Ph3 Mathematica Homework: Week 1

Eric D. Black
California Institute of Technology

v1.3

1 Obtaining, installing, and starting Mathematica

Exercise 1: If you don't already have Mathematica, download it and install it onto your computer. Caltech has a site license, so as long as you are a student here you don't have to pay anything extra for it. Go to `http://imss.caltech.edu/software`, and log in using your IMSS username and password. Scroll down to the bottom of the page, under the section titled "Wolfram Research," and select a version of Mathematica appropriate for your computer. Follow the directions from there to download, install, and validate your copy.

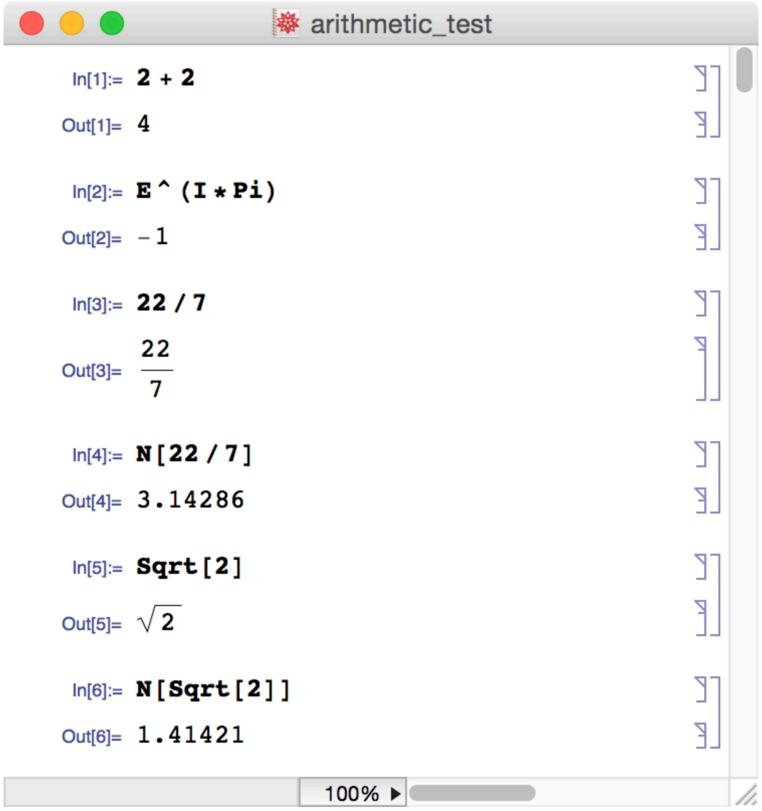
If you can't (or don't want to) install Mathematica on your own computer, you can use the computers in the lab (four iMacs and one Windows machine). Just be aware that you won't always have access to the room, and plan your work accordingly.

Mathematica documents are called *Notebooks*, and each notebook is organized into *cells.* Mathematica evaluates only what is in a given cell at any one time. It does remember the results of all of the cells it has evaluated during a session, but unlike a spreadsheet it does not update them all as soon as one cell is edited.

Exercise 2: Start Mathematica. The program opens with either a new, empty notebook or a dialog box asking you what you want to do. If you get the dialog box, click on "New Document," and a new notebook window will appear. Type in the following simple math problem, and (making sure that the cursor is still in the cell with your expression) hit either your Enter key, which is usually part of your numeric keypad, or Shift-Return. This will tell Mathematica to evaluate the expression you have just typed. (Just hitting Return makes a new line inside the cell.)

```
2+2
```

You should see the answer appear below your input, as in the first cell of Figure 1.



```
In[1]:= 2 + 2

Out[1]= 4

In[2]:= E ^ (I * Pi)

Out[2]= - 1

In[3]:= 22 / 7

         22
Out[3]= ---
         7

In[4]:= N[22 / 7]

Out[4]= 3.14286

In[5]:= Sqrt[2]

Out[5]= √2

In[6]:= N[Sqrt[2]]

Out[6]= 1.41421
```

Figure 1: A few examples of arithmetic in Mathematica. The square brackets on the right edge of the window delineate cells, and each cell has in input and an output portion, also delineated by brackets with different forms.

I've included a few other basic calculations to demonstrate how things work. Notice how Mathematica preserves the exact form of an expression whenever possible, as opposed to evaluating it and giving you a numerical approximation. You can force this evaluation using the `N` function, with the thing you want to evaluate enclosed in square brackets.

