

# Package ‘RcppFastAD’

September 24, 2024

**Type** Package

**Title** 'Rcpp' Bindings to 'FastAD' Auto-Differentiation

**Version** 0.0.4

**Date** 2024-09-24

**Description** The header-only 'C++' template library 'FastAD' for automatic differentiation <<https://github.com/JamesYang007/FastAD>> is provided by this package, along with a few illustrative examples that can all be called from R.

**URL** <https://github.com/eddelbuettel/rcppfastad>

**BugReports** <https://github.com/eddelbuettel/rcppfastad/issues>

**License** GPL (>= 2)

**Suggests** tinytest

**Encoding** UTF-8

**RoxygenNote** 6.0.1

**Imports** Rcpp

**LinkingTo** Rcpp, RcppEigen

**NeedsCompilation** yes

**Author** Dirk Eddelbuettel [aut, cre] (<<https://orcid.org/0000-0001-6419-907X>>),  
James Yang [aut] (<<https://orcid.org/0000-0002-0015-7812>>)

**Maintainer** Dirk Eddelbuettel <edd@debian.org>

**Repository** CRAN

**Date/Publication** 2024-09-24 12:50:06 UTC

## Contents

RcppFastAD-package . . . . .	2
black_scholes . . . . .	2
linear_regression . . . . .	3
quadratic_expression . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

---

RcppFastAD-package     *'Rcpp' Bindings to 'FastAD' Auto-Differentiation*

---

### Description

The header-only 'C++' template library 'FastAD' for automatic differentiation <<https://github.com/JamesYang007/FastAD>> is provided by this package, along with a few illustrative examples that can all be called from R.

### Package Content

Index of help topics:

RcppFastAD-package	'Rcpp' Bindings to 'FastAD' Auto-Differentiation
black_scholes	Black-Scholes valuation and first derivatives via Automatic Differentiation
linear_regression	Evaluate a squared-loss linear regression at a given parameter value
quadratic_expression	Compute the value and derivate of a quadratic expression $X' * \text{Sigma} * X$

### Maintainer

Dirk Eddelbuettel <[edd@debian.org](mailto:edd@debian.org)>

### Author(s)

Dirk Eddelbuettel [aut, cre] (<<https://orcid.org/0000-0001-6419-907X>>), James Yang [aut] (<<https://orcid.org/0000-0002-0015-7812>>)

---

black\_scholes     *Black-Scholes valuation and first derivatives via Automatic Differentiation*

---

### Description

This example illustrate how to use automatic differentiation to calculate the delte of a Black-Scholes call and put. It is based on the same example in the FastAD sources.

### Usage

```
black_scholes(spot = 105, strike = 100, vol = 5, r = 1.25/100,
tau = 30/365)
```

**Arguments**

spot	A double with the spot price, default is 105 as in Boost example
strike	A double with the strike price, default is 100 as in Boost example
vol	A double with the (annualized) volatility (in percent), default is 5 (for 500 percent) as in Boost example
r	A double with the short-term risk-free rate, default is 0.0125 as in Boost example
tau	A double with the time to expiration (in fractional years), default is 30/365 as in Boost example

**Value**

A matrix with rows for the call and put variant, and columns for option value, delta and vega

**Examples**

```
black_scholes()
```

---

linear_regression	<i>Evaluate a squared-loss linear regression at a given parameter value</i>
-------------------	---

---

**Description**

Not that this function does not actually fit the model. Rather it evaluates the squared sum of residuals and 'gradient' of parameters.

**Usage**

```
linear_regression(X, y, theta_hat, initial_lr = 1e-04, max_iter = 100L,  
tol = 1e-07)
```

**Arguments**

X	Matrix with independent explanatory variables
y	Vector with dependent variable
theta_hat	Vector with initial 'guess' of parameter values
initial_lr	[Optional] Scalar with initial step-size value, default is 1e-4
max_iter	[Optional] Scalar with maximum number of iterations, default is 100
tol	[Optional] Scalar with convergence tolerance, default is 1e-7

**Value**

A list object with the 'loss', 'theta' (parameters), 'gradient' and 'iter' for iterations

**Examples**

```
data(trees) # also used in help(lm)
X <- as.matrix(cbind(const=1, trees[, c("Girth", "Height")]))
y <- trees$Volume
linear_regression(X, y, rep(0, 3), tol=1e-12)
coef(lm(y ~ X - 1)) # for comparison
```

---

quadratic\_expression    *Compute the value and derivate of a quadratic expression  $X' * \text{Sigma} * X$*

---

**Description**

Compute the value and derivate of a quadratic expression  $X' * \text{Sigma} * X$

**Usage**

```
quadratic_expression(X, Sigma)
```

**Arguments**

X	A 2 element vector
Sigma	A 2 x 2 matrix

**Value**

A list with two elements for the expression evaluated for X and Sigma as well as

**Examples**

```
X <- c(0.5, 0.6)
S <- matrix(c(2, 3, 3, 6), 2, 2)
quadratic_expression(X, S)
```

# Index

## \* **package**

RcppFastAD-package, [2](#)

black\_scholes, [2](#)

linear\_regression, [3](#)

quadratic\_expression, [4](#)

RcppFastAD (RcppFastAD-package), [2](#)

RcppFastAD-package, [2](#)