

# Package ‘JacobiEigen’

May 7, 2026

**Type** Package

**Title** Classical Jacobi Eigenvalue Algorithm

**Version** 0.3-4

**Date** 2021-04-17

**Author** Bill Venables

**Imports** Rcpp

**Maintainer** Bill Venables <Bill.Venables@gmail.com>

**Description** Implements the classical Jacobi algorithm for the eigenvalues and eigenvectors of a real symmetric matrix, both in pure 'R' and in 'C++' using 'Rcpp'. Mainly as a programming example for teaching purposes.

**License** GPL (>= 2)

**LinkingTo** Rcpp

**Suggests** stats, knitr, dplyr, tidyr, ggplot2, rbenchmark, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** yes

**RoxygenNote** 6.1.1

**Repository** CRAN

**Date/Publication** 2021-04-17 04:50:09 UTC

## Contents

Jacobi	2
JacobiR	3
JacobiS	4
<b>Index</b>	<b>5</b>

---

Jacobi

*The Jacobi Algorithm using Rcpp*

---

## Description

The Classical Jacobi Algorithm

## Usage

```
Jacobi(x, symmetric = TRUE, only.values = FALSE, eps = 0)
```

## Arguments

<code>x</code>	A real symmetric matrix
<code>symmetric</code>	a logical value. Is the matrix symmetric? (Only symmetric matrices are allowed.)
<code>only.values</code>	A logical value: do you want only the eigenvalues?
<code>eps</code>	an error tolerance. 0.0 implies <code>.Machine\$double.eps</code> and <code>sqrt(.Machine\$double.eps)</code> if <code>only.values = TRUE</code>

## Details

Eigenvalues and optionally, eigenvectors, of a real symmetric matrix using the classical Jacobi algorithm, (Jacobi, 1854)

## Value

a list of two components as for `base::eigen`

## Examples

```
V <- crossprod(matrix(runif(40, -1, 1), 8))
Jacobi(V)
identical(Jacobi(V), JacobiR(V))
all.equal(Jacobi(V)$values, base::eigen(V)$values)
```

---

JacobiR

*The Jacobi Algorithm in Pure R*

---

## Description

The Jacobi Algorithm

## Usage

```
JacobiR(x, symmetric = TRUE, only.values = FALSE, eps = if  
(!only.values) .Machine$double.eps else sqrt(.Machine$double.eps))
```

## Arguments

x	a real symmetric matrix
symmetric	a logical value. Is the matrix symmetric? (Only symmetric matrices are allowed.)
only.values	A logical value: Do you want only the eigenvalues?
eps	a small positive error tolerance

## Details

Eigenvalues and optionally, eigenvectors of a real symmetric matrix using the classical Jacobi algorithm, (Jacobi, 1854)

## Value

a list of two components as for `base::eigen`

## Examples

```
V <- crossprod(matrix(rnorm(25), 5))  
JacobiR(V)  
identical(Jacobi(V), JacobiR(V))  
all.equal(Jacobi(V)$values, base::eigen(V)$values)
```

---

 JacobiS

*The Jacobi Algorithm using Rcpp with a stagewise rotation protocol*


---

**Description**

The Classical Jacobi Algorithm with a stagewise protocol

**Usage**

```
JacobiS(x, symmetric = TRUE, only.values = FALSE, eps = 0)
```

**Arguments**

<code>x</code>	A real symmetric matrix
<code>symmetric</code>	a logical value. Is the matrix symmetric? (Only symmetric matrices are allowed.)
<code>only.values</code>	A logical value: do you want only the eigenvalues?
<code>eps</code>	an error tolerance. 0.0 implies <code>.Machine\$double.eps</code> and <code>sqrt(.Machine\$double.eps)</code> if <code>only.values = TRUE</code>

**Details**

Eigenvalues and optionally, eigenvectors, of a real symmetric matrix using the classical Jacobi algorithm, (Jacobi, 1846) using a stagewise rotation protocol

**Value**

a list of two components as for `base::eigen`

**Examples**

```
V <- crossprod(matrix(runif(40, -1, 1), 8))
JacobiS(V)
all.equal(JacobiS(V)$values, Jacobi(V)$values)
zapsmall(crossprod(JacobiS(V)$vectors, Jacobi(V)$vectors))
```

# Index

Jacobi, [2](#)  
JacobiR, [3](#)  
JacobiS, [4](#)