Intermediate Systems, *American Mathematical Monthly* 86 (1979).
14. M. Seven and T. Zagar, *The Engineering Clinic Guidebook,* Harvey Mudd College, 1975.
15. D. Smith, A Seminar in Mathematical Modelbuilding, *American Mathematical Monthly* 86 (1979).
16. J. Spanier, The Mathematics Clinic: An Innovative Approach to Realism Within an Academic Environment, *American Mathematical Monthly* 83 (1976).

Books on Mathematical Modeling

For further references, see Applications section of *A Basic Library List,* Mathematical Association of America, 1976.

A. GENERAL MODELING

1. J. Andrew and R. McLone, ed., *Mathematical Modeling,* Butterworth, 1976.
2. R. Aris, *Mathematical Modeling Techniques,* Pitman, 1978.
3. E. Beltrami, *Mathematics for Dynamic Modeling,* Academic Press, 1987.
4. E. Bender, *An Introduction to Mathematical Modeling,* Wiley, 1978.
5. G. Carrier, *Topics in Applied Mathematics,* Vol. I and II, MAA summer seminar lecture notes, Mathematical Association of America, 1966.
6. C. Coffman and G. Fix, ed., *Constructive Approaches to Mathematical Models,* Academic Press, 1980.
7. R. DiPrima, ed., *Modern Modeling of Continuous Phenomena,* American Mathematical Society, 1977.
8. C. Dym and E. Ivey, *Principles of Mathematical Modeling,* Academic Press, 1980.
9. B. Friedman, *Lectures on Applications-oriented Mathematics,* Holden-Day, 1969.
10. F. Giordano and M. Weir, *A First Course in Mathematical Modeling,* Brooks/Cole, 1985.
11. P. Lancaster, *Mathematics Models of the Real World,* Prentice Hall, 1976.
12. D. Maki and M. Thompson, *Mathematical Models and Applications,* Prentice Hall, 1976.
13. F. Roberts, *Discrete Mathematical Models,* Prentice Hall, 1976.
14. T. Saaty, *Thinking with Models,* AAAS Study Guides on Contemporary Problems No. 9, 1974.

B. MODELING IN VARIOUS DISCIPLINES

Mathematical modeling is such an integral part of physics and engineering that any text in these fields is implicitly a mathematical modeling book.

1. P. Abell, *Model Building in Sociology,* Shocken, 1971.
2. R. Aggarwal and I. Khera, *Management Science Cases and Applications,* Holden-Day, 1979.
3. R. Atkinson, et al., *Introduction to Mathematical Learning Theory,* Krieger Publishing, 1965.
4. D. Bartholomew, *Stochastic Models for Social Processes,* Wiley, 1973.
5. M. Bartlett, *Stochastic Population Models,* Methuen, 1960.
6. R. Barton, *A Primer on Simulation and Gaming,* Prentice Hall, 1970.
7. S. Brams, *Game Theory and Politics,* The Free Press, 1975.
8. C. Clark, *Mathematical Bioeconomics,* Wiley, 1976.
9. J. Coleman, *Introduction to Mathematical Sociology,* Free Press, 1964.
10. P. Fishburn, *The Theory of Social Choice,* Princeton University Press, 1973.
11. J. Frauenthal, *Introduction to Population Modeling,* UMAP Monograph, 1979.
12. H. Gold, *Mathematical Modeling of Biological Systems,* Wiley, 1977.
13. S. Goldberg, *Some Illustrative Examples of the Use of Undergraduate Mathematics in the Social Sciences,* Mathematical Association of America, CUPM Report, 1977.

