

Package ‘ratioOfQsprays’

July 28, 2024

Title Fractions of Multivariate Polynomials with Rational Coefficients

Version 1.1.0

Description Based on the 'qspray' package, this package introduces the new type 'ratioOfQsprays'. An object of type 'qspray' represents a multivariate polynomial with rational coefficients while an object of type 'ratioOfQsprays', defined by two 'qspray' objects, represents a fraction of two multivariate polynomials with rational coefficients. Arithmetic operations for these objects are available, and they always return irreducible fractions. Other features include: differentiation, evaluation, conversion to a function, and fine control of the way to print a 'ratioOfQsprays' object. The 'C++' library 'CGAL' is used to make the fractions irreducible.

License GPL-3

URL <https://github.com/stla/ratioOfQsprays>

BugReports <https://github.com/stla/ratioOfQsprays/issues>

Depends qspray (>= 3.1.0)

Imports gmp, methods, Rcpp, Ryacas, utils

Suggests testthat (>= 3.0.0)

LinkingTo BH, qspray, Rcpp, RcppCGAL

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

SystemRequirements C++17, gmp, mpfr

Collate 'RcppExports.R' 'creation.R' 'evaluation.R' 'internal.R' 'ratioOfQsprays.R' 'queries.R' 'show.R' 'transformation.R'

NeedsCompilation yes

Author Stéphane Laurent [aut, cre]

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2024-07-28 05:20:02 UTC

Contents

as.function.ratioOfQsprays	2
as.ratioOfQsprays	3
changeVariables	4
derivRatioOfQsprays	5
dRatioOfQsprays	6
evalRatioOfQsprays	6
getDenominator	7
getNumerator	8
involvedVariables	8
isConstant	9
isPolynomial	9
isUnivariate	10
numberOfVariables	10
permuteVariables	11
ratioOfQsprays-unary	12
ratioOfQsprays_from_list	12
rRatioOfQsprays	13
showRatioOfQsprays	13
showRatioOfQspraysOption<-	14
showRatioOfQspraysX1X2X3	15
showRatioOfQspraysXYZ	16
substituteRatioOfQsprays	17
swapVariables	18

Index 19

as.function.ratioOfQsprays
Ratio of multivariate polynomials as function

Description

Coerces a ratioOfQsprays polynomial to a function.

Usage

```
## S3 method for class 'ratioOfQsprays'
as.function(x, N = FALSE, ...)
```

Arguments

x	object of class ratioOfQsprays
N	Boolean, whether the function must numerically approximate the result
...	ignored

Value

A function having the same variables as the polynomial. If N=FALSE, it returns a string. If N=TRUE, it returns a number if the result does not contain any variable, otherwise it returns a R expression.

Examples

```
library(ratioOfQsprays)
x <- qlone(1); y <- qlone(2)
roq <- (x^2/2 + y^2 + x*y - 1) / (x + 1)
f <- as.function(roq)
g <- as.function(roq, N = TRUE)
f(2, "3/7")
g(2, "3/7")
f("x", "y")
g("x", "y")
# the evaluation is performed by (R)yacas and complex numbers are
# allowed; the imaginary unit is denoted by \code{I}:
f("2 + 2*I", "Sqrt(2)")
g("2 + 2*I", "Sqrt(2)")
```

as.ratioOfQsprays *Coercion to a 'ratioOfQsprays' object*

Description

Coercion to a 'ratioOfQsprays' object

Usage

```
## S4 method for signature 'character'
as.ratioOfQsprays(x)

## S4 method for signature 'ratioOfQsprays'
as.ratioOfQsprays(x)

## S4 method for signature 'qspray'
as.ratioOfQsprays(x)

## S4 method for signature 'numeric'
as.ratioOfQsprays(x)

## S4 method for signature 'bigz'
as.ratioOfQsprays(x)

## S4 method for signature 'bigq'
as.ratioOfQsprays(x)
```

Arguments

`x` a `ratioOfQsprays` object, a `qspray` object, or an object yielding a quoted integer or a quoted fraction after an application of `as.character`, e.g. a big number

Value

This returns `x` if `x` already is a `ratioOfQsprays` object, otherwise this returns the `ratioOfQsprays` object whose numerator is the coercion of `x` to a `qspray` object and whose denominator is the unit `qspray` object.

