

# Package: manifold (via r-universe)

May 22, 2026

**Title** Operations for Riemannian Manifolds

**Version** 0.1.2

**Description** Implements operations for Riemannian manifolds, e.g., geodesic distance, Riemannian metric, exponential and logarithm maps, etc. Also incorporates random object generator on the manifolds. See Dai, Lin, and Müller (2021)  [<doi:10.1111/biom.13385>](https://doi.org/10.1111/biom.13385).

**Depends** R (>= 3.2.1)

**Imports** Rcpp, Matrix

**LinkingTo** Rcpp, RcppArmadillo

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 2

**NeedsCompilation** yes

**Author** Xiongtao Dai [aut, cre, cph], Zhenhua Lin [aut]

**Maintainer** Xiongtao Dai <xiongtao.dai@hotmail.com>

**Repository** <https://crossd.r-universe.dev>

**Date/Publication** 2025-12-01 16:40:25 UTC

**RemoteUrl** <https://github.com/cran/manifold>

**RemoteRef** HEAD

**RemoteSha** 22e18188f4c2daf03d5796ceab3a9404d9b67241

## Contents

axisAngleRep . . . . .	2
basisTan.Euclidean . . . . .	3
coordToTanV . . . . .	4
createM . . . . .	5

dimensions . . . . .	5
distance.Euclidean . . . . .	6
ExpM . . . . .	7
frechetMean . . . . .	8
frechetMedian . . . . .	9
geodesicCurve . . . . .	10
GetSettingName . . . . .	11
is.finiteDim . . . . .	11
listAvailMfd . . . . .	12
LogM . . . . .	12
LogMSPD . . . . .	13
MakeSym . . . . .	13
manifold . . . . .	14
metric.Euclidean . . . . .	14
norm.Euclidean . . . . .	15
Normalize . . . . .	17
origin.Euclidean . . . . .	17
project.Euclidean . . . . .	19
projectTangent.Euclidean . . . . .	20
rieExp.Euclidean . . . . .	21
rieLog.Euclidean . . . . .	23
rmfd . . . . .	24
runifSphere . . . . .	25
tanVToCoord . . . . .	26
<b>Index</b>	<b>27</b>

---

axisAngleRep	Returns the angle representation of $SO(3)$ matrices c.f. <a href="https://en.wikipedia.org/wiki/Axis-angle_representation">https://en.wikipedia.org/wiki/Axis-angle_representation</a>
--------------	---

---

## Description

Returns the angle representation of  $SO(3)$  matrices c.f. [https://en.wikipedia.org/wiki/Axis-angle\\_representation](https://en.wikipedia.org/wiki/Axis-angle_representation)

## Usage

```
axisAngleRep(mfd, X)
```

## Arguments

mfd	A manifold object created by <a href="#">createM</a>
X	A matrix holding a vectorized $SO(3)$ matrix in each column

**Value**

A matrix with 4 rows and the same number of columns as X. The first row contains the angles in rads (theta), and the last three rows correspond to the the axes with respect to which the rotations are performed.

---

basisTan.Euclidean	<i>Obtain an orthonormal basis on the tangent space</i>
--------------------	---

---

**Description**

Parametrize the tangent space at location p, so that the parameterized version contains an open neighborhood around the origin. (The dimension of v is potentially reduced).

**Usage**

```
## S3 method for class 'Euclidean'
basisTan(mfd, p)

basisTan(mfd, p)

## S3 method for class 'SO'
basisTan(mfd, p)

## S3 method for class 'SPD'
basisTan(mfd, p)

## S3 method for class 'Sphere'
basisTan(mfd, p)

## S3 method for class 'FlatTorus'
basisTan(mfd, p)
```

**Arguments**

mfd	A manifold object created by <a href="#">createM</a>
p	A vector for a base point on the manifold

**Value**

An orthonormal basis matrix D, whose columns contain the basis vectors, so that  $D^T v$  give the coordinates  $v_0$  for a tangent vector  $v$ , and  $D$