14. M. Gross, *Mathematical Models in Linguistics,* Prentice Hall, 1972.
15. R. Haberman, *Mathematical Models, Mechanical Vibrations, Population Dynamics and Traffic Flow,* Prentice Hall, 1977.
16. F. Hoppensteadt, *Mathematical Theories of Populations: Demographics and Epidemics,* SIAM, 1976.
17. J. Kemeny and L. Snell, *Mathematical Models in the Social Sciences,* MIT Press, 1973.
18. C. Lave and J. March, *An Introduction to Models in the Social Sciences,* Harper and Row, 1975.
19. C. Lin and L. Segal, *Mathematics Applied to Deterministic Problems in the Natural Sciences,* Macmillan, 1974.
20. D. Ludwig, *Stochastic Population Theories,* Springer, 1974.
21. J. Maynard-Smith, *Models in Ecology,* Cambridge University Press, 1974.
22. B. Noble, *Applications of Undergraduate Mathematics to Engineering,* Mathematical Association of America, 1976.
23. M. Olinik, *An Introduction to Mathematical Models in the Social and Life Sciences,* Addison Wesley, 1978.
24. E. Pielou, *Mathematical Ecology,* Wiley, 1977.
25. H. Pollard, *Mathematical Introduction to Celestial Mechanics,* Mathematical Association of America, 1977.
26. J. Pollard, *Mathematical Models for the Growth of Human Populations,* Cambridge University Press, 1973.
27. D. Riggs, *The Mathematical Approach to Physiological Problems,* Macmillan, 1979.
28. T. Saaty, *Topics in Behavioral Mathematics,* MAA summer seminar lecture notes, Mathematical Association of America, 1973.
29. H. Scarf, et al., *Notes on Lectures on Mathematics in the Behavioral Sciences,* MAA summer seminar lecture notes, Mathematical Association of America, 1973.
30. C. von Lanzenauer, *Cases in Operations Research,* Holden Day, 1975.
31. H. Williams, *Model Building in Mathematical Programming,* Wiley, 1978.

Subpanel Members

WILLIAM LUCAS, CHAIR, Cornell University.
ROBERT BORRELLI, Harvey Mudd College.
RALPH DISNEY, VPI & SU.
DONALD DREW, RPI.
SAM GOLDBERG, Oberlin College.

Statistics

This chapter contains the report of the Subpanel on Statistics of the CUPM Panel on a General Mathematical Sciences Program, reprinted with minor changes from Chapter VI of the 1981 CUPM report entitled RECOMMENDATIONS FOR A GENERAL MATHEMATICAL SCIENCES PROGRAM.

Introductory Course

Statistics is the methodological field of science that deals with collecting data, organizing and summarizing data, and drawing conclusions from data. Although statistics makes essential use of mathematical tools, especially probability theory, it is a misrepresentation of statistics to present it as essentially a subfield of mathematics.

The Statistics Subpanel believes that an introductory course in probability and statistics should concentrate on data and on skills and mathematical tools motivated by the problems of collecting and analyzing data. The traditional undergraduate course in statistical theory has little contact with statistics as it is practiced and is not a suitable introduction to the subject. Such a course gives little attention to data collection, to analysis of data by simple graphical techniques, and to checking assumptions such as normality.

The field of statistics has grown rapidly in applied areas such as robustness, exploratory data analysis, and use of computers. Some of this new knowledge should appear in a first course. It is now inexcusable to present the two-sample t-test for means and the F-test for variances as equally legitimate when a large literature demonstrates that the latter is so sensitive to nonnormality as to be of little practical value, while the former (at least for equal sample sizes) is very robust (e.g., see Pearson and Please, *Biometrika* 62 (1975), pp. 223-241, for an effective demonstration). However, the Statistics Subpanel does not believe that a course in "exploratory data analysis" is a suitable introduction to statistics, nor does it advocate replacing (say) least squares regression by a more robust procedure in a first course. But it does think that new knowledge renders a course devoted solely to the theory of classical parametric procedures out of date.

While the Statistics Subpanel prefers a two-semester introductory sequence in probability and statistics, enrollment data shows that most students take only a single course in this area. The course proposed below gives students a representative introduction to both the data-oriented nature of statistics and the mathematical concepts underlying statistics. These broad objectives raise several issues that require preliminary comment. One year of calculus is assumed for this course. The course should use Minitab or a similar interactive statistical package.

The Place of Probability

Probability is an essential tool in several areas of the mathematical sciences. It is not possible to compress a responsible introduction to probability and coverage of statistics into a single course. The Statistics Subpanel therefore recommends that probability topics be divided between the courses on probability and statistics, discrete methods, and modeling/operations research as follows:

- *Probability and statistics course:* Axioms and basic properties; random variables; univariate probability functions and density functions; moments; standard distributions; Laws of Large Numbers and Central Limit Theorem.
- *Discrete methods course:* Combinatorial enumeration problems in discrete probability.
- *Modeling/operations research course:* Conditional probability and several-stage models; stochastic processes.

This division is natural in the sense that the respective parts of probability are motivated by and applied to the primary concerns of these courses.

