Math 615 - 09/30/2019
Introduction to Mathematica

Evaluating expressions

Begin by evaluating a simple arithmetic expression. You will type the following, and then you will *evaluate* it by hitting “shift” and “return” at the same time. Try doing other calculations.

```
In[∘]:= 1 + (17 - 5) * 2 / 5
```

Mathematica’s built-in commands all start with a CAPITAL letter. This includes built-in constants such as π and ⅇ. Try evaluating an expression containing one of these. What happens if you use “e” instead of “E”?

```
In[∘]:= 5 Pi ^ 2 - 1 / E
```

By default, *Mathematica* will keep these answers *exact*. To get a numerical approximation, use the function N.

As a general rule, parentheses () are used when you are writing algebraic expressions, brackets [] are used for functions, and braces {} are used for lists, matrices, and ranges.

```
In[∘]:= N[5 Pi ^ 2 - 1 / E]
        N[5 Pi ^ 2 - 1 / E, 20]
```

There are two other ways to do this, and they generalize to other functions.

The symbol % refers to the last output.

() // () will make Mathematica input the expression on the left into the function on the right. The following expressions all accomplish the same thing in different ways.

Oh, and *semicolons* perform a line but do not display the output.

```
In[∘]:= 5 Pi ^ 2 - 1 / E
        N[%]

        5 Pi ^ 2 - 1 / E // N

        5 Pi ^ 2 - 1 / E;
        % // N
```

Be careful, because % refers to the expression evaluated most recently, not simply the expression immediately preceding within the text. To see this, make the following all distinct cells, and then evaluate them in different orders.

```
In[∘]:=  17^(1/2)

In[∘]:=  N[%]

In[∘]:=  E^(2 Pi)
```

Expressions and functions

Complicated expressions can be easily stored as variable names. Suppose that we were solving the equation 3x^2 - 14x - 174π. The solutions are a bit cumbersome, but we can do the following.

```
In[∘]:= a1 = (14 + Sqrt[14^2 - (4) (3) (-174 Pi)]) / (2 (3))
        a2 = (14 - Sqrt[14^2 - (4) (3) (-174 Pi)]) / (2 (3))
```

Let’s check to see how this behaves.

```
In[∘]:= a1

In[∘]:= a1 + a2

In[∘]:= N[a1]

In[∘]:= 3 a1^2 - 14 a1 - 174 Pi

In[∘]:= 3 a1^2 - 14 a1 - 174 Pi // N

In[∘]:= Simplify[3 a1^2 - 14 a1 - 174 Pi]
```

Functions are defined using the following conventions.

```
In[∘]:= f[x_] := 3 x^2 - 14 x - 174 Pi

In[∘]:= f[0]

In[∘]:= f[x + h]

In[∘]:= f[a1]

In[∘]:= % // Simplify
```

You can also replace variables occurring within an expression using the command /. and the arrow ->

```
In[∘]:= g = 2 x^2 - 4

In[∘]:= g /. x → 4

In[∘]:= g /. x → (x - 2)
```

You may be tempted to do something like the following, but let’s see what goes wrong.

```
In[∘]:= x = 4

In[∘]:= g

In[∘]:= h = 120 Sin[x] - 1
```

Try opening a new notebook and creating an expression with x. Arghh! *Mathematica* will automatically replace x by 4 everywhere, even when we move on to another expression or a new window. To stop this from happening, we can clear the variable by

```
In[∘]:= Clear[x]

In[∘]:= x
```

If things get bad, you can clear everything from *Mathematica*’s memory by quitting the kernel, located in the Evaluation menu. By the way, you shouldn't define functions as "=", so let's go ahead and clear our definition of g and h from above.

```
In[∘]:= ClearAll[f, g]
```

Lists (of lists (of lists...))

