CALCULUS

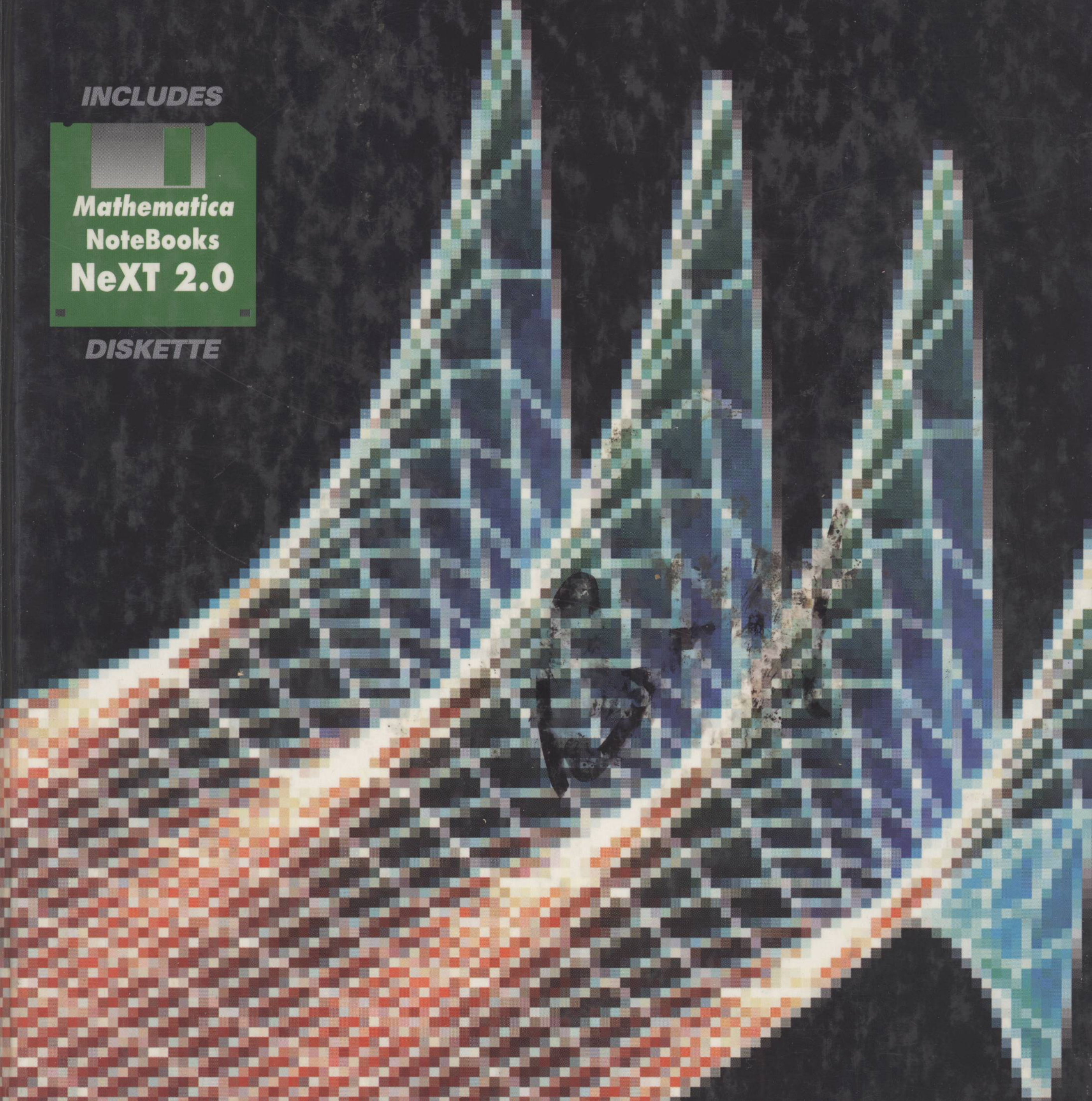
K. D. STROYAN

USING MATHEMATICA

INCLUDES

Mathematica
NoteBooks
NeXT 2.0

DISKETTE



CALCULUS USING *MATHEMATICA*

K. D. STROYAN

University of Iowa
Iowa City, Iowa

Academic Press, Inc.
A Division of Harcourt Brace and Company

Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper. ♾

ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101-4311

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24–28 Oval Road, London NW1 7DX

