CALCULUS PROJECTS USING

M·a·t·h·e·m·a·t·i·c·a

ndrew · G. L. Cain
T. D. Morley

Calculus Projects Using *Mathematica*

Alfred D. Andrew
G. L. Cain
S. Crum
T. D. Morley

Georgia Institute of Technology

The McGraw-Hill Companies, Inc.
New York St. Louis San Francisco Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan Montreal
New Delhi San Juan Singapore Sydney Tokyo Toronto

McGraw-Hill
A Division of The **McGraw-Hill** *Companies*

Calculus Projects Using *Mathematica*

1 2 3 4 5 6 7 8 9 0 SEM SEM 9 0 9 8 7 6 5

ISBN 0-07-001790-5

The editor was Jack Shira;
the production supervisor was Annette Mayeski.
Quebecor-Semline was printer and binder.

Library of Congress Catalog Card Number: 95-81741

Calculus Projects Using *Mathematica*

To the Student

There's no question that the world we live in is changing very rapidly. The advent of modern computers, software, and scientific and programmable calculators has influenced the way practicing mathematicians, scientists, and engineers work and the problems they address. It is appropriate for these tools to influence the way we teach and learn as well.

This book contains projects based on the software package *Mathematica*, a system for doing mathematics by computer, developed by Wolfram Research, Inc. *Mathematica* is a powerful tool for numerical, graphical, and symbolic calculations. It is superb at the mechanical operations of algebra and calculus, and using *Mathematica* frees you from much of the burden of calculations, allowing you to use your mind for what it does best -- thinking, reasoning, and understanding.

The projects in this book are designed to familiarize you with *Mathematica*, to help you understand the fundamental ideas of calculus, and to show you the power which calculus brings to bear on problems in science, mathematics, and engineering. In this book you'll use calculus to understand the formation of rainbows, to study the flight of a baseball, to design some electrical circuits, to analyze an amusement park ride, to explain the reflections in a coffee cup, to design a rotary engine, and to solve many other interesting scientific problems. You'll work projects designed to enhance your understanding of mathematical topics such as curves described by parametric and polar equations, functions of several variables, and surfaces.

Don't be concerned if you don't have great familiarity with computers. There are four projects which contain an introduction to the basic operation of *Mathematica*, and there is a tutorial which you can use as well. We will, however, assume you know how to turn on (or log onto) the computers in your environment and start *Mathematica*.

Don't regard *Mathematica* as a tool to be used only on these calculus projects. It is a powerful system which you can bring to bear on your mathematics and other coursework. Its graphical capability is an extremely useful aid to understanding. We particularly recommend experimenting with plots of all kinds of functions -- these can be done easily and in large numbers.

Finally, have fun. We hope you enjoy working these projects as much as we enjoyed writing them. We welcome your suggestions, and can be reached by electronic mail at andrew@math.gatech.edu.

To the Instructor

This book contains the *Mathematica* -based projects used in calculus at the Georgia Institute of Technology, beginning with the 1991-92 Academic Year. Among our aims when writing these projects were

- To capture student interest through projects closely tied to their mathematics, science, and engineering curricula.
- To demonstrate the applicability and effectiveness of mathematics in solving clearly relevant applied problems.
- To use computing not as a gimmick but as a genuine tool on problems where it really helps.
- To teach students with the type of computing tools they would use as professionals.
- To explain mathematics from graphic, numeric, and analytic viewpoints in a pedagogically sound way.
- To encourage group learning through teamwork on open-ended projects.

We invite you to use these projects in your classes, and we welcome your criticisms, suggestions, and comments. You will find a variety of types of projects here. Some deal largely with graphical experimentations involving mathematical concepts, but almost all contain an open-ended problem in which computing plays an essential part. These projects contain *real* applications and thus use *real* numbers. One effect of this is to reduce the number of exercises (after all, the distance from the earth to the sun doesn't change much from semester to semester), but some alternative problems are available from the publisher.

These projects have been used in both large lecture formats (approximately 175 students) and small classes (45 students). In both cases the classes were taught by a faculty-teaching assistant team with three days lecture and two days recitation per week. *Mathematica* was not generally used in the classroom, but each recitation section was given at least one fifty minute class period of supervised *Mathematica* instruction each quarter. In Fall 1991 we began by using *Mathematica* projects in Calculus I classes taken by 350 students, and expanded to the point where nearly 4000 calculus students (counted according to multiplicity) were using *Mathematica* projects in their courses during the 1992-93 Academic Year.

