

Package ‘mrtsSphere’

April 29, 2026

Type Package

Title Multi-Resolution Thin-Plate Splines on the Sphere

Version 0.1.2

Description Constructs multi-resolution thin-plate spline basis functions on the sphere for use in spatial regression and large-scale spatial prediction problems. Implements the basis system described in Huang, Huang, and Ing (2025) ‘Multi-Resolution Spatial Methods on the Sphere: Efficient Prediction for Global Data’, *Environmetrics*, <[doi:10.1002/env.70092](https://doi.org/10.1002/env.70092)>. Heavy computations are written in ‘C++’ via ‘Rcpp’ with optional ‘OpenMP’ parallelism.

License GPL (>= 2)

Encoding UTF-8

Depends R (>= 4.0)

Imports Rcpp, RSpecra

LinkingTo Rcpp, RcppEigen, RcppNumerical

Suggests fields, testthat (>= 3.0.0)

SystemRequirements GNU make

RoxygenNote 7.3.2

URL <https://github.com/STLABTW/multi-resolution-sphere>

BugReports <https://github.com/STLABTW/multi-resolution-sphere/issues>

Config/testthat/edition 3

NeedsCompilation yes

Author Hao-Yun Huang [aut, cre] (ORCID: <<https://orcid.org/0009-0004-8384-3261>>),
Hsin-Cheng Huang [aut] (ORCID: <<https://orcid.org/0000-0002-5613-349X>>),
Ching-Kang Ing [aut] (ORCID: <<https://orcid.org/0000-0003-1362-8246>>)

Maintainer Hao-Yun Huang <hhuscout@gmail.com>

Repository CRAN

Date/Publication 2026-04-29 18:40:07 UTC

Contents

mrts_sphere	2
Index	4

mrts_sphere	<i>Multi-resolution thin-plate spline basis on the sphere</i>
-------------	---

Description

Builds a set of k multi-resolution thin-plate spline (MRTS) basis functions on the sphere from a set of knot locations, and evaluates them at the prediction locations X .

Usage

```
mrts_sphere(knot, k, X)
```

Arguments

knot	Numeric matrix with two columns giving knot locations as (latitude, longitude) in degrees.
k	Integer. Number of basis functions to construct (the rank of the basis). Must satisfy $2 \leq k \leq \text{nrow}(\text{knot})$.
X	Numeric matrix with two columns giving prediction locations as (latitude, longitude) in degrees, where the basis is evaluated.

Details

The first basis function is constant ($\sqrt{1/n}$); the remaining $k - 1$ basis functions are obtained from the eigen-decomposition of the centered knot kernel matrix, following the construction described in the reference.

Value

A list with one element:

`mrts` An $\text{nrow}(X) \times k$ numeric matrix whose columns are the basis functions evaluated at the rows of X .

References

Multi-resolution approximations of Gaussian processes for large spatial datasets on the sphere. *Environmetrics*, 2025. doi:10.1002/env.70092

Examples

```

## Build a small global grid in (lat, lon) degrees.
n_lon <- 12
n_lat <- 8
lon_seq <- seq(-180, 150, length.out = n_lon)
lat_seq <- seq(-80, 80, length.out = n_lat)
grid <- as.matrix(expand.grid(lat = lat_seq, lon = lon_seq))

## Pick 30 knots and evaluate the MRTS basis at every grid point.
set.seed(1)
knots <- grid[sample(nrow(grid), 30), ]
res <- mrts_sphere(knots, k = 5, X = grid)
dim(res$mrts) # nrow(grid) x k

## Recovering a simulated spherical exponential field with the basis.
if (requireNamespace("fields", quietly = TRUE)) {
  # Great-circle distance (km) -> exponential covariance.
  d_grid <- fields::rdist.earth(grid[, 2:1], miles = FALSE)
  cov_field <- exp(-d_grid / 2000)
  y_true <- as.numeric(t(chol(cov_field + diag(1e-8, nrow(grid)))) %*%
    rnorm(nrow(grid)))

  # Noisy observations at the knot locations.
  obs_idx <- match(data.frame(t(knots)), data.frame(t(grid)))
  z_obs <- y_true[obs_idx] + rnorm(nrow(knots), sd = 0.3)

  # Project into the MRTS basis (least squares) and predict on the grid.
  B_obs <- res$mrts[obs_idx, , drop = FALSE]
  beta_hat <- qr.solve(B_obs, z_obs)
  y_hat <- res$mrts %*% beta_hat
  sqrt(mean((y_hat - y_true)^2)) # RMSE
}

```

Index

mrts_sphere, [2](#)