ISBN: Macintosh: 0-12-672971-9
IBM/DOS: 0-12-672972-7
NeXT: 0-12-672973-5

Printed in the United States of America
93 94 95 96 MV 9 8 7 6 5 4 3 2 1

CALCULUS USING *MATHEMATICA*

Contents of Mathematica NoteBooks
for
Calculus Using *Mathematica*

Chapter 1. Introduction

aMathcaIntro.ma introduces the Mathematica 'front end' or NoteBook Editor with open and closed cells. It also gives a brief tour of the various kinds of calculations that are possible in Mathematica and leads into the work of Chapters 2 and 3.

Exact Arithmetic

Floating Point (Approximate) Arithmetic

Symbolic Computations

Graphics

Lists

Part I. Differentiation in One Variable

Chapter 2. Using Calculus to Model Epidemics

FirstS-I-R.ma checks the hand calculations done in solving the first S-I-R model. It also provides an introduction to variable assignment and simple editing in Mathematica. It leads into the loop calculation for the second model.

SecondS-I-R.ma does more sophisticated calculations with the S-I-R model by varying the step size and producing graphs of the solutions. It also provides an introduction to some Mathematica programming structures like the Do loop.

EpidemicRoots.ma describes how to calculate limiting values of the S-I-R model. It does this with Mathematica's numerical root finding algorithm, which it introduces.

Chapter 3. Numerics, Symbolics and Graphics in Science

Functions.ma provides an introduction to function notation in Mathematica. Examples of numerical calculations with functions are given as well as symbolic computations. An example of a function that is actually a procedure is also given.

SlideSquash.ma introduces animations. A series of graphs representing parabolas with a parameter varied are generated and combined into a movie. This provides a dynamic representation of translation and expansion and shows how these are represented analytically.

ExpGth.ma (*see also* Chapter 8) is a demonstration of how rapid exponential growth is. Starting from a simple model with algae cells doubling every 6 hours, the NoteBook demonstrates that 1000 algae cells would completely fill Lake Michigan in only 15 days.

LogGth.ma (*see also* Chapter 8), in contrast to the previous NoteBook, demonstrates how slow logarithmic growth is. A computer adding ten billion terms of the harmonic series every second still takes 3.1 x 10^6 ages of the solar system to get to 100.

Chapter 4. Linearity vs. Local Linearity

Zoom.ma produces an animation of a graph expanding. The section of the graph to be blown up is surrounded by a small box and local coordinate axes are displayed. This is the live geometric version of the main idea of differential calculus: smooth curves appear linear under powerful magnification.

NonDiffble.ma shows Weierstrass's nowhere differentiable function. Not all functions are smooth and this one is 'kinky' at every point.

Chapter 5. Direct Computation of Increments

Differences.ma illustrates the difference quotient limit approaching the derivative function. It is another way to see the main approximation of differential calculus.

SymbolicIncrem.ma calculates symbolic increments using Mathematica.

Microscope1D.ma animates the main idea of differential calculus, namely that small changes in differentiable functions are locally linear. It also shows how to pre-compute the linear functions with Mathematica. In effect, calculus lets us 'see' with one eye in the microscope without opening the other eye to see the whole graph. Rules tell us what we will see in the microscope.

Chapter 6. Symbolic Differentiation

DiffRules.ma defines rules for a function that allow the function to perform symbolic differentiation. The rules are defined in the same order as the rules for differentiation are presented in the text, so that at any point in learning rules, this symbolic differentiator can only do the problems to which those rules apply.

Chapter 7. Basic Applications of Differentiation

Dfdx.ma shows how to use the built-in Mathematica function for differentiation.

