Mathematica Tips, Tricks, and Techniques

When Things Go Wrong (and they will) ...

Michael A. Morrison

(Version 1.2: February 2, 2000)

Contents

1 First Things First

No matter what goes wrong, check first for these extremely common errors:

- Did I forget to start the name with a capital letter?
- Did I use parentheses instead of square brackets to enclose the arguments to the command?
- Have I misspelled the command name?
- If the name consists of more than one word, did I forget to use capital letters for all words in the name? Did I accidentally put spaces between the words?
- Did I forget to tell *Mathematica* a necessary argument, such as the variable of integration or the independent variable in an equation solving command?
- Did I type two symbols right next to one another rather than using an asterisk `*` to denote multiplication?

If your commands pass these tests, try the troubleshooting strategies in the next section.

2 Now what?

2.1 I typed in a command and hit `Enter` but *Mathematica* didn't do anything.

What's going on: `Enter` just takes you to the next line in the current cell. To send a command to the kernel, hit `Shift-Enter`.

How to fix the problem: Put the cursor anywhere in the cell that contains the command you want to execute, then hit `Shift-Enter`.

2.2 *Mathematica* has given me an incomprehensible error message.

What's going on: Almost certainty, you have a syntax error. The error is probably a bit subtle, since *Mathematica*'s error messages for simple syntax errors are reasonably clear.

How to fix the problem: Highlight the name of the command you typed (not including the argument list). Then hit `F1`. This will invoke the on-line Help for the command. Read the description of the command syntax carefully and look at one or two examples. If you can spot your error, fix it. You might try pasting one of the examples into your notebook and changing its arguments to match your task. If, however, you think you've typed the command and are using the arguments correctly and *Mathematica* still gives errors, try the following. Position the cursor in a clean cell. There retype the command, just as you initially typed it. Sometimes this works.

2.3 *Mathematica* appears to be running but won't accept input from the keyboard or the mouse.

What's going on: Either *Mathematica* is busy executing a command you entered or it's entered some kind of internal loop from which it may never emerge or it (or your operating system) is hung.

How to fix the problem: First try to abort the command by holding down the `Alt` key and hitting a period. If this doesn't work, try to quit the kernel by clicking on `Kernel:Quit Kernel:Local`. If this doesn't work, then either *Mathematica* or your system is hung. You may be able to recover by hitting `Cntl-Alt-Del` or you may have to perform a cold reboot. In either case, you'll loose all your work.

2.4 *Mathematica* returned my command unevaluated but didn't issue an error message.

What's going on: Since you didn't get an error message, your syntax is correct. So either *Mathematica* did not know the command you entered or it didn't know how to evaluate it for the input you gave. (For example, *Mathematica* can't `Solve` all polynomial equations; nor can it evaluate all integrals analytically.)

How to diagnose the problem: To find out whether *Mathematica* knows the command you entered, type a `?` then the command name *exactly as you entered it before.* If *Mathematica* types `Information::"notfound":` and then says that your command (which it calls "Symbol") is not found, then either you have entered a built-in command with a spelling error or a user-defined command that you neglected to define (or have misspelled).

How to fix the problem: First, `Clear` the command that gave you the error. This removes it from *Mathematica*'s global symbol list. If the command was one you have defined, type `?Global‘*`. *Mathematica* will produce a list that should contain the correct name of the command. If the command is a built-in one, highlight the command name and hit `F1`. *Mathematica*'s on-line help will try to find the correct command name for you. If it can't, try typing the name of the command in the dialog box next to "Go To" in the on-line help. As you type each letter, watch the screen just below this dialog box: *Mathematica* will gradually home in on the command you want. Read the documentation *carefully* to check spelling and use of arguments (i.e., whether you left out some arguments).

2.5 *Mathematica* tells me I've not supplied the right number of arguments.

What's going on: Exactly what it sounds like. *Mathematica* needs more information to perform the chore you've given it.

How to diagnose the problem: Highlight the name of the command (only—not the argument list). Then click `Input:Make Template` and *Mathematica* will print an entire argument list in your notebook.

How to fix the problem: If you know how, just give *Mathematica* the additional information and re-enter the command. If not—if you need more information or an example—just highlight the command name and hit `F1` to access the Help Browser.

2.6 I can't remember the precise name of a command I need.

How to diagnose the problem: Keep in mind that the command you recall may be a built-in ("System") command or it may come in one of the Standard Packages distributed with *Mathematica*. First try to find it among the system commands, as described below. If you can't find it, bring up the Help Browser, click on the `Master Index` radio button, and start typing whatever you remember of the command name in the box next to "Go To." To browse the packages, click the `Add Ons` button and click on "Standard Packages" in the list that appears below.

How to fix the problem: Type as much of the name as you remember, being careful to start it with a capital letter. The click on `Input:CompleteSelection`. *Mathematica* will give you a menu of all names that start with whatever you typed. Select the one you want, and access the Help Browser if you need information or examples.

2.7 I tried to plot a function, but *Mathematica* returned several lines of gibberish and then drew empty axes.

What's going on: Almost certainly the argument you gave the plot command contains one or more symbols that don't currently have numerical values. The other, more remote possibility is that your function is defined correctly but wasn't purely numerical when *Mathematica* tried to evaluate it. If you received this error message *and Mathematica plotted your function* (rather than giving you empty axes), then your function doesn't yield a finite real number at every value of the independent variable (e.g., it has a singularity or becomes complex at one or more points).

