

Mathomatic version 16.0.5 Man Pages

George Gesslein II

Table of Contents

MATHOMATIC	1
RMATH	4
MATHO-MULT	6
MATHO-PASCAL	7
MATHO-PRIMES	8
MATHO-SUM	10
MATHO-SUMSQ	11
PRIMORIAL	12
matho_clear	13
matho_init	14
matho_parse	15
matho_process	16

NAME

mathomatic – a computer algebra system

SYNOPSIS

mathomatic [**-abcdehqrtuvwx**] [**-s** level:time] [**-m** number] [input_files or input]

DESCRIPTION

Mathomatic is a general-purpose computer algebra system (CAS) that can symbolically solve, simplify, combine, and compare algebraic equations, perform standard, complex number, modular, and polynomial arithmetic, etc. It does some calculus and handles all elementary algebra, except logarithms. Trigonometry and function expansion are supported in a separate program called **rmath**(1). Plotting expressions with **gnuplot** is also supported.

mathomatic is the main Mathomatic application that does interactive symbolic-numeric mathematics through a simple command-line interface. Readline or editline support is usually compiled into this application, making it easy to edit input and recall previous input with the cursor keys. The numeric arithmetic is double precision floating point with about 14 decimal digits accuracy. Many results will be exact, because symbolic math is an exact math, and because multiple floating point numbers can be combined for a single mathematical value; for example: $2^{(1/3)}$, which is the cube root of 2 exactly.

OPTIONS

- a** Enable alternative colors. Ansi color mode will be enabled in MS-Windows, if this option is specified and color mode is on.
- b** Enable bold colors. Color mode will be turned on and colors will be brighter if this option is specified. Same as the "set bold color" command.
- c** Toggle color mode. This mode outputs ANSI terminal escape sequences to make each level of parentheses a different color, for easier reading. Requires a terminal emulator that supports ANSI color escape sequences. If the colors are too hard to see, use the **-b** option to increase the color brightness.
- d** Set demo mode. Currently this mode only bypasses loading the startup (rc) file, and ignores the pause command. It also allows using the calculate command without prompting for the values of any of the variables.
- e** Process mathematical expressions and Mathomatic commands instead of input files on the shell command line, and then quit. Unquoted space characters are the line separators on the Mathomatic input that follows this option. Works similar to entering it into the Mathomatic main prompt, except the autoselect option is turned off. Useful for quick command-line calculations. The startup messages are not displayed with this option. Follow this option with "—" so that expressions can start with a minus sign (-).
- h** Display a brief help message listing all of these options and then exit.
- m number**
Change the memory size of equation spaces. It is followed by a decimal, floating point number which is a multiplier of the default equation space size. This allows larger equation spaces so that manipulating extremely large expressions will succeed without getting the "Expression too large" error. Specifying a number higher than 100 may make Mathomatic unresponsive.

- q** Set quiet mode. The startup messages and prompts are not displayed. This is useful when piping or redirecting input into Mathomatic, because the input won't be displayed, so prompt output should be turned off. This option does the same thing as the "set no prompt" command.
- r** Disable readline or editline input processing. Readline, and the editline drop-in replacement library, allow line input editing using the cursor keys, and output terminal control codes, all of which can be turned off with this option.
- s level:time**
Set the enforced security level for the user's Mathomatic session. Level 0 is the default with no security. Level 1 disallows shelling out (forking). Level 2 disallows shelling out and writing files. Level 3 disallows shelling out and reading/writing files. Level 4 is the highest security level and is the same as compiling with the `-DSECURE` option. This run-time option was created for use on open public servers. Specifying a colon, then a time in seconds, will time limit the application for that session.
- t** Set test mode. Used when testing and comparing output. Bypasses loading startup (rc) file, turns off color mode and readline, sets wide output mode, ignores the pause command, etc. It also allows using the calculate command without prompting for the values of any of the variables.
- u** Guarantee that standard output and standard error output are unbuffered. Also echoes all line input if not in quiet mode (`-q` option). Useful when piping.
- v** Display program name and version number, then exit successfully.
- w** Set wide output mode for an unlimited width output device like the "set wide" command does. Sets infinite screen columns and rows so that 2D (two-dimensional) expression output will always succeed and not be downgraded to 1D output when it doesn't fit in the display area. Use when redirecting output or with a terminal emulator that doesn't wrap lines. This mode only affects 2D output.
- x** Enable HTML output mode (which is also valid XHTML). This makes Mathomatic output suitable for inclusion in a web page. Color and bold mode affect this mode, allowing HTML color output. Wide output mode is also set by this option, meaning expressions will always be displayed in 2D.