Lists are probably Mathematica's most fundamental data structure. Lists can include numbers, variables, functions, and even more lists. All lists are ordered, and the i-th entry can be obtained in the following way.

```
In[∘]:= L1 = {a, b, 1, 3, c, 5}
        L2 = {7, 2, 5, 2, 5, 1, c, b}
        L1[[2]]
        L2[[4]]
```

There are lots of list functions, but here are a couple useful ones.

```
In[∘]:= Length[L1]
        Join[L1, L2]
```

Matrix basics

Mathematica works with matrices as lists of lists. Fortunately, there are nice input/output features. (At least in the newer versions of *Mathematica*), the easiest way for a novice to create a matrix is to use the "Classroom Assistant," which can be found under the "Palettes" menu. Under Basic Commands, select the matrix looking button, and then add rows and columns to create the following matrix

```
            ⎛0  1   2    3⎞
In[∘]:= A = ⎜4  5   6    7⎟
            ⎝8  9  10  100⎠
        A[[2]]
        A[[2, 3]]
```

```
In[∘]:= MatrixForm[A]
```

```
In[∘]:= RowReduce[A] // MatrixForm
```

Be careful with MatrixForm. You always want to store matrices as matrices (lists) and display them using MatrixForm, but you do not want to store them in MatrixForm.

```
In[∘]:= B = {{2, 1}, {1, 4}} // MatrixForm
        B
        RowReduce[B] // MatrixForm
        (B = {{2, 1}, {1, 4}}) // MatrixForm
        B
        RowReduce[B] // MatrixForm
```

Matrix algebra is easy. Addition uses “+“, multiplication uses “.”, and there are several built-in functions such as Det[], Transpose[], IdentityMatrix[n], and MatrixPower[A,n].
(Try using capital C for the matrix below, and you will see why it is a good idea to start variable names with lower-case letters.)

In[∘]:= $c = \begin{pmatrix} 1 & -1 \\ 0 & 4 \end{pmatrix}$

```
(A + B ) // MatrixForm
(B + c) // MatrixForm
(B.c.B) // MatrixForm
(MatrixPower[B, 3]) // MatrixForm
(Det[c]) // MatrixForm
(Transpose[A]) // MatrixForm
(B + IdentityMatrix[2]) // MatrixForm
```

Before quitting.

If you save right now, *Mathematica* will save all of the output data. For what we have done, this is not very much. However, once we introduce graphs or dynamic objects, this makes the file sizes unnecessarily bloated. It is best to click “Delete All Output” from the Cell menu and then save your file.

In fact, if you intend to use this notebook at a later date or turn it in, you should check that it will rerun correctly by deleting all the outputs and re-evaluating all the cells, which can be done quickly by clicking “Evaluate Notebook” in the Evaluation menu. This is what I will do when I grade your notebooks.

Mathematica Exercises 1

1. Evaluate the following expression to a few decimal places: $3 \sin(.5) - \sqrt{17}$

2. Evaluate π to 142 decimal places.

3. Consider the quadratic equation 5x^2 + $(17e^{-2})$ x - 2 cos(1) = 0.
Store the two solutions as “a1” and “a2”. Verify that your a1 and a2 are actually solutions to the equation.

4. Find rref for the following matrices.

$$\begin{pmatrix} 2 & -3 & 6 & 1 & -2 \\ -4 & -1 & 0 & 1 & 6 \\ 7 & 3 & -1 & 0 & 7 \\ -2 & -3 & 2 & 5 & 8 \end{pmatrix}, \begin{pmatrix} -4 & -3 & 6 & -6 & 3 & -4 \\ 1 & -4 & -7 & 3 & 2 & 5 \\ 7 & 2 & 6 & -4 & -1 & 2 \\ 8 & 6 & -3 & -6 & -4 & -1 \\ -3 & 6 & -1 & 3 & 8 & 9 \end{pmatrix}$$

Input Types

Text fields: You can enter text into a cell by declaring it to be a “Text” cell. This will prevent it from being evaluated, and hence it will not produce any unwanted error messages. Click “Format,” then “Style,” then “Text.” You can also create titles, subsections, etc.

```
In[∘]:= f[x_] := Cos[2 Pi x]
        f[1]
```

