

# The Special Functions Reference Manual

---

Special functions in Common Lisp, version 1.1

Steve Nunez <[steve@symbolics.tech](mailto:steve@symbolics.tech)>

---

# Table of Contents

<b>1</b>	<b>Systems .....</b>	<b>1</b>
1.1	special-functions .....	1
<b>2</b>	<b>Files .....</b>	<b>3</b>
2.1	Lisp .....	3
2.1.1	special-functions.asd .....	3
2.1.2	special-functions/pkgdcl.lisp .....	3
2.1.3	special-functions/utils.lisp .....	3
2.1.4	special-functions/erf.lisp .....	3
2.1.5	special-functions/gamma.lisp .....	3
2.1.6	special-functions/log-gamma.lisp .....	4
2.1.7	special-functions/factorial.lisp .....	4
<b>3</b>	<b>Packages .....</b>	<b>5</b>
3.1	special-functions .....	5
<b>4</b>	<b>Definitions .....</b>	<b>7</b>
4.1	Exported definitions .....	7
4.1.1	Functions .....	7
4.2	Internal definitions .....	8
4.2.1	Constants .....	8
4.2.2	Special variables .....	8
4.2.3	Functions .....	8
<b>Appendix A</b>	<b>Indexes .....</b>	<b>11</b>
A.1	Concepts .....	11
A.2	Functions .....	12
A.3	Variables .....	13
A.4	Data types .....	14



# 1 Systems

The main system appears first, followed by any subsystem dependency.

## 1.1 special-functions

**Author** Steve Nunez <steve@symbolics.tech>

**License** MS-PL

**Description**

Special functions in Common Lisp

**Version** 1.1

**Dependencies**

- num-utils
- float-features

**Source** [special-functions.asd], page 3, (file)

**Directory** s:/src/special-functions/

**Components**

- [pkgdcl.lisp], page 3, (file)
- [utils.lisp], page 3, (file)
- [erf.lisp], page 3, (file)
- [gamma.lisp], page 3, (file)
- [log-gamma.lisp], page 4, (file)
- [factorial.lisp], page 4, (file)



## 2 Files

Files are sorted by type and then listed depth-first from the systems components trees.

### 2.1 Lisp

#### 2.1.1 special-functions.asd

**Location** /src/special-functions/special-functions.asd

**Systems** [special-functions], page 1, (system)

#### 2.1.2 special-functions/pkgdcl.lisp

**Parent** [special-functions], page 1, (system)

**Location** pkgdcl.lisp

**Packages** [special-functions], page 5,

#### 2.1.3 special-functions/utils.lisp

**Dependency**

[pkgdcl.lisp], page 3, (file)

**Parent** [special-functions], page 1, (system)

**Location** utils.lisp

**Internal Definitions**

- [+square-root-2-pi+], page 8, (constant)
- [decode-float64], page 8, (function)
- [encode-float64], page 8, (function)
- [sin-pi], page 9, (function)

#### 2.1.4 special-functions/erf.lisp

**Dependency**

[utils.lisp], page 3, (file)

**Parent** [special-functions], page 1, (system)

**Location** erf.lisp

**Exported Definitions**

- [erf], page 7, (function)
- [erfc], page 7, (function)
- [inverse-erf], page 7, (function)
- [inverse-erfc], page 7, (function)

**Internal Definitions**

[inverse-error], page 9, (function)

#### 2.1.5 special-functions/gamma.lisp

**Dependency**

[erf.lisp], page 3, (file)

**Parent** [special-functions], page 1, (system)

**Location**    `gamma.lisp`

**Exported Definitions**

`[gamma]`, page 7, (function)

**Internal Definitions**

- `[gamma-inverse-small]`, page 8, (function)
- `[gamma-medium]`, page 8, (function)
- `[maxgamd]`, page 8, (constant)
- `[sign-gamma]`, page 9, (function)
- `[stirling]`, page 9, (function)
- `[tiny]`, page 8, (constant)

### 2.1.6 `special-functions/log-gamma.lisp`

**Dependency**

`[gamma.lisp]`, page 3, (file)

**Parent**      `[special-functions]`, page 1, (system)

**Location**    `log-gamma.lisp`

**Exported Definitions**

`[log-gamma]`, page 7, (function)

### 2.1.7 `special-functions/factorial.lisp`

**Dependency**

`[log-gamma.lisp]`, page 4, (file)

**Parent**      `[special-functions]`, page 1, (system)

**Location**    `factorial.lisp`

**Exported Definitions**

`[factorial]`, page 7, (function)

**Internal Definitions**

- `[factorial-table]`, page 8, (special variable)
- `[ramanujan]`, page 9, (function)
- `[sam-ramanujan]`, page 9, (function)

## 3 Packages

Packages are listed by definition order.