GENERAL

After any options, text files may be specified on the shell command line that will be automatically read in with the read command, unless the `-e` option is specified.

Mathomatic is best run from within a terminal emulator. It uses console line input and output for the user interface. First you type in your mathematical equations in standard algebraic notation, then you can solve them by typing in the variable name at the prompt, or perform operations on them with simple English commands. Type "help" or "?" for the help command, "help examples" to get started. If the command name is longer than 4 letters, you only need to type in the first 4 letters. Most commands operate on the current equation by default.

A command preceded by an exclamation point (such as "!!s") is taken to be a shell command and is passed unchanged to the shell (/bin/sh). "!" by itself invokes the default shell, which is specified in the SHELL environment variable. "!" is also the factorial operator.

Complete documentation is available in HTML and PDF formats; see the local documentation directory or online at "<http://mathomatic.org/math/doc/>" for the latest Mathomatic documentation.

ENVIRONMENT

EDITOR

The EDITOR environment variable specifies which text editor to use for the edit command.

FILES

~/.mathomaticrc

Optional startup file containing Mathomatic set command options. It should be a text file with one or more set options per line. For example, the line "no color" will make Mathomatic default to non-color mode, which is useful if you aren't using a supported color device.

AUTHOR

Mathomatic has been written by George Gesslein II (gesslein@mathomatic.org), with help from the Internet community.

REPORTING BUGS

The command to take the limit of an expression is partially functional and experimental. All else should work perfectly; if not, please report it as a bug to the author or on the Launchpad website: "<https://launchpad.net/mathomatic>".

SEE ALSO

rmath(1), **matho-primes(1)**, **primorial(1)**, **matho-mult(1)**, **matho-sum(1)**, **matho-pascal(1)**, **matho-sumsq(1)**

NAME

rmath – a computer algebra system with functions and readline

matho – a computer algebra system with functions

SYNOPSIS

rmath [input_files]

matho [input_files]

DESCRIPTION

Mathomatic is a general-purpose computer algebra system (CAS) that can symbolically solve, simplify, combine, and compare algebraic equations, perform standard, complex number, modular, and polynomial arithmetic, etc. It does some calculus and handles all elementary algebra, except logarithms. Plotting expressions with **gnuplot** is also supported.

rmath and **matho** are shell scripts that allow you to use Mathomatic with input of functions like **sin**(x) and **sqrt**(x) automatically expanded to equivalent algebraic expressions by the **m4** macro preprocessor. A matching pair of parentheses is required around the parameters for all functions in m4 Mathomatic; m4 requires this. **rmath** also runs the **rlwrap** readline wrapper utility if available, to provide readline input editing support similar to that provided by **mathomatic**(1).

rmath and **matho** define and enable named math functions in Mathomatic. Most functions enabled here should be real number, complex number, and symbolically capable. One exception is the **abs**(x) function, which doesn't work with complex numbers, because it is defined in Mathomatic as $((x^2)^.5)$.

The following general functions are defined when using **rmath** or **matho**: **sqrt**(x), **cbrt**(x), **exp**(x), **pow**(x,y), **abs**(x), **sgn**(x), **factorial**(x), **gamma**(x), **floor**(x), **ceil**(x), **int**(x), and **round**(x).