Chapter 8. The Natural Logarithm and Exponential

EulerApprox.ma shows the discrete Euler approximations to $dy = y\ dt$ converging to e^t. This illustrates the 'official' definition of the natural exponential function.

ExpDeriv.ma (*see also* Mathematical Background) approximates $d(b^t)/dt$ directly to find Euler's $e = 2.71828 \ldots$ as the base that has constant of proportionality 1.

ExpGth.ma (*see also* Chapter 3) is a demonstration of how rapid exponential growth is. Starting from a simple model with algae cells doubling every 6 hours, the NoteBook demonstrates that 1000 algae cells would completely fill Lake Michigan in only 15 days.

Chapter 8 contains the mathematical 'order of infinity' result that says exponentials grow faster than powers.

LogGth.ma (*see also* Chapter 3), in contrast to the previous notebook, demonstrates how slow logarithmic growth is. A computer adding ten billion terms of the harmonic series every second still takes 3.1 x 10^6 ages of the solar system to get to 100.

Chapter 8 contains the mathematical 'order of infinity' result that says powers grow faster than logs.

Chapter 9. Graphs and the Derivative

PlanckL.ma solves for Wein's Law of Radiation using the wavelength form of Planck's Law.

PlanckF.ma solves for Wein's Law of Radiation using the frequency form of Planck's Law.

Chapter 10. Velocity, Acceleration and Calculus

Gravity.ma contains data describing the fall of a body in vacuum. By performing operations on this data, the students can derive Galileo's Law that the acceleration of a falling body is constant (in the absence of air resistance). Students are also asked to reject Galileo's first conjecture that speed is proportional to the distance fallen.

AirResistance.ma (*see also* Scientific Projects) contains data on the fall of a body influenced by air resistance. By performing calculations with the data, students are able to calculate the coefficient of air resistance and develop a model for the fall of a body under the effects of air resistance.

Chapter 11. Maxima and Minima in One Variable

SolveEquations.ma Mathematica root finding is used in student-written NoteBooks to solve max-min problems that are intractable by hand, such as the distance from a point to a curve. See also the Geometric Optimization Exercises in the Mathematical Background.

Chapter 12. Discrete Dynamical Systems

FirstDynSys.ma solves difference equations by iterating the initial condition. It also draws graphs and shows the 'cobweb' iteration diagram.

InitialConds.ma solves difference equations with several initial conditions and shows the 'flow' of simultaneous solutions.

Whales.ma (*see also* Science Projects) solves a difference equation of order 9. This NoteBook comes from a report to the International Whaling Commission and is used in a scientific project.

Part 2: Integration in One Variable

Chapter 13. Basic Integration

Sums.ma shows how Mathematica handles the algebra of summation and contains exercises needed to build the theory of integration.

GraphIntAprx.ma shows various graphical approximations to integrals by left and middle rectangles and by trapezoids.

NumIntAprx.ma computes various numerical approximations to integrals.

Chapter 14. Symbolic Integration

SymbolicIntegr.ma shows how to use Mathematica symbolic integration and discusses several non-elementary integrals.

Chapter 15. Applications of Integration

SliceXamples is a folder of NoteBooks containing many 3-dimensional examples of slicing figures into disks, segments, etc.

Part 3: Vector Geometry

Chapter 16. Basic Vector Geometry

Vectors.ma draws vectors.

Chapter 17. Analytical Vector Geometry

Circles.ma is a basic introduction to Mathematica's ParametricPlot[] function, beginning with an animation of the parametric circle.

Cycloid.ma draws cycloids, epicycloids and hypocycloids, also known as spirographs. The idea is to trace the path of a point on a rolling wheel. The technical tools are the parametric circle and vector addition.

EpiCycAnimate.ma animates a wheel of radius 1 rolling around a wheel of radius 3. Contrary to a hasty 'intuitive' solution, the small wheel turns 4 times in one loop around the big wheel.

Chapter 18. Linear Functions and Graphs in Several Variables

Surface Slices builds up a surface plot with the curves of successive slices and shows the result in an animation.