**Methods (by class)**

- `basisTan(Euclidean)`: An identity matrix
- `basisTan(SO)`: An identity matrix
- `basisTan(SPD)`: The basis is obtained from enumerating the (non-strict) lower-triangle of a square matrix. If  $i \neq j$ , the  $(i, j)$ th entry is mapped into a matrix with  $1/\sqrt{2}$  in the  $(i,j)$  and  $(j,i)$  entries and 0 in other entries; if  $(i == j)$ , it is mapped to a matrix with 1 in the  $i$ th diagonal element and 0 otherwise. The mapped matrix is then vectorized to obtain the basis vector.
- `basisTan(Sphere)`: The basis at the north pole is  $[0, \dots, 1, \dots, 0]$  where the 1 is at the  $j = 2, \dots, d$  location. The basis at a point  $p$  is obtained through rotating the basis from the north pole to  $p$  along the shortest geodesic.
- `basisTan(FlatTorus)`: An identity matrix

---

 coordToTanV

*Transform the coordinates for the tangent space into a tangent vector*


---

**Description**

Transform the coordinates w.r.t. a basis returned by `[basisTan()]` to tangent vectors

**Usage**

```
coordToTanV(mfd, p, V0, ...)
```

```
## Default S3 method:
coordToTanV(mfd, p, V0, ...)
```

**Arguments**

<code>mfd</code>	A manifold object created by <code>createM</code>
<code>p</code>	A vector for the base point
<code>V0</code>	A matrix or a vector containing the coordinates
<code>...</code>	Passed into specific methods

**Value**

A matrix in which each column correspond to a different tangent vector

**Methods (by class)**

- `coordToTanV(default)`: Method

---

createM	<i>Create an object to symbolize the manifold.</i>
---------	--

---

### Description

Create an object to symbolize the manifold. Supports the Euclidean, the unit sphere, special orthogonal group, and symmetric positive definite (SPD) matrices with either the affine invariant or the log-Euclidean metric.

### Usage

```
createM(mfdName)
```

### Arguments

mfdName	One of 'Euclidean', 'Sphere', 'SO', 'AffInv', or 'LogEu'. The name is case sensitive.
---------	---

### Details

Overall, most of the arguments in the manifold functions respect recycling rules. If the input arguments are matrices with  $n$  rows corresponding to  $n$  points, then the manifold function will be applied to each of the  $n$  pairs; the returned value will be a matrix. If one input corresponds to  $n$  points and the other just one point, then the one point will be recycled. If each component corresponds to one point, the returned value will correspond to just one point.

### Value

A structure 1 with class being 'mfdName'

---

dimensions	<i>Dimensions in this package</i>
------------	-----------------------------------

---

### Description

'calcGeomPar' calculates geometric parameter. 'calcIntDim' calculates the intrinsic dimensions. 'calcAmbDim' calculates the ambient dimensions. 'calcTanDim' calculates the number of tuples used to represent a tangent vector.

### Usage

```
calcGeomPar(mfd, dimAmbient, dimIntrinsic, dimTangent)
```

```
calcIntDim(mfd, geomPar, dimAmbient, dimTangent)
```

```
calcAmbDim(mfd, geomPar, dimIntrinsic, dimTangent)
```

```
calcTanDim(mfd, geomPar, dimAmbient, dimIntrinsic)
```

**Arguments**

mfd	A manifold object created by <a href="#">createM</a>
dimAmbient	Dimension of the ambient space.
dimIntrinsic	Intrinsic dimension of the manifold.
dimTangent	The length of a tangent vector
geomPar	The geometric parameter (e.g., SO(*))

**Value**

A scalar value for the geometric parameter

A scalar value for the intrinsic dimension

A scalar value for the ambient dimension

A scalar value for the number of components in the (implementation of the) tangent vector

---

distance.Euclidean     *Geodesic distance*

---

**Description**

Geodesic distance

**Usage**

```
## S3 method for class 'Euclidean'
distance(mfd, X, Y, ...)

distance(mfd, X, Y, ...)

## S3 method for class 'L2'
distance(mfd, X, Y, ...)

## S3 method for class 'HS'
distance(mfd, X, Y, ...)

## S3 method for class 'Dens'
distance(mfd, X, Y, ...)

## S3 method for class 'SO'
distance(mfd, X, Y, ...)

## S3 method for class 'LogEu'
distance(mfd, X, Y, assumeLogRep = FALSE, ...)

## S3 method for class 'AffInv'
distance(mfd, X, Y, ...)
```

```
## S3 method for class 'Sphere'
distance(mfd, X, Y, ...)
```

### Arguments

mfd	A manifold object created by <a href="#">createM</a>
X, Y	Matrices with n columns. The distance between each pair of columns is calculated. If either X or Y is a vector then it is recycled.
...	Passed into specific methods
assumeLogRep	Whether to assume the input are already the representations under the logarithm map

### Value

A vector with n entries containing the distances between pairs of points

### Methods (by class)

- distance(Euclidean): Method
- distance(L2): Method
- distance(HS): Method
- distance(Dens): Method
- distance(SO): Method
- distance(LogEu): Method
- distance(AffInv): Method
- distance(Sphere): Method

---

ExpM

*Matrix exponential*

---

### Description

ExpM(X) computes the matrix exponential using Armadillo.

