

Math 1191

Mathematica Introduction

Lab 1

Fall, 2005

Overview of Mathematica

- Starting Mathematica
 - A Mathematica session is started when you run the program from the Windows desktop (we won't cover command-line interfaces but leave that for the more geeky types)
 - Each session starts with an empty *notebook*; notebooks together with the graphical interface compose the *frontend*
 - Behind the scenes, where the calculations are actually carried out, is the *kernel*
 - Depending on the network setup, it may be possible to have a frontend running on your local machine but all the calculations carried out on another (more powerful) machine
- Notebooks and Cells
 - Mathematica documents are called *notebooks*—file extension of `.nb`
 - A notebook is made up of a series of *cells*—commands are entered in cells and output (results of computations) are displayed in their own cells. You can also enter comments or other text that may be useful for displaying the notebook.
 - Cells are indicated by a triangle bracket symbol on the right side of the notebook.
 - Multiple commands may be entered in a single cell
 - Pressing **ENTER** in a cell will *not* tell Mathematica to evaluate the expressions—it is simply a carriage return
 - To evaluate the expressions in a cell, activate the cell (place the cursor in it) and press **SHIFT-ENTER**
 - After a cell is evaluated, it is labeled with `In[n] :=` where n is the number of the input cell (always starting at 1 and increasing). The corresponding output is labeled `Out[n] :=`.
 - Cells may be grouped with other cells, and opened or closed.

- Palettes
 - Also part of the frontend, palettes contain groups of symbols for mathematical operations and Greek letters
 - Enables the user to enter mathematical expressions or characters with a WYSIWYG interface
 - May be easier to use at first, and certainly easier for another person to read the notebook, but can also be more time-consuming than simply typing the corresponding command
- The Mathematica Kernel
 - Almost all computations are carried out by the *kernel*, then sent to the front-end for display
 - All notebooks opened during a Mathematica session use the same kernel by default (though that can be changed by the user)
 - Since notebooks share a kernel, results of computations carried out in one notebook are accessible to other notebooks—this can be confusing if the two notebooks use the same variable names for different concepts
 - The kernel can be restarted (clearing all variable definitions and computation results) in lieu of restarting Mathematica

General Comments

- All built-in functions, command names and constants contain no spaces and start with a Capital letter such as `Exp[E]` and `Sin[Pi]`. Some command names are composed of multiple words; usually the second word is capitalized as well, such as `ArcSin[0]`. All names are case-sensitive.
- Arguments to functions are enclosed in brackets `[]` *not parentheses*.
- Commands that end in `;` will not produce an output.
- You can place short comments in a cell between `(*` and `*)`.
- The results of the last command can be referenced with `%`. Results of the next to last command can be referenced with `%.` And the results of output n can be referenced with `%n` or with `Out[n]`. NOTE: You should use `%` and its variants sparingly and with caution. They can be handy for ‘quick and dirty’ calculations, but can give misleading results if not used carefully.

```

In[1] := 4*2 + 1
Out[1] := 9
In[2] := Sqrt[%]
Out[2] := 3
In[3] := (%1)^2
Out[3] := 81
In[4] := Out[2] + 10
Out[4] := 13

```

- Most arithmetic operations can be entered via the keyboard or palette just as you would expect. A number adjacent to a variable is interpreted as multiplication, such as $4x$ and $-6a$. A space between quantities is also interpreted as multiplication but this use is *not recommended!*
- Results are returned in exact form (e.g., the output for $6^{(1/2)}$ is $\sqrt{6}$). Use the function `N[]` to display a numerical approximation or append `//N` to the end of a command. By default, only 6 significant digits are displayed, but you can specify how many you would like. Examples:

Input	Output
<code>N[Pi]</code>	3.14159
<code>3*7^(1/2)//N</code>	7.93725
<code>N[Pi, 25]</code>	3.141592653589793238462643

Making Assignments

Any expression—an arithmetic calculation, an algebraic expression, even a graphics object such as the plot of a function—can be named by assigning it to a variable. Variable names can be of any length, but must start with a letter. Values assigned to variables are retained throughout the entire Mathematica session. You can clear a variable with `=.` or `Clear[]`, as in `x =.` and `Clear[a]`.

Assigning variables to symbols can be done in three ways:

- With = A single equal sign `=` is an *immediate* assignment. For many situation, such as defining constants, this will be sufficient. However, the expression assigned will *not* be evaluated in later uses of the variable.

```

In[1] := a = 3;
In[2] := 5a
Out[2] := 15

```

```

In[3] := x = a; (* x is assigned the value of a *)
In[4] := a = 10;
In[5] := x
Out[5] := 3

```

- With := The operator `:=` is a *delayed* assignment. It will not place a value into the variable until the variable is actually invoked. At that point, any other variable values

or expressions will be evaluated.

```
In[1] := a = 3;
```

```
In[2] := 5a
```

```
Out[2] := 15
```

```
In[3] := x: = a; (* x will be whatever a is at the time that x is used *)
```

```
In[4] := a = 10;
```

```
In[5] := x
```

```
Out[5] := 10
```

- With -> The operator `->` is an *assignment rule*; it replaces instances of the left hand side with the right hand side, but does not change the contents of any variable. The command `/.` (read as ‘given that’) actually carries out the replacement rules. You will see commonly see assignment rules when solving equations.

```
In[1] := p = x + y + y2;
```

```
In[2] := p/.{y->2a}
```

```
Out[2] := 2a + 4a2 + x
```

```
In[3] := %/.{a->2,x->5}
```

```
Out[3] := 25
```

Lists

Lists are a common Mathematica object. They are, not surprisingly, lists of objects such as numbers or characters, or even more lists. Lists are defined using braces: `{` and `}`.

