Package 'RcppFaddeeva'

August 29, 2016

Type Package

Title 'Rcpp' Bindings for the 'Faddeeva' Package

Version 0.1.0

Date 2015-06-07

Description Access to a family of Gauss error functions for arbitrary complex arguments is provided via the 'Faddeeva' package by Steven G. Johnson (see <http://ab-initio.mit.edu/wiki/index.php/Faddeeva_Package> for more information).

License GPL (>= 2)

Imports Rcpp (>= 0.11.0), knitr

Suggests testthat

VignetteBuilder knitr

LinkingTo Rcpp

NeedsCompilation yes

Author Baptiste Auguie [aut, cre],
Dirk Eddelbuettel [aut],
Steven G. Johnson [aut] (Author of Faddeeva)

Maintainer Baptiste Auguie <baptiste.auguie@gmail.com>

Repository CRAN

Date/Publication 2015-06-09 07:48:31

R topics documented:

RcppFaddeeva-package — *RcppFaddeeva*

Description

Rcpp Bindings for the "Faddeeva" Library

Details

Access to error functions for arbitrary complex arguments is provided via the Faddeeva package by Steven G. Johnson.

References

The Faddeeva Package wiki page details the algorithms implemented by Steve G. Johnson, http://ab-initio.mit.edu/wiki/index.php/Faddeeva_Package

Faddeeva_w — *Faddeeva family of error functions of the complex variable*

Description

the Faddeeva function

the scaled complementary error function

the error function of complex arguments

the imaginary error function

the complementary error function

the Dawson function

Usage

```
Faddeeva_w(z, relerr = 0)

erfcx(z, relerr = 0)

erf(z, relerr = 0)

erfi(z, relerr = 0)

erfc(z, relerr = 0)

Dawson(z, relerr = 0)
```

Arguments

`z`	complex vector
`relerr`	double, requested error

Value

complex vector

Functions

- `Faddeeva_w`: compute w(z) = exp(-z^2) erfc(-iz)
- `erfcx`: compute erfcx(z) = exp(z^2) erfc(z)
- `erf`: compute erf(z)
- `erfi`: compute erfi(z) = -i erf(iz)
- `erfc`: compute erfc(z) = 1 - erf(z)
- `Dawson`: compute Dawson(z) = sqrt(pi)/2 * exp(-z^2) * erfi(z)

Examples

```
Faddeeva_w(1:10 + 1i)
erfcx(1:10 + 1i)
erf(1:10 + 1i)
erfi(1:10 + 1i)
erfc(1:10 + 1i)
Dawson(1:10 + 1i)
```

`Voigt` *The Voigt function, corresponding to the convolution of a lorentzian and a gaussian distribution*

Description

Voigt distribution

Lorentzian distribution

Gaussian distribution

Usage

```
Voigt(x, x0, sigma, gamma, real = TRUE, ...)

Lorentz(x, x0, gamma)

Gauss(x, x0, sigma)
```

Arguments

`x`	numeric vector
`x0`	scalar, peak position
`sigma`	parameter of the gaussian
`gamma`	parameter of the lorentzian
`real`	logical, return only the real part of the complex Faddeeva
`...`	passed to Faddeeva_w

Value

numeric or complex vector

Functions

- `Voigt`: Voigt lineshape function
- `Lorentz`: Lorentzian lineshape function
- `Gauss`: Gaussian lineshape function

Author(s)

baptiste Auguie

Examples

```
## should integrate to 1 in all cases
integrate(Lorentz, -Inf, Inf, x0=200, gamma=100)
integrate(Gauss, -Inf, Inf, x0=200, sigma=50)
integrate(Voigt, -Inf, Inf, x0=200, sigma=50, gamma=100)

## visual comparison
x <- seq(-1000, 1000)
x0 <- 200
l <- Lorentz(x, x0, 30)
g <- Gauss(x, x0, 100)
N <- length(x)
c <- convolve(Gauss(x, 0, 100),
              rev(Lorentz(x, x0, 30)), type="o")[seq(N/2, length=N)]
v <- Voigt(x, x0, 100, 30)
matplot(x, cbind(v, l, g, c), t="l", lty=c(1,2,2,1), xlab="x", ylab="")
legend("topleft", legend = c("Voigt", "Lorentz", "Gauss", "Convolution"), bty="n",
       lty=c(1,2,2,1), col=1:4)
```

Index