### Usage

```
ExpM(X)
```

### Arguments

X	A square matrix
---	-----------------

### Value

A matrix having the same size as the input

---

 frechetMean

*Calculate the Fréchet mean*


---

### Description

Calculate the Fréchet mean

### Usage

```
frechetMean(mfd, X, ...)
```

```
## Default S3 method:
```

```
frechetMean(
  mfd,
  X,
  mu0,
  weight = NULL,
  tol = 1e-09,
  maxit = 1000,
  lam = 0.99,
  ...
)
```

### Arguments

<code>mfd</code>	A manifold object created by <a href="#">createM</a>
<code>X</code>	A matrix with each column being a point on the manifold.
<code>...</code>	Passed into specific methods
<code>mu0</code>	A matrix of starting points. Each column corresponds to a starting point. If there are multiple columns, then a multistart algorithm is used.
<code>weight</code>	A vector of weights for the observations
<code>tol</code>	The threshold for determining convergence
<code>maxit</code>	Maximum iteration
<code>lam</code>	The step length

### Value

A matrix with 1 column containing the Fréchet mean

### Methods (by class)

- `frechetMean(default)`: Default method

---

frechetMedian	<i>Calculate the Fréchet median</i>
---------------	-------------------------------------

---

**Description**

Calculate the Fréchet median

**Usage**

```
frechetMedian(mfd, X, ...)
```

```
## Default S3 method:
```

```
frechetMedian(  
  mfd,  
  X,  
  mu0,  
  weight = NULL,  
  tol = 1e-09,  
  maxit = 1000,  
  alpha = 1,  
  ...  
)
```

**Arguments**

mfd	A manifold object created by <a href="#">createM</a>
X	A matrix with each column being a point on the manifold.
...	Passed into specific methods
mu0	A matrix of starting points. Each column corresponds to a starting point. If there are multiple columns, then a multistart algorithm is used.
weight	A vector of weights for the observations
tol	The threshold for determining convergence
maxit	Maximum iteration
alpha	The step size parameter. See Fletcher et al. (2008) CVPR

**Value**

A matrix with 1 column containing the Fréchet median

**Methods (by class)**

- `frechetMedian(default)`: Default method

---

geodesicCurve      *Obtain the geodesic curve*

---

### Description

Obtain the geodesic curve

### Usage

```
geodesicCurve(mfd, p, h, t)

## S3 method for class 'HS'
geodesicCurve(mfd, p, h, t)

## S3 method for class 'Sphere'
geodesicCurve(mfd, p, h, t)
```

### Arguments

mfd	A class instance that represents the Riemannian manifold
p	The starting point of the geodesic curve
h	A matrix with each column representing a tangent vector. If there is only one tangent vector is supplied, then it is replicated to match the length of $t$
t	A vector, the time points where the curve is evaluated.

### Details

The curve is  $\gamma(t) = \text{Exp}_p(th)$

### Value

A matrix of points lying on a geodesic. Different columns correspond to different time points

A matrix with each column representing a point on the manifold

### Methods (by class)

- geodesicCurve(HS): Method
- geodesicCurve(Sphere): Geodesic curve starting at a point

---

GetSettingName	<i>Helper function for simulations</i>
----------------	--

---

**Description**

Get the name of the settings from a named list good for saving into a data frame or use as file name

**Usage**

```
GetSettingName(settings, digits = 3, display = c("short", "pretty", "tiny"))
```

**Arguments**

settings	A named list. The names corresponding to the setting parameter names, and the values are the parameter values
digits	How many digits to use to format the numerical values
display	Which version of the setting name to produce

**Value**

A character vector of setting names

---

is.finiteDim	<i>Tell whether a manifold is a finite-dimensional one</i>
--------------	--

---

**Description**

Tell whether a manifold is a finite-dimensional one

**Usage**

```
is.finiteDim(mfd)
```

**Arguments**

mfd	A manifold object created by <a href="#">createM</a>
-----	--

**Value**

A logical scalar indicating whether the manifold is finite-dimensional

---

<code>listAvailMfd</code>	<i>List all available manifold names</i>
---------------------------	--

---

**Description**

List all available manifold names

**Usage**

```
listAvailMfd()
```

**Value**

A character vector of the names of available manifolds

---

<code>LogM</code>	<i>Matrix logarithm</i>
-------------------	-------------------------

---

**Description**

`LogM(X)` computes the matrix logarithm of a general matrix using Armadillo. The returned value is a complex matrix but converted into a real one.

