Calc Manual

Last updated, November, The 27th, 2023

Table of Contents

I. Introduction

Calc is a fully-featured calculator written in Rust for education purpose, it was designed to be minimalistic but then went off the rails and a lot of feature where implemented.

Now **Calc** is a powerful calculator capable of exact rational computation, matrix and vectors algebra, bindings to gnuplot and terminal plotting, with dozens of updates and currently (as of writing this manual) in version **2.11.4**.

If you prefer a website you may want to read **The Online Book** which is always up to date.

I.1. Install

You can install it via cargo

```bash
1  cargo install mini-calc
```
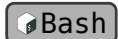
or via the source

```bash
1  git clone https://github.com/coco33920/calc
2  cd calc
3  cargo build --release
4  ./target/release/mini-calc
```
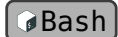
Visit **Calc** to see all the install page

I.2. Contributors

Name	Role	Website
Charlotte THOMAS	Main developer/ Maintener	Personal Page
Léana 江	Help, cleanup	Website/Blog

II. Usage

II.1. Basic operators

Calc have the basic operators which are

- `+` for the addition
- `-` for the substraction
- `*` for the multiplication
- `/` for the division (or for a rational)
- `^` for the exponentation
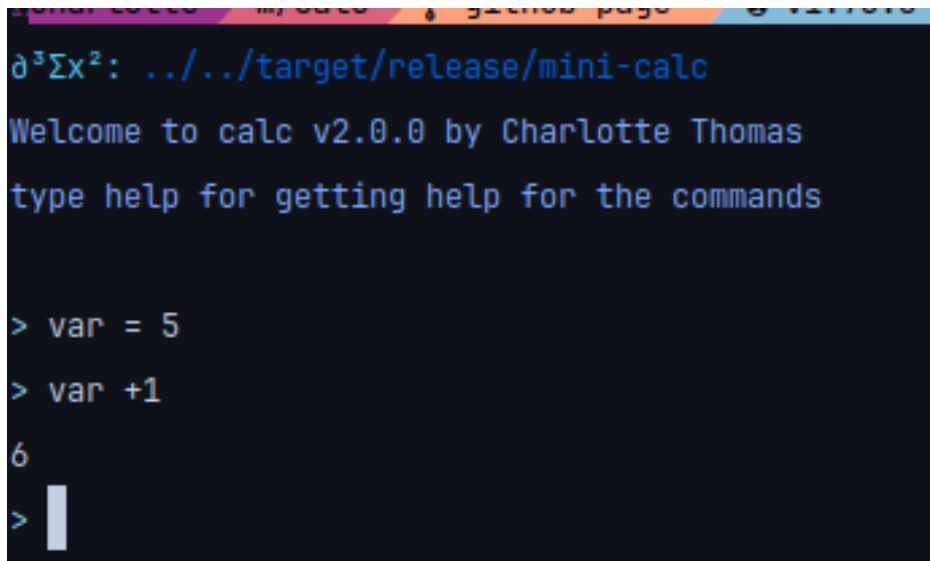
II.2. Variables

It also supports variable the syntax is

```
1   myvar = value
```

for example

```
1   var = (2+2)
```


Figure 1: Example of setting a variable

II.3. Built-in variables

The following variables are built-in:

- `pi` is pi as a double precision float
- `e` is e as a double precision float

III. Functions

III.1. Implemented

The following functions are currently implemented:

Trigonometry

- sin (vectorized)
- cos (vectorized)
- tan (vectorized)

Hyperbolic trigonometry

- sinh (vectorized)
- cosh (vectorized)
- tanh (vectorized)

Reverse trigonometry

- acos (vectorized)
- asin (vectorized)
- atan (vectorized)

Exponentiation

- exp (vectorized)
- ln (alias: log) (vectorized)

Vectors

- norm

Matrices

- det
- invert

Plot

- plot
- termplot

Other

- sqrt (vectorized)
- factorial (alias: fact)
- abs
- ceil
- floor
- round

III.2. Trigonometry

For trigonometry, the input is assumed to be in radians, if it is in degrees you need to add `false` or `true` as a second argument, example shown bellow.

```
charlotte    …/calc    add_functions !    v1.73.0    16:03    11s
∂³Σx²: ../../target/release/mini-calc
Welcome to Calc v2.3.0 by Charlotte Thomas, type help for help
> help
 Calc v2.3.0 Help
 > info : show infos
 > exit : exit the program
 > help : print this help
 > verbose : toggle the verbose
 > version : prints the version

> cos(pi)
-1
> cos(180,true)
-1
> cos(180)
-0.5984600690578581
>
```

