Mathomatic User's Manual

Introduction

Mathomatic is a generalized symbolic math interpreter that can:

- combine and solve equations containing many variables;
- completely simplify and compare expressions and equations;
- do calculus transformations and series;
- perform standard, complex number, and polynomial arithmetic;
- generate efficient C, Java, or Python language code from equations;
- carry out quick calculations, etc.

Mathomatic development started in 1986 and it was originally written in Microsoft C for MS-DOS and released as shareware. It was later ported to the GNU C Compiler under Linux and became free software, by publishing under the GNU Lesser General Public License (LGPL version 2.1).

Requirements

Building from source requires a C compiler with the standard C libraries. If compiled with the GCC C compiler, no changes need to be made to the source code.

Mathomatic can easily be ported to any computer with at least 1 megabyte of free RAM. In the standard distribution, found on the Mathomatic home page, the maximum memory usage defaults to 100 megabytes (the version command tells this). Maximum memory usage is not reached unless all equation spaces are filled.

4 megabytes of disk space is required to compile, install, and run Mathomatic.

Startup

SYNOPSIS

mathomatic [options] [input_files]

To start the interactive Mathomatic interpreter, type "mathomatic" at the shell prompt.

Color mode is toggled by the "-c" option. ANSI color mode outputs ANSI escape sequences to make each level of parentheses a different color, to improve readability. ANSI color mode is on by default, requiring a terminal emulator. If the colors are too hard to see, try typing "set bold" at the Mathomatic prompt to increase the brightness.

The other options are described in the man page. After any options, file names may be specified that will be read in with the read command.

It is recommended that the name *mathomatic* be shortened to *am* for quicker access. This can be done in the Bash shell by adding the following line to your *~/.bashrc* file:

```
alias am=mathomatic
```

Then just typing "am" at the shell prompt will bring up Mathomatic. If color mode doesn't work right, use this instead:

```
alias am="mathomatic -c"
```

to turn off color. "am" stands for algebraic manipulator.

Equations and Expressions

Mathematical equations and expressions are entered into **equation spaces**. The maximum number and size of available equation spaces is displayed every time Mathomatic starts up. When an expression grows larger than half the equation space size, processing stops and the "Expression too large" message is displayed, returning you to the main prompt.

Each equation space is successively numbered with an **equation number** (starting at 1). The main prompt "1->" contains the equation number of the current equation space.

To enter an equation or expression into the first available equation space and make it the current equation, simply type it in at the main prompt. Equations consist of a Left Hand Side (LHS) and a Right Hand Side (RHS), separated by an equals sign (**=**). An equation side consists of an algebraic mathematical expression, which is a mix of constants, variables, and operators, mostly in standard infix notation. Parentheses are used to override operator precedence and group things together. Valid parentheses characters are () and {}.

Shown below is a valid equation with its parts labeled:

```
        equation
-----------------------
| variables   constant|
|--------------     | |
||     |      |     | |
 a  =  b  -  (c  +  2)
| |   |   |      |    |
| |   |   --------    |
| |   |   operators   |
---   -----------------
LHS          RHS
```

In the above equation, the variable **a** is called the dependent variable because its value depends on the variables **b** and **c**. **b** and **c** are called independent variables. In Mathomatic, any variable can be made the dependent variable by simply typing the variable name in at the prompt. This will solve the equation for that variable and, if successful, make that variable the LHS.

Here is the above equation entered into Mathomatic and solved for **b**:

```
1—> a=b-(c+2)

#1: a = b - c - 2

1—> b

#1: b = 2 + c + a

1—>
```

The "#1:" listed in front of each displayed equation always indicates the equation space number it is stored in.

Constants

All constants are stored internally as IEEE 754 standard 64-bit (8 bytes) double precision floating point numbers. They may be entered in normal, scientific, or hexadecimal notation. They are displayed in decimal (up to 14 digits) in normal or scientific notation, whichever is shortest. Results exceeding 14 digits are always rounded to 14 digits and are usually accurate from 12 to 14 digits, due to accumulated round-off error.

Excepting constants with a name (like "inf" for the infinity constant), constants always start with a decimal digit (0..9) or a period. To enter a constant in hexadecimal, prepend it with "0x".

Examples of equivalent constants follow:

Normal Notation	Scientific Notation	Hexadecimal Notation
10	1e1 (1.0 times 10^{1})	0xa
.125	1.25e-1 (1.25 times 10^{-1})	0x.2
255	2.55e2 (2.55 times 10^{2})	0xff

- The largest value of a constant is ±1.7e+308.
- The smallest value of a constant is ±2.3e-308.