An issue we had to face right from the start was that of transfer students -- not only from other institutions, but from "non-*Mathematica*" sections at Georgia Tech. Thus you'll find projects containing an introduction to basic *Mathematica* commands at strategic points in the book. The mathematical concepts illustrated in these introductions vary with the students' familiarity with calculus. These introductions, together with the help of teammates, very quickly converted *Mathematica* novices into *Mathematica* experts. There is also a *Mathematica* tutorial which we have used in some organized laboratory sessions.

The projects are ordered so that they may be used in conjunction with a typical four-quarter or three-semester calculus sequence. There are more than enough projects per term, and Projects 1, 2, 7, 15, and 21 contain comprehensive introductions to *Mathematica*. While most of the projects contain some descriptions of new *Mathematica* functions, there are several (Projects 3, 9, 10, 13, 17, 18, and 25, for example) which are devoted almost exclusively to the solution of open-ended problems and contain little or no new *Mathematica* code. Project 5 is a computer project for calculus which uses no calculus and no computing!

As mentioned, our students work on the projects in teams, generally of three to four students, and turn in a single written report per team for each project. There is a sample project and a sample project report to help students see what is required of them. Students can learn a lot from their friends, and explaining their ideas to team members improves their understanding and communication skills. The Georgia Tech students have been able to work on their projects in several computer laboratories equipped with *Mathematica* on Macintosh or PC compatible computers and several varieties of UNIX workstations. Students were remarkably quick to learn how to use *Mathematica*, and easily mastered aspects not originally discussed in the projects. Beginning students generally used the computer lab run by our School of Mathematics since expert help was available, while experienced students soon used other campus labs because of the more flexible hours of operation.

We acknowledge the help and support of the National Science Foundation and a great many individuals, including the many people at Georgia Tech's Office of Information Technology who helped set up and maintain computing labs. We wish to thank the thousands of Tech students who have worked these projects, and the faculty and graduate students who have taught with them. We particularly acknowledge the help of the late Dr. Robert A. Pierotti, formerly Dean of the Georgia Tech's College of Sciences and founding Director of the Georgia Tech's Center for Education Integrating Science, Mathematics, and Computing. He had a genuine concern for students, and through this Center he created an environment conducive to developing curricular innovations and improvements.

About the Authors

Fred Andrew grew up in Connecticut, and received a bachelor's degree from Yale University. After serving with the U. S. Army in Viet Nam, he attended Stanford University, from which he received a Ph.D. He is currently Professor and Associate Chairman of the School of Mathematics at Georgia Tech, where he has been a member of the mathematics department for some twenty years, interrupted only by a one year visiting lectureship at the University of California at Davis. Fred's research interests are in functional analysis, mostly in matters regarding the structure of Banach spaces. In addition to his research, he has always maintained a deep and lively interest in teaching and learning. He continues to be one of the leaders of Georgia Tech's efforts in introducing and integrating microcomputers and computer algebra systems into the basic mathematics offerings: calculus, linear algebra, and differential equations.

George Cain came North from his native North Carolina to study at M.I.T., from which institution he received a bachelor's degree. After several years as a mathematician and computer programmer at Lockheed Aircraft Corporation, he returned to graduate school and received his Ph.D. from the Georgia Institute of Technology. He is now a Professor in the School of Mathematics at Georgia Tech, where he has been a member of the faculty for almost thirty years. George's research interests have been mostly in general topology. While at Georgia Tech, he has been almost continually involved in matters of curriculum and teaching, and has a special interest in the teaching of mathematics to engineering students.

Tom Morley, born in Georgia, grew up in Maryland, in suburban Washington, D. C. After receiving a bachelor's degree from the University of Maryland, he attended Carnegie-Mellon University, where he obtained his Ph.D. After a brief stay in the mathematics department of the University of Illinois, he joined the mathematics department at Georgia Tech, where he is now a Professor. His research has been primarily in functional analysis and operations research. He has been at the forefront of the use of microcomputers in classroom settings and is a pioneer in the use of *Mathematica* in the teaching of elementary calculus and linear algebra.

Sheryl Crum earned her bachelor's degree at Lock Haven University. She also has a master's degree from the University of South Carolina and an Ed.S. from Georgia State University. She is currently a mathematics teacher at Parkview High School in Gwinett County, Georgia. Prior to joining Parkview High, she was a mathematics teacher and chair of the mathematics department at North Gwinnett High School, in Suwanee, Georgia, an instructor at Georgia State University, and a mathematics teacher in high schools in the Columbia, South Carolina, metropolitan area. During several recent summers, Sheryl has held a Georgia Industrial Fellowship for Teachers in the mathematics department at Georgia Tech.