Notice also that Mathematica has built-in fundamental constants, and that they are represented with capital letters.

$$\texttt{Pi} = \pi \approx 3.14159$$

$$\texttt{I} = i = \sqrt{-1}$$

2 Constants and variables

It is a general convention that built-in (protected) symbols and functions in Mathematica start with capital letters. You can define your own variables and functions. Just be sure to use lower-case letters.

Exercise 3: Assign the following values to the variables `a` and `b`.

```
a = 22/7
b = Pi
```

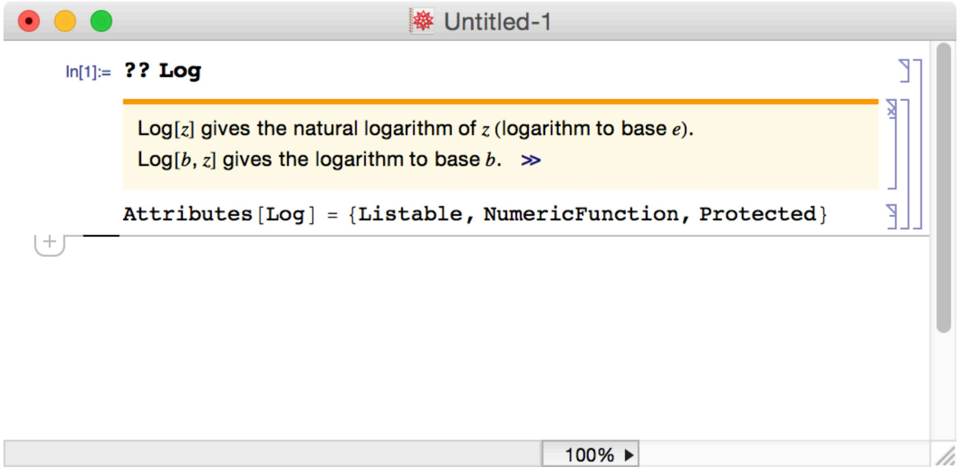
and evaluate the fractional difference between them using

```
N[(a-b)/b]
```

The capital `N[]` is a function that says, "evaluate the expression inside the square brackets, and give a numerical answer." It is a built-in or *reserved* function in Mathematica, which means it is already defined when you start up the program, just like the constants `Pi`, `E`, and `I`. Mathematica has many built-in functions, and their names are all more or less what you'd expect, except that they all start with capital letters. Some examples are `Sin[]`, `Log[]`, and `Exp[]`, but there are many others. If you know or can guess the name of a function, you can get help on its syntax with double question marks.

You can define your own functions as well, just as you can define your own constants and variables. Like constants and variables, you should use lower-case names for your functions. We'll talk more about functions and how to define them in subsequent chapters.

Always use round parentheses `()` in your mathematical expressions. Square brackets `[]` are reserved for enclosing the arguments of a function, and curly brackets `{}` format lists, which we'll talk about in a minute.



```
In[1]:= ?? Log
```

Log[*z*] gives the natural logarithm of *z* (logarithm to base *e*).
Log[*b*, *z*] gives the logarithm to base *b*. »

```
Attributes[Log] = {Listable, NumericFunction, Protected}
```

Figure 2: Double question marks followed by a function name gives you help on that function.

Exercise 4: You can specify the number of significant figures `N` returns by adding a comma and the number of sig-figs you want inside the square brackets. Evaluate the fractional difference between $\sqrt{10}$ and π to four significant figures.

```
N[(Sqrt[10] - Pi)/Pi, 4]
```

Mathematica will happily evaluate an expression to any number of significant figures. Try the following to get the answer to greater precision.

```
N[(Sqrt[10] - Pi)/Pi, 40]
N[(Sqrt[10] - Pi)/Pi, 400]
```

You don't need to type all three lines out, though you may if you prefer. If you want you can just edit the part of the cell you want to change, in this case the precision, and then re-evaluate.

3 Units and uncertainties

3.1 Units

You can assign units to a quantity using the `Quantity` command. Syntax is

```
y = Quantity[8, "Meters"]
```

which produces $y = 8$ m, indicating that Mathematica understood what you meant and is now using the abbreviation for the units you specified. You can try typing in the abbreviation in the `Quantity` argument, and sometimes Mathematica will get what you mean. Most units have the expected names, and of course Mathematica will offer to auto-complete your typing so you don't have to memorize all of the specific unit names.