### 3.1 special-functions

**Source**      [pkgdcl.lisp], page 3, (file)

**Nicknames**

- specfun
- spfn

**Use List**

- num-utils.arithmetic
- num-utils.polynomial
- common-lisp

**Exported Definitions**

- [erf], page 7, (function)
- [erfc], page 7, (function)
- [factorial], page 7, (function)
- [gamma], page 7, (function)
- [inverse-erf], page 7, (function)
- [inverse-erfc], page 7, (function)
- [log-gamma], page 7, (function)

**Internal Definitions**

- [+square-root-2-pi+], page 8, (constant)
- [decode-float64], page 8, (function)
- [encode-float64], page 8, (function)
- [factorial-table], page 8, (special variable)
- [gamma-inverse-small], page 8, (function)
- [gamma-medium], page 8, (function)
- [inverse-error], page 9, (function)
- [maxgamd], page 8, (constant)
- [ramanujan], page 9, (function)
- [sam-ramanujan], page 9, (function)
- [sign-gamma], page 9, (function)
- [sin-pi], page 9, (function)
- [stirling], page 9, (function)
- [tiny], page 8, (constant)





## 4 Definitions

Definitions are sorted by export status, category, package, and then by lexicographic order.

### 4.1 Exported definitions

#### 4.1.1 Functions

**erf** *N* [Function]

Returns the error function of *n*

**Package** [special-functions], page 5,

**Source** [erf.lisp], page 3, (file)

**erfc** *X* [Function]

Returns the complementary error function of *x*

**Package** [special-functions], page 5,

**Source** [erf.lisp], page 3, (file)

**factorial** *X* [Function]

Return the factorial value *X*! for *X* ≤ MAX-FACTORIAL; DOUBLE-FLOAT-POSITIVE-INFINITY if *x* < 0. *X* must be an INTEGER.

**Package** [special-functions], page 5,

**Source** [factorial.lisp], page 4, (file)

**gamma** *X* [Function]

Return gamma(*x*), *x* ≤ MAXGAMD; NAN/RTE if *x* is a non-positive integer

**Package** [special-functions], page 5,

**Source** [gamma.lisp], page 3, (file)

**inverse-erf** *X* [Function]

Return the inverse function of erf: (erf (inverse-erf *x*)) = *x*, -1 < *x* < 1

**Package** [special-functions], page 5,

**Source** [erf.lisp], page 3, (file)

**inverse-erfc** *X* [Function]

Return the inverse function of erfc: (erfc (inverse-erfc *x*)) = *x*, 0 < *x* < 2

**Package** [special-functions], page 5,

**Source** [erf.lisp], page 3, (file)

**log-gamma** *N* [Function]

Return the logarithm of gamma(*x*)

**Package** [special-functions], page 5,

**Source** [log-gamma.lisp], page 4, (file)

## 4.2 Internal definitions

### 4.2.1 Constants

`+square-root-2-pi+` [Constant]

**Package** [special-functions], page 5,

**Source** [utils.lisp], page 3, (file)

`maxgamd` [Constant]

Maximum argument for gamma

**Package** [special-functions], page 5,

**Source** [gamma.lisp], page 3, (file)

`tiny` [Constant]

**Package** [special-functions], page 5,

**Source** [gamma.lisp], page 3, (file)

### 4.2.2 Special variables

`factorial-table` [Special Variable]

Table of factorials for integer values up to 100

**Package** [special-functions], page 5,

**Source** [factorial.lisp], page 4, (file)

### 4.2.3 Functions

`decode-float64` *X* [Function]

Convert the (unsigned-byte 64) bit representation into a native double-float

**Package** [special-functions], page 5,

**Source** [utils.lisp], page 3, (file)

`encode-float64` *X* [Function]

Returns the bit representation of the double-float *X* as an (unsigned-byte 64)

**Package** [special-functions], page 5,

**Source** [utils.lisp], page 3, (file)

`gamma-inverse-small` *X* [Function]

Return  $1/\text{gamma}(x)$  for  $|x| < 0.03125$

**Package** [special-functions], page 5,

**Source** [gamma.lisp], page 3, (file)

`gamma-medium` *X* [Function]

