

# Mathomatic Quick Reference Sheet

\*\*\*\*\*

## Mathomatic Version 16.0.4 Command Summary

-----

approximate - Approximate all numerical values in equation spaces.

Usage: `approximate` [equation-number-ranges]

"repeat approximate" approximates more, like calculate.

calculate - Temporarily plug in values for variables and approximate well.

Usage: `calculate` ["factor"] [equation-number-range] [variable iterations]

"repeat calculate" repeatedly prompts for any input.

clear - Delete expressions stored in memory so equation spaces can be reused.

Usage: `clear` [equation-number-ranges]

Tip: Use "clear all" to quickly restart Mathomatic.

code - Output C, Java, or Python code for the specified equations.

Usage: `code` ["c" or "java" or "python" or "integer"] [equation-number-ranges]

Related commands: `simplify`, `optimize`, and `variables`

compare - Compare two equation spaces for mathematical equivalence.

Usage: `compare` ["symbolic" "approx"] equation-number ["with" equation-number]

This command may be preceded with "repeat" for full simplify.

copy - Duplicate the contents of the specified equation spaces.

Usage: `copy` ["select"] [equation-number-ranges]

With select, the first copy is made the current equation.

derivative - Symbolically differentiate and simplify, order times.

Usage: `derivative` ["nosimplify"] variable or "all" [order]

Alternate name for this command: `differentiate`

display - Display expressions in pretty, 2D multi-line fraction format.

Usage: `display` ["factor"] ["simple" or "mixed"] [equation-number-ranges]

divide - Divide 2 numbers or polynomials. Give detailed result and GCD.

Usage: `divide` [base-variable] [dividend divisor]

"repeat divide" repeatedly prompts for any input.

echo - Output a line of text, followed by a newline.

Usage: `echo` [text]

This command may be preceded with "repeat".

edit - Edit all equation spaces or an input file, then read them in.

Usage: `edit` [file-name]

Editor name in EDITOR environment variable.

eliminate - Substitute the specified variables with solved equations.

Usage: `eliminate` variables or "all" ["using" equation-number]

This command may be preceded with "repeat".

extrema - Show where the slope of the current equation equals zero.

Usage: `extrema` [variable] [order]

Helps with finding the minimums and maximums.

factor - Factor variables in equation spaces or factorize given integers.

Usage: `factor` ["number" [integers]] or ["power"] [equation-number-range] [variables]

Alternate name for this command: `collect`

for - Evaluate and display the current expression for each value of variable.

Usage: `for` variable start end [step-size]

Same syntax as the `sum` and `product` commands.

## Mathomatic Quick Reference Sheet

`fraction` - Convert expression to a single simple algebraic fraction.

Usage: `fraction` ["numerator" "denominator"] [equation-number-range]

This command may be preceded with "repeat".

`help` - Short, built-in help and reference.

Usage: `help` [topics or command-names]

Alternate name for this command: ?

`imaginary` - Fully expand and copy the imaginary part of the current expression.

Usage: `imaginary` [variable]

Related command: `real`

`integrate` - Symbolically integrate polynomials order times, then simplify.

Usage: `integrate` ["constant" or "definite"] variable [order [lower and upper-bounds]]

Alternate name for this command: `integral`

`laplace` - Compute the Laplace or inverse Laplace transform of polynomials.

Usage: `laplace` ["inverse"] variable

This command only works with polynomials.

`limit` - Take the limit as variable goes to expression.

Usage: `limit` variable expression

This limit command is experimental.

`list` - Display equation spaces in single-line (one-dimensional) format.

Usage: `list` ["export" or "maxima" or "gnuplot" or "hex"] [equation-number-ranges]

Options to export expressions to other math programs.

`nintegrate` - Do numerical definite integration using Simpson's rule.

Usage: `nintegrate` ["trapezoid"] variable [partitions [lower and upper-bounds]]

This command cannot integrate over singularities.

`optimize` - Split up equations into smaller, more efficient equations.

Usage: `optimize` [equation-number-range]

`pause` - Display a line of text and wait for user to press the Enter key.

Usage: `pause` [text]

`plot` - Automatically plot multiple expressions in 2D or 3D with gnuplot.

Usage: `plot` [equation-number-ranges] [xyz-ranges] [gnuplot-expressions,]

Plots variable x; if expression contains y, do a 3D surface plot.

`product` - Compute the product as variable goes from start to end.

Usage: `product` variable start end [step-size]

Related command: `sum`

`push` - Push equation spaces or text into readline history for editing.

Usage: `push` [equation-number-ranges or text-to-push]

Available only if readline is enabled.

`quit` - Terminate this program without saving.

Usage: `quit` [exit-value]

Alternate name for this command: `exit`

`read` - Display/change directory, or read in a text file as if it was typed in.

Usage: `read` [file-name or directory]

"repeat read" will read in a file repeatedly until failure.

