

# Package ‘Carlson’

June 28, 2022

**Type** Package

**Title** Carlson Elliptic Integrals and Incomplete Elliptic Integrals

**Version** 1.1.0

**Date** 2022-06-28

**Author** Stéphane Laurent

**Maintainer** Stéphane Laurent <laurent\_step@outlook.fr>

**Description** Evaluation of the Carlson elliptic integrals and the incomplete elliptic integrals with complex arguments. The implementations use Carlson's algorithms <doi:10.1007/BF02198293>. Applications of elliptic integrals include probability distributions, geometry, physics, mechanics, electrodynamics, statistical mechanics, astronomy, geodesy, geodesics on conics, and magnetic field calculations.

**License** GPL-3

**URL** <https://github.com/stla/Carlson>

**BugReports** <https://github.com/stla/Carlson/issues>

**Suggests** gsl, testthat

**Encoding** UTF-8

**RoxygenNote** 7.0.2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-06-28 10:10:02 UTC

## R topics documented:

Carlson_RC . . . . .	2
Carlson_RD . . . . .	3
Carlson_RF . . . . .	3
Carlson_RG . . . . .	4
Carlson_RJ . . . . .	4

elliptic_E	5
elliptic_F	6
elliptic_PI	6
elliptic_Z	7
<b>Index</b>	<b>8</b>

---

Carlson_RC	<i>Carlson elliptic integral RC</i>
------------	-------------------------------------

---

### Description

Evaluate the Carlson elliptic integral RC.

### Usage

```
Carlson_RC(x, y, minerror = 2 * .Machine$double.eps)
```

### Arguments

x, y	real or complex numbers, with y different from 0
minerror	bound on the relative error passed to <a href="#">Carlson_RF</a>

### Value

A complex number, the value of the Carlson elliptic integral  $R_C(x, y)$ .

### Note

The function returns a value when x or y are negative real numbers, but this value is not the one of the Carlson integral.

### Examples

```
Carlson_RC(5, 2)
gsl::ellint_RC(5, 2)
```

---

Carlson_RD	<i>Carlson elliptic integral RD</i>
------------	-------------------------------------

---

**Description**

Evaluate the Carlson elliptic integral RD.

**Usage**

```
Carlson_RD(x, y, z, minerror = 2 * .Machine$double.eps)
```

**Arguments**

x, y, z	real or complex numbers; at most one can be 0
minerror	bound on the relative error

**Value**

A complex number, the value of the Carlson elliptic integral  $R_D(x, y, z)$ .

**Note**

The function returns a value when x, y or z are negative real numbers, but this value is not the one of the Carlson integral.

**Examples**

```
Carlson_RD(5, 2, 3)
gsl::ellint_RD(5, 2, 3)
```

---

Carlson_RF	<i>Carlson elliptic integral RF</i>
------------	-------------------------------------

---

**Description**

Evaluate the Carlson elliptic integral RF.

**Usage**

```
Carlson_RF(x, y, z, minerror = 2 * .Machine$double.eps)
```

**Arguments**

x, y, z	real or complex numbers; at most one can be 0
minerror	bound on relative error

**Value**

A complex number, the value of the Carlson elliptic integral  $R_F(x, y, z)$ .

**Note**

The function returns a value when  $x$ ,  $y$  or  $z$  are negative real numbers, but this value is not the one of the Carlson integral.

**Examples**

```
Carlson_RF(5, 2, 3)
gsl::ellint_RF(5, 2, 3)
```

---

Carlson_RG	<i>Carlson elliptic integral RG</i>
------------	-------------------------------------

---

**Description**

Evaluate the Carlson elliptic integral RG.

**Usage**

```
Carlson_RG(x, y, z, minerror = 2 * .Machine$double.eps)
```

**Arguments**

$x, y, z$	real or complex numbers; they can be zero
minerror	bound on the relative error passed to <a href="#">Carlson_RF</a> and <a href="#">Carlson_RD</a>

**Value**

A complex number, the value of the Carlson elliptic integral  $R_G(x, y, z)$ .