**Usage**

```
LogM(X)
```

**Arguments**

<code>X</code>	A square matrix
----------------	-----------------

**Value**

A matrix having the same size as the input

---

LogMSPD	<i>Matrix logarithm of a symmetric positive definite</i>
---------	--

---

**Description**

LogMSPD(X) computes the matrix logarithm of an SPD matrix using Armadillo. The returned value is a real matrix.

**Usage**

LogMSPD(X)

**Arguments**

X	An SPD matrix
---	---------------

**Value**

A symmetric matrix

---

MakeSym	<i>Make a symmetric matrix by specifying a near-symmetric matrix M, or the lower triangular elements lowerTri with diagonal.</i>
---------	--

---

**Description**

Make a symmetric matrix by specifying a near-symmetric matrix M, or the lower triangular elements lowerTri with diagonal.

**Usage**

MakeSym(M, lowerTri, doubleDiag = FALSE)

**Arguments**

M	A near-symmetric matrix
lowerTri	A vector containing the lower triangular elements of the matrix. This is an alternative way to specify the matrix.
doubleDiag	Only meaningful for lowerTri is not missing. Whether the diagonal elements should be multiplied by sqrt(2) for doubling the squared norm of the lower triangle

**Value**

A symmetric matrix

---

manifold	<i>manifold: Operations for Riemannian manifolds</i>
----------	--

---

**Description**

Implements operations for Riemannian manifolds, e.g., geodesic distance, Riemannian metric, exponential and logarithm maps, etc. Also incorporates random object generator on the manifolds

**Details**

References: Dai X, Lin Z, Müller HG. Modeling sparse longitudinal data on Riemannian manifolds. *Biometrics*. 2021;77(4):1328–41. Dai X, Lopez-Pintado S. Tukey’s depth for object data. *Journal of the American Statistical Association*. 2021;In press.

Maintainer: Xiongtao Dai <xiongtao.dai@hotmail.com>

**Author(s)**

Xiongtao Dai <xiongtao.dai@hotmail.com> Zhenhua Lin

---

metric.Euclidean	<i>Returns the Riemannian metric</i>
------------------	--------------------------------------

---

**Description**

Returns the Riemannian metric

**Usage**

```
## S3 method for class 'Euclidean'
metric(mfd, p, U, V)

metric(mfd, p, U, V)

## Default S3 method:
metric(mfd, p = NULL, U, V)

## S3 method for class 'HS'
metric(mfd, p, U, V)

## S3 method for class 'Dens'
metric(mfd, p, U, V)

## S3 method for class 'SO'
metric(mfd, p, U, V)
```

```

## S3 method for class 'LogEu'
metric(mfd, p, U, V)

## S3 method for class 'AffInv'
metric(mfd, p, U, V)

## S3 method for class 'Sphere'
metric(mfd, p, U, V)

```

### Arguments

mfd	A manifold object created by <a href="#">createM</a>
p	A vector containing the (single) base point on the manifold.
U, V	Matrices with n columns each containing a tangent vector. The metric of each pair of n columns is calculated.

### Value

A vector with n entries containing the metric values

### Methods (by class)

- `metric(Euclidean)`: Method
- `metric(default)`: Method
- `metric(HS)`: Method
- `metric(Dens)`: Method
- `metric(SO)`: Method
- `metric(LogEu)`: Method
- `metric(AffInv)`: Method
- `metric(Sphere)`: Method

