

Package ‘logStirling2’

September 7, 2026

Title Fast Stirling Numbers of the Second Kind

Version 0.3.0

Date 2026-09-07

Description Provides efficient tools for calculating Stirling numbers of the second kind and their logarithms. Includes an exact arbitrary-precision implementation using 'gmp' that avoids numerical cancellation, a fast C++ backend with internal caching for log-scale calculations, and Temme's asymptotic approximation for very large inputs.

License GPL (>= 3)

URL <https://github.com/jblood94/logStirling2>

BugReports <https://github.com/jblood94/logStirling2/issues>

Encoding UTF-8

Depends R (>= 4.1.0)

Imports gmp, Rcpp

LinkingTo Rcpp

Suggests testthat (>= 3.0.0),

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Jon Blood [aut, cre]

Maintainer Jon Blood <jblood94@gmail.com>

Repository CRAN

Date/Publication 2026-09-07 17:00:02 UTC

Contents

Eulerian2All	2
get_state_data	2
logStirling2	3
Stirling2All	4
stirling2direct	5

Eulerian2All	<i>Second-Order Eulerian Numbers</i>
--------------	--------------------------------------

Description

Calculates the exact values of the second-order Eulerian numbers $\langle\langle n, k \rangle\rangle$ for a fixed row n and all $k \in \{0, \dots, n - 1\}$.

Usage

Eulerian2All(n)

Arguments

n A positive integer specifying the row index.

Value

A bigz vector of length n containing the sequence $\langle\langle n, 0 \rangle\rangle, \langle\langle n, 1 \rangle\rangle, \dots, \langle\langle n, n - 1 \rangle\rangle$.

get_state_data	<i>Download and Cache Stirling State Data</i>
----------------	---

Description

Downloads the pre-computed long-double state blocks from GitHub and saves them to the user's local data directory. Once downloaded, logStirling2 will automatically detect and use these states for accelerated calculations.

Usage

get_state_data(force = FALSE)

Arguments

force Logical; if TRUE, re-downloads the data even if it already exists locally.

Value

Invisible TRUE on success.

logStirling2

*Logarithms of Stirling Numbers of the Second Kind***Description**

Calculates the natural logarithm of Stirling numbers of the second kind, $S(n, k)$, which represent the number of ways to partition a set of n elements into k non-empty subsets.

Usage

```
logStirling2(n, k = NULL, as.matrix = TRUE, ones = TRUE)
```

```
logStirling2Temme(n, k = NULL, as.matrix = TRUE, ones = TRUE, twoterms = TRUE)
```

Arguments

n	Integer vector of set sizes. Coerced to natural numbers (floor).
k	Integer vector of subset sizes. Coerced to natural numbers (floor). If NULL, returns all available k for each n .
as.matrix	Logical; if TRUE, returns a matrix where rows correspond to n and columns to k . If FALSE, returns a flat vector.
ones	Logical; if FALSE, excludes the trivial cases where $k = 1$ and $k = n$ (where $S(n, k) = 1$). This is automatically set to TRUE if as.matrix is TRUE, k is explicitly provided, or if any($n < 3$) is TRUE.
twoterms	Logical; if TRUE, uses Temme's two-term approximation. If FALSE, uses the one-term approximation.

Details

The function dispatches to one of three C++ routines (Row_C, All_C, or Mult_C) depending on the sparsity of the input vector n .

For systems supporting 16-byte long double precision, if $n \geq 1000$, the function automatically searches for pre-computed state blocks. If found in the user's data directory (tools::R_user_dir), these blocks are used to dramatically accelerate calculations. If missing, the full table is computed on-the-fly. If unsupported (e.g., Apple Silicon/ARM64), the full table is computed using standard double precision.

logStirling2Temme provides a high-speed asymptotic approximation based on Temme's method, which is functionally identical in interface but trades exactness for performance at very large n .

Value

A numeric matrix or vector containing $\ln(S(n, k))$. For $k > n$, values are returned as NA_real_.

References

Temme, N. M. (1993). Asymptotic estimates of Stirling numbers. *Studies in Applied Mathematics*, 89(3), 233-243.

Examples

```
# 1. Matrix output for specified n and k
logStirling2(n = 5:8, k = 2:5, as.matrix = TRUE)

# 2. Vector output with 'ones' filtered
# This returns only the "non-trivial" values (1 < k < n)
logStirling2(n = 8:10, k = NULL, as.matrix = FALSE, ones = FALSE)

# 3. Full row with large n
s <- logStirling2(n = 1e3, as.matrix = FALSE)
length(s)
s[10:13]

# 4. Temme's asymptotic approximation – fast even for very large n
s <- logStirling2Temme(n = 1e5, as.matrix = FALSE)
s[1000:1003]
```

Stirling2All

Stirling Numbers of the Second Kind

Description

Calculates the exact values of the Stirling numbers of the second kind $S(n, k)$ for a fixed n and all $k \in \{1, \dots, n\}$.

Usage

```
Stirling2All(n)
```

Arguments

n A positive integer specifying the set size.

Value

A bigz vector of length n containing the Stirling numbers of the second kind $S(n, 1), S(n, 2), \dots, S(n, n)$.

stirling2direct	<i>Stirling Numbers of the Second Kind (Exact)</i>
-----------------	--

Description

Calculates the exact value of $S(n, k)$ using bigz integers.

Usage

```
stirling2direct(n, k)
```

Arguments

n	Positive integer set size.
k	Integer subset size in 1:n.

Details

Implements the explicit formula for positive arguments:

$$\begin{aligned}
 S(n, k) &= \frac{1}{k!} \sum_{j=1}^k (-1)^{k-j} \binom{k}{j} j^n \\
 &= \frac{1}{k!} \sum_{j=1}^k \binom{-(j+1)}{k-j} j^n
 \end{aligned}$$

This is a "direct" calculation similar to `gmp::Stirling2(method = "direct")`, but without cancellation errors for "large" n .

For extreme cases where $k < 4$ or $n - k < 5$, the function utilizes fast closed-form polynomial expressions. Additionally, for large k where the direct summation is slow, the function dynamically branches to a high-performance C++ algorithm utilizing second-order Eulerian numbers.

Value

A bigz object.

See Also

[logStirling2](#) for log-scale calculations accepting vectors for n and k .

Examples

```
# Basic usage
stirling2direct(5, 3)

# Comparison with the log version
mapply(\(k) log(stirling2direct(200, k)), 10:20)
logStirling2(200, 10:20)
```

Index

Eulerian2All, [2](#)

get_state_data, [2](#)

logStirling2, [3](#), [5](#)

logStirling2Temme (logStirling2), [3](#)

Stirling2All, [4](#)

stirling2direct, [5](#)