The SPECIAL-FUNCTIONS Reference Manual

Mathematical Special Functions, version 1.2.0

Steve Nunez `<steve@symbolics.tech>`

This manual was generated automatically by Declt 4.0b2.

Table of Contents

Copying

1 Systems

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

1.1 `special-functions`

Special functions in Common Lisp

Long Name
Mathematical Special Functions

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

Source Control
`(GIT https://github.com/Lisp-Stat/special-functions.git)`

Bug Tracker
`https://github.com/Lisp-Stat/special-functions/issues`

License MS-PL

Long Description
Special functions written in common lisp with accuracy equal to Boost, Python and Cephes.

Version 1.2.0

Dependencies

- `num-utils` (system).
- `float-features` (system).
- `let-plus` (system).
- `cephes` (system).

Source [`special-functions.asd`], page 5.

Child Components

- [`pkgdcl.lisp`], page 5 (file).
- [`utils.lisp`], page 5 (file).
- [`erf.lisp`], page 5 (file).
- [`gamma.lisp`], page 6 (file).
- [`lanczos.lisp`], page 6 (file).
- [`log-gamma.lisp`], page 7 (file).
- [`factorial.lisp`], page 7 (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/special-functions.asd

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

ASDF Systems
[special-functions], page 3.

2.1.2 special-functions/pkgdcl.lisp

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

Packages [special-functions], page 9.

2.1.3 special-functions/utils.lisp

Dependency
[pkgdcl.lisp], page 5 (file).

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

Internals

- [+square-root-2-pi+], page 13 (constant).
- [decode-float64], page 14 (function).
- [encode-float64], page 14 (function).
- [sin-pi], page 16 (function).

2.1.4 special-functions/erf.lisp

Dependency
[utils.lisp], page 5 (file).

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

Public Interface

- [erf], page 11 (function).
- [erfc], page 11 (function).
- [inverse-erf], page 12 (function).
- [inverse-erfc], page 12 (function).

Internals

- [erfc-scaled], page 14 (function).
- [inverse-error], page 14 (function).

2.1.5 special-functions/gamma.lisp

Dependency
[erf.lisp], page 5 (file).

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

Public Interface

- [gamma], page 11 (function).
- [incomplete-gamma], page 11 (function).
- [lower-incomplete-gamma], page 12 (function).
- [regularised-gamma-prefix], page 12 (function).
- [upper-incomplete-gamma], page 12 (function).

Internals

- [gamma-aux], page 14 (function).
- [gamma-inverse-small], page 14 (function).
- [gamma-medium], page 14 (function).
- [gamma-p-derivative], page 14 (function).
- [log-maximum-double-value], page 13 (constant).
- [log-minimum-double-value], page 13 (constant).
- [maxgamd], page 13 (constant).
- [p-taylor], page 15 (function).
- [pq-asymptotic], page 15 (function).
- [q-fraction], page 15 (function).
- [q-gamma-half], page 15 (function).
- [q-gamma-integer], page 15 (function).
- [q-taylor], page 15 (function).
- [regularised-gamma-prefix*], page 15 (function).
- [regularised-gamma-prefix-], page 16 (function).
- [sign-gamma], page 16 (function).
- [stirling], page 16 (function).

2.1.6 special-functions/lanczos.lisp

Dependency
[gamma.lisp], page 6 (file).

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

Packages [lanczos], page 10.

Public Interface

- [g], page 11 (special variable).
- [g-1/2], page 11 (special variable).
- [lanczos-sum], page 12 (function).

- [n], page 11 (special variable).

Internals

- [boost-denominator], page 13 (special variable).
- [boost-numerator], page 13 (special variable).
- [boost-numerator-scaled], page 13 (special variable).
- [c-to-rat], page 14 (function).
- [floatify-coefficients], page 14 (function).
- [lanczos-13-denominator], page 13 (special variable).
- [lanczos-13-numerator], page 13 (special variable).
- [lanczos-13-numerator-scaled], page 13 (special variable).
- [rationalize-coefficients], page 15 (function).

2.1.7 special-functions/log-gamma.lisp

Dependency
[lanczos.lisp], page 6 (file).

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

Public Interface
[log-gamma], page 12 (function).

2.1.8 special-functions/factorial.lisp

Dependency
[log-gamma.lisp], page 7 (file).

Source [special-functions.asd], page 5.

Parent Component
[special-functions], page 3 (system).

Public Interface
[factorial], page 11 (function).

Internals

- [factorial-table], page 13 (special variable).
- [ramanujan], page 15 (function).
- [sam-ramanujan], page 16 (function).