The following standard trigonometric functions are defined: **sin**(x), **cos**(x), **tan**(x), **cot**(x), **sec**(x), and **csc**(x). **sinc**(x) is the normalized sinc function, defined as $\sin(\pi*x)/(\pi*x)$.

The following standard hyperbolic trigonometric functions are defined: **sinh**(x), **cosh**(x), **tanh**(x), **coth**(x), **sech**(x), and **csch**(x).

The following universal constants are defined: π , e , i (the imaginary unit), *euler* (the Euler-Mascheroni constant), *omega*, and *phi* (the golden ratio).

GENERAL

Text files may be specified on the shell command line that will be automatically read in through the m4 preprocessor into Mathomatic. After any files are read in, Mathomatic prompts for input from the console.

Mathomatic is best run from within a terminal emulator. It uses console line input and output for the user interface. First you type in your mathematical equations in standard algebraic notation, then you can solve them by typing in the variable name at the prompt, or perform operations on them with simple English commands. Type "help" or "?" for the help command, "help examples" to get started. If the command name is longer than 4 letters, you only need to type in the first 4 letters. Most commands operate on the current equation by default.

Complete documentation is available in HTML and PDF formats; see the local documentation directory or online at "<http://mathomatic.org/math/doc/>" for the latest Mathomatic documentation.

FILES

~/.mathomaticrc

Optional startup file containing Mathomatic set command options. It should be a text file with one or more set options per line. For example, the line "no color" will make Mathomatic default to non-color mode, which is useful if you aren't using a supported color device.

AUTHOR

Mathomatic has been written by George Gesslein II (gesslein@mathomatic.org), with help from the Internet community.

REPORTING BUGS

Please report any bugs to the author or on the Launchpad website: "<https://launchpad.net/mathomatic>".

SEE ALSO

mathomatic(1), **matho-primes(1)**, **primorial(1)**, **matho-mult(1)**, **matho-sum(1)**, **matho-pascal(1)**, **matho-sumsq(1)**

NAME

matho-mult – multiply large integers

SYNOPSIS

matho-mult [integers]

DESCRIPTION

This command-line utility is optionally part of the **mathomatic(1)** package. It uses Python to multiply many large integers separated by spaces or newlines. The size of the integers is only limited by the available memory of the computer. The single integer result is output to standard output, followed by a newline.

The integers to multiply may be specified on the command line or read from standard input.

AUTHOR

George Gesslein II (gesslein@mathomatic.org) at "<http://www.mathomatic.org>".

REPORTING BUGS

If you find a bug, please report it to the author or at "<https://launchpad.net/mathomatic>".

SEE ALSO

mathomatic(1), **primorial(1)**, **matho-sum(1)**

NAME

matho-pascal – display Pascal’s triangle

SYNOPSIS

matho-pascal [number-of-lines]

DESCRIPTION

This command-line utility is optionally part of the **mathomatic**(1) package. It calculates up to 1000 lines of Pascal’s triangle using floating point arithmetic, dumping the lines to standard output. The default is to center one screen full.

Every number inside Pascal’s triangle is the sum of the two numbers immediately above it.

Each line of Pascal’s triangle is the same as the binomial coefficients for a given power.

The sum of all numbers in each line of Pascal’s triangle is a power of 2.

AUTHOR

George Gesslein II (gesslein@mathomatic.org) at "<http://www.mathomatic.org>".

REPORTING BUGS

If you find a bug, please report it to the author or at "<https://launchpad.net/mathomatic>".

SEE ALSO

mathomatic(1), **matho-primes**(1), **matho-sumsq**(1)

NAME

matho-primes – generate consecutive prime numbers

SYNOPSIS

matho-primes [start [stop] or "all"] ["twin"] ["pal" [base]]

matho-primes [-htuv] [-c count] [-m number] [-p base] [start [stop]]

DESCRIPTION

This command-line utility is optionally part of the **mathomatic**(1) package. It quickly computes any number of consecutive prime numbers using a windowing, memory efficient sieve of Eratosthenes algorithm, dumping them to standard output. They are displayed one prime per line in ascending order, unless the "twin" option is specified, which displays only twin primes, two primes per line.

