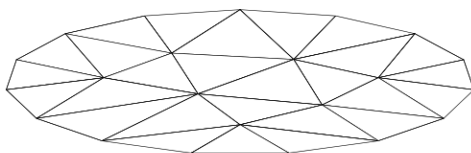AN INTRODUCTION TO

Mathematica **Programming**

FOR UNDERGRADUATE RESEARCH

Robert D. French   Dr. Samuel Jator

December 15, 2012

Contents

Preface

This book is written especially for the students of MATH 3120 and 3130 at Austin Peay State University in Clarksville, TN, but we hope it reaches a wider audience.

If you have never programmed before. . .

This pamphlet assumes you have done at least a little programming before, possibly in C, C++, FORTRAN, Visual Basic, or some other language. If not, *that is completely fine.* We will walk you through a little basic programming here so that you can get your feet wet, and then you will be ready to take on chapter 1!

1. Notebooks and Cells

Notebooks contain many cells. Cells contain code.

2. Fancy Typing and the Palette

You can make your mathematica code look more mathematical with this stuff

2.1. Subscripts, Exponents, and Fractions. For the sake of clarity, or to enhance the presentation of your code, you may want to make it look a bit more like normal mathematics. For example, to write x^2 in Mathematica, type "x" and then "Ctrl+6" and it will elevate the cursor and create a box above the x.

2.2. π, θ, and *alpha*. Using the "Esc" button, you can generate these symbols. "Esc+pi+Esc" will get you the π symbol.

2.3. Sums, Integrals, and Derivatives.

CHAPTER 1

A Review of Some Stuff You've Probably Seen Before

Before beginning this chapter, it is assumed that you know a few things about programming already. Specifically, you will need to know how to

(1) assign values to variables
(2) evaluate *Mathematica* cells

and if you don't, that's quite alright. Just see the section "*If you have never programmed before. . .*" on page , and even if you have done a bit of programming before, we will walk you through some basic *Mathematica*.

1. Variable Assignment

The first thing we need to know in any language is how to assign variables. Probably you are thinking "*I learned this in CSCI 1010!*", but *Mathematica* is a subtle language, and does not always work as you might expect if you are coming from C++ or FORTRAN.

```
i = 1;
j = 2
k := 3;
l := 4
```

Looking at this example, we can see that there are four slightly different ways to "assign" values to a variable, so let's discuss this a bit. If you put this code into a *Mathematica* cell, you will see that, upon evaluating the cell, it will output:

```
2
4
```

So, if we assigned four numbers to four variables, why do we only see two outputs? The key here is the semicolon (;) at the end of the first and third lines. This tells *Mathematica* to suppress the output of that calculation. Generally, it is appropriate to put a semicolon at the end of every line of code in a cell except for the last one. This is because you *usually* want to group your code into cells in such a way that each cell achieves one result, computes one item (or related set of items), or builds one data structure. When debugging, it can sometimes be handy to remove individual semicolons in order to investigate whether each line behaves as you expect.

Now, what is the deal with the `:=` operator? This is called the *delayed evaluation* operator. This is different from the standard `=` sign in one important way: it does not assign the value to that variable immediately. Rather, it tells *Mathematica*

to wait until your program is running[1] and then evaluate the right hand side of that expression.

What that means, in terms of the above example, is that, at the moment of evaluation, `i` and `j` are numeric variables that contain the values 1 and 2, but `k` and `l` are just symbols that do not yet contain any value. Let's investigate this:

```
Head[i]
Head[k]
```

You can see that the `Head` function returns different results for `i` and `j`. This should give you some indication that they are fundamentally different structures, even though the code used to create them looks very similar.

Delayed Evaluation is most frequently used when defining functions (See Defining Your Own Functions). For now, this is as far as we need to go with this topic. You might be thinking: "*Then why did we even bother?!*" but, the misuse of this operator is the cause of many bugs, and much of the *Mathematica* code you are likely to find on the internet contains a wild `:=` when it shouldn't. Straightening this out now will spare you headaches, and will promote friendship between you and your code.

2. Invoking Functions

Mathematica has tons of built-in functions.

---

[1]This is called *runtime.* Computer Scientists are a clever lot, aren't they?

CHAPTER 2

The Kernel, Variable State, and Scope

Before beginning this chapter, it is assumed that you know a few things about *Mathematica* programming already. Specifically, you will need to know how to

(1) assign values to variables
(2) evaluate *Mathematica* cells

and if you don't, that's quite alright. Just see the section "*If you have never programmed before...*" on page iii, and even if you have done a bit of programming before, we will walk you through some basic *Mathematica*.

1. What is the Kernel?

In *Mathematica*, all of your coding is done in a *notebook*, and all of the output of your code is displayed there as well. However, the calculations themselves are done in *an entirely separate program*[1], and this program is called a *kernel*. There are many reasons for doing these calculations in a separate program from your notebooks:

(1) You can still edit your notebooks while long computations are running
(2) You can share variables and data between notebooks
(3) You can manage multiple kernels, (and thus multiple long-running computations) from a single notebook
(4) You can run computations on multiple kernels *on other computers*

So we see that this separation of kernel and notebook is very powerful. But what does it mean in terms of your research? Specifically, while you are working on your code, the values you calculate and the variables you assign them to will be stored in your "Local Kernel". This assignment of values to variables is called "State", and it's just a fancy computer science term for "The values of your variables at a given time".