Figure 2: Usage of trigonometry

III.3. Exp/ln
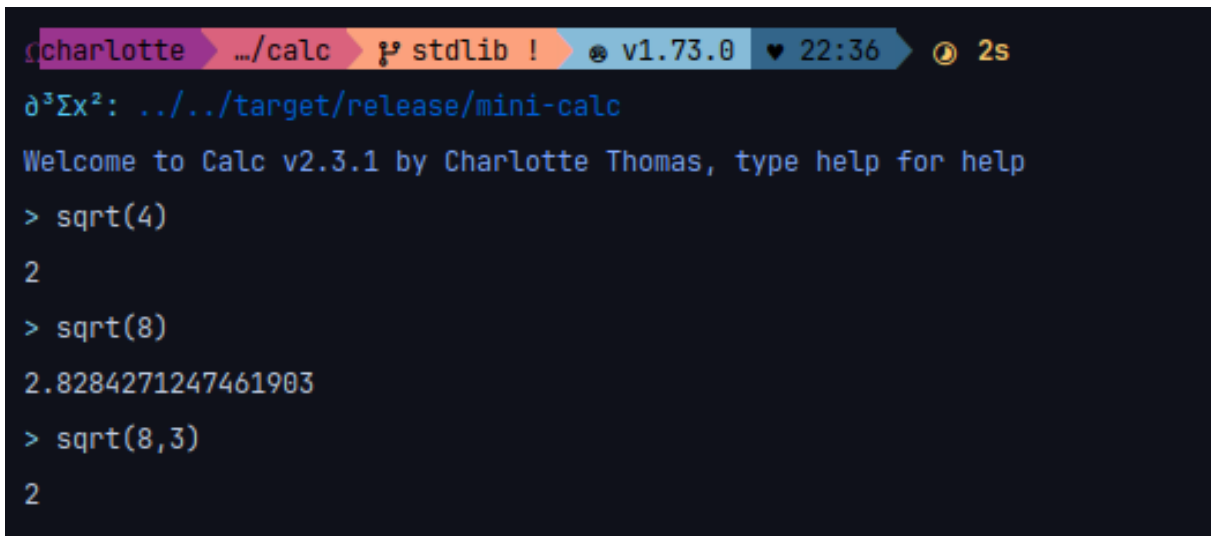
If you use the exp function you can pass as a second argument the base you want to use if no second arguments are passed it will used the natural base.

Figure 3: Usage of exp/ln

III.4. Root

You can specify in second argument an integer to take
the nth root, if not it take the square root.
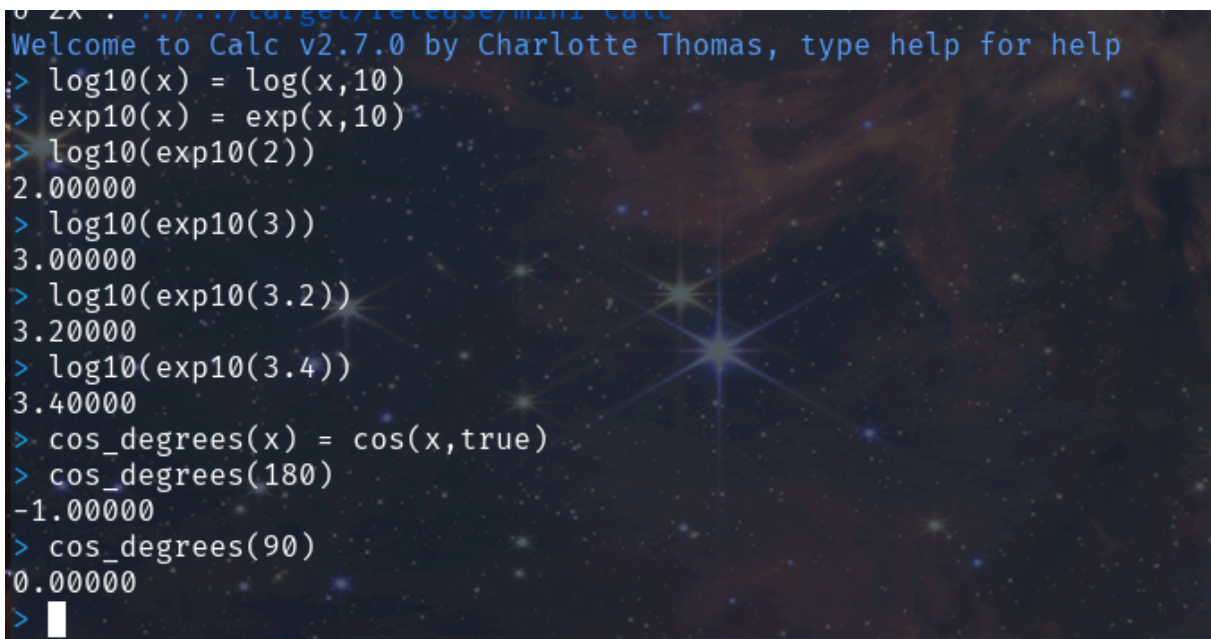
Figure 4: Usage of sqrt