`real` - Fully expand and copy the real part of the current expression.

Usage: `real` [variable]

Related command: `imaginary`

## Mathomatic Quick Reference Sheet

replace - Substitute variables in the current equation with expressions.

Usage: `replace` [variables ["with" expression]]

This command may be preceded with "repeat".

roots - Display all the roots of a complex number.

Usage: `roots` root real-part imaginary-part

"repeat roots" repeatedly prompts for any input.

save - Save all equation spaces in a text file.

Usage: `save` file-name

Related command: read

set - Display, set, or "save" current session options.

Usage: `set` [{"no"} option [value]] ...

"set" by itself will show all current option settings.

simplify - Completely simplify expressions.

Usage: `simplify` ["sign" "symbolic" "quick[est]" "fraction"] [equation-number-ranges]

This command may be preceded with "repeat" for full simplify.

solve - Solve the specified equations for a variable or for zero.

Usage: `solve` ["verify" or "verifiable"] [equation-number-range] [{"for"} expression]

The verify options check all returned solutions for correctness.

sum - Compute the summation as variable goes from start to end.

Usage: `sum` variable start end [step-size]

Related command: product

tally - Add entries, specified and prompted for, showing total.

Usage: `tally` ["average"] [equation-number-ranges]

taylor - Compute the Taylor series expansion of the current expression.

Usage: `taylor` ["nosimplify"] variable order point

unfactor - Algebraically expand (multiply out) expressions.

Usage: `unfactor` ["count" "fraction" "quick" "power"] [equation-number-range]

Alternate name for this command: expand

variables - Show all variable names used within the specified expressions.

Usage: `variables` ["c" "java" "integer" "count"] [equation-number-ranges]

Related command: code

version - Display Mathomatic version, status, and compiler information.

Usage: `version` ["status"]

End of command list. Total of 43 different commands.

\*\*\*\*\*

Help main or prompt:

-----

At the Mathomatic main prompt, you may enter:

- \* a numerical expression, which is instantly evaluated and displayed with the calculate command (autocalc) (see "help constants" and autodelete),
- \* an algebraic expression or equation, which is stored and made the current equation (see "help expressions"),
- \* a variable to solve the current equation for (autosolve),
- \* an equation number to select as the current equation (autoselect),
- \* a slash (/) or pound sign (#) followed by a variable name to search all equations spaces forward or backward for, respectively,

## Mathomatic Quick Reference Sheet

- \* a Mathomatic command (see "help all"),
- \* a question mark (?) for quick, short help (same as the help command),
- \* a semicolon (;) followed by a line comment (everything on a line after a semicolon is ignored),
- \* or an exclamation point (!) followed by a shell or system command. "!" by itself invokes the default shell. "!" is also the factorial operator.

If a colon (:) starts the line, preceding any of the above input to the main prompt, it will always return with successful status, preventing any current read command operation from aborting due to an error return status.

\*\*\*\*\*

### Help equations:

-----

To enter an expression or equation, type or copy/paste it in at the prompt. Operators have precedence decreasing as indicated:

```
! factorial      (same as gamma(x+1) function; highest precedence)
** or ^ power   (exponentiation; high precedence)
* multiply      / divide      % modulus      // integral divide
+ add           - subtract or negate
= equate        (denotes equivalence; lowest precedence)
```

Multiple operators of the same precedence level are grouped left to right. Parentheses are used to override operator precedence and group things together. Valid parentheses characters are () and {}. [] are reserved for arrays.

Variables consist of any combination of letters, digits, and underscores (\_), and "set special\_variable\_characters". Variables never start with a digit. Predefined variables follow ("help constants" shows predefined constants):

```
sign, sign1, sign2, ... - may only be ±1
integer, integer1, ... - may be any integer value
```

Absolute value notation "|x|" and dual polarity "+/-x" are understood.

\*\*\*\*\*

### Help constants:

-----

Constants are displayed as decimal, double precision floating point values, rounded to 14 decimal digits. They are displayed in standard or scientific notation, whichever is shortest. They can be entered in standard, scientific, or hexadecimal notation. Any constant can be entered in hexadecimal (base 16) by starting it with "0x". Excepting named constants, constants always start with a decimal digit (0..9), a period, or a dash (-).

Named constants follow:

```
e, ê, or e# - the universal constant e (2.718281828...)
pi or pi# - the universal constant pi (3.1415926...)
i, î, or i# - the imaginary unit ( $\sqrt{-1}$ )
```

The above constants may also be used anywhere variables are required.

```
inf - floating point infinity constant
NaN - invalid floating point result
```

Double precision floating point limits:

The largest valid constant is +/-1.797693e+308 (slightly less than  $2^{1024}$ ).  
The smallest valid constant is +/-2.225074e-308 or 0.

## Mathomatic Quick Reference Sheet

For more information, visit [www.mathomatic.org](http://www.mathomatic.org)