Explicit Plots teaches you how to use the Plot3D[] function and draw the explicit surface graphs of the functions in text Exercise 15.6.

Contour Plots contains the contour graphs of the functions from text Exercise 15.6. You are to compare these with your solution to the NoteBook ExplicitSurfaces.ma. To make things interesting, one of the contour plots is deliberately WRONG. You are to find it.

FlyBy Surfaces The Mathematica function FlyBy[] helps to visualize a surface. It creates an animation that is analogous to getting in a plane and flying by the surface.

Part 4: Differentiation in Several Variables

Chapter 19. Differentiation of Functions of Several Variables

Microscopes in 3-D 'zooms' into the graph of a function of two variables. It does this for the explicit graph, the contour graph, or a density graph of the function.

Total DifflGrfs computes and plots the total differentials of the functions in Exercise 16.8.

Chapter 20. Maxima and Minima in Several Variables

Maxmin gives Mathematica solutions to the various equations on max-min examples from the text and can be used as a template for further multivariable max-min.

Part 5: Differential Equations

Chapter 21. Continuous Dynamical Systems

EulerApprox.ma shows an animation of better and better discrete "Euler" approximations to the solution of a differential equation.

EULER&Exact.ma shows a discrete Euler approximation and the exact solution of a differential equation.

AccDEsoln.ma is an accurate differential equation solver that is functionally similar to the Euler methods studied in detail by students. Without laboring the details, this is a way to get more accurate numerical solutions.

DirField.ma plots direction fields for two dimensional dynamical systems and plots 3-D vectors along the gradient tangent to a surface.

Flow1D shows an animation of a one-dimensional flow and the associated explicit solutions of a differential equation.

Flow2D makes an animated flow associated with a pair of autonomous differential equations. (It is based on a fixed step Runge–Kutta method.)

Xamples

CowSheepFlow shows a case of competition between two species.

FoxRabbitFlow This flow animation is the classical Lotka–Volterra predtor–prey system, which is studied in more detail in the projects.

Chapter 22. Autonomous Linear Dynamical Systems

UnforcdSpring.ma makes an animation of the oscillation of a spring without external forcing. The motion is given by the differential equation

$$m\,x'' + c\,x' + s\,x = 0.$$

Chapter 23. Equilibria of Continuous Dynamical Systems

Five Cases of Linear Equilibria

The following separate NoteBooks animate the main types of dynamic equilibria.

NegativeRoots. Negative characteristic roots make the origin an attractor.

OppositeRoots. Opposite sign characteristic roots make the origin attractive in one direction and repulsive in others.

RepeatedRoots. A repeated negative root makes the origin an attractor, but with one invariant line.

ImaginaryRoots. Purely imaginary roots make the origin a neutrally stable equilibrium; perturbations oscillate indefinitely.

ComplexRoots. Complex roots with negative real parts make the origin a spiral attractor.

LocalStability.ma may be used to compute the symbolic criteria for stability of equilibria in nonlinear systems.

Part 6: Infinite Series

Chapter 24. Geometric Series

ConvergSeries.ma illustrates convergence of the Taylor series of some of the most basic functions.

Chapter 25. Power Series

FormalTSeries.ma shows how to use Mathematica's built-in formal Taylor series.

BesselSeries.ma illustrates the Taylor Series approximation of Bessel function.

AbsSeries.ma The absolute value function has a series which can be built up from powers, however, this is not in the form we call "power series." This NoteBook shows an animation of this convergence. The limit of this series is not differentiable at the kink in the absolute value graph.

Chapter 26. The Edge of Convergence

FourierSeries.ma *(see also* Math Background) shows animations of convergence for several Fourier series.

Scientific Projects

co2 notebook.ma (*see* Chapter 2, Linearity vs. Local Linearity) contains data on the percentage of carbon dioxide in the atmosphere measured since 1958. The NoteBook approximates the data by a simple linear function and uses the linear function to make predictions.

This NoteBook shows the danger of using a linear approximation to project too far into the future. The linear approximation of calculus is only 'local.'