---

norm.Euclidean

*Norm on the tangent space induced by the Riemannian metric*

---

### Description

Norm on the tangent space induced by the Riemannian metric

**Usage**

```

## S3 method for class 'Euclidean'
norm(mfd, p, U)

norm(mfd, p, U)

## Default S3 method:
norm(mfd, p = NULL, U)

## S3 method for class 'L2'
norm(mfd, p, U)

## S3 method for class 'HS'
norm(mfd, p, U)

## S3 method for class 'Dens'
norm(mfd, p, U)

## S3 method for class 'SO'
norm(mfd, p, U)

## S3 method for class 'LogEu'
norm(mfd, p, U)

## S3 method for class 'AffInv'
norm(mfd, p, U)

## S3 method for class 'Sphere'
norm(mfd, p, U)

```

**Arguments**

mfd	A manifold object created by <a href="#">createM</a>
p, U	Matrices with n columns. p stands for the base point(s) and U the tangent vector(s). The norm of each column in U is calculated w.r.t. the corresponding column in p.

**Value**

A vector with n entries containing the norms of the tangent vectors

**Methods (by class)**

- norm(Euclidean): Method
- norm(default): Method
- norm(L2): Method
- norm(HS): Method

- norm(Dens): Method
- norm(SO): Method
- norm(LogEu): Method
- norm(AffInv): Method
- norm(Sphere): Method

---

 Normalize

*Normalize a vector*


---

### Description

Normalize a vector

### Usage

```
Normalize(v, tol = 1e-10)
```

### Arguments

v	A vector to be normalized to have a unit norm
tol	The tolerance value for deciding the v is actually 0, cannot be normalized, and thus return just the 0 vector

### Value

A vector with the same length as 'v' but having unit Euclidean norm

---

 origin.Euclidean

*Returns the origin of the manifold*


---

### Description

Each manifold defined in this package has been designated a more or less reasonable choice of origin.

**Usage**

```
## S3 method for class 'Euclidean'
origin(mfd, dimIntrinsic, ...)

origin(mfd, dimIntrinsic, ...)

## S3 method for class 'L2'
origin(mfd, dimIntrinsic, ...)

## S3 method for class 'HS'
origin(mfd, dimIntrinsic, ...)

## S3 method for class 'Dens'
origin(mfd, dimIntrinsic, ...)

## S3 method for class 'SO'
origin(mfd, dimIntrinsic, ...)

## S3 method for class 'SPD'
origin(mfd, dimIntrinsic, ...)

## S3 method for class 'Sphere'
origin(mfd, dimIntrinsic, ...)
```

**Arguments**

<code>mfd</code>	A manifold object created by <code>createM</code>
<code>dimIntrinsic</code>	An integer for the intrinsic dimension
<code>...</code>	Passed into specific methods

**Value**

A matrix with 1 column, a designation of the origin on the manifold

**Methods (by class)**

- `origin(Euclidean)`: Method
- `origin(L2)`: Method
- `origin(HS)`: Method
- `origin(Dens)`: The uniform density
- `origin(SO)`: The origin has 1 in the first ambient coordinate and 0 otherwise.
- `origin(SPD)`: The origin is the identity matrix but vectorized.
- `origin(Sphere)`: The origin has 1 in the first ambient coordinate and 0 otherwise.

---

project.Euclidean      *Project data points in the ambient space onto the manifold*

---

### Description

Project data points in the ambient space onto the manifold

### Usage

```
## S3 method for class 'Euclidean'  
project(mfd, p)
```

```
project(mfd, p)
```

```
## S3 method for class 'L2'  
project(mfd, p)
```

```
## S3 method for class 'HS'  
project(mfd, p)
```

```
## S3 method for class 'Dens'  
project(mfd, p)
```

```
## S3 method for class 'SO'  
project(mfd, p)
```

```
## S3 method for class 'AffInv'  
project(mfd, p)
```

```
## S3 method for class 'LogEu'  
project(mfd, p)
```

```
## S3 method for class 'SPD'  
project(mfd, p)
```

```
## S3 method for class 'Sphere'  
project(mfd, p)
```

### Arguments

mfd	A manifold object created by <a href="#">createM</a>
p	A matrix of points to be projected to the manifold.