Contents

A *Mathematica* Primer

▩ Introduction

Mathematica is a powerful system for doing mathematics by computer. This tutorial is designed to give you a brief introduction to the system, and can be used anytime as a refresher. It was written using the Macintosh version, but can be used with any graphical interface version of *Mathematica.*

While the syntax may seem unforgiving at first, *Mathematica* is amazingly consistent, and this consistency makes *Mathematica* easy to use. To see some of this consistency, notice the similarities in formats for the commands you use in this tutorial. After a short while you'll be able to "guess" the form of a new command.

This Primer is organized in a *Mathematica* notebook. In a *Mathematica* notebook you can include titles, word-processed text, live *Mathematica* code, and graphics in a single document. You can use the notebook to organize your document, just as is done here. The major topics covered are

1. Try This First -- a general introduction
2. Some Basic Operations and Functions
3. Getting Help
4. Word Processing
5. More Advanced Operations and Functions

Some things to keep in mind throughout are

1. *Mathematica* is case sensitive and **ALL *Mathematica* FUNCTIONS BEGIN WITH A CAPITAL LETTER. IF THE FUNCTION NAME IS TWO WORDS, THEY ARE BOTH CAPITALIZED.**

2. Parentheses ( ), braces { }, and brackets [ ] all have special uses. Pay attention to them.

3. Multiplication is denoted by juxtaposition (writing things next to each other), but **xy** will be interpreted as a single variable and not a product. You need a space between x and y (**x y**) to multiply x times y.

4. Commands are executed with the ENTER (or SHIFT-RETURN) key.

▩ Try This First

Notice how similar these commands look. This similarity is what makes *Mathematica* easy to use. Type the command carefully, press the ENTER (or SHIFT-RETURN) key, and notice what happens.

To plot the graph of a function

```
Plot[x^2 + 3 x, {x,-2,2}]
```

To find the definite integral of a function

```
Integrate[x^2 + 3 x,{x,-2,2}]
```

To approximate the definite integral using a numerical method

```
NIntegrate[x^2 + 3 x,{x,-2,2}]
```

To generate a **list**

```
listOfSquares = Table[j^2,{j,1,10}]
```

To plot the list

```
ListPlot[listOfSquares]
```

To plot the surface defined by a function of two variables

```
Plot3D[(1/3) x^3 - x y^2, {x,-1,1},{y,-1,1}]
```

To make a contour map of the surface defined by a function of two variables

```
ContourPlot[(1/3) x^3 - x y^2, {x,-1,1},{y,-1,1}]
```

Notice the capitalization in the next one. What do you think **Sin** and **Pi** are?

```
Play[Sin[440 2 Pi t],{t,0,2}]
```

To plot several functions on the same set of axes.

```
Plot[{Sqrt[13 x], E^x, x^2 - x},{x,0,2}]
```

Notice that

1. Parentheses ( ) are used to group terms for algebraic operations like **(1/3) x^3**
2. Braces { } group **lists** like **{x,-1,1}**
3. Brackets [ ] surround the arguments of a **function** like **Plot[...]** or **Sin[..]**.

▩ Some Basic Operations and Functions

■ Algebraic and Trigonometric Operations

Here are illustrations of basic algebraic and trigonometric manipulations. Please try them out and see what they do.

```
7 + 5 6 /3^4

(7 + 5 6 /3^4)//N

N[7 + 5 6 /3^4,25]
```

Try these symbolic operations. Notice the effect of parentheses. Multiplication and division have equal priority, but like many computer languages, *Mathematica* reads left to right.

```
a / b  c
```

The division (on the left) was done before the multiplication (on the right).

```
a / (b c)
```

The parentheses instructed *Mathematica* to do the multiplication before the division.

```
a / bc
```

Mathematica interpreted **bc** as a single variable. It can't read your mind.

```
FactorInteger[4557141]

3^4 127^1 443^1
```

For a good time, call 455-7141 in Atlanta.

```
Factor[x^2 + 3 x + 2]

Expand[(1+x) (2 + x)]

1/(x + 3)  +  2/(x + 2)
```

To put an expression over a common denominator

```
Together[1/(x + 3)  +  2/(x + 2)]
```

To expand an expression into partial fractions (break up that common denominator)

```
Apart[(8 + 3 x)/((2 + x)(3 + x))]
```