FluDataHelp.ma helps you work the project on matching the theoretical predictions of the S-I-R model to the actual 1968 Hong Kong Flu epidemic as described in the Scientific Projects chapter on Epidemiological Applications.

LadderHelp.ma is a NoteBook to help solve the project on the ladder from the Scientific Projects chapter on Applications to Mechanics.

AirResistance.ma contains data on the fall of body influenced by air resistance. By performing calculations with the data, you can calculate the coefficient of air resistance and develop a model for the fall of a body under the effects of air resistance. This is a project in the Scientific Projects chapter on Applications to Mechanics.

BungeeHelp.ma helps you to solve the project on bungee diving from the Scientific Projects chapter on Applications to Mechanics.

OpticsHelp.ma is a Mathematica package to reflect parallel 'rays of light.'

DrugData.ma is a Mathematica NoteBook for the project on fitting data for drug concentration to the biexponential model.

BytieHelp.ma helps you work the project on discrete price adjustment in the model economy the Scientific Projects on Applications to Economics.

WhaleHelp.ma solves a difference equation of order 9. This NoteBook comes from a report to the International Whaling Commission and is used to study the sustainable harvest level for this particular species of whale. See the section on sustained harvest of whales in the Scientific Projects on Applications to Ecology.

Forced Springs

SpringFriction shows the effect of varying the forcing frequency on a linear oscillator: mass, spring, and shock absorber.

Resonance shows the effect of varying the forcing frequescy on a linear oscillator. A maximal response is called a resonant frequency.

Jupiter.ma shows how to use Jupiter as a slingshot to send a satellite out of the solar system.

Lissajous.ma helps you compute trajectories of the spring pendulum from the Scientific Projects chapter on Differential Equations from Vector Geometry.

Mathematical Background

ExpDeriv.ma helps you approximate *d(b^t)/dt* directly to find Euler's $e = 2.71828 \ldots$ as the base that has constant of proportionality 1.

CoolHelp.ma examines some students' data on cooling and compares it with Newton's Law of Cooling. It is the project suggested in the Mathematica Background Section of the Canary Resurrected.

InvFctHelp.ma calculates the inverse of the Sine function directly and is intended to help you find the inverse to the function *y* = *x*^*x* needed in the project in the Mathematical Background chapter on Inverse Functions.

LeastSquares.ma goes with the Mathematical Background Chapter 9 on Least Squares fit of data.

VanderFlow.ma The Van der Pol nonlinear oscillator is mathematically very interesting because it has a single stable limit oscillation, rather than an attracting equilibrium point. This NoteBook illustrates the flow described in The Mathematical Background chapter on Theory of Initial Value Problems.

Fourier Series Animations showing convergence of several Fourier series.

Preface

Calculus is primarily important because it is the language of science. It is profound mathematics and a key to understanding physical science and engineering, but calculus also has an expanding role in economics, ecology, and some of the most quantitative parts of business and social science. Beginning college students should learn calculus, because without calculus they close the door to many scientific and technical careers. This book is intended for beginning students who want to become users of calculus in any one of these areas.

A vast majority of our students are convinced of the importance of calculus by working on problems that they find interesting. The primary goal of this new calculus curriculum is to show students first hand how calculus acts as the language of change and to answer their question, 'What good is it?' We 'show' students this by developing their basic skills and then having them apply those skills to their choice of topics from a wide variety of scientific and mathematical projects. Students themselves answer such questions as,

Why did we eradicate polio by vaccination, but not measles?

They present their solution in the form of term papers. (Three large papers and several small ones per year.) This core text does NOT stand on its own; rather it is one of four parts of our materials:

Core text:	***Calculus using Mathematica***
Science projects:	***Scientific Projects for Calculus using Mathematica***
Computing:	***Mathematica NoteBooks for Calculus using Mathematica***
Math projects:	***Mathematical Background for Calculus using Mathematica***

Computing with *Mathematica* has changed both the topics we treat and the way we present old topics. It allows us to achieve our primary goal of having students work real scientific and mathematical projects within the first semester. A number of topics formerly considered too advanced are a major part of our course. *Mathematica* can numerically solve basic differential equations and create a movie animating its 'flow.' This allows us to treat deep and important applications in a wide variety of areas (ecology, epidemiology, mechanics), while only developing basic skills about traditional exponential functions. The ingredients in studying nonlinear 2-D systems are high school math for describing the law of change and exponentials for local analysis.

