Package ‘jacobi’

July 19, 2022

Type Package

Title Jacobi Theta Functions and Related Functions

Version 2.0.0

Description Evaluation of the Jacobi theta functions and related functions: Weierstrass elliptic function, Weierstrass sigma function, Weierstrass zeta function, Klein j-function, Dedekind eta function, lambda modular function, Jacobi elliptic functions, Neville theta functions, and Eisenstein series. Complex values of the variable are supported.

License GPL-3

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

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

Imports Carlson, Rcpp (>= 1.0.8), rgl, Rvcg

Suggests testthat (>= 3.0.0), elliptic

LinkingTo Rcpp

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation yes

Author Stéphane Laurent [aut, cre],
Mikael Fremling [aut] (author of the original Fortran code for the theta functions)

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2022-07-19 12:00:09 UTC
\section*{R topics documented:}

\begin{itemize}
  \item \texttt{agm} \hspace{1cm} 2
  \item \texttt{am} \hspace{1cm} 3
  \item \texttt{CostaMesh} \hspace{1cm} 3
  \item \texttt{EisensteinE} \hspace{1cm} 4
  \item \texttt{eta} \hspace{1cm} 4
  \item \texttt{jellip} \hspace{1cm} 5
  \item \texttt{jtheta1} \hspace{1cm} 6
  \item \texttt{jtheta2} \hspace{1cm} 6
  \item \texttt{jtheta3} \hspace{1cm} 7
  \item \texttt{jtheta4} \hspace{1cm} 8
  \item \texttt{kleinj} \hspace{1cm} 8
  \item \texttt{lambda} \hspace{1cm} 9
  \item \texttt{theta.s} \hspace{1cm} 10
  \item \texttt{wp} \hspace{1cm} 10
  \item \texttt{wpinv} \hspace{1cm} 11
  \item \texttt{wsigma} \hspace{1cm} 12
  \item \texttt{wzeta} \hspace{1cm} 13
\end{itemize}

\section*{Index}

\begin{itemize}
  \item \texttt{agm} \hspace{1cm} \textit{Arithmetic-geometric mean}
\end{itemize}

\section*{Description}

Evaluation of the arithmetic-geometric mean of two complex numbers.

\section*{Usage}

\begin{verbatim}
agm(x, y)
\end{verbatim}

\section*{Arguments}

\begin{itemize}
  \item \texttt{x, y} \hspace{1cm} complex numbers
\end{itemize}

\section*{Value}

A complex number, the arithmetic-geometric mean of \texttt{x} and \texttt{y}.

\section*{Examples}

\begin{verbatim}
agm(1, sqrt(2))
2*pi^(3/2)*sqrt(2) / gamma(1/4)^2
\end{verbatim}
**am**  
*Amplitude function*

**Description**
Evaluation of the amplitude function.

**Usage**
```
am(u, m)
```

**Arguments**
- `u` complex number
- `m` square of elliptic modulus, a complex number

**Value**
A complex number.

**Examples**
```
library(Carlson)
phi <- 1 + 1i
m <- 2
u <- elliptic_F(phi, m)
am(u, m) # should be phi
```

---

**CostaMesh**  
*Costa surface*

**Description**
Computes a mesh of the Costa surface.

**Usage**
```
CostaMesh(nu = 50L, nv = 50L)
```

**Arguments**
- `nu`, `nv` numbers of subdivisions

**Value**
A triangle `rgl` mesh (object of class `mesh3d`).
Examples

```r
library(jacobi)
library(rgl)

mesh <- CostaMesh(nu = 250, nv = 250)
open3d(windowRect = c(50, 50, 562, 562), zoom = 0.9)
bg3d("#15191E")
shade3d(mesh, color = "darkred", back = "cull")
shade3d(mesh, color = "orange", front = "cull")
```

---

**EisensteinE**  
_Eisenstein series_

**Description**

Evaluation of Eisenstein series with weight 2, 4 or 6.

**Usage**

```r
EisensteinE(n, q)
```

**Arguments**

- `n`  
  the weight, can be 2, 4 or 6
- `q`  
  nome, complex number with modulus smaller than one, but not a negative real number

**Value**

A complex number, the value of the Eisenstein series.

---

**eta**  
_Dedekind eta function_

**Description**

Evaluation of the Dedekind eta function.

**Usage**

```r
eta(tau)
```

**Arguments**

- `tau`  
  a complex number with strictly positive imaginary part
Value

A complex number.

Examples

\[
\begin{align*}
\eta(2i) \\
\gamma(1/4) / 2^{11/8} / \pi^{3/4}
\end{align*}
\]

jellip

**Jacobi elliptic functions**

Description

Evaluation of the Jacobi elliptic functions.