Generates up to 18 decimal digit primes, or whatever is the number of digits of precision for a floating point **long double** in the C compiler used to compile this utility. Note that this utility might be compiled to use only double precision floating point, if long double precision is not fully supported by the C compiler or hardware, allowing at most 15 decimal digit primes in that case.

Ways to verify that this utility is working are to pipe the output into the Unix "factor" utility, or compare the output with the BSD Games "primes" utility, using the supplied shell script: **examples/testprimes**.

All numbers displayed by this utility are decimal (base 10) prime numbers. A prime number is an integer that cannot be factored.

A range may be specified on the command line, otherwise the starting number and the number of primes to output is prompted for. The range is **start** to **stop** inclusive, and **stop** must be greater than or equal to **start**.

If the **-c** option is specified, the number of lines of primes displayed is limited to the decimal count that follows this option.

If the **-t** or "twin" option is specified on the command line, only **twin primes** will be displayed. Twin primes are two primes that differ in value by 2. Each twin pair is displayed together on the same line separated by a space character.

If the **-p** or "pal" option is specified on the command line, only **palindromic primes** are displayed. Palindromes are symmetrical, they read exactly the same forward and backward. The palindromic number **base** may be specified, the default is base 10. The **base** can be any integer greater than 1. Primes are always displayed in decimal (base 10).

The version number and short help on the allowed command-line parameters and usage information are displayed when given the **-h** option.

With the **-u** option, all output (standard output and standard error output) is set to be unbuffered, making all output happen immediately, instead of when the output buffer is full or when the program terminates or waits for input.

The **-m** option changes the memory size of the prime number sieve window. It is followed by a decimal, floating point number which is a multiplier of the default window size (2 megabytes). It is possible that changing the memory size may speed up the total run time a bit; otherwise there is no reason to use this option, and its use is not recommended.

The **-v** option simply displays the program name and version number, and then exits successfully.

AUTHOR

George Gesslein II (gesslein@mathomatic.org) at "<http://www.mathomatic.org>".

REPORTING BUGS

If you find a bug, please report it to the author or at "<https://launchpad.net/mathomatic>".

SEE ALSO

rmath(1), **mathomatic(1)**, **primorial(1)**, **matho-mult(1)**, **matho-sum(1)**, **matho-pascal(1)**, **matho-sumsq(1)**

NAME

matho-sum – sum large integers

SYNOPSIS

matho-sum [integers]

DESCRIPTION

This command-line utility is optionally part of the **mathomatic**(1) package. It uses Python to sum many large integers separated by spaces or newlines. The size of the integers is only limited by the available memory of the computer. The single integer result is output to standard output, followed by a newline.

The integers to sum may be specified on the command line or read from standard input.

AUTHOR

George Gesslein II (gesslein@mathomatic.org) at "<http://www.mathomatic.org>".

REPORTING BUGS

If you find a bug, please report it to the author or at "<https://launchpad.net/mathomatic>".

SEE ALSO

mathomatic(1), **primorial**(1), **matho-mult**(1)

NAME

matho-sumsq – Find the minimum sum of the squares for integers

SYNOPSIS

matho-sumsq [*numbers*]

DESCRIPTION

This command-line utility is optionally part of the **mathomatic**(1) package. It finds the minimum number of positive integers that when squared and added together, equal the given number. There is a proof that no more than 4 squares summed together are required to represent any positive integer.

The command-line may contain positive integers to find the minimum squares of, they must be less than 2147483648 (2^{31}) on 32-bit systems or less than 9223372036854775808 (2^{63}) on 64-bit systems. If "+" is appended to the given number, the program counts up from the given number. If the minimum number of squares is 2, this program displays all possible combinations with 2 squares for the given number, otherwise it just displays the first combination it finds.

If no command-line arguments are given, the programs reads the numbers from standard input.

AUTHOR

George Gesslein II (gesslein@mathomatic.org) at "<http://www.mathomatic.org>".

