Package 'tdaunif'

October 14, 2022

Title Uniform Manifold Samplers for Topological Data Analysis

Version 0.1.0

Description Uniform random samples from simple manifolds, sometimes with noise, are commonly used to test topological data analytic (TDA) tools. This package includes samplers powered by two techniques: analytic volume-preserving parameterizations, as employed by Arvo (1995) <doi:10.1145/218380.218500>, and rejection sampling, as employed by Diaconis, Holmes, and Shahshahani (2013) <doi:10.1214/12-IMSCOLL1006>.

Depends R (>= 3.3.0)

Suggests knitr, rmarkdown, testthat, vdiffr (>= 0.2)

License GPL-3

Encoding UTF-8

URL https://corybrunson.github.io/tdaunif/

BugReports https://github.com/corybrunson/tdaunif/issues

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Author Jason Cory Brunson [aut, cre], Brandon Demkowicz [aut], Sanmati Choudhary [aut]

Maintainer Jason Cory Brunson <cornelioid@gmail.com>

Repository CRAN

Date/Publication 2020-10-26 09:20:02 UTC

R topics documented:

`arch-spirals` *Sample (with noise) from archimedean spirals and swiss rolls*

Description

These functions generate uniform samples from archimedean spirals in 2-dimensional space or from swiss rolls in 3-dimensional space, optionally with noise.

Usage

```
sample_arch_spiral(n, ar = 1, arms = 1L, min_wrap = 0, max_wrap = 1, sd = 0)

sample_swiss_roll(
  n,
  ar = 1,
  arms = 1L,
  min_wrap = 0,
  max_wrap = 1,
  width = 2 * pi,
  sd = 0
)
```

Arguments

`n`	Number of observations.
`ar`	Aspect ratio of spiral (ratio of width/height)
`arms`	Number of spiral arms.
`min_wrap`	The wrap of the spiral from which sampling begins.
`max_wrap`	The wrap of the spiral at which sampling ends.
`sd`	Standard deviation of (independent multivariate) Gaussian noise.
`width`	Width of the swiss roll (with respect to the radius of the spiral at one wrap).

Details

The archimedean spiral starts at the origin and wraps around itself such that radial distances between all the spiral branches are equal. The specific parametrization was taken from Koeller (2002). The uniform sample is generated through a rejection sampling process as described by Diaconis, Holmes, and Shahshahani (2013).

The swiss roll sampler is patterned after one in drtoolbox and extended from the archimedian spiral sampler.

References

Koeller, J. (2002). Spirals. Retrieved July 18, 2019, from http://www.mathematische-basteleien.de/spiral.htm

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

Examples

```
set.seed(77151L)

#Uniformly sampled archimedean spiral in 2-space, with 1 wrap
x <- sample_arch_spiral(360, min_wrap = 0, max_wrap = 1)
plot(x, asp = 1, pch = 19, cex = .5)

#Uniformly sampled archimedean spiral in 2-space, with 1 wrap
#and aspect ratio of 2:1
x <- sample_arch_spiral(360, ar = 2, min_wrap = 0, max_wrap = 1)
plot(x, asp = 1, pch = 19, cex = .5)

#Uniformly sampled archimedean spiral in 2-space, with 1 wrap
#and aspect ratio of 1:2
x <- sample_arch_spiral(360, ar = 0.5, min_wrap = 0, max_wrap = 1)
plot(x, asp = 1, pch = 19, cex = .5)

#Uniformly sampled archimedean spiral in 2-space, with 5 wraps
x <- sample_arch_spiral(360, min_wrap = 0, max_wrap = 5)
plot(x, asp = 1, pch = 19, cex = .5)

#Uniformly sampled archimedean spiral in 2-space, with 5 wraps and starting from
#2 wraps
x <- sample_arch_spiral(360, min_wrap = 2, max_wrap = 5)
plot(x, asp = 1, pch = 19, cex = .5)

#Uniformly sampled archimedian spiral, from 1 to 2 wraps, with 3 arms
x <- sample_arch_spiral(360, arms = 3, min_wrap = 1, max_wrap = 2)
plot(x, asp = 1, pch = 19, cex = .5)

#Uniformly sampled archimedean spiral in 2-space, with 1 wrap and noise with a
#standard deviation of 0.1
x <- sample_arch_spiral(360, min_wrap = 0, max_wrap = 1, sd = 0.1)
```

```
plot(x, asp = 1, pch = 19, cex = .5)

#Uniformly sampled swiss roll in 3-space, from 0 to 1 wraps and width 2*pi
x <- sample_swiss_roll(720, width = 2*pi)
pairs(x, asp = 1, pch = 19, cex = .5)
pca <- prcomp(x)
plot(x %*% pca$rotation, asp = 1, pch = 19, cex = .5)

#Uniformly sampled swiss roll in 3-space, from 0 to 1 wraps and width 2*pi
x <- sample_swiss_roll(720, ar = 2, width = 2*pi)
pairs(x, asp = 1, pch = 19, cex = .5)
pca <- prcomp(x)
plot(x %*% pca$rotation, asp = 1, pch = 19, cex = .5)
```

