
 Sept. 09 2005: Lecture 2:

Introduction to Mathematica

Expressions and Evaluation

There are very many ways to learn how to use MATHEMATICA[®]. Nearly all of the best ways involve performing examples from the very beginning. That is how we are going to start—with examples. Using MATHEMATICA[®]'s FrontEnd, you may execute a command by pressing Shift-Enter.

Mathematica's syntax will feel fairly natural after a while. Use the following notebook to get started. Execute a few commands until you get a sense for what output MATHEMATICA[®] will produce; try editing the commands; try to make MATHEMATICA[®] do something strange—just try playing with it and you will soon get the hang of what is going on.

MATHEMATICA[®] Example: (notebook) Lecture-02
 MATHEMATICA[®] Example: (html) Lecture-02
 MATHEMATICA[®] Example: (mathml + xml) Lecture-02

Assignment (=) 1. Set some variables and then perform simple operations
 2. Show how the history of the session will change the variables—they try to figure out what is “current.”

Exactness and Numerical Representation 1. Symbols and rational numbers are exact.
 2. Mathematica will try to remain exact until numerical expressions get mixed in.

From the above, we see that one way to use MATHEMATICA[®] is simply as a calculator that allows symbols to get carried along. A number is not returned until all of the symbols in an expression are defined. MATHEMATICA[®] will try to be exact—it does not calculate $\frac{1}{3} + \frac{1}{2}$ by adding $0.33333\cdots + 0.5 = 0.83333\cdots$, it has an algorithm for adding rational numbers and gives $\frac{5}{6}$.

MATHEMATICA[®] Example: (notebook) Lecture-02
 MATHEMATICA[®] Example: (html) Lecture-02
 MATHEMATICA[®] Example: (mathml + xml) Lecture-02

Equality (==) 1. Equality is a logical test.

Clearing Assignments 1. Sometimes you want to use a symbol again, it is good practice to clear variables when you are through with them.
 2. Another practice (which I favor) is to use very descriptive variable names.
 3. To clear *all* assigned variables, use `Clear["Global`*"]`.

For beginners, most bugs probably crawl into MATHEMATICA[®] from not clearing symbols. Either get into the habit of clearing your definitions regularly, or use long descriptive symbol names that are unlikely to get used twice. There are other options, such as localizing variables and giving them contexts—but these are subjects better left until we have mastered the basics.

MATHEMATICA[®]’s built-in functions make it very powerful:

MATHEMATICA[®] Example: (notebook) Lecture-02
 MATHEMATICA[®] Example: (html) Lecture-02
 MATHEMATICA[®] Example: (mathml + xml) Lecture-02

Mathematica Functions

1. Identify the square-bracket syntax.
2. Demonstration of a few familiar examples.
3. Demonstrate the alternative syntax—the double forward slash.

Functions that Operate descriptions

1. Expand, Simplify
2. Show example of spelling error and point out the help feature. variable names.

Functions are really just symbols; the fact that a symbol is identified with a function comes from the square-bracket that follows it..

MATHEMATICA[®] Example: (notebook) Lecture-02
 MATHEMATICA[®] Example: (html) Lecture-02
 MATHEMATICA[®] Example: (mathml + xml) Lecture-02

Calculus

1. Examples of integration
2. The appearance of strange functions from integration
3. Demonstration of the fundamental theorem of calculus.
4. Demonstration of a function that MATHEMATICA[®] does not know how to integrate or differentiate.
5. Demonstration of derivative of integral

MATHEMATICA[®] does calculus pretty well. It can integrate functions faster and more accurately than humans and it takes derivatives. I think one of the most useful aspects of MATHEMATICA[®] is its ability to do Taylor expansions of a function about a specified point.

MATHEMATICA[®] Example: (notebook) Lecture-02
MATHEMATICA[®] Example: (html) Lecture-02
MATHEMATICA[®] Example: (mathml + xml) Lecture-02

Plotting

1. Examples of plotting a few functions.
2. Example of how to look at details that do not show up unless scale is specified.
3. An example, unexplained, but hopefully copied, of how to produce a pretty plot.

MATHEMATICA[®] has quite a few different plotting packages and this multiplies its power as a math exploration package. We will explore graphics in more depth later.

MATHEMATICA[®] Example: (notebook) Lecture-02
MATHEMATICA[®] Example: (html) Lecture-02
MATHEMATICA[®] Example: (mathml + xml) Lecture-02

Lists {,} and Matrices {{},{}}

1. A list is a way of keeping related information together.
2. The double bracket [[]] is a way to get at particular members of a list.
3. There are a number of operations operate on lists and return the list in a new form, such as Sort.
4. List elements can be selected with logical statements, such as Select.
5. A matrix is a list of lists. This idea can be logically extended to any depth.
6. A matrix is easier to view using the function MatrixForm.
7. Selecting items from a matrix is logically the same as selecting from a vector, but requires a bit more concentration and care.

Picking out parts of a matrix is a bit tedious until you get the hang of it. Remember that, when using the double bracket notation, that an integer argument will give you a position [row first, column second]. A list in either position will give you the selection(s) associated with the list. All is a special list—it is the one that spans the entire dimension associated with that row or column.

MATHEMATICA[®] Example: (notebook) Lecture-02
 MATHEMATICA[®] Example: (html) Lecture-02
 MATHEMATICA[®] Example: (mathml + xml) Lecture-02

Rules ($\rightarrow$) and Replacement ($/.$)

1. A rule $lhs \rightarrow rhs$ is *similar* to assignment in that it associates a new symbol (lhs), but it should not be as assignment—it does not effect future values of lhs .
2. The rule may be applied to lhs when it appears in an expression ($expr$) by application of a Replace operation to $expr$. Don't confuse Replace with Apply. Apply is reserved for the application of a function to arguments.
3. A rule can be applied with the function Replace, but the syntax ($.$) is typically used instead.
4. Rules can be collected into lists.
5. Rules are necessary for manipulations in MATHEMATICA[®], but can be used to generate “mistakes.” Think of Rule and Replace acting on an expression as “What would the expression be if a certain rule were applied to it?” If the rule is wrong, the resulting expression will be as well.

We will see that many of MATHEMATICA[®] functions return rules as a result.

☺ Getting Help on Mathematica

MATHEMATICA[®]'s built-in help functions are very useful. This was true even before the whole MATHEMATICA[®] manual was incorporated into the Help Browser. In the old days, one would memorize large portions of the MATHEMATICA[®] book—which has grown continuously heavier since its first publication in the early 1990's—and rely on the useful “?” and “??” operators. The use of “?” with the wildcard “*” enabled a beginning user to track down almost any MATHEMATICA[®] function. The Options function is also a very efficient way to discover alternative ways of getting results.

I would have recommended ‘scanning’ the *entire* MATHEMATICA[®] manual in a single three hour sitting (about 600 pages per hour) as an effective way to acquire a working familiarity with the software, but I don't because the built-in browser is so easy to use.

I encourage you to idly explore the MATHEMATICA[®] Help Browser. You will not only learn about MATHEMATICA[®], but also about mathematics.