### Value

A matrix in which columns correspond to points on the manifold

**Methods (by class)**

- project(Euclidean): Method
- project(L2): Method
- project(HS): Method
- project(Dens): Method
- project(SO): Method
- project(AffInv): Method
- project(LogEu): Method
- project(SPD): Method
- project(Sphere): Method

---

projectTangent.Euclidean

*Project data points in the ambient space onto the tangent space*

---

**Description**

Project data points in the ambient space onto the tangent space

**Usage**

```
## S3 method for class 'Euclidean'
projectTangent(mfd, p, X, projMatOnly = FALSE, ...)

projectTangent(mfd, p, X, projMatOnly, ...)

## S3 method for class 'L2'
projectTangent(mfd, p, X, projMatOnly = FALSE, ...)

## S3 method for class 'HS'
projectTangent(mfd, p, X, projMatOnly = FALSE, ...)

## S3 method for class 'Dens'
projectTangent(mfd, p, X, projMatOnly, ...)

## S3 method for class 'SO'
projectTangent(mfd, p, X, projMatOnly = FALSE, ...)

## S3 method for class 'SPD'
projectTangent(mfd, p, X, projMatOnly = FALSE, ...)

## S3 method for class 'Sphere'
projectTangent(mfd, p, X, projMatOnly = FALSE, ...)
```

**Arguments**

mfd	A manifold object created by <code>createM</code>
p	A vector containing the base point on the manifold. Data X will be projected onto the tangent space of p.
X	A vector or a matrix containing the data points in terms of the coordinates in the ambient space.
projMatOnly	Whether to only return the projection matrix (FALSE) or the projected data points (TRUE, the default)
...	Passed into specific methods

**Value**

A matrix in which columns correspond to points on the tangent spaces

**Methods (by class)**

- `projectTangent(Euclidean)`: Method
- `projectTangent(L2)`: Method
- `projectTangent(HS)`: Method
- `projectTangent(Dens)`: Method
- `projectTangent(SO)`: Method
- `projectTangent(SPD)`: Method
- `projectTangent(Sphere)`: Method

---

<code>rieExp.Euclidean</code>	<i>Riemannian exponential map</i>
-------------------------------	-----------------------------------

---

**Description**

Riemannian exponential map

**Usage**

```
## S3 method for class 'Euclidean'
rieExp(mfd, p, V, ...)

rieExp(mfd, p, V, ...)

## S3 method for class 'L2'
rieExp(mfd, p, V, ...)

## S3 method for class 'HS'
rieExp(mfd, p, V, ...)
```

```

## S3 method for class 'Dens'
rieExp(mfd, p, V, ...)

## S3 method for class 'SO'
rieExp(mfd, p, V, tol = 1e-10, ...)

## S3 method for class 'LogEu'
rieExp(mfd, p, V, ...)

## S3 method for class 'AffInv'
rieExp(mfd, p, V, ...)

## S3 method for class 'Sphere'
rieExp(mfd, p, V, ...)

```

### Arguments

mfd	A manifold object created by <a href="#">createM</a>
p, V	Matrices with n columns. The exponential map of each column in V is calculated w.r.t. the corresponding column in p.
...	Passed into specific methods
tol	Tolerance for 'rieExp.SO'

### Value

A matrix in which columns correspond to points on the manifold

### Methods (by class)

- `rieExp(Euclidean)`: Method
- `rieExp(L2)`: Method
- `rieExp(HS)`: Method
- `rieExp(Dens)`: Method
- `rieExp(SO)`: Method
- `rieExp(LogEu)`: Method
- `rieExp(AffInv)`: Method
- `rieExp(Sphere)`: Method

---

rieLog.Euclidean      *Riemannian logarithm map*

---

### Description

Riemannian logarithm map

### Usage

```
## S3 method for class 'Euclidean'
rieLog(mfd, p, X, ...)

rieLog(mfd, p, X, ...)

## S3 method for class 'L2'
rieLog(mfd, p, X, ...)

## S3 method for class 'HS'
rieLog(mfd, p, X, tol = 1e-07, ...)

## S3 method for class 'Dens'
rieLog(mfd, p, X, ...)

## S3 method for class 'SO'
rieLog(mfd, p, X, tol = 1e-10, ...)

## S3 method for class 'LogEu'
rieLog(mfd, p, X, ...)

## S3 method for class 'AffInv'
rieLog(mfd, p, X, ...)

## S3 method for class 'Sphere'
rieLog(mfd, p, X, tol = 1e-10, ...)
```

### Arguments

mfd	A manifold object created by <a href="#">createM</a>
p, X	Matrices with n columns. The logarithm map of each column in X is calculated w.r.t. the corresponding column in p.
...	Passed into specific methods
tol	Tolerance for 'rieLog.SO'