Mathematica has accessible 3-D graphics, which it can also animate, so we can study problems in more than one variable in the first year. Most problems in science have more than one variable and several parameters.

Mathematica has a convenient front end editor (called NoteBooks) that helps us keep the 'intellectual overhead' to a minimum. Our intention is to use computing to study deep mathematics and applications, not to let the tail wag the dog. We weave *Mathematica* into the fabric of the course and introduce the technical features gradually.

Mathematica also helps students learn the central mathematics of calculus. Our students learn the basic skills of differentiation and integration, but also learn how to use *Mathematica* to perform very elaborate symbolic and numerical computations. We don't labor some of the esoteric 'techniques of integration,' or complicated differentiations. If students' basic skills are backed up with modern computing, it is not necessary to drill them ad nauseam in order to make them proficient mathematical thinkers and users of calculus. Our students demonstrate this in several major term papers on large projects. In addition, their performance on traditional style tests is very good (though the tests only comprise half of their grade.) Good understanding of the main computations and knowledge of how to use them with help from modern graphic, numeric and symbolic computation focuses our students' efforts on the important issues.

How Much Does it Count?

Our course grades were first derived in the following way, though we are shifting credit away from exams as we go.

Traditional Skill Exams (computers not available)

- Exam #1 -15%
- Exam #2 -15%
- Final Exam - 20%

Daily Homework (about 3/4 traditional drill and word problems) - 15%

Weekly electronic homework (submitted electronically) - 10%

Term paper length projects (almost all use computing) - 25%

Suggested Course Syllabi

Two suggested course syllabi that we have used are included in the *Mathematica* NoteBooks. The accelerated syllabus is for students who are very well prepared in high school math. We have used it at Iowa and BYU. The regular syllabus is for students with ordinary preparation for mainstream calculus in college. We have used it at Iowa, UNC and UW-L.

Acknowledgments

Many people contributed ideas and effort toward developing these materials. The National Science Foundation made it all possible.

Undergraduate students at the University of Iowa, The University of Northern Colorado, The University of Wisconsin - LaCrosse, and Brigham Young University have worked hard in response to our taking them seriously. They showed us which applications of calculus they find interesting. Almost all the projects were developed in close collaboration with students in the course or with undergraduate assistants who took the course.

Faculty and graduate students at these schools also contributed a great deal. The faculty leaders were Walter Seaman at Iowa, Steve Leth at UNC, John Unbehaun at UW-L and Gerald Armstrong at BYU. We have been blessed at Iowa with many marvelous graduate assistant teachers. Francisco Alarcon, Randy Wills, Asuman Oktac, Robert Dittmar, Monica Meissen, Srinivas Kavuri, Kathy Radloff, Robert Doucette, Oki Neswan and others helped make the course a success from the start.

Ideas from other new calculus projects also contributed to our developments. Frank Wattenberg at the University of Massachusetts was especially helpful. The philosophy of the Five College Project, which Wattenberg's materials spin off, contributed a lot to our course. Lang Moore at Duke, Elgin Johnston and Jerry Mathews at Iowa State traded ideas for student projects with us. Deborah Hughes-Hallett, Andrew Gleason and David Lomen of the Harvard project shared rough ideas and offered encouragement at many meetings. Jerry Uhl, Horacio Porta and William Davis of the *Calculus & Mathematica* project did likewise. Our calculus projects share a common goal to show students that calculus is important and thus help train the 21^{st} century's scientists, engineers, mathematicians and technical managers. We are all doing this in different ways and at different levels. The successful feature common to our new courses is: more actively involved students.

Contents