Return  $\text{gamma}(x)$ ,  $|x| \leq 13$ , *x* negative integer produces div by 0

**Package** [special-functions], page 5,

**Source** [gamma.lisp], page 3, (file)

- inverse-error** *P Q* [Function]  
Return value of inverse error function: `erf_inv(p)` if  $p \leq 0.5$ , `erfc_inv(q)` otherwise  
**Package** [special-functions], page 5,  
**Source** [erf.lisp], page 3, (file)
- ramanujan** *X* [Function]  
Ramanujan's original approximation of  $n!$   
**Package** [special-functions], page 5,  
**Source** [factorial.lisp], page 4, (file)
- sam-ramanujan** *X* [Function]  
Modification of Ramanujan's approximation of  $n!$  by Sidney A. Morris  
**Package** [special-functions], page 5,  
**Source** [factorial.lisp], page 4, (file)
- sign-gamma** *X* [Function]  
Return `sign(gamma(x))`, invalid for 0 or negative integer  
**Package** [special-functions], page 5,  
**Source** [gamma.lisp], page 3, (file)
- sin-pi** *X* [Function]  
Returns `(sin (* pi x))`  
**Package** [special-functions], page 5,  
**Source** [utils.lisp], page 3, (file)
- stirling** *X* [Function]  
Return `(gamma x)` for  $x > 13$   
**Package** [special-functions], page 5,  
**Source** [gamma.lisp], page 3, (file)



## Appendix A Indexes

### A.1 Concepts

#### F

File, Lisp, `special-functions.asd` ..... 3  
 File, Lisp, `special-functions/erf.lisp` ..... 3  
 File, Lisp, `special-functions/factorial.lisp` .... 4  
 File, Lisp, `special-functions/gamma.lisp` ..... 3  
 File, Lisp, `special-functions/log-gamma.lisp` .... 4  
 File, Lisp, `special-functions/pkgdcl.lisp` ..... 3  
 File, Lisp, `special-functions/utils.lisp` ..... 3

#### L

Lisp File, `special-functions.asd` ..... 3  
 Lisp File, `special-functions/erf.lisp` ..... 3  
 Lisp File, `special-functions/factorial.lisp` .... 4  
 Lisp File, `special-functions/gamma.lisp` ..... 3  
 Lisp File, `special-functions/log-gamma.lisp` .... 4

Lisp File, `special-functions/pkgdcl.lisp` ..... 3  
 Lisp File, `special-functions/utils.lisp` ..... 3

#### S

`special-functions.asd` ..... 3  
`special-functions/erf.lisp` ..... 3  
`special-functions/factorial.lisp` ..... 4  
`special-functions/gamma.lisp` ..... 3  
`special-functions/log-gamma.lisp` ..... 4  
`special-functions/pkgdcl.lisp` ..... 3  
`special-functions/utils.lisp` ..... 3

## A.2 Functions

### D

decode-float64..... 8

### E

encode-float64..... 8

erf..... 7

erfc..... 7

### F

factorial..... 7

Function, decode-float64..... 8

Function, encode-float64..... 8

Function, erf..... 7

Function, erfc..... 7

Function, factorial..... 7

Function, gamma..... 7

Function, gamma-inverse-small..... 8

Function, gamma-medium..... 8

Function, inverse-erf..... 7

Function, inverse-erfc..... 7

Function, inverse-error..... 9

Function, log-gamma..... 7

Function, ramanujan..... 9

Function, sam-ramanujan..... 9

Function, sign-gamma..... 9

Function, sin-pi..... 9

Function, stirling..... 9

### G

gamma..... 7

gamma-inverse-small..... 8

gamma-medium..... 8

### I

inverse-erf..... 7

inverse-erfc..... 7

inverse-error..... 9

### L

log-gamma..... 7

### R

ramanujan..... 9

### S

sam-ramanujan..... 9

sign-gamma..... 9

sin-pi..... 9

stirling..... 9

### A.3 Variables

+

+square-root-2-pi+ ..... 8

C

Constant, +square-root-2-pi+ ..... 8

Constant, maxgamd ..... 8

Constant, tiny ..... 8

F

factorial-table ..... 8

M

maxgamd ..... 8

S

Special Variable, factorial-table ..... 8

T

tiny ..... 8



## A.4 Data types

### P

Package, `special-functions` ..... 5

### S

`special-functions` ..... 1, 5  
System, `special-functions` ..... 1