`circles` *Sample (with noise) from circles*

Description

These functions generate uniform and stratified samples from configurations of circles of radius 1 in 2- or 3-dimensional space, optionally with noise.

Usage

```
sample_circle(n, bins = 1L, sd = 0)

sample_circles_interlocked(n, bins = 1L, sd = 0)
```

Arguments

`n`	Number of observations.
`bins`	Number of intervals to stratify by. Default set to 1, which generates a uniform sample.
`sd`	Standard deviation of (independent multivariate) Gaussian noise.

Details

The function `sample_circle()` uses the usual sinussoidal parameterization from the unit interval to the unit circle.

The function `sample_circles_interlocked()` effectively samples from a pair of circles and rotates them in 3-dimensional space so that they are interlocked (perpendicular to each other).

Both functions are length-preserving and admit stratification. If `bins = 2`, the stratification for the latter function will simply be between the two interlocked circles.

Examples

```
set.seed(14312L)

# circle in 2-space
x <- sample_circle(120, sd = .1)
plot(x, asp = 1, pch = 19, cex = .5)

# interlocked circles in 3-space
x <- sample_circles_interlocked(120, sd = .1)
pairs(x, asp = 1, pch = 19, cex = .5)
```

---

disks *Sample (with noise) from disk*

Description

These functions generate uniform samples from a disk in 2-dimensional space, optionally with noise.

Usage

```
sample_disk(n, bins = 1L, sd = 0)
```

Arguments

n	Number of observations.
bins	Number of intervals per dimension to stratify by. Default set to 1, which generates a uniform sample.
sd	Standard deviation of (independent multivariate) Gaussian noise.

Details

The sample is generated by an area-preserving parameterization of the disk. This parametrization was derived through the method for sampling 2-manifolds as described by Arvo (2001).

References

J Arvo (2001) Stratified Sampling of 2-Manifolds. *SIGRAPH 2001 (State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis)*, Course Notes, Vol. 29. https://www.cs.princeton.edu/courses/archive/fall04/cos526/papers/course29sig01.pdf

Examples

```
set.seed(99812L)

# Uniformly sampled unit disk in 2-space
x <- sample_disk(1800, sd = 0)
plot(x, asp = 1, pch = 19, cex = .5)

# Uniformly sampled unit disk in 2-space with Gaussian noise
x <- sample_disk(1800, sd = .1)
plot(x, asp = 1, pch = 19, cex = .5)
```

---

ellipses *Sample (with noise) from ellipses*

Description

These functions generate uniform samples from configurations of ellipses of major or minor radius 1 in 2- or 3-dimensional space, optionally with noise.

