Package: RcppFaddeeva (via r-universe)

September 8, 2024

Type Package

Title 'Rcpp' Bindings for the 'Faddeeva' Package

Version 0.2.3

Date 2022-05-28

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)

Encoding UTF-8

Imports Rcpp (>= 0.11.0), stats

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/nano-optics/RcppFaddeeva

BugReports https://github.com/nano-optics/RcppFaddeeva/issues

LinkingTo Rcpp

NeedsCompilation yes

RoxygenNote 7.2.3

Repository https://nano-optics.r-universe.dev

RemoteUrl https://github.com/nano-optics/RcppFaddeeva

RemoteRef HEAD

RemoteSha 344008f7e53529a0617165c1d8b88fcfd6331924

Contents

RcppFaddeeva-package *RcppFaddeeva*

Description

'Rcpp' Bindings for the 'Faddeeva' Package

Details

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).

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

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 $w(z) = \exp(-z^2)\,\mathrm{erfc}(-iz)$
- `erfcx()`: compute $\mathrm{erfcx}(z) = \exp(z^2)\,\mathrm{erfc}(z)$
- `erf()`: compute $\mathrm{erf}(z)$
- `erfi()`: compute $\mathrm{erfi}(z) = -i\,\mathrm{erf}(iz)$
- `erfc()`: compute $\mathrm{erfc}(z) = 1 - \mathrm{erf}(z)$
- `Dawson()`: compute $\mathrm{Dawson}(z) = \sqrt{\pi}/2\exp(-z^2)\,\mathrm{erfi}(z)$

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