

# Package ‘rationalfun’

July 23, 2025

**Version** 0.1-1

**Date** 2022-03-05

**Title** Manipulation of Rational Functions

**Description** Functions to manipulate rational functions, including basic arithmetic operators, derivatives, and integrals with EXPLICIT forms.

**Author** Yixuan Qiu <yixuan.qiu@cos.name>

**Maintainer** Yixuan Qiu <yixuan.qiu@cos.name>

**Depends** polynom

**Imports** stats

**URL** <https://github.com/yixuan/rationalfun>

**BugReports** <https://github.com/yixuan/rationalfun/issues>

**License** GPL

**Collate** 'internals.R' 'rf\_base.R' 'rf\_deriv\_integral.R'  
'rf\_operators.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-03-05 15:50:04 UTC

## Contents

|                                    |   |
|------------------------------------|---|
| as.character.rationalfun . . . . . | 2 |
| as.function.rationalfun . . . . .  | 2 |
| deriv.rationalfun . . . . .        | 3 |
| int2fun . . . . .                  | 4 |
| integral.rationalfun . . . . .     | 5 |
| Ops.rationalfun . . . . .          | 6 |
| predict.rationalfun . . . . .      | 6 |
| print.rationalfun . . . . .        | 7 |
| rationalfun . . . . .              | 8 |
| simplify . . . . .                 | 9 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>11</b> |
|--------------|-----------|

as.character.rationalfun

*Convert object to character*

---

### Description

This function converts an object of class "rationalfun" to a character string.

### Usage

```
## S3 method for class 'rationalfun'  
as.character(x, ...)
```

### Arguments

|     |                                  |
|-----|----------------------------------|
| x   | an object of class "rationalfun" |
| ... | not used in this function        |

### Value

A character string representing the rational function.

### See Also

[as.character.polynomial](#)

### Examples

```
r <- rationalfun(c(1, 1), c(3, 2, 1))  
as.character(r)
```

---

as.function.rationalfun

*Convert object to function*

---

### Description

This function converts an object of class "rationalfun" to a function.

### Usage

```
## S3 method for class 'rationalfun'  
as.function(x, ...)
```

**Arguments**

|     |                                  |
|-----|----------------------------------|
| x   | an object of class "rationalfun" |
| ... | not used in this function        |

**Value**

A function with one argument which could be a real or complex vector.

**See Also**

[as.function.polynomial](#)

**Examples**

```
r <- rationalfun(c(1, 1), c(3, 2, 1))
r
f <- as.function(r)
f
f(1:10)
f(1:10 + (0+2i))
```

---

|                   |                                          |
|-------------------|------------------------------------------|
| deriv.rationalfun | <i>Differentiate a rational function</i> |
|-------------------|------------------------------------------|

---

**Description**

Calculate the derivative of a rational function. The returned value result is still an object of class "rationalfun".

**Usage**

```
## S3 method for class 'rationalfun'
deriv(expr, ...)
```

**Arguments**

|      |                                  |
|------|----------------------------------|
| expr | an object of class "rationalfun" |
| ...  | not used in this function        |

**Value**

An object of class "rationalfun" representing the derivative of the original rational function.

**See Also**

[deriv.polynomial](#), [deriv](#)

**Examples**

```
# (x + 1) / (x^2 + x + 1)
r <- rationalfun(c(1, 1), c(1, 1, 1))
deriv(r)
```

---

|         |                                     |
|---------|-------------------------------------|
| int2fun | <i>Convert a call to a function</i> |
|---------|-------------------------------------|

---

**Description**

Convert a function call to a function in R. In this package, the function is typically used to convert the result of [integral.rationalfun\(\)](#) to a function with one argument.

**Usage**

```
int2fun(expr)
```

**Arguments**

expr                    a function call, typically returned by [integral.rationalfun\(\)](#).

**Value**

A function with one argument which could be a real or complex vector.