Examples

```
library(qspray)
as.ratioOfQsprays(2)
as.ratioOfQsprays("1/3")
( qspray <- 5*qlone(1) + qlone(2)^2 )
as.ratioOfQsprays(qspray)
# show options are inherited:
showQsprayOption(qspray, "x") <- "A"
as.ratioOfQsprays(qspray)
```

changeVariables

Change of variables in a 'ratioOfQsprays' fraction of polynomials

Description

Replaces the variables of a `ratioOfQsprays` fraction of polynomials with some `qspray` polynomials. E.g. you have a fraction of polynomials $R(x, y)$ and you want the fraction of polynomials $R(x^2, x + y + 1)$.

Usage

```
## S4 method for signature 'ratioOfQsprays,list'
changeVariables(x, listOfQsprays)
```

Arguments

`x` a `ratioOfQsprays` fraction of polynomials

`listOfQsprays` a list containing at least `n` `qspray` objects, or objects coercible to `qspray` objects, where `n` is the number of variables of the `ratioOfQsprays` fraction of polynomials given in the `x` argument; if this list is named, then its names will be used in the show options of the result

Value

The `ratioOfQsprays` fraction of polynomials obtained by replacing the variables of the fraction of polynomials given in the `x` argument with the `qspray` polynomials given in the `listOfQsprays` argument.

Examples

```
library(ratioOfQsprays)
f <- function(x, y) {
  (x^2 + 5*y - 1) / (x + 1)
}
x <- qlone(1)
y <- qlone(2)
R <- f(x, y)
X <- x^2
Y <- x + y + 1
S <- changeVariables(R, list(X, Y))
S == f(X, Y) # should be TRUE
```

derivRatioOfQsprays *Partial derivative*

Description

Partial derivative of a ratioOfQsprays.

Usage

```
derivRatioOfQsprays(roq, i, derivative = 1)
```

Arguments

roq	object of class ratioOfQsprays
i	integer, the dimension to differentiate with respect to, e.g. 2 to differentiate with respect to y
derivative	integer, how many times to differentiate

Value

A ratioOfQsprays object.

Examples

```
library(ratioOfQsprays)
x <- qlone(1)
y <- qlone(2)
roq <- (2*x + 3*x*y) / (x^2 + y^2)
derivRatioOfQsprays(roq, 2) # derivative w.r.t. y
```

dRatioOfQsprays *Partial differentiation*

Description

Partial differentiation of a ratioOfQsprays polynomial.

Usage

```
dRatioOfQsprays(roq, orders)
```

Arguments

roq	object of class ratioOfQsprays
orders	integer vector, the orders of the differentiation; e.g. $c(2, 0, 1)$ means that you differentiate two times with respect to x , you do not differentiate with respect to y , and you differentiate one time with respect to z

Value

A ratioOfQsprays object.

Examples

```
library(ratioOfQsprays)
x <- qlone(1)
y <- qlone(2)
roq <- (x + 2*y + 3*x*y) / (x + 1)
dRatioOfQsprays(roq, c(1, 1))
derivRatioOfQsprays(derivRatioOfQsprays(roq, 1), 2)
```

evalRatioOfQsprays *Evaluate a 'ratioOfQsprays' object*

Description

Evaluation of the fraction of multivariate polynomials represented by a ratioOfQsprays object.

Usage

```
evalRatioOfQsprays(roq, values_re, values_im = NULL)
```

Arguments

roq	a ratioOfQsprays object
values_re	vector of the real parts of the values; each element of as.character(values_re) must be a quoted integer or a quoted fraction
values_im	vector of the imaginary parts of the values; each element of as.character(values_im) must be a quoted integer or a quoted fraction

Value

A bigq number if values_im=NULL, a pair of bigq numbers otherwise: the real part and the imaginary part of the result.

