Package 'resultant'

July 31, 2024

Title Utilities for Multivariate Polynomials with Rational Coefficients

Version 1.0.0

Description Computation of resultant, subresultants, greatest common divisor, integral division (aka division without remainder) of two multivariate polynomials with rational coefficients, Sturm-Habicht sequence and square-free factorization of a multivariate polynomial with rational coefficients. The computations are performed by the 'C++' library 'CGAL' (<https://www.cgal.org/>). Resultants have applications in polynomial systems solving, number theory, and algebraic geometry. The package also contains some functions computing the number of real roots of a univariate polynomial with rational coefficients, and a function computing the division with remainder of two univariate polynomials with rational coefficients.

License GPL-3

URL https://github.com/stla/resultant

BugReports https://github.com/stla/resultant/issues

Depends qspray (>= 3.1.0)

Imports Rcpp, gmp, utils

Suggests testthat (>= 3.0.0)

LinkingTo BH, Rcpp, RcppCGAL

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

SystemRequirements C++17, gmp, mpfr

NeedsCompilation yes

Author Stéphane Laurent [aut, cre]

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2024-07-31 10:50:02 UTC

Contents

`gcd` *Greatest common divisor of two polynomials*

Description

Greatest common divisor of two polynomials with rational coefficients.

Usage

```
gcd(qspray1, qspray2, utcf = FALSE)
```

Arguments

`qspray1`, `qspray2`	two `qspray` polynomials with at more nine variables
`utcf`	Boolean, whether to get the greatest common divisor up to a constant factor (this can be faster)

Value

A `qspray` polynomial.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
g <- x^2 + 2*x*y + 1
p <- g * (y^2 + x^2)
q <- g * (y + x^3 + 2)
gcd(p, q)
```

`integralDivision` *Integral division of two polynomials*

Description

Integral division (division without remainder) of two polynomials with rational coefficients.

Usage

```
integralDivision(qspray1, qspray2, check = TRUE)
```

Arguments

`qspray1`, `qspray2`	two `qspray` polynomials having at most nine variables
`check`	Boolean, whether to check that `qspray2` divides `qspray1`

Value

If `check=TRUE`, this returns `NULL` if `qspray2` does not divide `qspray1`, otherwise this returns a `qspray` polynomial, the quotient of `qspray1` by `qspray2`. If `check=FALSE`, this always returns a `qspray` polynomial, which is the quotient of `qspray1` by `qspray2` if `qspray2` divides `qspray1`, otherwise it is an undefined polynomial. So you can use `check=FALSE` only when you are sure that `qspray2` divides `qspray1`.

See Also

`univariateDivision`, `qsprayDivision`.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
q <- x^2 + 2*x*y + 1
qspray1 <- q * (x^4 + y^2 + 2)
qspray2 <- x^4 + y^2 + 2
integralDivision(qspray1, qspray2) == q # should be TRUE
```

numberOfRealRoots *Number of real roots*

Description

Number of distinct real roots of a univariate polynomial.

Usage

```
numberOfRealRoots(qspray)
```

Arguments

qspray	a univariate qspray polynomial

Value

An integer, the number of real roots of the polynomial.

Note

The roots are not counted with their multiplicity.

Examples

```
library(resultant)
x <- qlone(1)
P <- 2*x^4 + x^3 - 3*x^2 - x + 1
numberOfRealRoots(P)
```

numberOfRealRootsInInterval *Number of real roots in an interval*

Description

Number of distinct real roots of a univariate polynomial in a given interval.

Usage

```
numberOfRealRootsInInterval(qspray, lower, upper, closed = TRUE)
```

Arguments

qspray	a univariate qspray polynomial
lower, upper	the bounds of the interval, bigq numbers or objects coercible to bigq numbers, and it is also possible to set lower = -Inf and upper = Inf
closed	Boolean, whether to consider the interval is closed or open

Value

An integer, the number of real roots of the polynomial in the interval.

Note

The roots are not counted with their multiplicity.

Examples

```
library(resultant)
x <- qlone(1)
P <- 2*x^4 + x^3 - 3*x^2 - x + 1
numberOfRealRootsInInterval(P, 0, 1)
```

---

principalSturmHabicht *Principal Sturm-Habicht sequence of a polynomial*

Description

Principal Sturm-Habicht sequence of a polynomial with rational coefficients.