III.5. Partial function

The calculator's language supports partial function.

Figure 5: Example of a partial function

III.6. Vectorization

Functions have been vectorized.

Welcome to Calc v2.10.0 by Charlotte Thomas, type help for help
> sqrt([1,4,9,16,25])
[1.00000,2.00000,3.00000,4.00000,5.00000]
>

Figure 6: Example of a vectorized function

III.7. User defined function

You can defined your own function

∂³Σx²: ../../target/release/mini-calc

Welcome to Calc v2.5.0 by Charlotte Thomas, type help for help
> test(x) = 2*x
> test(2)
4
> let(x,y) = x+y
> let(1,2)
3
> let(test(2),2)
6
> let(cos(pi),1)
0
>

Figure 7: Definition of function

IV. Configuration

You can configure the general color, greeting message, greeting color, prompt and prompt color in a toml file found for example on linux in

```bash
1  ~/.config/mini-calc/mini-calc.toml
```

```toml
1  general_color = 'purple'
2
3  [greeting]
4  greeting_message = 'Welcome to Calc %version% by %author%, type help for help'
5  greeting_color = 'blue'
6
7  [prompt]
8  prompt = '> '
9  prompt_color = 'cyan'
```

Figure 8: Example of the default configuration

IV.1. Colors

Available colors are

- blue
- black
- purple
- green
- cyan
- red
- yellow
- white
- an hexadecimal color (ex: #f7d8a8)

The default color (or if your colour can't be parsed) is cyan

IV.2. Example of a modified configuration


```
  mini-calc.toml    ●
1 general_color = 'cyan'
2
3 [greeting]
4 greeting_message = 'Heya! This is calc! Version %version%, and I think coded by %author% but who knows!'
5 greeting_color = '#f7a8d8'
6
7 [prompt]
8 prompt = 'λπ: '
9 prompt_color = '#55cdf'
```

Figure 9: Example of a modified config

it looks like


```
Heya! This is calc! Version v2.2.2, and I think coded by Charlotte Thomas but who knows!

λπ: 1+1

2

λπ: pi

3.141592653589793

λπ: e

2.718281828459045

λπ:
```

Figure 10: Modified configuration in action

IV.3. Interact in the command line

You can interact in the command line with the config, the commands are

- config: show the config help
- config reload: reload the config from the file
- config reset: reset the config
- config show: show the current config
- config set <category> <value>

categories are:

- greeting_message
- greeting_color
- prompt_color
- prompt
- general_color

```
Welcome to Calc v2.8.0 by Charlotte Thomas, type help for help
> config
 The greeting colour is set to blue which prints
 Welcome to Calc %version% by %author%, type help for help
 The prompt is >  in cyan
 Main color is purple which looks like
 This is the general colour
 If you've modified your config and it doesn't look good, the author (Charlotte Thomas) declines any responsabilities.

> config show
 The greeting colour is set to blue which prints
 Welcome to Calc %version% by %author%, type help for help
 The prompt is >  in cyan
 Main color is purple which looks like
 This is the general colour
 If you've modified your config and it doesn't look good, the author (Charlotte Thomas) declines any responsabilities.

> config reload
Your configuration has been reloaded
> config set
You need more argument for this command
> config set general_color #f7a8d8
Greeting color has been set to #f7a8d8 , reload for this to take effect
> config reload
Your configuration has been reloaded
> config set greeting_message Heya! %author% I love you!
Prompt has been updated to Heya! %author% I love you! , reload for this to take effect
> config reload
Your configuration has been reloaded
> config
 The greeting colour is set to blue which prints
 Heya! %author% I love you!
 The prompt is >  in cyan
 Main color is #f7a8d8  which looks like
 This is the general colour
 If you've modified your config and it doesn't look good, the author (Charlotte Thomas) declines any responsabilities.

> exit

 charlotte   …/calc   github-page $   v1.73.0   13:49   1m4s
∂³Σx²: mini-calc
Heya! Charlotte Thomas I love you!
> config reset
Your config has been reseted to default settings
> exit

 charlotte   …/calc   github-page $   v1.73.0   13:49   10s
∂³Σx²: mini-calc
Welcome to Calc v2.8.0 by Charlotte Thomas, type help for help
>
```