1.1. Quitting the Kernel. . This is kindof like an emergency reset for your program. Quitting the Kernel will basically erase the values for all the variables in your notebook (because they are stored in this separate program which you are about to quit).

---

[1]This is an example of a Service Oriented Architecture, and if you are interested in software engineering, it would behoove you to investigate that topic further.

Defining Your Own Functions

1. Plotting Your Functions

CHAPTER 4

Lists and Map

This chapter introduces what computer scientists refer to as *functional pro-gramming.*

CHAPTER 5

Plotting

1. ListPLot

2. 3DListPlot

Symbols, Expressions, and Replacement Rules

1. Replacement Rules

. When you solve a system of equations . . .

2. Transforming Expressions

. You can apply a set of replacement rules to an expression. . .

3. Performance Issues

. If part of your code that uses replacement rules seems to be running slowly, you can speed things up by preparing a *dispatch table*.

Finding Help in *Mathematica*

1. Documentation Center

Shift+F1.

2. Stack Overflow

One of the greatest resources for *Mathematica* help on the internet is StackOverflow/Mathematica. Also becoming very popular is mathematica.stackexchange.com, which is a separate site dedicated enitrely to *Mathematica* issues. Both of these should serve you well in your quest to find answers to your problems.

StackOverflow is a forum in which users are encouraged to give insightful answers in order to receive *points*, which amount to social capital. Thus, the answers you will find on StackOverflow are consistently of a higher quality than those you will find on other programming-related websites.

On StackOverflow, all questions are *tagged* according to which programming language or platform they pertain to. The link above will take you directly to the *Mathematica* questions, and from there you can search for more specific information about your question.

Generally, the probability that the issue you have run into when programming in *Mathematica* (or any language) is unique is very low, so the odds are in your favor that someone has encountered a very similar problem before you. Thus, the challenge is to alter your search terms judiciously until you stumble on a problem that seems to fit the issue you are dealing with.

One way to help with this process is to speculate a few guess about what the problem might be. For example, are you using some functions whose behavior you don't quite understand? Maybe you getting a weird output that doesn't look like what you think it should? Searching on StackOverflow for things like "ListPlot no graph" is more likely to get you useful results than "no graph".

2.1. Asking Questions on Stack Overflow. Accounts on StackOverflow are free, and joining this community will help you learn a great deal about both *Mathematica* and programming in general. Searching through other people's questions and the suggested answers can be very informative, and it is a good intellectual exercise to try to solve some on your own (and you may even be rewarded with profile points!).

Of course, membership on StackOverflow also allows you to post questions. Understand though that the community expects you to have done some work before you post a question. Generally, before posting new questions, it is advisable to:

(1) Try your query on a major search engine like Google or Bing. At least try all the links on the first page to see if they have anything helpful to offer.

(2) Try searching WolframAlhpa. While this is not exactly a search engine in the usual sense, its results usually contain *Mathematica* code which is occasionally helpful.

(3) Try the *Mathematica* documentation. You can access this by pressing Shift+F1 while running *Mathematica*. Specifically, you will want to look at the examples that are available on the documentation page for each function that you might have questions about. Frequently functions can take different arguments, or give different output depending on parameters or options, so it may be that you need to invoke your function in a slightly different way.

If you have done these things and still not found the answer to your question, then it will be cool to post your question on Stack Overflow. One thing to note is that while people on Stack Overflow are usually very eager to help, they need enough information about what you are trying to do to be able to understand where you might be running into trouble. Generally, posting a single line of code may not be enough. Also, keep in mind that folks on the internet are very unlikely to be familiar with your research, so it is important to track down the issue as specifically as possible.

Using GitHub to Collaborate with Teammates

If you are working on a research project with a team, you may decide simply to email the mathematica notebooks you are using back and forth every time either of you makes a change. This seems like a good idea, and indeed it is the simplest way to share changes, but it can lead to some surprisingly hairy problems very quickly.

For example, let us suppose that Nell, Ben, and Raman are on a team together. They start with one document, *research1.nb*, which contains all of their initial code that they plan to use for their project, and each person takes a copy of the document home with them. Later that week, Nell and Ben both make separate changes to the document and email their new documents to Raman. When Raman checks her email, she sees that Nell and Ben have both come up with neat ideas, but in order to get them to work together, she has to cut and paste their changes by hand. This seems straightforward enough, until Raman tries to run the code and she encounters a bug!

What might have happened? Did she make a typo while rearranging the code? Do Nell and Ben's changes conflict in some subtle way? Did the documents they emailed her match their latest changes (that is, did either Nell or Ben accidentally send an old document)? Emailing documents leads to many, many difficulties of this nature. Fortunately, programmers before us have encountered this same issue, and developed tools to solve it.

1. The Value of Version Control

Version Control is, in general, a way of keeping track of changes to files across a team of people (even if it is only a team of one!). Version Control systems are programs that you can install on your computer that will note the changes you make to your code in a way that allows you to access specific versions of your code at any time.

2. Using Git

Git is awesome.

3. Sharing Repositories on GitHub

Use GitHub or git left behind.

A Sample *Mathematica* Project: Deriving Euler's Method

Here we will look at some sample code and understand how one can derive Euler's Method for solving 1st-order ODEs. We will go through this piece by piece.