Alternative Arrangements

The subpanel is convinced that two semesters are required for a firm introduction to both probability and statistics. Many institutions now offer such a two-semester sequence in which probability is followed by statistics. The subpanel prefers this structure. In this sequence the statistics course should be revised to incorporate the topics and flavor of the data analysis section of the proposed unified course. With probability first, added material in statistics can also be covered, such as Neyman-Pearson theory, distribution-free tests, robust procedures, and linear models.

Institutions will vary considerably in their choice of material for this statistics course, but the subpanel reiterates its conviction that the traditional "theory-only" statistics course is not a wise choice. If experience shows that many students drop out in the middle of a two-course sequence, the unified course outlined below should be adopted, followed by one of the elective courses suggested in Section 3 of this chapter.

Instructor Preparation

Since the recommended outline is motivated by data and shaped by the modern practice of statistics, many mathematically trained instructors will be less prepared to teach this course than a traditional statistical theory course. Growing interest in "applied" statistics has, of course, led many instructors to broaden their knowledge. Some background reading is provided for others who wish to do so. The publications listed here contain material that can be incorporated in the recommended course, but none is suitable as a course text. In order of ascending level:

1. Tanur, Judith, *et al.*, eds., *Statistics: A Guide to the Unknown,* Second Edition, Holden-Day, 1978.
   An elementary volume describing important applications of statistics and probability in many fields of endeavor.
2. Moore, David, *Statistics: Concepts and Controversies,* W.H. Freeman, 1979.
   A paperback with good material on data collection, statistical common sense, appealing examples, and the logic of inference.
3. Freedman, David; Pisani, Robert; Purves, Roger, *Statistics,* W.W. Norton, 1978.
   A careful introduction to elementary statistics written with conceptual richness, attention to the real world, and awareness of the treachery of data.
4. Mosteller, Frederick and Tukey, John, *Data Analysis and Regression,* Addison-Wesley, 1977.
   Good ideas on exploratory data analysis, robustness and regression.
5. Box, George; Hunter, William; Hunter, J. Stuart, *Statistics for Experimenters,* Wiley, 1978.
   Applied statistics explained by experienced practical statisticians. Some specialized material, but much of the book will repay careful reading.
6. Efron, Bradley, "Computers and the Theory of Statistics: Thinking the Unthinkable," *SIAM Review,* October, 1979.
   A superb article on some new directions in statistics, written for mathematicians who are not statisticians.

Course Outline

I. Data **(about 2 weeks)**

- *Random sampling.* Using a table of random digits; simple random samples, experience with sampling variability of sample proportions and means; stratified samples as a means of reducing variability.
- *Experimental design.* Why experiment; motivation for statistical design when field conditions for living subjects are present; the basic ideas of control and randomization (matching, blocking) to reduce variability.

COMMENTS: Data collection is an important part of statistics. It meets practical needs (see Moore) and justifies the assumptions made in analyzing data (see Box, Hunter and Hunter). Experience with variability helps motivate probability and the difficult idea of a sampling distribution. Students should see for themselves the results of repeated random sampling from the same population and the variability of data in simple experiments such as comparing 3-minute performance of egg timers (see W.G. Hunter, *American Statistician,* 1977, pp. 12-17, for suggestions).

II. Organizing and Describing Data **(about 2 weeks)**

- *Tables and graphs.* Frequency tables and histograms; bivariate frequency tables and the misleading effects of too much aggregation; standard line and bar graphs and their abuses; box plots; spotting outliers in data.
- *Univariate descriptive statistics.* Mean, median and percentiles; variance and standard deviation; a few more robust statistics such as the trimmed mean.
- *Bivariate descriptive statistics.* Correlation; fitting lines by least squares. If computer resources permit, least-square fitting need not be restricted to lines.

COMMENTS: In addition to simple skills, students must be trained to look at data and be aware of pitfalls. Freedman, Pisani and Purves have much good material on this subject, such as the perils of aggregation (pp. 12-15). The impressive effect on a correlation of keypunching 7.314 as 731.4 should be pointed out. Simple plots are a powerful tool and should be stressed throughout the course as part of good practice.

III. Probability **(about 4 weeks)**

- *General probability.* Motivation; axioms and basic rules, independence.
- *Random variables.* Univariate density and probability functions; moments; Law of Large Numbers.

- *Standard distributions.* Binomial, Poisson, exponential, normal; Central Limit Theorem (without proof).
- *More experience with randomness.* Use in computer simulation to illustrate Law of Large Numbers and Central Limit Theorem.