Figure 11: Example of interaction in the command line of config

V.1. Implemented operators

The following operators have been implemented:

- or (alias: ||)
- and (alias: &&)
- geq (alias: >=)
- leq (alias: <=)
- gt (alias : >)
- lt (alias: <)

V.2. Example

```
Welcome to Calc v2.4.0 by Charlotte Thomas, type help for help
> true
true
> false
false
> true && false
false
> true || false
true
> 1 > 2
false
> 1 < (1+1)
true
> (1+2) >= cos(1)
true
> let = false
> !let
true
> !let && (1+2) >= cos(1)
true
>
```

Figure 12: Example of logic

VI. Plot

You can plot, the backend is provided by GNUPlot, so it should work great on linux and macos, the behaviour on windows is unknown.

VI.1. Help

To display the help just type `plot()`

Figure 13: Help of plot

VI.2. Plot

VI.2.1. Default

It's easy to plot a function just type `plot(fn)`
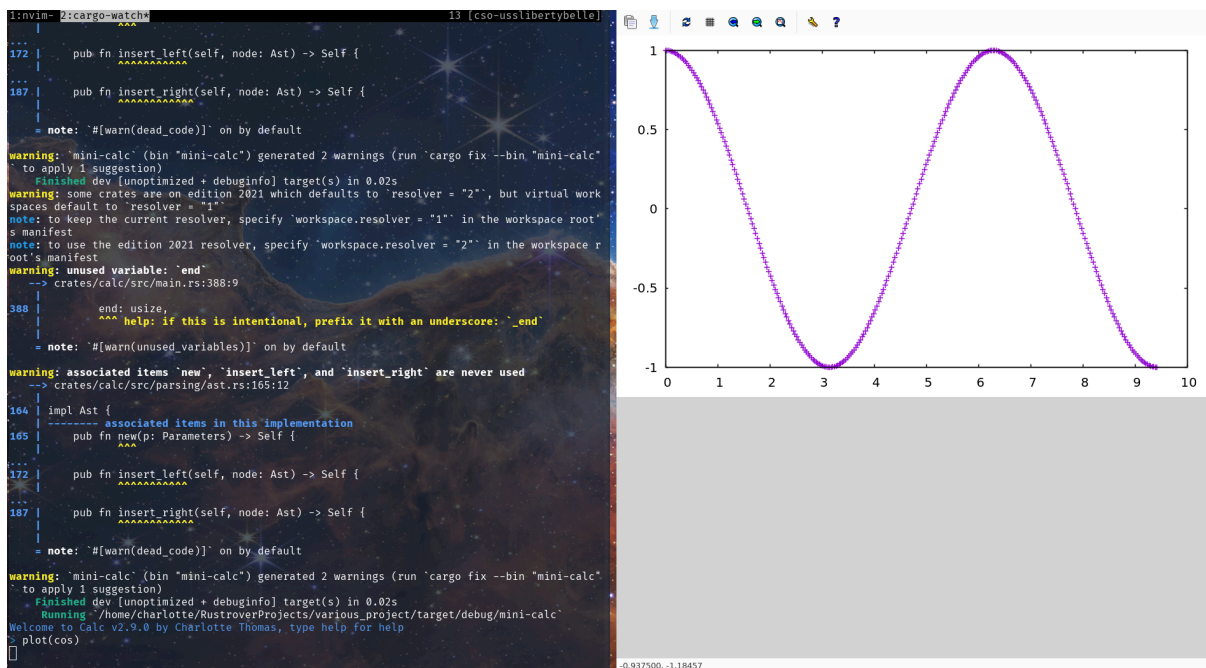
Figure 14: Plot of the cos function with default values