Examples

```
x <- qlone(1); y <- qlone(2)
roq <- 2*x / (x^2 + 3*y^2)
evalRatioOfQsprays(roq, c("2", "5/2", "99999")) # "99999" will be ignored
```

getDenominator	<i>Get the denominator of a 'ratioOfQsprays'</i>
----------------	--

Description

Get the denominator of a ratioOfQsprays object, preserving the show options.

Usage

```
getDenominator(roq)
```

Arguments

roq	a ratioOfQsprays object
-----	-------------------------

Value

A qspray object.

getNumerator	<i>Get the numerator of a 'ratioOfQsprays'</i>
--------------	--

Description

Get the numerator of a ratioOfQsprays object, preserving the show options.

Usage

```
getNumerator(roq)
```

Arguments

roq a ratioOfQsprays object

Value

A qspray object.

involvedVariables	<i>Variables involved in a 'ratioOfQsprays'</i>
-------------------	---

Description

Variables involved in a ratioOfQsprays object.

Usage

```
## S4 method for signature 'ratioOfQsprays'
involvedVariables(x)
```

Arguments

x a ratioOfQsprays object

Value

A vector of integers. Each integer represents the index of a variable involved in x.

See Also

[numberOfVariables](#).

Examples

```
x <- qlone(1); z <- qlone(3)
r0Q <- 2*x/z + x/(x+z) + z^2/x
involvedVariables(r0Q) # should be c(1L, 3L)
```

isConstant	<i>Whether a 'ratioOfQsprays' is constant</i>
------------	---

Description

Checks whether a ratioOfQsprays object defines a constant fraction of polynomials.

Usage

```
## S4 method for signature 'ratioOfQsprays'  
isConstant(x)
```

Arguments

x a ratioOfQsprays object

Value

A Boolean value.

isPolynomial	<i>Whether a 'ratioOfQsprays' is polynomial</i>
--------------	---

Description

Checks whether a ratioOfQsprays actually is polynomial, that is, whether its denominator is a constant qspray polynomial (and then it should be equal to one).

Usage

```
isPolynomial(roq)
```

Arguments

roq a ratioOfQsprays object

Value

A Boolean value.

Examples

```
x <- qlone(1)  
y <- qlone(2)  
roq <- (x^2 - y^2) / (x - y)  
isPolynomial(roq)  
roq == x + y
```

isUnivariate	<i>Whether a 'ratioOfQsprays' is univariate</i>
--------------	---

Description

Checks whether a ratioOfQsprays object defines a univariate fraction of polynomials.

Usage

```
## S4 method for signature 'ratioOfQsprays'
isUnivariate(x)
```

Arguments

x a ratioOfQsprays object

Value

A Boolean value.

Note

The ratioOfQsprays object $y / (1 + y)$ where $y = \text{qlone}(2)$ is not univariate, although it involves only one variable. The function returns TRUE when only $\text{qlone}(1)$ is involved or when no variable is involved.

numberOfVariables	<i>Number of variables in a 'ratioOfQsprays'</i>
-------------------	--

Description

Number of variables involved in a ratioOfQsprays object.

Usage

```
## S4 method for signature 'ratioOfQsprays'
numberOfVariables(x)
```

Arguments

x a ratioOfQsprays object

Value

An integer.

Note

The number of variables in the ratioOfQsprays object $y / (1 + y)$ where $y = \text{qlone}(2)$ is 2, not 1, although only one variable occurs. Rigorously speaking, the function returns the maximal integer d such that $\text{qlone}(d)$ occurs in the 'ratioOfQsprays'.

See Also

[involvedVariables](#)

permuteVariables *Permute variables*

Description

Permute the variables of a ratioOfQsprays fraction of polynomials.

