Mathematica Reference Sheet

Peter Williams

December 1, 2005

1 Basic Input

Operators Addition, subtraction, multiplication, division, and exponentiation are given by `+`, `-`, `*`, `/`, and `^` (caret).

Functions Standard mathematical functions are written like `ArcTan[x]`. Others include `Log[x]`, `Exp[x]`, `BesselJ[n, x]`, etc. Press `F1` with the cursor over a function name to get help on how to use it.

Output Mathematica will let you write several lines of math, separated by hitting `Enter`. To get Mathematica to actually evaluate what you've typed, you need to hit `Shift-Enter`. If you end a line that you've typed with `;` (a semicolon), Mathematica will hide the result for that line.

Precision Mathematica will avoid calculating actual numbers for as long as possible. For instance, evaluating `Log[10]` will just give you `Log[10]` back out. To force numerical evaluation, wrap your command in the `N` function: `N[Log[10]]` gives 2.30259.

Mathematica gives you a palette that you can use to type mathematical symbols that don't appear on your keyboard. It is probably worthwhile to learn the fast ways of writing them, though:

Keys	**Result**	**Keys**	**Result**	**Keys**	**Result**
`Esc a Esc`	α	`Esc b Esc`	β	`Esc g Esc`	γ
`Esc pi Esc`	π	`Esc ee Esc`	e (2.71828...)	`Esc ii Esc`	i ($\sqrt{-1}$)
`Esc inf Esc`	∞	`Esc pd Esc`	∂	`Esc elem Esc`	$\in$
`Esc dintt Esc`	$\int_?^? ?\mathrm{d}?$	`Ctrl-/`	$\frac{?}{?}$	`Ctrl-2`	$\sqrt{?}$
`Ctrl-6`	$x^?$	`Ctrl-Minus`	$x_?$	`Ctrl-Space`	Leave current group

2 Variables and Functions

There are actually two ways to set variables in Mathematica. You can of course just write `a = 3x^2`. But there is a danger to this. If `x` has been given some fixed value, that value will be embedded in the value of `a`. For instance, if you have `x = 2`, you will find `a = 12`. No problem so far. But if you then write `x = 4`, you will still find `a = 12`, *not* `a = 48`.

You can avoid this by using the second way to set a variable in Mathematica. If you write `a := 3x^2`, the value of `a` will be remembered as `3x^2` even if `x` already has a value. (If you write `a = 3x^2` and `x` hasn't been given a value, it's as if you had used the `:=` form.) Sometimes, after you've set a variable to a particular value, you realize that you'd rather not specify it. You can tell Mathematica to forget the variable with `Clear[`*var*`]`.

The `:=` form is also used to define your own functions: `f[a_, b_] := Sin[a] + Sqrt[b]`. You need to put in the underscores after the variable names to make Mathematica happy.

3 The Most Important Tools

`FullSimplify[`*expr*`]`	Simplify the expresssion *expr* as much as possible.
`Simplify[`*expr*`]`	Like the above. Tries fewer tricks but runs faster.
`Collect[`*expr*`, `*x*`]`	Collects terms in the same power of x in expression *expr*.
`TrigReduce[`*expr*`]`	Tries to combine several trig functions in *expr* into one function with a more complicated argument.

`Solve[`*eqns*`, `*vars*`]`	Tries to solve the equations *eqns* algebraically for the variables *vars*. Both *eqns* and *vars* can be single or multiple. `Solve[Sqrt[x] == 3, x]` is the simplest case, but you can also do something like `Solve[{y + 3 == x, x^2 == 3y + 2}, {x, y}]` to solve this pair of equations for `x` and `y`. It is often impossible to solve an equation algebraically, in which case you should use `FindRoot`.
`FindRoot[`*eqn*`, {`*x*`, `*x0*`}]`	Numerically solves the equation *eqn* for variable x, starting from an approximate guess of `x = x0`.