REPORTING BUGS

If you find a bug, please report it to the author or at "<https://launchpad.net/mathomatic>".

SEE ALSO

mathomatic(1), **matho-pascal**(1), **matho-primes**(1)

NAME

primorial – calculate large primorials

SYNOPSIS

primorial integers

DESCRIPTION

This command-line utility is optionally part of the **mathomatic(1)** package. It uses Python and **matho-primes(1)** and **matho-mult(1)** to calculate and display large primorials.

A primorial is the product of all primes up to the given integer. The integers to show the primorials of are given on the command line.

The calculated primorials are output to standard output. The size is limited by the amount of computer memory available.

AUTHOR

George Gesslein II (gesslein@mathomatic.org) at "<http://www.mathomatic.org>".

REPORTING BUGS

If you find a bug, please report it to the author or at "<https://launchpad.net/mathomatic>".

SEE ALSO

mathomatic(1), **matho-mult(1)**, **matho-primes(1)**

NAME

matho_clear - Erase all equation spaces so they can be reused

SYNOPSIS

```
void matho_clear(void);
```

DESCRIPTION

Mathomatic only has a limited number of equation spaces. Similar to a restart, recommended after each group of symbolic math operations. Currently this is the same as entering the command "clear all".

matho_init(3) must have been called only one time before this to initialize the Mathomatic symbolic math engine.

FILE

lib.c

NAME

matho_init - Initialize the Mathematic symbolic math library

SYNOPSIS

```
int matho_init(void);
```

DESCRIPTION

Call this only once before calling any Mathematic code. This must be called exactly once upon program startup and not again, unless **free_mem()** is called.

Returns true if successful. If this returns false, there was not enough memory available and Mathematic cannot be used.

FILE

lib.c

NAME

matho_parse - Process Mathomatic expression or equation input

SYNOPSIS

```
int matho_parse(char *input, char **outputp);
```

DESCRIPTION

Parse a mathematical equation or expression and store in the next available equation space, making it the current equation. Afterwards, it can be operated on by Mathomatic commands using **matho_process**(3).

matho_init(3) must have been called only one time before this to initialize the Mathomatic symbolic math engine. Use **matho_clear**(3) as many times as you want to restart Mathomatic for the next group of operations.

The *input* and output ASCII strings are expressions, if successful. The expression or equation string to enter is in "*input*", the resulting output string is stored in "**outputp*". The equation number of the equation space that the output expression is additionally stored in (if any) is available in the global "result_en", otherwise result_en = -1.

Works the same as **matho_process**(3), except commands are not allowed, so that variables are not ever confused with commands. In fact, this function is currently set to only allow entry and storage of expressions and equations.

Returns true (non-zero) if successful.

FILE

lib.c

NAME

matho_process - Process Mathematic command or expression input

SYNOPSIS

```
int matho_process(char *input, char **outputp);
```

DESCRIPTION

Process a Mathematic command or enter an expression into an equation space. The command or expression ASCII string is given as "*input*", the resulting output string is stored in "**outputp*".

matho_init(3) must have been called only one time before this to initialize the Mathematic symbolic math engine. Use **matho_clear**(3) as many times as you want to restart Mathematic for the next group of operations.

This function works just like typing something into the Mathematic prompt. To only parse any expression or equation and store it, use **matho-parse**(3).

If this returns true (non-zero), the command or *input* was successful, and the resulting expression output string is stored in "**outputp*". That is a **malloc**(ed) text string which must be **free**(d) after use to return the memory used by the string. The equation number of the equation space that the output expression is additionally stored in (if any) is available in the global "result_en", otherwise result_en = **-1**.

If this returns false, the command or *input* failed and a text error message is always stored in "**outputp*". The error message is a constant string and should NOT be **free**(d).

Some commands, like the set command, will return no output when successful, setting "**outputp*" to NULL.

The resulting output string can safely be ignored by calling this function with "*outputp*" set to NULL.

FILE

lib.c