---

Carlson_RJ	<i>Carlson elliptic integral RJ</i>
------------	-------------------------------------

---

**Description**

Evaluate the Carlson elliptic integral RJ.

**Usage**

```
Carlson_RJ(x, y, z, p, minerror = 2 * .Machine$double.eps)
```

**Arguments**

`x, y, z, p`      real or complex numbers; at most one can be 0  
`minerror`      bound on the relative error

**Value**

A complex number, the value of the Carlson elliptic integral  $R_J(x, y, z, t)$ .

**Note**

The function returns a value when `x, y, z` or `p` are negative real numbers, but this value is not the one of the Carlson integral.

**Examples**

```
Carlson_RJ(5, 2, 3, 4)
gsl::ellint_RJ(5, 2, 3, 4)
```

---

elliptic\_E

*Incomplete elliptic integral of the second kind*


---

**Description**

Evaluate the incomplete elliptic integral of the second kind.

**Usage**

```
elliptic_E(phi, m, minerror = 2 * .Machine$double.eps)
```

**Arguments**

`phi`              amplitude, real or complex number  
`m`                 parameter, real or complex number  
`minerror`        the bound on the relative error passed to [Carlson\\_RF](#) and [Carlson\\_RD](#)

**Value**

A complex number, the value of the incomplete elliptic integral  $E(\phi, m)$ .

**Examples**

```
elliptic_E(1, 0.2)
gsl::ellint_E(1, sqrt(0.2))
```

---

elliptic\_F                      *Incomplete elliptic integral of the first kind*

---

**Description**

Evaluate the incomplete elliptic integral of the first kind.

**Usage**

```
elliptic_F(phi, m, minerror = 2 * .Machine$double.eps)
```

**Arguments**

phi	amplitude, real or complex number
m	parameter, real or complex number
minerror	the bound on the relative error passed to <a href="#">Carlson_RF</a>

**Value**

A complex number, the value of the incomplete elliptic integral  $F(\phi, m)$ .

**Examples**

```
elliptic_F(1, 0.2)
gsl::ellint_F(1, sqrt(0.2))
```

---

elliptic\_PI                      *Incomplete elliptic integral of the third kind*

---

**Description**

Evaluate the incomplete elliptic integral of the third kind.

**Usage**

```
elliptic_PI(phi, n, m, minerror = 2 * .Machine$double.eps)
```

**Arguments**

phi	amplitude, real or complex number
n	characteristic, real or complex number
m	parameter, real or complex number
minerror	the bound on the relative error passed to <a href="#">Carlson_RF</a> and <a href="#">Carlson_RJ</a>

**Value**

A complex number, the value of the incomplete elliptic integral  $\Pi(\phi, n, m)$ .

**Examples**

```
elliptic_PI(1, 0.8, 0.2)
gsl::ellint_P(1, sqrt(0.2), -0.8)
```

---

elliptic\_Z

*Jacobi zeta function*

---

**Description**

Evaluate the Jacobi zeta function.

**Usage**

```
elliptic_Z(phi, m, minerror = 2 * .Machine$double.eps)
```

**Arguments**

phi	amplitude, real or complex number
m	parameter, real or complex number
minerror	bound on relative error passed to <a href="#">elliptic_E</a> and <a href="#">elliptic_F</a>

**Value**

A complex number, the value of the Jacobi zeta function  $Z(\phi, m)$ .

# Index

Carlson\_RC, 2  
Carlson\_RD, 3, 4, 5  
Carlson\_RF, 2, 3, 4–6  
Carlson\_RG, 4  
Carlson\_RJ, 4, 6  
  
elliptic\_E, 5, 7  
elliptic\_F, 6, 7  
elliptic\_PI, 6  
elliptic\_Z, 7