VI.2.2. Options

A more difficult operation is with values, `plot(sin,-pi,pi, 0.01,"sin","x(rad)","y","line")`.
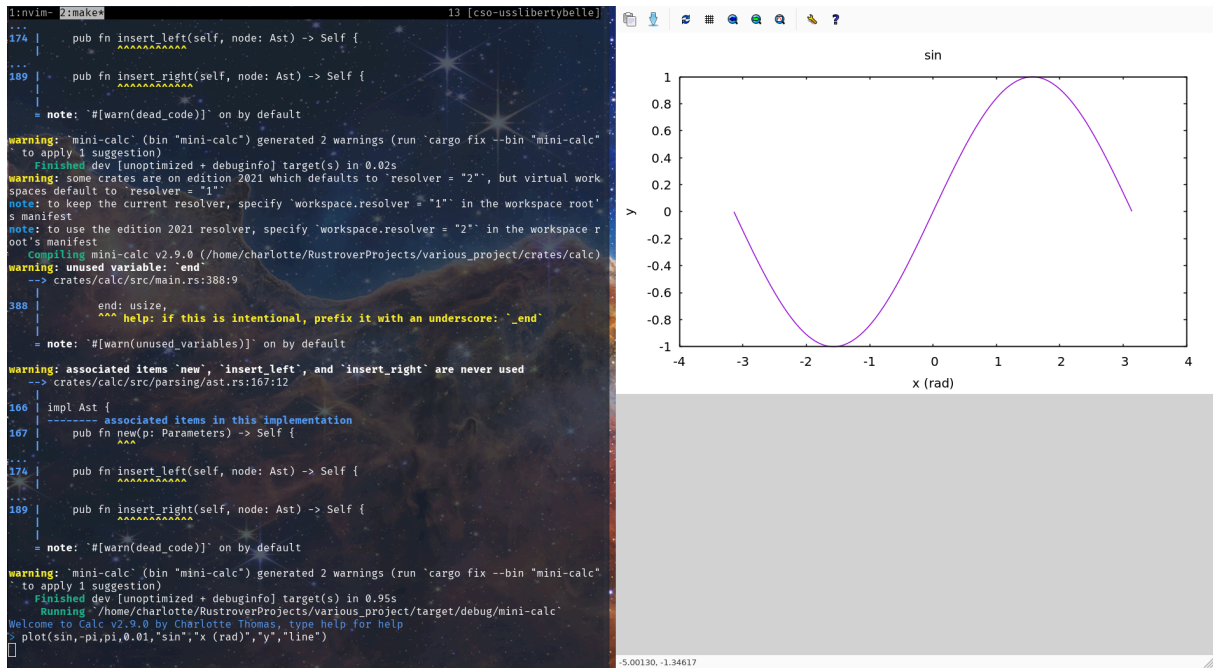
Figure 15: Plot with overloading of default values

VI.2.3. Plot your own function

You can plot your own defined functions!
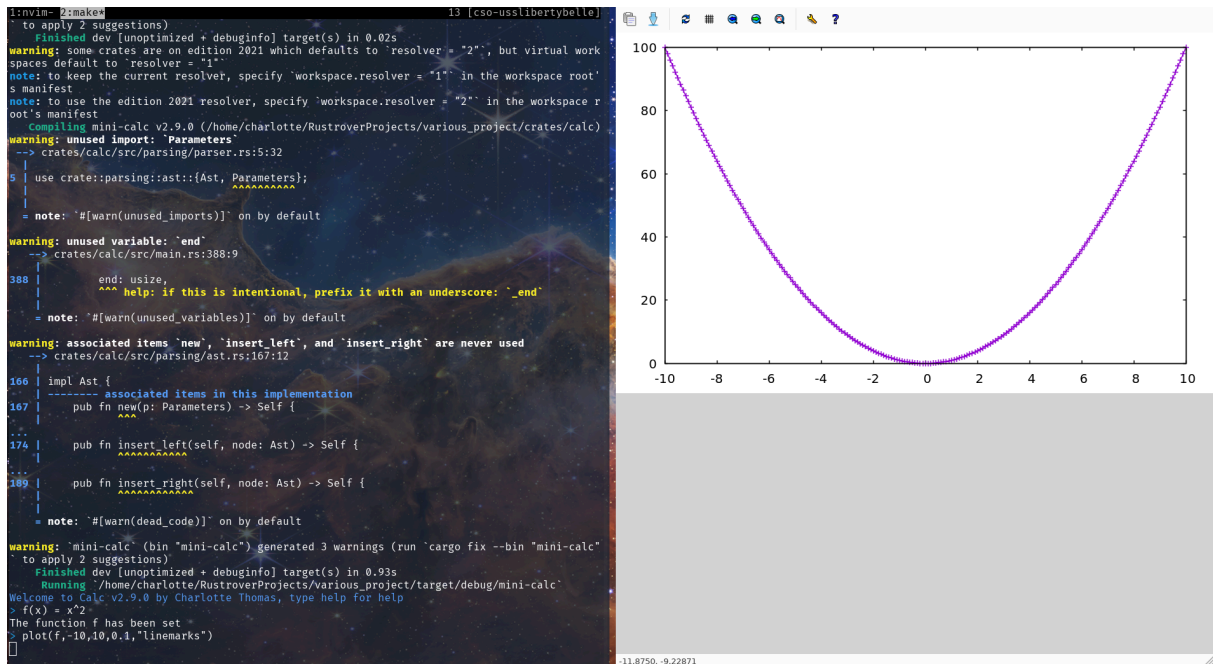
Figure 16: Plot of an user-defined function

