

Package: gmpoly (via r-universe)

August 22, 2024

Type Package

Title Multivariate Polynomials with Rational Coefficients

Version 1.1.0

Date 2022-02-21

Author Stéphane Laurent

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Description Symbolic calculation (addition or multiplication) and evaluation of multivariate polynomials with rational coefficients.

License GPL-3

Encoding UTF-8

Imports gmp, purrr, mvp, english

RoxygenNote 7.1.2

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

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

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://stla.r-universe.dev>

RemoteUrl <https://github.com/stla/gmpoly>

RemoteRef HEAD

RemoteSha 705d52ecac24d03126064577662bf7f7f49066ed

Contents

gmpoly	2
gmpoly2mvp	3
gmpolyConstant	3
gmpolyEval	4
gmpolyGrow	5
Ops.gmpoly	5
print.gmpoly	6

gmpoly	<i>Define a multivariate polynomial</i>
--------	---

Description

Define a multivariate polynomial from a string or from its coefficients and powers.

Usage

```
gmpoly(string, coeffs = NULL, powers = NULL)
```

Arguments

string	a string such as " $x^{(1,2,1)} + 4 x^{(0,2,3)}$ ", or you can define the polynomial with <code>coeffs</code> and <code>powers</code> ; in this case set <code>string</code> to <code>NULL</code> or to nothing (i.e. missing)
coeffs	the coefficients of the polynomial, in case you don't define it with the <code>string</code> argument; this must be a vector of <code>bigq</code> numbers
powers	the powers of the polynomial, in case you don't define it with the <code>string</code> argument; this must be a matrix of integers, one row for each term (hence <code>nrow(powers)</code> must equal <code>length(coeffs)</code>), and the number of columns is the number of variables

Value

A `gmpoly` object.

Examples

```
library(gmpoly)
gmpoly("5/2 x^(2,3) + 3 x^(1,1)")
gmpoly("5/2 x^(2,3) - 3 x^(1,1)")
gmpoly("-x^(1,2,3) + 4/7 x^(3,1,1)")
library(gmp)
gmpoly(
  coeffs = as.bigq(c(5, 7), c(8, 9)),
  powers = rbind(c(1, 1, 1), c(2, 2, 2))
)
```

gmpoly2mvp	<i>Conversion to 'mvp' polynomial</i>
------------	---------------------------------------

Description

Convert a [gmpoly](#) polynomial to a [mvp](#) polynomial.

Usage

```
gmpoly2mvp(pol)
```

Arguments

pol a [gmpoly](#) object

Value

A [mvp](#) object.

Examples

```
library(gmpoly)
pol <- gmpoly("5/2 x^(2,2,3) + 3 x^(1,0,1)")
gmpoly2mvp(pol)
```

gmpolyConstant	<i>Constant multivariate polynomial</i>
----------------	---

Description

Constructs a constant multivariate polynomial.

Usage

```
gmpolyConstant(m, value)
```

Arguments

m number of variables, a strictly positive integer
value the constant value of the polynomial; the [as.bigq](#) function is applied to this argument, so it can be e.g. an integer or a character string such as "2/3" (avoid decimal numbers)

Value

A [gmpoly](#) object.

Examples

```
library(gmpoly)
gmpolyConstant(3, "2/3")
```

gmpolyEval

Evaluation of a multivariate polynomial

Description

Evaluates a gmpoly multivariate polynomial for given values of the variables.

Usage

```
gmpolyEval(pol, x)
```

Arguments

`pol` a [gmpoly](#) object

`x` either a [bigq](#) vector, or a [bigq](#) matrix; in the later case, the evaluation is performed for each row of the matrix

Value

A [bigq](#) number or vector.

Examples

```
library(gmpoly)
library(gmp)
pol <- gmpoly("5/2 x^(2,3) + 3 x^(1,1)")
gmpolyEval(pol, as.bigq(c(1, 1)))
x <- rbind(
  t(as.bigq(c(1, 1))),
  t(as.bigq(c(3, 4), c(4, 3)))
)
gmpolyEval(pol, x)
```

gmpolyGrow	<i>Grow polynomial</i>
------------	------------------------

Description

Grow a multivariate polynomial by including a new variable.

Usage

```
gmpolyGrow(pol)
```

Arguments

pol a [gmpoly](#) object

Value

The "same" multivariate polynomial as pol, except that it has an additional variable.

Examples

```
library(gmpoly)
pol <- gmpoly("3 x^(1,2) - 1/7 x^(5,3)")
gmpolyGrow(pol)
```

Ops.gmpoly	<i>Arithmetic operators for multivariate polynomials</i>
------------	--

Description

Arithmetic operators for multivariate polynomials

Usage

```
## S3 method for class 'gmpoly'
Ops(e1, e2 = NULL)
```

Arguments

e1, e2 for an unary operator, only e1 must be given, a [gmpoly](#) object; for a binary operator, at least one of e1 and e2 must be a [gmpoly](#) object, and the other must be a [gmpoly](#) object as well or a scalar; the power operator (^) is an exception: one can only raise a [gmpoly](#) object to a positive integer power

Value

A [gmpoly](#) object.

Examples

```
library(gmpoly)
pol <- gmpoly("4 x^(2, 1, 1) + 1/2 x^(0,1,0)")
+pol
-pol
2 * pol
pol / 2
pol + 5
pol - 5
pol^2
pol1 <- gmpoly("2 x^(1,1) - 5/3 x^(0,1)")
pol2 <- gmpoly("-2 x^(1,1) + 3 x^(2,1)")
pol1 + pol2
pol1 * pol2
pol1 == pol2
pol1 != pol2
```

print.gmpoly

Print a multivariate polynomial

Description

Print a multivariate polynomial of class gmpoly.

Usage

```
## S3 method for class 'gmpoly'
print(x, ...)
```

Arguments

x	a gmpoly object
...	ignored

Value

No value, just prints the polynomial.

Index

`as.bigq`, 3

`bigq`, 2, 4

`gmpoly`, 2, 3–6

`gmpoly2mvp`, 3

`gmpolyConstant`, 3

`gmpolyEval`, 4

`gmpolyGrow`, 5

`mvp`, 3

`Ops.gmpoly`, 5

`print.gmpoly`, 6