3 Packages

Packages are listed by definition order.

3.1 special-functions

Source [pkgdcl.lisp], page 5.

Nicknames

- specfun
- spfn

Use List

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

Used By List

- distributions.
- special-functions-tests.

Public Interface

- [erf], page 11 (function).
- [erfc], page 11 (function).
- [factorial], page 11 (function).
- [gamma], page 11 (function).
- [incomplete-gamma], page 11 (function).
- [inverse-erf], page 12 (function).
- [inverse-erfc], page 12 (function).
- [log-gamma], page 12 (function).
- [lower-incomplete-gamma], page 12 (function).
- [regularised-gamma-prefix], page 12 (function).
- [upper-incomplete-gamma], page 12 (function).

Internals

- [+square-root-2-pi+], page 13 (constant).
- [decode-float64], page 14 (function).
- [encode-float64], page 14 (function).
- [erfc-scaled], page 14 (function).
- [factorial-table], page 13 (special variable).
- [gamma-aux], page 14 (function).
- [gamma-inverse-small], page 14 (function).
- [gamma-medium], page 14 (function).
- [gamma-p-derivative], page 14 (function).
- [inverse-error], page 14 (function).
- [log-maximum-double-value], page 13 (constant).
- [log-minimum-double-value], page 13 (constant).

- [`maxgamd`], page 13 (constant).
- [`p-taylor`], page 15 (function).
- [`pq-asymptotic`], page 15 (function).
- [`q-fraction`], page 15 (function).
- [`q-gamma-half`], page 15 (function).
- [`q-gamma-integer`], page 15 (function).
- [`q-taylor`], page 15 (function).
- [`ramanujan`], page 15 (function).
- [`regularised-gamma-prefix*`], page 15 (function).
- [`regularised-gamma-prefix-`], page 16 (function).
- [`sam-ramanujan`], page 16 (function).
- [`sign-gamma`], page 16 (function).
- [`sin-pi`], page 16 (function).
- [`stirling`], page 16 (function).

3.2 `lanczos`

Source [`lanczos.lisp`], page 6.

Use List `common-lisp`.

Public Interface

- [`g`], page 11 (special variable).
- [`g-1/2`], page 11 (special variable).
- [`lanczos-sum`], page 12 (function).
- [`n`], page 11 (special variable).

Internals

- [`boost-denominator`], page 13 (special variable).
- [`boost-numerator`], page 13 (special variable).
- [`boost-numerator-scaled`], page 13 (special variable).
- [`c-to-rat`], page 14 (function).
- [`floatify-coefficients`], page 14 (function).
- [`lanczos-13-denominator`], page 13 (special variable).
- [`lanczos-13-numerator`], page 13 (special variable).
- [`lanczos-13-numerator-scaled`], page 13 (special variable).
- [`rationalize-coefficients`], page 15 (function).

4 Definitions

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

4.1 Public Interface

4.1.1 Special variables

`g` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`g-1/2` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`n` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

4.1.2 Ordinary functions

`erf` (*n*) [Function]

Returns the error function of n

Package [`special-functions`], page 9.

Source [`erf.lisp`], page 5.

`erfc` (*x*) [Function]

Return the complementary error function erfc(x) = 1-erf(x)

Package [`special-functions`], page 9.

Source [`erf.lisp`], page 5.

`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 9.

Source [`factorial.lisp`], page 7.

`gamma` (*x*) [Function]

Return gamma(x), x <= +MAXGAMD+; NAN/RTE if x is a non-positive integer

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`incomplete-gamma` (*a x* **&key** *compute-prefix*) [Function]

Return the normalised incomplete gamma functions P and Q, a>=0, x>=0 P(a,x) = integral(exp(-t)*t^(a-1), t=0..x)/gamma(a)
Q(a,x) = integral(exp(-t)*t^(a-1), t=x..Inf)/gamma(a))
dax = x^a*exp(-x)/gamma(a) (prefix factor)

Returns three values:
P is the first value, Q the second, DAX the third, e.g. (values p q dax)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`inverse-erf` (*x*) [Function]

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

Package [`special-functions`], page 9.

Source [`erf.lisp`], page 5.

`inverse-erfc` (*x*) [Function]

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

Package [`special-functions`], page 9.

Source [`erf.lisp`], page 5.

`lanczos-sum` (*x* **&key** *unscaled*) [Function]

Return the Lanczos sum for x, exp(g). If UNSCALED is non-nil, return the unscaled result

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`log-gamma` (*n*) [Function]

Return the logarithm of gamma(x)

Package [`special-functions`], page 9.

Source [`log-gamma.lisp`], page 7.