### Value

A matrix in which columns correspond to points on the tangent spaces

**Methods (by class)**

- `rieLog(Euclidean)`: Method
- `rieLog(L2)`: Method
- `rieLog(HS)`: Method
- `rieLog(Dens)`: Method
- `rieLog(S0)`: Method
- `rieLog(LogEu)`: Method
- `rieLog(AffInv)`: Method
- `rieLog(Sphere)`: Method

---

 rmfd

*Generate random variables on the manifold*


---

**Description**

By default, random variables are generated by mapping isotropic Gaussian distributions on the tangent space back to the manifold using the exponential map

**Usage**

```
rmfd(mfd, n, dimIntrinsic, ...)
```

```
## Default S3 method:
```

```
rmfd(
  mfd,
  n,
  dimIntrinsic,
  p,
  dist = c("norm", "unif", "exp"),
  totalVar = 1,
  ...
)
```

**Arguments**

<code>mfd</code>	A manifold object created by <a href="#">createM</a>
<code>n</code>	Sample size
<code>dimIntrinsic</code>	The intrinsic dimension of the target manifold
<code>...</code>	Passed into specific methods
<code>p</code>	Optionally, specify the base point of the tangent space, on which random tangent vectors will be generated. Default to the origin of 'mfd'.

dist	Either a string or a function, describing the distributions. If it is character, then independent univariate r.v. following that distribution with total variance 1 is generated on the tangent space. If it is a function, it should specify the generation of the samples on the tangent space <code>_coordinates_</code> (with dimension <code>dimIntrinsic</code> ), and it must return a matrix for which the columns are the coordinates. Then it will be mapped to the tangent space at the origin and then to the manifold.
totalVar	Total variance (sum of diagonal of the covariance matrix) of the tangent random vector

**Value**

A matrix with `n` columns, each containing a random sample on the manifold

**Methods (by class)**

- `rmfd(default)`: Default method

---

runifSphere	<i>Generate uniform random variables on the unit sphere</i>
-------------	---

---

**Description**

Generate uniform random variables on the unit sphere

**Usage**

```
runifSphere(n, dimAmbient)
```

**Arguments**

<code>n</code>	Sample size
<code>dimAmbient</code>	The dimension of the ambient space

**Value**

A ‘`dimAmbient`’ by ‘`n`’ matrix. Each column is a random observation

---

tanVToCoord

*Transform tangent vectors to their coordinates on the tangent space*


---

**Description**

Transform the tangent vectors to coordinates w.r.t. the basis returned by [basisTan()]

**Usage**

```
tanVToCoord(mfd, p, V, ...)
```

```
## Default S3 method:
```

```
tanVToCoord(mfd, p, V, ...)
```

**Arguments**

mfd	A manifold object created by <a href="#">createM</a>
p	A vector for the base point
V	A matrix or a vector containing the tangent vectors
...	Passed into specific methods

**Value**

A matrix in which each column correspond to a coordinate

**Methods (by class)**

- tanVToCoord(default): Method

# Index

axisAngleRep, 2

basisTan (basisTan.Euclidean), 3  
basisTan.Euclidean, 3

calcAmbDim (dimensions), 5  
calcGeomPar (dimensions), 5  
calcIntDim (dimensions), 5  
calcTanDim (dimensions), 5  
coordToTanV, 4  
createM, 2–4, 5, 6–9, 11, 15, 16, 18, 19,  
21–24, 26

dimensions, 5  
distance (distance.Euclidean), 6  
distance.Euclidean, 6

ExpM, 7

frechetMean, 8  
frechetMedian, 9

geodesicCurve, 10  
GetSettingName, 11

is.finiteDim, 11

listAvailMfd, 12  
LogM, 12  
LogMSPD, 13

MakeSym, 13  
manifold, 14  
manifold-package (manifold), 14  
metric (metric.Euclidean), 14  
metric.Euclidean, 14

norm (norm.Euclidean), 15  
norm.Euclidean, 15  
Normalize, 17

origin (origin.Euclidean), 17  
origin.Euclidean, 17

project (project.Euclidean), 19  
project.Euclidean, 19  
projectTangent  
    (projectTangent.Euclidean), 20  
projectTangent.Euclidean, 20

rieExp (rieExp.Euclidean), 21  
rieExp.Euclidean, 21  
rieLog (rieLog.Euclidean), 23  
rieLog.Euclidean, 23  
rmfd, 24  
runifSphere, 25

tanVToCoord, 26