Usage

\[
jellip(\text{kind}, u, \tau = \text{NULL}, m = \text{NULL})
\]

Arguments

kind

a string with two characters among "s", "c", "d" and "n"; this string specifies the function: the two letters respectively denote the basic functions \(sn, cn, dn\) and 1, and the string specifies the ratio of two such functions, e.g. \(ns = 1/sn\) and \(cd = cn/dn\)

u

a complex number, vector or matrix

tau

complex number with strictly positive imaginary part; it is related to \(m\) and only one of them must be supplied

m

the "parameter", square of the elliptic modulus; it is related to \(\tau\) and only one of them must be supplied

Value

A complex number, vector or matrix.

Examples

\[
\begin{align*}
u & \leftarrow 2 + 2i \\
\tau & \leftarrow 1i \\
jellip("cn", u, \tau)^2 + jellip("sn", u, \tau)^2 & \# \text{ should be } 1
\end{align*}
\]
**jtheta1**

*Jacobi theta function one*

**Description**

Evaluates the first Jacobi theta function.

**Usage**

\[
\text{jtheta1}(z, \tau = \text{NULL}, q = \text{NULL})
\]

\[
\text{ljtheta1}(z, \tau = \text{NULL}, q = \text{NULL})
\]

**Arguments**

- **z**: complex number, vector, or matrix
- **\(\tau\)**: lattice parameter, a complex number with strictly positive imaginary part; the two complex numbers \(\tau\) and \(q\) are related by \(q = \exp(1i\pi\tau)\), and only one of them must be supplied
- **\(q\)**: the nome, a complex number whose modulus is strictly less than one, and which is not zero nor a negative real number

**Value**

A complex number, vector or matrix; \text{jtheta1} evaluates the first Jacobi theta function and \text{ljtheta1} evaluates its logarithm.

**Examples**

\[
\text{jtheta1}(1 + 1i, q = \exp(-\pi/2))
\]

---

**jtheta2**

*Jacobi theta function two*

**Description**

Evaluates the second Jacobi theta function.

**Usage**

\[
\text{jtheta2}(z, \tau = \text{NULL}, q = \text{NULL})
\]

\[
\text{ljtheta2}(z, \tau = \text{NULL}, q = \text{NULL})
\]
**jtheta3**

**Arguments**

- **z**
  complex number, vector, or matrix

- **tau**
  lattice parameter, a complex number with strictly positive imaginary part; the two complex numbers tau and q are related by $q = \exp(1i\pi\tau)$, and only one of them must be supplied

- **q**
  the nome, a complex number whose modulus is strictly less than one, and which is not zero nor a negative real number

**Value**

A complex number, vector or matrix; jtheta2 evaluates the second Jacobi theta function and ljtheta2 evaluates its logarithm.

**Examples**

```r
jtheta2(1 + 1i, q = exp(-pi/2))
```

---

**jtheta3**  

*Jacobi theta function three*

**Description**

Evaluates the third Jacobi theta function.

**Usage**

```r
jtheta3(z, tau = NULL, q = NULL)
ljtheta3(z, tau = NULL, q = NULL)
```

**Arguments**

- **z**
  complex number, vector, or matrix

- **tau**
  lattice parameter, a complex number with strictly positive imaginary part; the two complex numbers tau and q are related by $q = \exp(1i\pi\tau)$, and only one of them must be supplied

- **q**
  the nome, a complex number whose modulus is strictly less than one, and which is not zero nor a negative real number

**Value**

A complex number, vector or matrix; jtheta3 evaluates the third Jacobi theta function and ljtheta3 evaluates its logarithm.

**Examples**

```r
jtheta3(1 + 1i, q = exp(-pi/2))
```
**jtheta4** *Jacobi theta function four*

### Description
Evaluates the fourth Jacobi theta function.

### Usage
- `jtheta4(z, tau = NULL, q = NULL)`
- `ljtheta4(z, tau = NULL, q = NULL)`

### Arguments
- **z**: complex number, vector, or matrix
- **tau**: lattice parameter, a complex number with strictly positive imaginary part; the two complex numbers `tau` and `q` are related by `q = exp(1i*pi*tau)`, and only one of them must be supplied
- **q**: the nome, a complex number whose modulus is strictly less than one, and which is not zero nor a negative real number

### Value
A complex number, vector or matrix; `jtheta4` evaluates the fourth Jacobi theta function and `ljtheta4` evaluates its logarithm.

### Examples
- `jtheta4(1 + 1i, q = exp(-pi/2))`

---

**kleinj** *Klein j-function and its inverse*

### Description
Evaluation of the Klein j-invariant function and its inverse.

### Usage
- `kleinj(tau, transfo = FALSE)`
- `kleinjinv(j)`
lambda

Arguments

tau a complex number with strictly positive imaginary part, or a vector or matrix of such complex numbers; missing values allowed
transfo Boolean, whether to use a transformation of the values of tau close to the real line; using this option can fix some failures of the computation (at the cost of speed), e.g. when the algorithm reaches the maximal number of iterations
j a complex number

Value

A complex number, vector or matrix.

Note

The Klein-j function is the one with the factor 1728.

Examples

( j <- kleinj(2i) )
66^3
kleinjinv(j)

Description

Evaluation of the lambda modular function.