Another way to enter text is to click the “+“ sign above a cell. Notice this also gives you the option of “Free-form input.” Under this option, you type a command for *Mathematica* in plain English, and it will try to translate it into a formal *Mathematica* command.

```
In[∘]:= graph y=sin(x) from x=-2pi to x=2pi
        Plot[Sin[x], {x, -2 * Pi, 2 * Pi}]
```

Try to integrate a function.

```
In[∘]:= integral of x^2 - x^4 + e^(2 x)
        Integrate[x^2 - x^4 + E^(2 * x), x]
```

Teaching yourself Mathematica

Free-form input - This is a good way to discover possible commands and their usage. Clicking on the *Mathematica* input will automatically replace your input, and the “+“ symbol gives other possible expressions.

Classroom Assistant Palette - Open this under the Palettes menu. The “Basic Commands” contains most commands you will ever need, and the command will enter a template for you to fill in.

```
In[∘]:= Integrate[expr, var]
```

In[∘]:= $\int_{lower}^{upper} expr \, dvar$

Auto-Completion - If you begin typing a command, *Mathematica* will suggest the rest of the command. You can also see the syntax by clicking the down arrows. For example, begin typing Sum, and ...

```
In[∘]:= Sum[f, {i, imin, imax}]
```

Documentation Center - This contains all the information about a command’s usage along with examples. The sample code can be modified and rerun within the Documentation Center. For example, figure out how to graph both sin(x) and cos(x) on the same graph by looking up “Plot.”

```
In[∘]:= Plot[{Sin[x], Cos[x]}, {x, 0, 2 Pi}, PlotLegends → "Expressions"]
```

Question Mark - You can also evaluate ?Command in your workbook to see a brief description of a function and its syntax. For example, ?Plot gives the basic documentation of the Plot command.

```
In[∘]:= ? Plot
```

Graphs of functions

Function(s) of 1-variable.

```
In[∘]:= Plot[x^2, {x, -2, 2}]
```

```
In[∘]:= Plot[{x^2, 2 x - 4}, {x, -2, 2}]
```

Function(s) of 2-variables.

```
In[∘]:= Plot3D[x^2 + y^2, {x, -2, 2}, {y, -2, 2}]
```

```
In[∘]:= Plot3D[{x^2 + y^2, 2 x - 4 y }, {x, -2, 2}, {y, -2, 2}]
```

```
In[∘]:= Plot3D[{2 x - y + 5, 3 x - 2 + y, 4 x - 2 y + 10}, {x, -10, 10}, {y, -10, 10}]
```

2D and 3D Parametric Graphs

Function f: **R** -> **R^2**

```
In[∘]:= ParametricPlot[{2 Cos[t], 7 Sin[t]}, {t, 0, 2 Pi}]
```

Two functions Function f: **R** -> **R^2**

```
In[∘]:= ParametricPlot[{{2 Cos[t], 7 Sin[t]}, {Cos[t], Sin[t]}}, {t, 0, 2 Pi}]
```

Click on “More, Legends, Formulas” to get a legend like the following.

```
In[∘]:= ParametricPlot[{{2 Cos[t], 7 Sin[t]}, {Cos[t], Sin[t]}},
          {t, 0, 2 π}, PlotLegends → "Expressions"]
```

Graphing previously defined functions. Note: the semicolon; prevents the output from being printed on the screen.

```
In[∘]:= f[t_] := {2 Cos[t], 7 Sin[t]}
        g[t_] := {Cos[t], Sin[2 t]}
        ParametricPlot[f[t], {t, 0, 2 Pi}]
        ParametricPlot[{f[t], g[t]}, {t, 0, 2 Pi}]
```

Function **R -> R^3.**

```
In[•]:= ParametricPlot3D[{Cos[t], Sin[t], t}, {t, 0, 4 Pi}]
```

Some styling options.

```
In[•]:= f[x_] := x Sin[x^2 E^x];
        g[x_] := E^(Sin[3 x]);
        Plot[{f[x], g[x]}, {x, -1, 2},
         PlotStyle → {Directive[Red, Thickness[.005]], Directive[Blue, Thickness[.01]]}]
```

Visualizing spans of vectors.