COMMENTS: Probability must unavoidably be pressed in a unified course that includes data analysis. Instructors should repeatedly ask "What probability do I need for basic statistics?" and "What can the students learn within about four weeks?" It is certainly the case that combinatorics, moment generating functions, and continuous joint distributions must be omitted. Some instructors may be able to cover conditional probability and Bayes' theorem in addition to the outline material.

IV. Statistical Inference (about 6 weeks)

- *Statistics vs. probability.* The idea of a sampling distribution; properties of a random sample, e.g., it is normal for normal populations; the Central Limit Theorem.
- *Tests of significance.* Reasoning involved in alpha-level testing and use of P-values to assess evidence against a null hypothesis; cover one- and two-sample normal theory tests and (optional) chi-square tests for categorical data. Comment on robustness, checking assumptions, and the role of design (Part I) in justifying assumptions.
- *Point estimation methods.* Method of moments; maximum likelihood; least squares; unbiasedness and consistency.
- *Confidence intervals.* Importance of error estimate with point estimator; measure of size of effect in a test of significance.
- *Inference in simple linear regression.*

COMMENTS: A firm grasp of statistical reasoning is more important than coverage of a few additional specific procedures. For much useful material on statistical reasoning such as use of the "empirical rule" to assess normality, see Box, Hunter and Hunter. *Don't* just say, "We assume the sample consists of iid normal random variables." Applied statisticians favor P-values over fixed alpha tests; a comparative discussion of this issue appears in Moore.

RECOMMENDED TEXTS

The Subpanel is not aware of a text at the post-calculus level that fits the recommended outline closely. Instructors should seriously consider adopting a good post-calculus statistical methods text rather than a theoretical statistics text. A methods text is more likely to have examples and problems which have the ring of truth. Moreover, most instructors will find it easy to supplement a methods text with mathematical material and problems familiar from previous teaching. It is much harder to supply motivation and realistic problems, and it is psychologically difficult for both the teacher and student to skip much of the probability in a mathematical statistics text.

The following books are possible texts or reference material for the course described above. All of these have essentially the same shortcoming of being too "unmathematical." The appropriate combination of level of sophistication and content is not now available under a single cover. The class of books below fall in the "intermediate" level between an elementary statistics course and a first course in mathematical statistics.

1. Box, George; Hunter, William; Hunter, J. Stuart, *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building,* John Wiley & Sons, New York, 1978.
2. Moore, David and McCabe, George, *Introduction to the Practice of Statistics,* Freeman, San Francisco, 1989.
3. Ott, Lyman, *An Introduction to Statistical Methods and Data Analysis, Second Edition,* Duxbury, Boston, 1984.
4. Neter, John; Wasserman, William; Whitmore, G.A., *Applied Statistics,* Allyn and Bacon, Boston, 1978.

Additional Courses

Probability and Statistical Theory

CONTENT: Distribution functions; moment and probability generating functions; joint, marginal and conditional distributions; correlations; distributions of functions of random variables; Chebyschev's inequality; convergence in probability; limiting distributions; power test and likelihood ratio tests; introduction to Bayesian and nonparametric statistics; additional regression topics.

COMMENT: This course is designed to complete the traditional probability-then-statistics sequence. Since the students have already completed a semester of study, they should be capable of tackling a good text on mathematical statistics such as the one by DeGroot or by Hogg and Craig. The book by Bickel and Doksum is a little more difficult than the other two, and the teacher would have to supplement it with the topics in probability.

TEXTS:

1. Mendenhall, William; Schaeffer, Richard; Wackerly, Dennis, *Mathematical Statistics with Applications, Second Edition,* Duxbury, Boston, 1981.
2. Larsen, Richard and Marx, Morris, *An Introduction to Probability and its Applications,* Prentice-Hall, Englewood Cliffs, N. Jers., 1985.
3. DeGroot, Morris M., *Probability and Statistics,* Addison-Wesley, Reading, Mass., 1975.
4. Hogg, Robert and Craig, Allen, *Introduction to Mathematical Statistics,* Macmillan, New York, 1978.

Applied Statistics

CONTENT: This course uses statistical packages to analyze data sets. Topics include linear and multiple regression; nonlinear regression; analysis of variance; random, fixed and mixed models; expected mean squares; pooling, modifications under relaxed assumptions; multiple comparisons; variance of estimators; analysis of covariance.