Usage

```
## S4 method for signature 'ratioOfQsprays,numeric'  
permuteVariables(x, permutation)
```

Arguments

`x` a ratioOfQsprays object
`permutation` a permutation

Value

A ratioOfQsprays object.

Examples

```
library(ratioOfQsprays)  
f <- function(x, y, z) {  
  (x^2 + 5*y + z - 1) / (x + 1)  
}  
x <- qlone(1)  
y <- qlone(2)  
z <- qlone(3)  
R <- f(x, y, z)  
permutation <- c(3, 1, 2)  
S <- permuteVariables(R, permutation)  
S == f(z, x, y) # should be TRUE
```

ratioOfQsprays-unary *Unary operators for 'ratioOfQsprays' objects*

Description

Unary operators for ratioOfQsprays objects.

Usage

```
## S4 method for signature 'ratioOfQsprays,missing'
e1 + e2
```

```
## S4 method for signature 'ratioOfQsprays,missing'
e1 - e2
```

Arguments

e1	object of class ratioOfQsprays
e2	nothing

Value

A ratioOfQsprays object.

ratioOfQsprays_from_list
(internal) Make a 'ratioOfQsprays' object from a list

Description

This function is for internal usage. It is exported because it is also used for internal usage in other packages.

Usage

```
ratioOfQsprays_from_list(x)
```

Arguments

x	list returned by the Rcpp function returnRatioOfQsprays
---	---

Value

A ratioOfQsprays object.

rRatioOfQsprays	<i>Random 'ratioOfQsprays'</i>
-----------------	--------------------------------

Description

Generates a random ratioOfQsprays object.

Usage

```
rRatioOfQsprays(allow.zero = TRUE)
```

Arguments

allow.zero Boolean, whether to allow to get a null ratioOfQsprays

Value

A ratioOfQsprays object.

showRatioOfQsprays	<i>Print a 'ratioOfQsprays' object</i>
--------------------	--

Description

Prints a ratioOfQsprays object given a function to print a qspray object

Usage

```
showRatioOfQsprays(
  showQspray,
  quotientBar = " %//% ",
  lbracket = "[ ",
  rbracket = " ]"
)
```

Arguments

showQspray a function which prints a qspray object, which will be applied to the numerator and the denominator

quotientBar a string representing the quotient bar between the numerator and the denominator, including surrounding spaces, e.g " / "

lbracket, rbracket used to enclose the numerator and the denominator

Value

A function which takes as argument a `ratioOfQsprays` object and which prints it.

Note

The function returned by this function can be used as the option "showRatioOfQsprays" of the setter function `showRatioOfQspraysOption<-`. That said, one would more often uses `showRatioOfQspraysX1X2X3` or `showRatioOfQspraysXYZ` for this option, which are both built with `showRatioOfQsprays`.

See Also

[showRatioOfQspraysX1X2X3](#), [showRatioOfQspraysXYZ](#), [showRatioOfQspraysOption<-](#), [showQspray](#).

Examples

```
set.seed(666)
( roq <- rRatioOfQsprays() )
f <- showRatioOfQsprays(showQsprayX1X2X3("a"), " / ", "[[ ", " ]]")
f(roq)
# this is equivalent to
f <- showRatioOfQspraysX1X2X3("a", " / ", lbracket = "[[ ", rbracket = " ]]")
f(roq)
```

```
showRatioOfQspraysOption<-
```

Set a show option to a 'ratioOfQsprays'

Description

Set a show option to a `ratioOfQsprays` object.

Usage

```
showRatioOfQspraysOption(x, which) <- value
```

Arguments

<code>x</code>	a <code>ratioOfQsprays</code> object
<code>which</code>	which option to set; this can be "x", "quotientBar", "showQspray", or "showRatioOfQsprays"
<code>value</code>	the value of the option to be set

Value

This returns the updated `ratioOfQsprays`.

See Also

[showRatioOfQsprays](#).