VI.3. Terminal plotting

You can plot *right* into your terminal

VI.3.1. Default

The best example to show it is the square function between -5 and 5 with a 0.1 step. The x axis is automatically scaled but not the y axis for now.

Figure 17: Terminal plotting of an user defined function

VI.3.2. Options

The terminal supports labels as options

```
Welcome to Calc v2.9.1 by Charlotte Thomas, type help for help
> f(x) = x*x
The function f has been set
> termplot(f,-5,5,0.1,"x squared","x axis","y axis")
*******************************************************************************************
********************************************x squared**************************************
*|                                                                                       *
*|                                                                                       *
*|                                                                                       *
*|                                                                                      *
*|+                                                                                     *
*|+                                                                                  ++*
*|  +                                                                             +   *
*|   +                                                                           +    *
*|    +                                                                         +     *
y|     +                                                                       +      *
 |      +                                                                     +       *
a|       +                                                                   +        *
x|        +                                                                 +         *
i|         ++                                                             ++          *
s|           +                                                          +            *
*|            +                                                        +             *
*|             +                                                      +              *
*|              ++                                                  ++               *
*|                +                                              +                   *
*|                 ++                                         ++                     *
*|                   ++                                     ++                       *
*|                    +                                    +                        *
*|                     ++                                ++                         *
*|                       ++                            ++                           *
*|                         ++         +++          ++                              *
*|                           ++      + +         ++                                *
*|                             +++            +++                                  *
*|                              ++++       ++++                                    *
*|                                +++++++++++++++++++                              *
*|---------------------------------------------------------------------------------*
* -5                                      0                                      5 *
****************************************x axis*********************************************
>
```

Figure 18: Terminal plotting with options

VI.3.3. Auto y scaling

It scales automatically in y too!

Figure 19: Example of a plot with y auto-scaling

VII. Vectors computation

You can compute vectors using these functions,

- add vectors
- dot product (* operator)
- norm function

Figure 20: Example of vector computation

VIII. Matrices computation

As of 2.7.0 matrix algebra was added to the calculator you can

- add matrices
- multiply compatible matrices

functions added

- transpose
- invert
- 2023-11-26 14:49

Figure 21: Example of matrix computation

And as of 2.11.5 they are pretty printed with each column being aligned.

```
> [[1,2],[3,4]]
+---+
|1 2|
|3 4|
+---+
> a = [[1,2,3],[4,0,6],[7,8,9]]
 a = +-----+
|1 2 3|
|4 0 6|
|7 8 9|
+-----+
> a
+-----+
|1 2 3|
|4 0 6|
|7 8 9|
+-----+
> invert(a)
+--------------+
|-4/5 1/10  1/5 |
|1/10 -1/5 1/10 |
|8/15 1/10 -2/15|
+--------------+
> invert([[1,2,3,4],[5,0,6,0],[7,0,8,0],[9,10,11,12]])
+-------------------+
| 0     -4     3    0  |
|-3/4 17/4  -13/4 1/4 |
| 0    7/2  -5/2   0  |
|5/8  -15/4 11/4  -1/8|
+-------------------+
>
```

Figure 22: Pretty printed matrix

IX. Exact math

IX.1. Rational exact math

As of 2.11.0 rational exact math was added, supports for

- rational operations
- rational reduction
- rationalization of floats (precision: 10 digits)

IX.1.1. Examples

Figure 23: Example of rational computations

Figure 24: Example of rational in matrices