Mathematica - quick start

Spring semester 2025

https://www.phys.uconn.edu/~rozman/Courses/P2400_25S/

Last modified: January 20, 2025

1. *Mathematica* uses capitals for the first letter of its built-in functions, commands, options, etc. There are no spaces in the command names.

   `Plot[…], Sin[…], Sqrt[…], N[…], Integrate[…]`

2. Natural log base e is `E`, imaginary unit i is `I`, π is `Pi`, and ∞ is `Infinity`.
3. Power: x^y — `x^y`
4. (a) *Mathematica* uses `[]` to enclose the argument of a function: `Sin[x]`

   (b) *Mathematica* uses `{}` to enclose the contents of a list:

   `{x, 0, Infinity}`   `{Sin[t], Cos[t]}`

   (c) `()` are only used for grouping expressions: `Sin[x/(x+3)]`

   (d) `[, {, (` must be used in balanced pairs.
5. An equality must have 2 equal signs, ==, an assignment – only one, =.
6. `N[expression]` finds the numerical value of the `expression`.

```
Sqrt[Pi]
N[Sqrt[Pi]]
```

7. Roots of equations:

```
sol = Solve[x^2 + 2a == x, x]
```

```
sol = NSolve[x^3 + 2 == x]    (* numerical solution *)
s = x /. sol[[1]]   (* assign the 1st root to variable s; note [[..]] *)
```

8. Plot Command: `Plot[functions, {x, xmin, xmax}]`

```
Plot[Sin[x], {x, 0, 2Pi}]      (* one graph *)
```

```
Plot[{Sin[x], Cos[x]}, {x, 0, 2Pi}, GridLines -> Automatic,
 PlotLegends -> "Expressions"]  (* two graphs *)
```

```
LogPlot[{x^x, Exp[x], Factorial[x]}, {x, 1, 10}, GridLines -> Automatic,
 PlotLegends -> "Expressions"]  (* semilogy plot *)
```

```
LogLinearPlot[{Tanh[x]}, {x, 1, 100}, GridLines -> Automatic]  (* semilogx plot *)
```

```
LogLogPlot[{Log[x]^x, x^x}, {x, 0.1, 10}, GridLines -> Automatic,
 PlotLegends -> "Expressions"]  (* loglog plot *)
```

```
intensity[r_, theta_] = r * (Cos[theta])^2
PolarPlot[intensity[0.4, theta], {theta, 0, 2*Pi},
 AspectRatio -> Automatic, PolarGridLines -> Automatic,
 PolarAxes -> Automatic, PolarTicks -> {"Degrees", Automatic}] (* Polar plot *)
```

9. You can define your own function with `:=`. The definition must include the underscore after the function parameter name in the left hand side of the function definition:

```
addTwo[x_] := x + 2
Plot[addTwo[x], {x, -1, 1}, Frame -> True, GridLines -> Automatic]
```

10. Derivative: using the `D` command or prime notation

```
D[Sin[x], {x, 3}]      (* third derivative with respect to x *)
```

```
Sin'[x]     (* first derivative *)
```

11. Integration: `Integrate[function[var], {var, from, to}]`

$$f(x) = \int_{-x}^{x} e^{-y^3}\, \mathrm{d}y;$$

```
f[x_] := Integrate[Exp[-y^3], {y, -x, x}]
Plot[f[x], {x, 0, 3/2}, Frame -> True, GridLines -> Automatic]
```

$$f_1(x) = \int_{0}^{x} e^{-y^3}\, \mathrm{d}y, \quad f_2(x) = \int_{0}^{x} \left(1 - e^{-y^2}\right) \mathrm{d}y$$

```
f1[x_] := NIntegrate[Exp[-y^3], {y, 0, x}]
f2[x_] := NIntegrate[1 - Exp[-y^2], {y, 0, x}]
Plot[{f1[x], f2[x]}, {x, 0, 2}, Frame -> True, GridLines -> Automatic,
  PlotLegends -> "Expressions"]
```

12. Series expansion: `Series[fun[x], {x, x0, n}]`

Generates a power series expansion for `fun` about the point `x=x0` to order $(x - x0)^n$.

```
Series[1/(Sin[x])^5, {x, 0, 2}]
```

```
res = Series[Sin[x]/(x + 2), {x, 0, 4}]
```

Truncate higher order terms:

```
Normal[res]
```

13. Ordinary differential equations

$$y' + y = \sin(x), \quad y(0) = 0;$$

```
ode = {y'[x]+y[x] == Sin[x], y[0]==0}
sol = DSolve[ode, y[x], x]
y[x] /. sol[[1]] // Expand
```

14. Numerical solution of differential equations

Initial value problem:

$$y'' + y + y^3 = 0, \quad y(0) = 1, \quad y'(0) = 0, \quad 0 \le x \le 20;$$

```
ode = {y''[x]+y[x]+y[x]^3 == 0, y[0]==1, y'[0]==0}
sol = NDSolve[ode, y[x], {x, 0, 20}]
Plot[y[x] /. sol, {x, 0, 20}, Frame -> True, GridLines -> Automatic]
```

Boundary value problem:

$$0.01y'' + 0.1xy - y + e^x = 0, \quad y(0) = 2, \quad y(1) = 1, \quad 0 \le x \le 1;$$

```
ode = {0.01 *y''[x] + 0.1*x*y[x] - y[x] + Exp[x] == 0, y[0] == 2, y[1] == 1}
sol = NDSolve[ode, y[x], {x, 0, 1}]
Plot[y[x] /. sol, {x, 0, 1}, Frame -> True, GridLines -> Automatic]
```

15. Factorization:

```
Factor[-2/3 - x^3/3 + x]
```

16. Simplification:

```
Simplify[Sin[2*x]/Sin[x]]
```

or

```
(x^3 - 1)/(x -1) // Simplify
```

17. Expand symbolic expressions into real and imaginary parts:

```
f[z_] := Exp[(2 + z)/(3 + z)]
ComplexExpand[Re[f[Exp[I*t]]]] // Simplify
```

18. Limits:

```
Limit[Sin[x]/x, x -> 0]
```

```
Limit[Sinh[x]/Cosh[x], x -> Infinity]
```