Usage

lambda(tau, transfo = FALSE)

Arguments

tau a complex number with strictly positive imaginary part, or a vector or matrix of such complex numbers; missing values allowed
transfo Boolean, whether to use a transformation of the values of tau close to the real line; using this option can fix some failures of the computation (at the cost of speed), e.g. when the algorithm reaches the maximal number of iterations

Value

A complex number, vector or matrix.

Note

The lambda function is the square of the elliptic modulus.
Examples

\[
x <- 2
\]
\[
\text{lambda}(1i*\sqrt{x}) + \text{lambda}(1i*\sqrt{1/x}) \ # \text{should be one}
\]

---

**theta.s**  
*Neville theta functions*

**Description**

Evaluation of the Neville theta functions.

**Usage**

\[
\text{theta.s}(z, \tau = \text{NULL}, m = \text{NULL})
\]
\[
\text{theta.c}(z, \tau = \text{NULL}, m = \text{NULL})
\]
\[
\text{theta.n}(z, \tau = \text{NULL}, m = \text{NULL})
\]
\[
\text{theta.d}(z, \tau = \text{NULL}, m = \text{NULL})
\]

**Arguments**

- **z**  
a complex number, vector, or matrix
- **tau**  
complex number with strictly positive imaginary part; it is related to \(m\) and only one of them must be supplied
- **m**  
the "parameter", square of the elliptic modulus; it is related to \(\tau\) and only one of them must be supplied

**Value**

A complex number, vector or matrix.

---

**wp**  
*Weierstrass elliptic function*

**Description**

Evaluation of the Weierstrass elliptic function and its derivatives.

**Usage**

\[
\text{wp}(z, g = \text{NULL}, \omega = \text{NULL}, \tau = \text{NULL}, \text{derivative} = 0L)
\]
Arguments

- **z**: complex number, vector or matrix
- **g**: the elliptic invariants, a vector of two complex numbers; only one of g, omega and tau must be given
- **omega**: the half-periods, a vector of two complex numbers; only one of g, omega and tau must be given
- **tau**: the half-periods ratio; supplying tau is equivalent to supply omega = c(1/2, tau/2)
- **derivative**: differentiation order, an integer between 0 and 3

Value

A complex number, vector or matrix.

Examples

```r
omega1 <- 1.4 - 1i
omega2 <- 1.6 + 0.5i
omega <- c(omega1, omega2)
e1 <- wp(omega1, omega = omega)
e2 <- wp(omega2, omega = omega)
e3 <- wp(-omega1-omega2, omega = omega)
e1 + e2 + e3 # should be 0
```

Description

Evaluation of the inverse of the Weierstrass elliptic function.

Usage

```r
wpinv(w, g = NULL, omega = NULL, tau = NULL)
```
Value

A complex number.

Examples

```r
library(jacobi)
omega <- c(1.4 - 1i, 1.6 + 0.5i)
w <- 1 + 1i
z <- wpinv(w, omega = omega)
wp(z, omega = omega) # should be w
```

**Description**

Evaluation of the Weierstrass sigma function.

**Usage**

```r
wsigma(z, g = NULL, omega = NULL, tau = NULL)
```

**Arguments**

- `z` a complex number, vector or matrix
- `g` the elliptic invariants, a vector of two complex numbers; only one of `g`, `omega` and `tau` must be given
- `omega` the half-periods, a vector of two complex numbers; only one of `g`, `omega` and `tau` must be given
- `tau` the half-periods ratio; supplying `tau` is equivalent to supply `omega = c(1/2, tau/2)`

**Value**

A complex number, vector or matrix.

**Examples**

```r
wsigma(1, g = c(12, -8))
# should be equal to:
sin(1i*sqrt(3))/(1i*sqrt(3)) / sqrt(exp(1))
```
wzeta

Weierstrass zeta function

Description

Evaluation of the Weierstrass zeta function.

Usage

wzeta(z, g = NULL, omega = NULL, tau = NULL)

Arguments

z complex number, vector or matrix
g the elliptic invariants, a vector of two complex numbers; only one of g, omega and tau must be given
omega the half-periods, a vector of two complex numbers; only one of g, omega and tau must be given
tau the half-periods ratio; supplying tau is equivalent to supply omega = c(1/2, tau/2)

Value

A complex number, vector or matrix.

Examples

# Mirror symmetry property:
z <- 1 + 1i
g <- c(1i, 1+2i)
wzeta(Conj(z), Conj(g))
Conj(wzeta(z, g))
Index

agm, 2
am, 3
CostaMesh, 3
EisensteinE, 4
eta, 4
jellip, 5
jtheta1, 6
jtheta2, 6
jtheta3, 7
jtheta4, 8
kleinj, 8
kleinjinv (kleinj), 8
lambda, 9
ljtheta1 (jtheta1), 6
ljtheta2 (jtheta2), 6
ljtheta3 (jtheta3), 7
ljtheta4 (jtheta4), 8
theta.c (theta.s), 10
theta.d (theta.s), 10
theta.n (theta.s), 10
theta.s, 10
wp, 10
wpinv, 11
wsigma, 12
wzeta, 13