How to diagnose the problem: Look through the gibberish, searching for the phrase "is not a machine-size real number." Don't worry about any of the rest of the message. Now, in a separate cell ask *Mathematica* what it knows about every symbol in the function you're trying to plot (except the independent variables), using the `?` syntax. If this tactic doesn't reveal the culprit, position the cursor in a new cell just above the plot command, Into this cell, paste the function argument you gave the plot command. Execute the cell. Search through the output for a symbol other than the independent variables.

How to fix the problem: This is a very common error. The functions or expressions you feed to *Mathematica*'s plot commands must be entirely numerical except for the independent variables, the variables with respect to which you're plotting. To fix it, assign values to those symbols that appear in the argument but aren't numerical, either by executing immediate assignment statements before the plotting command or by using replacement rules in the argument. If this doesn't work, try wrapping the first argument to the plot command—the function you're trying to plot—in `Evaluate`.

2.8 I defined a function correctly but *Mathematica* didn't execute it correctly.

What's going on: The two most likely possibilities are these. First, that you forgot the underscore on one or more arguments in the argument list (inside square brackets) on the left-hand-side of the function definition. Second, there is a conflict between a symbol you used in the function definition and a prior use of this symbol in the notebook. *Mathematica* never forgets a symbol unless you tell it to.

How to fix the problem: This one's easy. If you forgot the underscores, just add them. Then carry out the following steps, which you should also use in the case of a conflict. Go to the cell in which you defined the function. Prior to the function definition, `Clear` the symbol you're using for the name of the function and the symbols for all the independent variables. Thus, if you're defining `{f[x_] := x^2`, then the cell should read

```
Clear[f,x]
f[x_] := x^2
```

2.9 I loaded a package which contains a command I want to execute, but when *Mathematica* loaded the package it gave me a stream of error messages about "shadowing."

What's going on: This error condition arises when you try to use a *Mathematica* command, *then* load the package that contains the command. *Mathematica* gets confused because when you first (unsuccessfully) used the command it created a symbol. Then when you loaded the package it encountered the same symbol in two different contexts—your session (the so-called `Global‘` context—and the context defined by the package.

How to fix the problem: You must first tell *Mathematica* to forget all about the symbol you inadvertently defined in your session, *then* load the package. Here are the steps:

1. Position the cursor to a new cell *just before* the `Needs` statement you used to try to import the package.
2. In the new cell, type `Remove[< >]`, inserting the name of the symbol inside the square brackets. Execute this cell.
3. Now execute the `Needs` command. The package should load with no error messages.
4. Now, delete the cell containing `Remove`.
5. Find the cell (earlier in your session) in which you tried to execute the command before loading the package. Delete this cell.

The last two steps are important; only if you do them will your package run properly next time you tried to execute it.

3 If you're trying to solve an equation ...

3.1 When I try to use one of *Mathematica*'s built-in equation solvers, I get incomprehensible error messages complaining that my equation is not "well-formed."

What's going on: Almost certainly you entered the equation using a single equal sign `=` the way you would in traditional mathematics notation, rather than the double equal sign `==` required by *Mathematica*'s syntax. (Remember that the single equal sign signifies an immediate assignment, not an equation.)

How to fix the problem: Add a second equal sign *immediately adjacent* to the one you typed, then re-execute the cell.

3.2 I've properly entered a `Solve` command but all *Mathematica* returns is an empty list!

What's going on: You've asked *Mathematica* to solve an equation it can't solve analytically. So instead of a list of solutions, it gives you an empty list. The same thing can happen, incidentally, with `NSolve`.

How to fix the problem: Try using `FindRoot` to solve the equation. First write the equation in the form

$$\text{expression} = 0. \tag{1}$$

Then use `Plot` to graph the expression. Use *Mathematica*'s coordinate locator to determine roughly where the zeros of the expression are. Feed these to `FindRoot` as initial guesses.

3.3 When I try to differentiate a function, *Mathematica* always returns zero for the derivative.

What's going on: Assuming that you specified the correct independent variable as the second argument to `D`), you probably forgot to explicitly show the functional dependence in the first argument. Suppose you've defined `f[x_] := x^2`, then type `D[f,x]`. You get zero, which is wrong, because *Mathematica* doesn't know that by `f` you mean `f[x]`.

How to fix the problem: Type `D[f[x],x]`.

3.4 I'm trying to use `DSolve` (or `NDSolve`) to solve one or more differential equations, but *Mathematica* keeps complaining that my equations are "not valid."

What's going on: Almost certainly there is a syntax error in one or more of the equations you fed *Mathematica*.

How to fix the problem: First, check that you used `==` for all equations. Then be sure you explicitly showed the functional dependence of the unknown function in all equations. That is, if one of your equations reads `y'[x] == 4x^2*y`, the solver won't work. You have to type `y[x]` on the right hand side, explicitly indicating that [y] is a function of `x`. If this isn't the problem, use `Remove` to tell *Mathematica* to forget about all the independent variables and function names, then re-execute the `Solve` command.