Parameterizing a line.

```
In[•]:= ParametricPlot[{2 - t, 1 + 3 t}, {t, -2, 2}]
```

Using scalar multiplication.

```
In[•]:= v0 = {2, 1};
        v = {-1, 3};
        ParametricPlot[v0 + r v, {r, -2, 2}]
```

Linear combinations of two vectors give a parametric function **R^2 -> R^2**.

```
In[•]:= v1 = {3, 1};
        v2 = {1, 2};
        ParametricPlot[r v1 + s v2, {r, -2, 2}, {s, -2, 2}]
```

Visualizing span(w1), span(w2), span(w1,w2).

```
In[•]:= w1 = {1, 1, 0};
        w2 = {-1, 2, 1};
        ParametricPlot3D[ r w1, {r, -2, 2}]
        ParametricPlot3D[ r w2, {r, -2, 2}]
        ParametricPlot3D[ r w1 + s w2, {r, -2, 2}, {s, -2, 2}]
```

What is the dimension of span(v1,v2,v3) for given vectors? Let’s check graphically.

```
In[•]:= v1 = {1, 0, 0};
        v2 = {0, 1, 0};
        v3 = {0, 0, 1};
        ParametricPlot3D[{r1 v1 + r2 v2, r1 v3}, {r1, -2, 2}, {r2, -2, 2}]
```

```
In[•]:= v1 = {1, 0, 0};
        v2 = {0, 1, 0};
        v3 = {1, 1, 0};
        ParametricPlot3D[{r1 v1 + r2 v2, r1 v3}, {r1, -2, 2}, {r2, -2, 2}]
```

Some assorted useful functions

```
In[∘]:= Integrate[x^2 - x, x]
```

```
In[∘]:= Integrate[x^2 - x, {x, 0, 1}]
```

```
In[∘]:= Integrate[E^(-x^2), {x, 0, 1}]
```

```
In[∘]:= Integrate[E^(-.5 x^2), {x, 0, 1}]
```

```
In[∘]:= FactorInteger[25]
```

```
In[∘]:= Sum[k^2, {k, 1, 10}]
        Sum[Pi^k, {k, 1, 3}]
```

```
In[∘]:= Poly2 = -4 x^6 - x^5 + 3 x^4 - 2 x^3 + x^2 - 8 x + 5
        Poly2Roots = N[Roots[Poly2 == 0, x], 10]
        Rules = {ToRules[Poly2Roots]}
        Poly2 /. Rules
```

```
In[∘]:= Factor[x^3 - 25 x]
```

Manipulate Introduction

Here are three examples of showing how different parameters affect the shape of a sinusoidal curve. The first example suffers from a couple of annoying problems that are fixed in the second incarnation. The third incarnation is a slightly fancier version of the second.

```
In[∘]:= Manipulate[
         Plot[ amp Sin[ 2 Pi freq t + phase], {t, -4, 4}],
         {amp, 0, 2}, {freq, .1, 4}, {phase, -10, 10}]
```

Better example: (Take note!)

```
In[∘]:= Manipulate[
         Plot[ amp Sin[ 2 Pi freq t + phase], {t, -4, 4}, PlotRange → {-2, 2}],
         {{amp, 1}, 0, 2}, {{freq, 1}, .1, 4}, {{phase, 0}, -10, 10}  ]
```

Example with nicer labeling.

```
In[∘]:= Manipulate[
         Plot[ amp Sin[ 2 Pi freq t + phase], {t, -4, 4},
           PlotRange → {-2, 2}, PlotLabel → "Graph of y=" <> ToString[amp] <>
             " Sin(2π*" <> ToString[freq] <> "t + " <> ToString[phase] <> ")"],
         {{amp, 1, "Amplitude"}, 0, 2, Appearance → "Labeled"},
         {{freq, 1, "Frequency"}, .1, 4,  Appearance → "Labeled"},
         {{phase, 0, "Phase"}, -10, 10,  Appearance → "Labeled"}  ]
```