`Integrate[`*expr*`, {`*x*`, `*xmin*`, `*xmax*`}]`	Symbolically integrate *expr* as a function of x from *xmin* to *xmax*. This can be entered visually by typing `Esc dintt Esc` *xmin* `Tab` *xmax* `Tab` *expr* `Tab` *x*, which yields something that looks like $$\int_0^{10} \texttt{Exp}[-\mathrm{x}/2]\, \mathrm{dx}.$$
`Integrate[`*expr*`, `*x*`]`	As above, but does an indefinite integral.
`D[`*expr*`, `*x*`]`	Take the partial derivative of *expr* with regards to x. This can be entered visually by typing `Esc pd Esc Ctrl-Minus` *x* `Ctrl-Space` *expr*, which yields something that looks like $$\partial_x \texttt{Sin}[\mathrm{x}].$$
`Dt[`*expr*`, `*x*`]`	Take the total derivative of *expr* with regards to x. You will usually actually want to use `D` even when actually doing a total derivative, because Mathematica is anal about variables: `Dt[a*x, x]` yields `a + x Dt[a, x]`.
`Series[`*expr*`, {`*x*`, `*x0*`, `*n*`}]`	Calculate an n-term power series for the expression *expr* in terms of the variable x centered around *x0*. This is probably the most useful function in Mathematica.

`Plot[`*func*`, {`*x*`, `*xmin*`, `*xmax*`}]`	Plot *func* as a function of x ranging from *xmin* to *xmax*. You can plot multiple functions on the same graph by having *func* be a list of functions: `Plot[{Sqrt[1+x], 1+x/2}, {x,0,0.1}]` plots $\sqrt{1+x}$ and its first-order Taylor approximation near 0.
`N[`*expr*`]`	Evaluate *expr* numerically.

4 Assumptions, Replacements and Other Tricks

Mathematica is sometimes too smart for its own good. For instance, I input:

```
Integrate[Exp[-A x^2], {x, -Infinity, Infinity}]
```

Mathematica spits back something confusing and scary:

If[Re[A] > 0, $\frac{\sqrt{\pi}}{\sqrt{A}}$, Integrate[$e^{-Ax^2}$, {x, $-\infty$, ∞}, Assumptions $\rightarrow$ Re[A] $\leq$ 0]]

If you try and read the output, you might see what Mathematica has noticed: if $A < 0$, the integral is undefined. Well, nine times out of ten we know that `A` is indeed positive. We can tell Mathematica this fact by adding an `Assumptions` statement to our input like so:

```
Integrate[Exp[-A x^2], {x, -Infinity, Infinity}, Assumptions -> {A > 0}]
```

Now we get a nice and simple

$$\frac{\sqrt{\pi}}{\sqrt{A}}$$

back. Note that the pretty graphical integral doesn't let you add assumptions like this; the `Integral` function is less attractive but does have this flexibility.

You often have to do the same sort of thing when complex numbers crop up. You can assert that a variable is real by having a line that says

```
Assumptions -> {A ∈ Reals}]
```

(You get the $\in$ symbol by typing `Esc elem Esc`.) The `Integrate` and `FullSimplify` functions are the two that you'll use the most that can be told to make assumptions like this.

Mathematica has a useful feature called "replacements" that can make your life a lot easier. In short, typing

expr `/. {a -> x, b-> y}`

will replace every instance of `a` in *expr* with `x`, every instance of `b` with `y`, and so on. This is really useful because it lets you avoid cluttering your equations until you really need to. For instance, say you want to integrate

$$\int_{-\infty}^{\infty} \frac{4\pi}{\alpha} e^{-\omega^2 t^2/a_0+\alpha} \mathrm{d}t$$