COMMENT: The new introductory course will probably attract more students from other fields than the traditional probability-then-statistics course. This course is an excellent follow-up for such non-mathematical sciences students. Its topics are among the more widely used statistical tools. Students should be expected to use a statistical computing package such as Minitab of SPSS for many of the analyses. The book by Miller and Wichern is a possible text for this course.

TEXTS:

1. Miller, Robert and Wichern, Dean, *Intermediate Business Statistics,* Holt, Rinehart and Winston, New York, 1977.
2. Neter, John; Wasserman, William; Kutner, Michael, *Applied Linear Statistical Models, Second Edition,* Irwin, 1985.
3. Morrison, Donald, *Applied Linear Statistical Methods,* Prentice-Hall, Englewood Cliffs, N. Jers., 1983.

Probability and Stochastic Processes

CONTENT: Combinatorics; conditional probability and independence; Bayes theorem; joint, marginal and conditional distributions; distribution functions; distributions of functions of random variables; probability and moment generating functions; Chebyschev's inequality; convergence in probability; convergence in distribution; random walks; Markov chains; introduction to continuous-time stochastic processes.

COMMENT: This is a fairly standard course and a number of texts are available. The book by Feller is a classic but covers only discrete probability. The book by Olkin, Gleser and Derman is at a slightly lower level and is more "applied" but will require the instructor to provide some supplementary materials. The book by Chung is excellent but must be read with a "grain of salt." The book by Breiman is also excellent but expects much of its reader. A new book by Johnson and Kotz also looks interesting but is restricted to discrete probability. The books by Chung, Feller and Breiman are difficult for the average student.

TEXTS:

1. Olkin, Ingram; Gleser, Leon J.; Derman, Cyrus, *Probability Models and Applications,* Macmillan, New York, 1980.
2. Larsen, Richard and Marx, Morris, *An Introduction to Probability and its Applications,* Prentice-Hall, Englewood Cliffs, N.J., 1985.
3. Ross, Sheldon, *A First Course in Probability, Second Edition,* Macmillan, New York, 1984.
4. Chung, Kai Lai, *Elementary Probability Theory with Stochastic Processes,* Springer-Verlag, New York, 1974.
5. Feller, William, *An Introduction to Probability Theory and Its Applications, Volume I,* John Wiley & Sons, New York, 1950.

Preparation for Graduate Study

There are a large number of career opportunities for statisticians in industry, government and teaching. For example as of 1977, the Federal Government employed over 3500 statisticians, plus 3500 statistical assistants and numerous other employees performing statistical duties but classified in different job series. A recent report by the U.S. Labor Department, reprinted in the *New York Times* National Recruitment Survey, predicts an increase of 35% in the demand for statisticians during the 1980's. This compares to a predicted increase of 9% for mathematicians and 30% for computer specialists.

Preparation for a career in statistics usually involves graduate study. An undergraduate major in statistics, computer science, or mathematical sciences is the recommended preparation for graduate study in statistics. It is desirable for such a major to include solid courses in matrix theory and real analysis, in addition to courses in probability and statistics. Most statistics graduate programs require matrix algebra and real analysis for fully matriculated admission. Either one of the sample programs in the report of the General Mathematical Sciences Panel in the first chapter would be adequate preparation for graduate study in statistics. However, major A is preferable to major B, and both should include at least one follow-on elective in probability and statistics.

In addition to courses in the mathematical sciences, a student preparing for graduate study in statistics should:

- Study in depth some subject where statistics is an important tool (physics, chemistry, economics, psychology, ...). In fact, a double major should be considered.
- Take as many courses as possible which are designed to enhance his or her communication skills. Statisticians in industry and government are often called upon to provide written reports and critiques; consulting requires clear oral communications.

A detailed discussion of preparation for a statistical career in industry can be found in [1] A similar report, [2], discusses preparation for a career in government.

1. Preparing Statisticians for Careers in Industry: Report of the ASA Section on Statistical Education. *The American Statistician,* 1980, pp. 65-80.
2. Preparing Statisticians for Careers in Government: Report of the ASA Section on Statistics in Government. Paper presented at the American Statistical Association meeting in August, 1980.

Panel Members:

RICHARD ALO, CHAIR, Lamar University.
RICHARD KLEBER, St. Olaf College.
DAVID MOORE, Purdue University.
MIKE PERRY, Appalachian State University.
TIM ROBERTSON, University of Iowa.