To simplify an expression

```
(1+x) (2 + x) + (1 + x) (3 + x)

Simplify[(1+x) (2 + x) + (1 + x) (3 + x)]
```

Some built-in functions

```
Sin[Pi/3]

Cos[Pi/3]

ArcSin[.45]
```

■ Calculus

□ Defining functions

There are two ways to define a function. The delayed assignment operator is := and the immediate assignment operator is =. Don't forget the **underscore**. It tells *Mathematica* what's a variable.

```
f[x_] := x Sin[x^2]
g[x_] = x^2 Sin[1/x]
h[x_] = x^3 + 2 x^2 + 1
f[2]
g[5]
g[5.]
```

"Slash dot" (/.) means "evaluate at."

```
g[x]/.{x->5}
```

□ Derivatives

Try these and see what you get. Remember that f, g, and h are defined above.

Here are some first derivatives

```
f'[x]
D[h[x],x]
```

Now some higher order derivatives

```
D[h[x],{x,2}]
D[g[x],{x,4}]
```

The next example shows some of the differences between immediate assignment (=) and delayed assignment (:=). In the second example, you're trying to differentiate g[3] with respect to 3.

```
fp[x_] = D[f[x],x]
fp[3]
gp[x_] := D[g[x],x]
gp[3]
```

□ Partial derivatives

It's wise to **Clear** function names from *Mathematica* 's memory before using them.

```
Clear[f,g,h]
```

```
f[x_,y_,z_] = x^3 y^4 z^5
```

To differentiate f[x,y,z] with respect to x (holding y and z constant)

```
D[f[x,y,z],x]
```

Here's a mixed partial derivative -- differentiate with respect to x twice, y four times, and z once.

```
D[f[x,y,z],{x,2},{y,4},{z,1}]
```

■ Solving Equations

□ Solving a single equation

Note the use of the double equals sign ==

```
Solve[3 x + 5 == 2,x]
```

```
Solve[x^2 + 3 x + 2 == 0,x]
```

□ Solving a system of equations

```
Solve[{3 x + 2 y == 5,
       2 x - 5 y == 4},{x,y}]
```

```
Solve[{x^2 + y^2 == 1, y == 9 x^2 - 5},
     {x,y}]
```

□ Multi-line statements

The last cell contained two examples of multi-line statements. Sometimes a multi-line statement is ambiguous. This usually happens when one of the lines is executable by itself. You can force *Mathematica* to continue to the next line by putting the continuation character \ at the end of the line before hitting the RETURN key. Breaking lines at a comma, as above, also forces *Mathematica* to move to the next line.

□ When "Solve" doesn't apply

Solve is going to have trouble with inverse trig functions.

```
Solve[  Sin[x] - 1/x == 0,x]
```

The *Mathematica* function **FindRoot** is basically Newton's Method. You must start it out with an estimate of where the root is located. Try to make it a good guess for the root you seek. Plotting the function first can help you here.

```
Plot[Sin[x] - 1/x,{x,.5,9}]
```

```
FindRoot[Sin[x] - 1/x, {x,6}]
```

FindRoot works on systems of equations as well. Give estimates for all variables.

```
FindRoot[{x^2 + y^2 == 1, y == 9 x^2 - 5},
     {x,.8},{y,.6}]
```

```
FindRoot[{x^2 + y^2 == 1, y == 9 x^2 - 5},
     {x,.4},{y,-.6}]
```

◻ Evaluating a function at the roots of an equation

Unless you really like typing (and making mistakes), this is worth learning.

```
Clear[f]
```

```
f[x_] = 4 x^2 + 3 x
```

```
roots = Solve[x^2 + 3 x + 2 == 0,x]
```

Notice that **roots** is the name assigned to this list of numbers.

```
roots
```

To evaluate f at the first root.

```
f[x]/.roots[[1]]
```

To evaluate f at the second root

```
f[x]/.roots[[2]]
```

To get them both at once

```
f[x]/.roots
```

■ Vectors

Vectors are just **lists**, enclosed by **braces** { }

```
v = {1,2,3}
```

```
w = {5,1,3}
```

Addition of vectors

```
v + w
```

Multiplication of a vector by a scalar is done by juxtaposition.

```
3 w
```

The **dot product** of vectors is not just juxtaposition. The **space dot space** is necessary.

```
v . w
```

▩ Getting Help

■ Type ? or ??

To learn a little bit about a *Mathematica* function, type **?FunctionName**. To learn more, type **??FunctionName**. For example