`lower-incomplete-gamma` (*x a*) [Function]

Return the normalised lower incomplete gamma function P(a,x), a>=0, x>=0 P(a,x) = integral(exp(-t)*t^(a-1), t=0..x)/gamma(a)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`regularised-gamma-prefix` (*a x*) [Function]

Return x^a * exp(-x) / gamma(a)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`upper-incomplete-gamma` (*x a*) [Function]

Return the normalised upper incomplete gamma function Q(a,x), a>=0, x>=0 Q(a,x) = integral(exp(-t)*t^(a-1), t=x..Inf)/gamma(a))

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

4.2 Internals

4.2.1 Constants

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

Package [`special-functions`], page 9.

Source [`utils.lisp`], page 5.

`log-maximum-double-value` [Constant]

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`log-minimum-double-value` [Constant]

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`maxgamd` [Constant]

Maximum argument for gamma

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

4.2.2 Special variables

`boost-denominator` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`boost-numerator` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`boost-numerator-scaled` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`factorial-table` [Special Variable]

Table of factorials for integer values up to 100

Package [`special-functions`], page 9.

Source [`factorial.lisp`], page 7.

`lanczos-13-denominator` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`lanczos-13-numerator` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`lanczos-13-numerator-scaled` [Special Variable]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

4.2.3 Ordinary functions

`c-to-rat` (*int frac*) [Function]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`decode-float64` (*x*) [Function]

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

Package [`special-functions`], page 9.

Source [`utils.lisp`], page 5.

`encode-float64` (*x*) [Function]

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

Package [`special-functions`], page 9.

Source [`utils.lisp`], page 5.

`erfc-scaled` (*x*) [Function]

p/q := exp(x^2)*erfc(x), 1<=x<=128

Package [`special-functions`], page 9.

Source [`erf.lisp`], page 5.

`floatify-coefficients` (*coeff*) [Function]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`gamma-aux` (*x*) [Function]

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`gamma-inverse-small` (*x*) [Function]

Return 1/gamma(x) for |x| < 0.03125

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`gamma-medium` (*x*) [Function]

Return gamma(x), |x| <= 13, x negative integer produces div by 0

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`gamma-p-derivative` (*a x*) [Function]

Partial derivative with respect to x of the incomplete gamma function

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`inverse-error` (*p q*) [Function]

Return value of inverse error function: erf_inv(p) if p <= 0.5, erfc_inv(q) otherwise

Package [`special-functions`], page 9.

Source [`erf.lisp`], page 5.

`p-taylor` (*a x dax*) [Function]

Temme/Gautschi code for P(a,x), dax = x^a*exp(-x)/gamma(a+1) Returns (values p q)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`pq-asymptotic` (*a x*) [Function]

Incomplete gamma functions for large A and A near X

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`q-fraction` (*a x dax*) [Function]

Continued fraction for Q(a,x)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`q-gamma-half` (*a x*) [Function]

Calculates normalised Q when a is a half-integer for a < min(30, x+1)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`q-gamma-integer` (*a x*) [Function]

Return Q(a,x) when A is an integer, A < min(30,x+1)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`q-taylor` (*a x*) [Function]

Temme/Gautschi code for Q(a,x) when x < 1

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`ramanujan` (*x*) [Function]

Ramanujan's original approximation of n!

Package [`special-functions`], page 9.

Source [`factorial.lisp`], page 7.

`rationalize-coefficients` (*coeff*) [Function]

Package [`lanczos`], page 10.

Source [`lanczos.lisp`], page 6.

`regularised-gamma-prefix*` (*a x*) [Function]

Return x^a * exp(-x) / gamma(a)

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`regularised-gamma-prefix-` (*a z*) [Function]

Return (z^a)(e^-z)/gamma(a), the power term prefix, using Lanczos summation Most of the error occurs in this function

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`sam-ramanujan` (*x*) [Function]

Modification of Ramanujan's approximation of n! by Sidney A. Morris

Package [`special-functions`], page 9.

Source [`factorial.lisp`], page 7.

`sign-gamma` (*x*) [Function]

Return sign(gamma(x)), invalid for 0 or negative integer

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

`sin-pi` (*x*) [Function]

Returns (sin (* pi x))

Package [`special-functions`], page 9.

Source [`utils.lisp`], page 5.

`stirling` (*x*) [Function]

Return gamma(x) for x > 13

Package [`special-functions`], page 9.

Source [`gamma.lisp`], page 6.

Appendix A Indexes

A.1 Concepts

(Index is nonexistent)

A.2 Functions

A.3 Variables

A.4 Data types