You can combine units in a logical manner, and Mathematica will understand you.

```
v = Quantity[17, "Meters"/"Seconds"]
```

which will produce the output $v = 17$ m/s.

3.2 Uncertainties

Uncertainties can be incorporated into a number using `Around`. For example,

```
x = Around[5, 2]
```

will tell Mathematica that the value for x should be 5 ± 2.

You can combine the two to assign both units and uncertainties.

```
v0 = Quantity[Around[17, 2], "Meters"/"Seconds"]
```

which will produce an output of $v0 = (17.0 \pm 2.0)$ m/s.

I'm not going to give you any exercises here on this, but remember it. We will use it next week, and of course you will find it very useful in analyzing actual laboratory data.

There are many other things `Around` can do. You can use it to define asymmetric error bars, for example, but you can also use it to calculate the mean and uncertainty of a distribution or a list of numbers. All these things are useful and interesting, but they are beyond the scope of this week's handout. You can look them up if you are interested.

4 Lists

Lists are collections of numbers, and they are the structure you are going to work with the most when analyzing data. You can store them in variables just as you would a constant, and you can name your variables pretty much anything you want, as long as you stay away from capital letters.

Exercise 5: Define a list of the first five integers using the following syntax.

```
firstfive = {1, 2, 3, 4, 5}
```

You can call the value of one particular *element* of a list using double square brackets. Type this in, and verify that it returns the fifth number in the list.

```
firstfive[[5]]
```

You can put numbers directly into lists too. Replace the value of the second number with π.

```
firstfive[[2]] = Pi
```

Type the name of the list into a new cell, and hit Enter to see all its values.

```
firstfive
```

Lists can be two- or more-dimensional as well, in which case they take the form

```
data = {{x1,y1}, {x2,y2}, {x3,y3}}
```

5 The Table command

One of the most useful commands you will learn for handling data is the `Table` command. Its syntax goes like this, `Table[expr, {i, min, max}]`, where `expr` is some expression, and the index `i` goes from `i=min` to `i=max`. To construct our list of the first five integers, for example, you could enter

```
firstfive = Table[i, {i, 1, 5}]
```

You can make tables of just about anything you can code for. To construct a two-dimensional list of the first five integers (with perhaps the second one replaced by π) and their squares, enter

```
squares = Table[{firstfive[[i]], firstfive[[i]]^2}, {i, 1, 5}]
```

Mathematica displays lists in bracketed form by default. If you want to see a table in a more conventional layout, use the `TableForm` function.

```
TableForm[squares]
```

Exercise 6: Construct a two-dimensional list of the first ten integers and their square roots. Calculate their numerical values to three significant figures, and display them as a table.

6 Comments

There are two ways to add comments to a *Mathematica* notebook, by lines or by cells. To add a comment line inside an executable cell, enclose the comment in parentheses and asterisks, `(* like this *)`. To create a text-only cell, create an empty cell, and with the cursor in the cell hold down the Command key, type the number seven (`<Cmd> 7`). You can also set the format of a cell using the `Format -> Style` pull-down menu.

7 Turning in your work

Print a copy of your notebook, write your name, section number, and the date on it, and hand it in to the Mathematica TA when you come to class. You can use a comment cell for your name and section information if you like, but handwritten is ok too.

You should either bring your computer to class or, if you'd rather use the lab computers, make sure you have access to your Mathematica notebook on a thumb drive or server in case you need to correct anything.

8 Reference

1. Evaluating a cell - Place the cursor inside the cell you want to evaluate and hit "Enter" or "Shift-Return."

2. Evaluating multiple cells - Select the cell (or cells) you want evaluated, pull down the "Evaluation" menu, and select "Evaluate Cells ..."

3. Evaluating the entire notebook - Pull down the "Evaluation" menu, and choose "Evaluate Notebook."

4. Capital letters - Reserved symbols and function names usually begin with capital letters. Stick with lower-case letters when you are naming your own variables and functions.

5. Parentheses - The different types of parentheses have different meanings.

   - Round brackets, or standard parentheses `()` are used in formatting mathematical expressions. Use only these when entering mathematical expressions.
   - Square brackets `[]` are used for the arguments of functions *e.g.* `N[22/7]`, or as formatting to show the boundaries of cells.
   - Curly brackets `{}` are used in lists.
   - Double square brackets `[[]]` are used to address specific elements of a list, *e.g.* `q[[i,j]]`.