Typing all this in to a Mathematica integral line would be a real pain in the butt. But we can recognize that this problem is of the form

$$\int_{-\infty}^{\infty} A e^{-Bt^2} \mathrm{d}t,$$

which is a lot cleaner to enter. If we try entering

```
i := Integrate[A Exp[-B t^2], {t, -Infinity, Infinity}, Assumptions -> {B > 0}]
```

we can save the integral in the variable `i`. We can then print out `i` to see what the general dependence on `A` and `B` is. We type in

```
i /. {A -> 4 π/aa*e^aa, B -> w^2/a0}
```

to get our particular answer. (Note that here I've typed `aa` instead of α, `w` instead of ω, and `a0` instead of a_0. The result isn't as pretty as if I had entered all the variables into Mathematica directly, but it's much faster to type. You can always do a replacement when you're done: `i /. {a ->` α`, w ->` ω`, a0 ->` a_0`}` ...) In this case, you can just set A and B to be variables, although if you want to change their values you'd have to go back and recalculate everything. But in some cases, you can't use variables. Consider evaluating a power series. We can't write

```
x = 0.3
Series[Exp[x^2 + Sin[x]], {x, 0, 3}]
```

because `x` is basically replaced with 0.3 wherever it appears, so Mathematica thinks you're trying to get a power series around the number $e^{0.3^2+\sin 0.3}$, not a function of `x`. We can avoid this by finding the series first, and then setting the value of `x` with a replacement:

```
Normal[Series[Exp[x^2 + Sin[x]], {x, 0, 3}]] /. {x -> 0.3}
```

Here `Normal` is a function that converts the output of `Series` into a polynomial expression. We then set `x` to 0.3 and we're done. (The output of `Series` *looks* like a polynomial, but it isn't, and you can't work with it like other equations. For instance, `Series[Cos[x], {x, 0, 4}] + 2` can't be evaluated.)

Note that several Mathematica functions like `FindRoot` and `Solve` return their results as replacements. This may seem a little odd but can be useful. For instance, say we want to evaluate

$$\sin(2xu) + e^{u^2x^2/3}$$

with u such that $ue^{-u/4}$ is maximized. We can accomplish this succinctly:

```
f[x_] := Sin[2x*u] + Exp[u^2x^2/3]
g[u_] := u*Exp[-u/4]
min = Solve[D[g[u], u] == 0, u]
f[x] /. min
```

The third line sets the variable `min` *to a replacement*, specifically `{u -> 4}`. The fourth line applies this replacement to our function to yield a simplified form.

Another useful thing to know is that `%` is a special symbol that means "the result of whatever command I just evaluated." For instance, if you type out some big integral that takes a long time to calculate, you can just write `ans = %` for your next line and save the result without needing to rerun the calculation.

Finally, the syntax *expr* `//` `Func` is just identical to `Func[`*expr*`]`. This can be useful if you've typed out a long line and want to `FullSimplify` it or something. You can combine this with the above feature to get two cryptic but useful commands: `% // N` and `% // FullSimplify` to numerically evaluate or simplify the result of whatever command you just ran.

5 Appendix: Installing Mathematica

- First, go to `http://www.fas.harvard.edu/cgi-bin/software/download.pl` and download a Mathematica instller for your operating system. Run it.
- The license number to enter is `L2482-2405`. Mathematica should then give you a "MathID" number that you should keep track of.
- Now go to `http://register.wolfram.com`. Say that you want to "register a product and generate a password." Enter your MathID. For the "machine name", put your FAS username.
- On the next registration page, put your "Harvard University" as your organization and "Harvard College" as your department. You don't need to give them your phone number.
- Now you need to wait for the Harvard Mathematica site administrator to email your password to you. This will take a few days. The guy is Steve Burns and he can be reached at `mathsw@fas.harvard.edu`, but give him a week or so before bothering him.
- Your password is the the fourth column of weird digits in the email that you get back from FAS. It looks something like `4218-262-283:2:8:20050801`. (Note that the last eight digits are the expiration date of your password.) Enter this into the Mathematica password window and get started!