The infinity constant is entered by typing "inf". Positive and negative infinity are distinct and understood, however division by zero produces one infinity value, not the two-valued ±infinity which would be more correct. Also, floating point overflow produces either positive or negative infinity.

```
1-> 1/0
Warning: Division by zero.

 answer = inf

1-> 0/0
Warning: Division by zero.

 answer = nan

1->
```

nan or *NaN* stands for **Not a Number** and it means an invalid or indeterminate floating point arithmetic result. *NaN* cannot be directly entered into Mathomatic. The appearance of *NaN* in an expression means the expression is unusable.

Fractions (such as **100/101**) are preserved if the numerator and denominator are not large. Fractions are always presented in fully reduced form; for example, **6/9** is converted to the irreducible fraction **2/3**. Constants which are exactly equal to a fraction are converted and displayed as fully reduced fractions; for example, **0.5** converts to **1/2**. Mathomatic internally converts a fraction to a single floating point value, then may convert it back to a fraction for display after all floating point arithmetic has been done.

Irrational numbers, such as **2^(1/2)**, are preserved and simplified, if possible. This can be turned off with the command "set no preserve_roots", **2^(1/2)** will then always be approximated as **1.4142135623731**.

Variables

Variables are what Mathomatic is all about. That is where the term "symbolic" comes from, because variables are symbolic in nature. They can represent known or unknown values, or any expression.

Variables consist of any combination of letters (a..z), digits (0..9), and underscores (_). They never start with a digit. By using the "set special_variable_characters" command, you can add to the allowed variable characters. By default, letters in variable names are case sensitive, meaning the alphabetic case of each letter in the variable name is important. For example, variables named "A1" and "a1" represent two different variables, unless "set no case" is entered beforehand.

The following variables are predefined and are not normal variables:

```
e or e# - the universal constant e (2.718281828…)
pi or pi# - the universal constant pi (3.1415926…)
i or i# - the imaginary number (square root of -1)
sign, sign1, sign2, … - may only be +1 or -1
```

To automatically enter multiplication by a unique, two-valued "sign" variable, precede any expression with "+/−".

Operators

Operators have precedence decreasing as indicated:

```
- negate
! factorial (gamma function)
** or ^ power (exponentiation)
* multiply      / divide       % modulus       // integral divide
+ add           - subtract
= equate
```

Multiple operators of the same precedence are evaluated left to right.

The negate operator (**-x**) may precede any expression and has the highest precedence of all operators, similar to the C programming language. This is different from other math programs, where negate has been given the same precedence as times and divide. So in Mathomatic **-2^x** will give the expected **(-2)^x**, and not **-1*(2^x)**, which are completely different.

The default operator for variables and constants is multiply (*). If a variable or constant is entered when an operator is expected, a multiply operator is automatically inserted.

The modulo operator **a % b** (spoken as "a modulo b") gives the remainder of the division **a / b**. Whether the result is always positive or has the same sign as the dividend depends on the "set true_modulus" option.

The integral divide operator **a // b** divides **a** by **b** and then truncates by zeroing the fractional part to make the result an integer. For example, **8 // 3** results in 2, which is useful when doing integer arithmetic.

Factorials (**x!**) use the gamma function (**gamma(x+1)**), so that they work with any real number, not just the positive integers.

Currently no rules of algebra are implemented for the integral divide operator (**//**) and factorials (**!**), and these two operators will not evaluate if an operand is a complex number.

Absolute value notation is allowed, **|x|** is converted to **(x^2)^.5**. This is not the same as standard absolute value where the real and imaginary parts of complex numbers are separated and then squared, but it works the same when given real number values.

Complex Numbers

Mathomatic automatically performs complex number addition, subtraction, multiplication, and division. It will also approximate roots and powers of complex numbers.

Complex numbers are in the form:

```
    a + b*i#
```

where **a** is the real part and **b** is the imaginary part. **i#** represents the square root of -1 ("-1^.5" in Mathomatic notation), and may be entered into Mathomatic as "i", it will be displayed as "i#".

The imaginary number **i#** may appear anywhere within an expression, as many times as you want, Mathomatic will handle and simplify it properly.

As an example of imaginary numbers being produced, **-2^.5** will be converted to **(2^.5)*i#**.

Roots of complex numbers, such as **i#^.5** and **.5^i#**, will be approximated, and only a single root will be produced, even though there may be many roots (see the roots command). This is called the "principal value" and may be unexpected and will often be inexact, therefore a warning is displayed when this is done.

Conjugation of all complex numbers in the current equation is accomplished by typing the following command:

```
   replace i with -i
```

Commands

Mathomatic has about 40 simple English commands that may be typed at the main prompt. Please consult the Mathomatic Command Reference, for detailed information on all commands.