- Vectors are lists; matrices are lists of lists.
- You can perform operations on an entire list and access the n th element of list with `[[n]]`. If you want to perform the same operation on every element of a list, first try performing the operation on the list itself.
- Lists are often used as a form of grouping in function arguments.
- Several functions (such as `Solve[]`) give output in the form of a list.
- For looping commands (to repeat an action some number of times), Mathematica uses a particular kind of list call an *iterator*.

```
In[1]:= A = {1,2,3,4}
```

```
Out[1]:= {1,2,3,4}
```

```
In[2]:= A = A * 2; (* multiply each element of A by 2 *)
```

```
In[3]:= A
```

```
Out[3]:= {2,4,6,8}
```

```
In[4]:= A[[3]]
```

```
Out[4]:= 6
```

```
In[5]:= B = 5 + Range[4] (* Range[n] is a list from 1 to n *)
```

```
Out[5]:= {6,7,8,9}
```

```
In[6]:= A + B
```

```
Out[6]:= {8,11,14,17}
```

```
In[7]:= Solve[{3x + 2y == 30, 6x - y == 0}, {x,y}]
```

```
Out[7]:= {x->2,y->12}
```

```
In[8]:= mymatrix = { {1,2,3}, {4,5,6}};
```

Some Common and Useful Functions and Constants

Sqrt[x]	\sqrt{x}
Exp[x] or E ^x	e^x
Log[x]	$\ln(x)$
Sin[x], Cos[x], etc.	$\sin(x)$, $\cos(x)$, etc. (x in radians)
ArcSin[x], etc.	$\sin^{-1}(x)$, etc.
n!	$n!$ (n factorial)
Abs[x]	absolute value of x , $ x $
N[x], x//N, N[x,n]	numerical value of x (to n significant digits)
Round[x]	round x to the nearest integer
Expand[p]	multiply any products in the expression p
Factor[p]	factor the expression p
Simplify[p]	simplify the expression p
Collect[p,x]	collect powers of x in expression p
Solve[p == q, x]	<p>solve the equation $p = q$ for the variable x</p> <p>note that there are <i>two</i> equal signs</p> <p>the first argument could be a list of equations, followed by a list of variables to solve for</p> <p>Solve[] may not work for complex equations; it searches for exact algebraic solutions</p>
NSolve[p == q, x]	numerically solve the equation $p = q$ for the variable x
FindRoot[p==q, {x,a}]	<p>finds a numerical solution to the equation $p = q$ for the variable x using an initial guess of a</p> <p>useful for equations that Solve[] or NSolve[] cannot handle</p>
Prepend[t,e]	put the element e at the start of list t
Append[t,e]	put the element e at the end of the list t
Join[s,t]	join the lists s and t
Sort[t]	sort the elements of t
Range[n]	returns the list $\{1, 2, \dots, n\}$
Max[t]	returns the maximum value in the list t
Min[t]	returns the minimum value in the list t
Last[t]	returns the last element of list t
First[t]	returns the first element of list t
Pi	the constant π
E	the constant e
Infinity	∞

Some examples:

```
In[1]:= p = x^2 - 3x + 2;  
In[2]:= Solve[p == 0, x]  
Out[2]:= {x->1,x->2}
```

```
In[3]:= q = 3x - 3;  
In[4]:= Solve[p == q, x]  
Out[4]:= {x->1,x->5}
```

```
In[5]:= Solve[p==0, q==0,x]  
Out[5]:= {x->1}
```

```
In[6]:= mylist := {2a,3a,a}  
No output! Why?
```

```
In[7]:= a = 5;  
In[8]:= Sort[mylist]  
Out[8]:= {5,10,15}
```

```
In[9]:= FindRoot[x^7 - x^4 + x^2 - x + 1 == 0, {x, -1}]  
Out[9]:= -1.09278
```

Assignment

1. Factor

$$-6x^4 + 2x^3y + 18x^2y^2 - 18xy^3 + 4y^4$$

2. Solve the equation

$$x^3 - y^2 + x^2 + y - 1 = 0$$

for y in terms of x . Using only %, -> and N[] operators, determine the approximate values of y when $x = 3$.

3. The test scores for last week's Cal III test included the following scores (out of 40):

23, 31, 32, 28, 30, 25, 31, 29

- Construct a list containing these scores.
 - Five points were added to each test score. Update the scores using your list, a single arithmetic operation and assignment.
 - A late test has come in with (adjusted) score of 30. Put it into the list of test scores.
 - Sort the list and find the maximum and minimum scores as well as the median (the middle score of the nine scores).
4. Solve the system of equations

$$3x - y + z = 3$$

$$x + y - z = 2$$

$$4x + 2y + z = 1$$

5. Find a numerical solution to the equation

$$2 \sin(x^2 + 2) - \ln(x) = 2$$

6. You may need to use the Help documentation to do the following:

- Find all divisors of 44234199136.
- Find the one millionth prime number.

- Find the exact value and a numerical approximation of $\sum_{k=1}^{\infty} \frac{1}{k^2}$.