Examples

```

set.seed(666)
( roq <- rRatioOfQsprays() )
showRatioOfQspraysOption(roq, "quotientBar") <- " / "
roq
showRatioOfQspraysOption(roq, "x") <- "a"
roq
showRatioOfQspraysOption(roq, "showQspray") <- showQsprayXYZ()
roq

```

```
showRatioOfQspraysX1X2X3
```

```
Print a 'ratioOfQsprays'
```

Description

Print a ratioOfQsprays object given a string to denote the non-indexed variables.

Usage

```
showRatioOfQspraysX1X2X3(var, quotientBar = " %//% ", ...)
```

Arguments

var	a string, usually a letter, to denote the non-indexed variables
quotientBar	a string representing the quotient bar between the numerator and the denominator, including surrounding spaces, e.g " / "
...	arguments other than quotientBar passed to showRatioOfQsprays

Value

A function which takes as argument a ratioOfQsprays object and which prints it.

Note

The function returned by this function can be used as the option "showRatioOfQsprays" of the setter function [showRatioOfQspraysOption<-](#). If you do not use the ellipsis arguments, this is equivalent to set the "x" option and the "quotientBar" option (see example).

See Also

[showRatioOfQspraysXYZ](#), [showRatioOfQspraysOption<-](#).

Examples

```

set.seed(666)
( roq <- rRatioOfQsprays() )
showRatioOfQspraysX1X2X3("x", " / ")(roq)
# setting a show option:
showRatioOfQspraysOption(roq, "showRatioOfQsprays") <-
  showRatioOfQspraysX1X2X3("x", " / ")
roq
# this is equivalent to set the "x" and "quotientBar" options:
showRatioOfQspraysOption(roq, "x") <- "x"
showRatioOfQspraysOption(roq, "quotientBar") <- " / "

```

```
showRatioOfQspraysXYZ Print a 'ratioOfQsprays'
```

Description

Print a `ratioOfQsprays` object given some letters to denote the variables, by printing monomials in the style of `"x^2.yz"`.

Usage

```

showRatioOfQspraysXYZ(
  letters = c("x", "y", "z"),
  quotientBar = " %//% ",
  ...
)

```

Arguments

<code>letters</code>	a vector of strings, usually some letters such as <code>"x"</code> and <code>"y"</code> , to denote the variables
<code>quotientBar</code>	a string representing the quotient bar between the numerator and the denominator, including surrounding spaces, e.g <code>" / "</code>
<code>...</code>	arguments other than <code>quotientBar</code> passed to showRatioOfQsprays

Value

A function which takes as argument a `ratioOfQsprays` object and which prints it.

Note

The function returned by this function can be used as the option `"showRatioOfQsprays"` of the setter function [showRatioOfQspraysOption<-](#). As another note, let us describe the behavior of this function in a case when the number of variables of the `ratioOfQsprays` object to be printed is bigger than the number of provided letters. In such a case, the output will be the same as an application of the function `showRatioOfQspraysX1X2X3(x)` with `x` being the first letter provided. See the example.

See Also

[showRatioOfQspraysX1X2X3](#), [showRatioOfQspraysOption<-](#).

Examples

```
set.seed(666)
( roq <- rRatioOfQsprays() )
showRatioOfQspraysXYZ(c("X", "Y", "Z"), " / ")(roq)
# now take a ratioOfQsprays with four variables:
roq <- roq * qlone(4)
# then the symbols X1, X2, X3, X4 denote the variables now:
showRatioOfQspraysXYZ(c("X", "Y", "Z"), " / ")(roq)
# this is the method used by default to print the ratioOfQsprays objects,
# with the initial letters x, y, z which then become x1, x2, x3, x4:
roq
```

substituteRatioOfQsprays

Partial evaluation of a 'ratioOfQsprays' fraction of polynomials

Description

Substitute some values to a subset of the variables of a ratioOfQsprays fraction of polynomials.