**See Also**

[integral.polynomial](#)

**Examples**

```
x <- rationalfun(c(-6, -1, -8, 15, -1, 8, -9, 2),
  c(8, 12, 16, 4, 4))
int <- integral(x)
fun <- int2fun(int)
fun(c(0, 1))
```

---

`integral.rationalfun`    *Integrate a rational function*

---

**Description**

Calculate the integral of a rational function. See "Details".

**Usage**

```
## S3 method for class 'rationalfun'  
integral(expr, ...)
```

**Arguments**

|                   |                                  |
|-------------------|----------------------------------|
| <code>expr</code> | an object of class "rationalfun" |
| <code>...</code>  | not used in this function        |

**Details**

The returned value is a function call with argument named "x". That is, the integral is an expression in R with an explicit form, which could be evaluated directly by calling `eval()`, or indirectly using the `int2fun()` function.

The algorithm is based on the Hermite-Ostrogradski formula which is discussed in the reference. See the article for more details.

**Value**

A function call representing the explicit form of the integral.

**References**

T. N. Subramaniam, and Donald E. G. Malm, How to Integrate Rational Functions, *The American Mathematical Monthly*, Vol. 99, No.8 (1992), 762-772.

**See Also**

[integral.polynomial](#)

**Examples**

```
# (x + 1) / (x^2 + x + 1)  
r <- rationalfun(c(1, 1), c(1, 1, 1))  
expr <- integral(r)  
# Evaluate the call directly  
eval(expr, list(x = 2))  
# Use int2fun()  
f <- int2fun(expr)  
f(2)
```

---

|                 |                                         |
|-----------------|-----------------------------------------|
| Ops.rationalfun | <i>Operators for rational functions</i> |
|-----------------|-----------------------------------------|

---

**Description**

Basic arithmetic operators for rational functions.

**Usage**

```
## S3 method for class 'rationalfun'
Ops(e1, e2)
```

**Arguments**

|    |                                                                               |
|----|-------------------------------------------------------------------------------|
| e1 | an object of class "rationalfun"                                              |
| e2 | for "^", a positive integer; in other cases, an object of class "rationalfun" |

**Value**

A new object of "rationalfun" class.

**Examples**

```
r1 <- rationalfun(c(1, 2), c(1, 2, 1))
r2 <- rationalfun(c(1, 1), c(1, -2, 1))
r1 + r2
r1 * r2
r1^2
```

---

|                     |                                     |
|---------------------|-------------------------------------|
| predict.rationalfun | <i>Evaluate a rational function</i> |
|---------------------|-------------------------------------|

---

**Description**

Evaluate a rational function at a real or complex vector.

**Usage**

```
## S3 method for class 'rationalfun'
predict(object, newdata, ...)
```

**Arguments**

|         |                                                                       |
|---------|-----------------------------------------------------------------------|
| object  | an object of class "rationalfun"                                      |
| newdata | a vector at which evaluation is requested.                            |
| ...     | not used in this function Both real and complex vectors are accepted. |

**Value**

A vector of evaluated results.

**See Also**

[predict.polynomial](#)

**Examples**

```
r <- rationalfun(c(1, 1), c(3, 2, 1))
predict(r, 1:10)
```

---

|                   |                                  |
|-------------------|----------------------------------|
| print.rationalfun | <i>Print a rational function</i> |
|-------------------|----------------------------------|

---

**Description**

Print a rational function in a fraction form.

**Usage**

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

**Arguments**

|     |                                  |
|-----|----------------------------------|
| x   | an object of class "rationalfun" |
| ... | not used in this function        |

**Value**

Invisible, the object itself.

**See Also**

[print.polynomial](#)

**Examples**

```
r <- rationalfun(c(1, 1), c(3, 2, 1))
print(r)
```

---

|             |                                           |
|-------------|-------------------------------------------|
| rationalfun | <i>Construction of rational functions</i> |
|-------------|-------------------------------------------|

---

### Description

Construction of rational functions.