Usage

```
principalSturmHabicht(qspray, var = 1)
```

Arguments

qspray	a qspray polynomial with at most nine variables
var	integer indicating with respect to which variable the resultant is desired (e.g. 1 for x and 2 for y)

Value

For a univariate polynomial, this returns a vector of bigq rational numbers. For a multivariate polynomial, this returns a list of qspray polynomials that do not involve the var-th variable.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
qspray <- x^3*y^2 - 5*x*y^2 + 7*x - 2
principalSturmHabicht(qspray, var = 1)
principalSturmHabicht(qspray, var = 2)
```

principalSubresultants *Principal subresultants of two polynomials*

Description

Principal subresultants of two polynomials with rational coefficients.

Usage

```
principalSubresultants(qspray1, qspray2, var = 1)
```

Arguments

qspray1, qspray2	two qspray polynomials with at most nine variables
var	integer indicating with respect to which variable the subresultants are desired (e.g. 1 for x and 2 for y)

Value

If both qspray1 and qspray2 are univariate polynomials, the function returns a vector of bigq rational numbers. Otherwise, it returns a list of qspray polynomials that do not involve the var-th variable.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
p <- x^2 * y * (y^2 - 5*x + 6)
q <- x^2 * y * (3*y + 2)
principalSubresultants(p, q, var = 1) # should be 0, 0, non-zero, ...
principalSubresultants(p, q, var = 2) # should be 0, non-zero, ...
```

resultant *Resultant of two polynomials*

Description

Resultant of two polynomials with rational coefficients.

Usage

```
resultant(qspray1, qspray2, var = 1)
```

Arguments

qspray1, qspray2	two qspray polynomials with at most nine variables
var	integer indicating with respect to which variable the resultant is desired (e.g. 1 for x and 2 for y)

Value

If both qspray1 and qspray2 are univariate polynomials, the function returns a bigq rational number. Otherwise, it returns a qspray polynomial that does not involve the var-th variable.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
f <- x^4 - x^3 + x^2 - 2*x*y^2 + y^4
g <- x - 2*y^2
resultant(f, g, var = 1)
resultant(f, g, var = 2)
```

---

squareFreeFactorization

Square-free factorization of a polynomial

Description

Square-free factorization of a polynomial with rational coefficients.

Usage

```
squareFreeFactorization(qspray)
```

Arguments

qspray	a qspray polynomial having at most nine variables

Value

A list with two fields constantFactor and nonConstantFactors. In the field constantFactor, there is a bigq rational number, the constant factor of the factorization. In the field nonConstantFactors, there is a list providing the square-free and pairwise coprime qspray polynomials of the factorization with their multiplicity.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
p <- x^8*y^7 + 2*x^7*y^5 + x^6*y^4 + 2*x^5*y^2
squareFreeFactorization(p)
```

`SturmHabicht` *Sturm-Habicht sequence of a polynomial*

Description

Sturm-Habicht sequence of a polynomial with rational coefficients.

Usage

```
SturmHabicht(qspray, var = 1)
```

Arguments

`qspray`	a `qspray` polynomial having at most nine variables
`var`	index of the variable with respect to which the Sturm-Habicht sequence will be computed

Value

A list of `qspray` polynomials, the Sturm-Habicht sequence of `qspray`, starting with the `0`-th Sturm-Habicht polynomial.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
SturmHabicht(x^3*y^2 + 2*x*y + 1)
```

subresultants *Subresultants of two polynomials*

Description

Subresultants of two polynomials with rational coefficients.

Usage

```
subresultants(qspray1, qspray2, var = 1)
```

Arguments

qspray1, qspray2	two qspray polynomials having at most nine variables
var	integer indicating with respect to which variable the subresultants will be computed (e.g. 1 for x and 2 for y)

Value

A list of qspray polynomials.

Examples

```
library(resultant)
x <- qlone(1)
y <- qlone(2)
p <- x^2 * y * (y^2 - 5*x + 6)
q <- x^2 * y * (3*y + 2)
subresultants(p, q, var = 1)
subresultants(p, q, var = 2)
```

univariateDivision *Division of univariate polynomials*

Description

Division with remainder of univariate polynomials with rational coefficients.

Usage

```
univariateDivision(qspray1, qspray2)
```

Arguments

qspray1, qspray2	two univariate qspray polynomials

Value

A list of two univariate qspray polynomials, the quotient of the division in the field Q of the list, and the remainder in the field R.

See Also

integralDivision, qsprayDivision.

Examples

```
library(resultant)
x <- qlone(1)
qspray1 <- 2*x^4 + x^3 - 3*x^2 - x + 1
qspray2 <- x^2 - 5*x + 10
division <- univariateDivision(qspray1, qspray2)
Q <- division[["Q"]]; R <- division[["R"]]
qspray1 == Q*qspray2 + R # should be TRUE
```

Index