Usage

```
substituteRatioOfQsprays(roq, values)
```

Arguments

roq	a ratioOfQsprays object
values	the values to be substituted; this must be a vector whose length equals the number of variables of roq, and whose each entry is either NA for non-substitution or a "scalar" x such that as.character(x) is a quoted integer or a quoted fraction, e.g. a bigq number

Value

A ratioOfQsprays object.

Examples

```
library(ratioOfQsprays)
x <- qlone(1)
y <- qlone(2)
z <- qlone(3)
roq <- (x^2 + y^2 + x*y*z - 1) / (x + 1)
substituteRatioOfQsprays(roq, c("2", NA, "3/2"))
```

swapVariables	<i>Swap variables</i>
---------------	-----------------------

Description

Swap two variables of a ratioOfQsprays.

Usage

```
## S4 method for signature 'ratioOfQsprays,numeric,numeric'  
swapVariables(x, i, j)
```

Arguments

x	a ratioOfQsprays object
i, j	indices of the variables to be swapped

Value

A ratioOfQsprays object.

Examples

```
library(ratioOfQsprays)  
f <- function(x, y, z) {  
  (x^2 + 5*y + z - 1) / (x + 1)  
}  
x <- qlone(1)  
y <- qlone(2)  
z <- qlone(3)  
R <- f(x, y, z)  
S <- swapVariables(R, 2, 3)  
S == f(x, z, y) # should be TRUE
```

Index

- `+`, `ratioOfQsprays`, missing-method
(`ratioOfQsprays-unary`), 12
- `-`, `ratioOfQsprays`, missing-method
(`ratioOfQsprays-unary`), 12
- `as.function.ratioOfQsprays`, 2
- `as.ratioOfQsprays`, 3
- `as.ratioOfQsprays`, `bigq`-method
(`as.ratioOfQsprays`), 3
- `as.ratioOfQsprays`, `bigz`-method
(`as.ratioOfQsprays`), 3
- `as.ratioOfQsprays`, `character`-method
(`as.ratioOfQsprays`), 3
- `as.ratioOfQsprays`, `numeric`-method
(`as.ratioOfQsprays`), 3
- `as.ratioOfQsprays`, `qspray`-method
(`as.ratioOfQsprays`), 3
- `as.ratioOfQsprays`, `ratioOfQsprays`-method
(`as.ratioOfQsprays`), 3
- `changeVariables`, 4
- `changeVariables`, `ratioOfQsprays`, `list`-method
(`changeVariables`), 4
- `derivRatioOfQsprays`, 5
- `dRatioOfQsprays`, 6
- `evalRatioOfQsprays`, 6
- `getDenominator`, 7
- `getNumerator`, 8
- `involvedVariables`, 8, 11
- `involvedVariables`, `ratioOfQsprays`-method
(`involvedVariables`), 8
- `isConstant`, 9
- `isConstant`, `ratioOfQsprays`-method
(`isConstant`), 9
- `isPolynomial`, 9
- `isUnivariate`, 10
- `isUnivariate`, `ratioOfQsprays`-method
(`isUnivariate`), 10
- `numberOfVariables`, 8, 10
- `numberOfVariables`, `ratioOfQsprays`-method
(`numberOfVariables`), 10
- `permuteVariables`, 11
- `permuteVariables`, `ratioOfQsprays`, `numeric`-method
(`permuteVariables`), 11
- `ratioOfQsprays-unary`, 12
- `ratioOfQsprays_from_list`, 12
- `rRatioOfQsprays`, 13
- `showQspray`, 14
- `showRatioOfQsprays`, 13, 14–16
- `showRatioOfQspraysOption<-`, 14
- `showRatioOfQspraysX1X2X3`, 14, 15, 17
- `showRatioOfQspraysXYZ`, 14, 15, 16
- `substituteRatioOfQsprays`, 17
- `swapVariables`, 18
- `swapVariables`, `ratioOfQsprays`, `numeric`, `numeric`-method
(`swapVariables`), 18