### Usage

```
rationalfun(number = c(0, 1), denom = c(1, 1, 1))

rfun(number = c(0, 1), denom = c(1, 1, 1))

rationalfun.poly(number = polynomial(c(0, 1)), denom = polynomial(c(1,
1, 1)))

rfun.poly(number = polynomial(c(0, 1)), denom = polynomial(c(1,
1, 1)))
```

### Arguments

|        |                                                                                                                                                                                                 |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| number | in <code>rationalfun()</code> , the coefficient vector of the numerator; in <code>rationalfun.poly()</code> , an object of class "polynom" in <b>polynom</b> package representing the numerator |
| denom  | similar to <code>number</code> , but for the denominator                                                                                                                                        |

### Details

A rational function object could be constructed either by calling `rationalfun()` or by calling `rationalfun.poly()`.

`rationalfun()` constructs a rational function from the coefficient vectors of the numerator and the denominator. For example, consider a rational function  $R(x) = P(x)/Q(x)$  where

$$P(x) = p_1 + p_2x + p_3x^2 + \cdots + p_kx^{k-1}$$

and

$$Q(x) = q_1 + q_2x + q_3x^2 + \cdots + q_mx^{m-1}$$

, you may call `rationalfun(p[1:k], q[1:m])` to build the object.

For `rationalfun.poly()`, it receives two objects of class "polynomial" from the **polynom** package, representing the polynomials of the numerator and the denominator respectively. Use this function if you already have objects of "polynomial" class, typically by calling `polynomial()`, `poly.calc()` or `poly.orth()`.

`rfun()` and `rfun.poly()` are aliases of `rationalfun()` and `rationalfun.poly()` in order to type fewer letters.

The value returned by `rationalfun()` and `rationalfun.poly()` is an object of class "rationalfun". You can coerce the object to a function, by calling `as.function.rationalfun()`, or to a character string, by calling `as.character.rationalfun()`.

Objects of "ratioanlfun" class support basic operators including "+", "-", "\*", "/" and "^". To evaluate a rational function at a given vector, use `predict.rationalfun()`. To compute the derivative and integral in **explicit** form, call `deriv.rationalfun()` and `integral.rationalfun()` respectively.

### Value

An object of class "rationalfun".

### See Also

`polynomial`, `poly.calc`, `poly.orth`

### Examples

```
# (x + 1) / (x^2 + 2 * x + 3)
r1 <- rationalfun(c(1, 1), c(3, 2, 1))
print(r1)
# Construct from objects of 'polynomial' class
if (require(polynom)) {
  p1 <- poly.calc(c(1, 2))
  p2 <- polynomial(rep(1, 5))
  r2 <- rfun.poly(p1, p2)
  print(r2)
}
```

---

simplify

*Simplify a rational function*


---

### Description

Simplify a rational function by dropping terms whose coefficients are close to zero, and then reducing it to an irreducible form.

### Usage

```
simplify(x, ...)
```

### Arguments

`x`                      an object of class "rationalfun"  
`...`                    currently not used in this function

### Value

A new object of class "rationalfun" representing the simplified rational function.

**Examples**

```
# (x + 1) / (x^2 + 2 * x + 1) ==> 1 / (x + 1)
r <- rationalfun(c(1, 1), c(1, 2, 1))
simplify(r)
```

# Index

## \* **symbolmath**

- as.character.rationalfun, 2
- as.function.rationalfun, 2
- deriv.rationalfun, 3
- int2fun, 4
- integral.rationalfun, 5
- Ops.rationalfun, 6
- predict.rationalfun, 6
- print.rationalfun, 7
- simplify, 9

- as.character.polynomial, 2
- as.character.rationalfun, 2, 8
- as.function.polynomial, 3
- as.function.rationalfun, 2, 8

- deriv, 3
- deriv.polynomial, 3
- deriv.rationalfun, 3, 9

- eval, 5

- int2fun, 4, 5
- integral.polynomial, 4, 5
- integral.rationalfun, 4, 5, 9

- Ops.rationalfun, 6

- poly.calc, 8, 9
- poly.orth, 8, 9
- polynomial, 8, 9
- predict.polynomial, 7
- predict.rationalfun, 6, 9
- print.polynomial, 7
- print.rationalfun, 7

- rationalfun, 8
- rfun (rationalfun), 8

- simplify, 9