Usage

```
sample_ellipse(n, ar = 1, sd = 0)

sample_cylinder_elliptical(n, ar = 1, width = 1, sd = 0)
```

Arguments

`n`	Number of observations.
`ar`	Aspect ratio for an ellipse (ratio of major and minor radii).
`sd`	Standard deviation of (independent multivariate) Gaussian noise.
`width`	Width of the cylinder (with respect to the fixed radius of the ellipse).

Details

The function `sample_ellipse()` uses the usual sinussoidal parameterization from the unit interval to an ellipse with radii 1 and `1/ar`. The uniform sample is generated through a rejection sampling process as described by Diaconis, Holmes, and Shahshahani (2013).

Examples

```
set.seed(97205L)

# ellipses in 2-space
x <- sample_ellipse(120, ar = 6)
plot(x, asp = 1, pch = 19, cex = .5)
x <- sample_ellipse(120, ar = 1/6)
plot(x, asp = 1, pch = 19, cex = .5)
```

```
# ellipses in 2-space
x <- sample_ellipse(120, ar = 6, sd = .1/6)
plot(x, asp = 1, pch = 19, cex = .5)
x <- sample_ellipse(120, ar = 1/6, sd = .1)
plot(x, asp = 1, pch = 19, cex = .5)

# cylinders in 3-space
x <- sample_cylinder_elliptical(120, ar = 1)
pairs(x, asp = 1, pch = 19, cex = .5)
x <- sample_cylinder_elliptical(120, ar = 3, width = 2*pi)
pairs(x, asp = 1, pch = 19, cex = .5)
```

`klein-bottles` *Sample (with noise) from Klein bottles*

Description

These functions generate uniform samples from Klein bottles in 4-dimensional space, optionally with noise.

Usage

```
sample_klein_tube(n, ar = 2, sd = 0)

sample_klein_flat(n, ar = 1, bump = 0.1, sd = 0)
```

Arguments

`n`	Number of observations.
`ar`	Aspect ratio for Möbius tube Klein bottle (ratio of major and minor radii) or flat torus-based Klein bottle (ratio of scale factors).
`sd`	Standard deviation of (independent multivariate) Gaussian noise.
`bump`	Bump constant for the flat torus-based Klein bottle.

Details

The function `sample_klein_tube()` uses the Möbius tube parameterization documented at the Encyclopédie des Formes Mathématiques Remarquables.

The function `sample_klein_flat()` uses a flat parameterization based on that of the torus, as presented on Wikipedia.

Both uniform samples are generated through a rejection sampling process as described by Diaconis, Holmes, and Shahshahani (2013).

References

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

Examples

```
set.seed(834L)

# Klein bottle tube embedding in 4-space
x <- sample_klein_tube(120, sd = .05)
pairs(x, asp = 1, pch = 19, cex = .5)

# Klein bottle flat torus-based embedding in 4-space
x <- sample_klein_flat(120, sd = .05)
pairs(x, asp = 1, pch = 19, cex = .5)
```

---

lemniscates *Sample (with noise) from lemniscates (figure eights)*

Description

These functions generate uniform samples from lemniscates (figure eights) in 2-dimensional space, optionally with noise.

Usage

```
sample_lemniscate_gerono(n, sd = 0)
```

Arguments

n	Number of observations.
sd	Standard deviation of (independent multivariate) Gaussian noise.

Details

These functions use a simple parameterization from the unit circle to the lemniscate of Gerono, as presented on Wikipedia. The uniform sample is generated through a rejection sampling process as described by Diaconis, Holmes, and Shahshahani (2013).

References

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

Examples

```
set.seed(12051L)

# Uniformly sampled figure eight in 2-space
x <- sample_lemniscate_gerono(720)
plot(x, asp = 1, pch = 19, cex = .5)

# Naively sampled figure eight, for comparison
theta <- runif(n = 720, min = 0, max = 2*pi)
x_naive <- cbind(x = cos(theta), y <- cos(theta) * sin(theta))
plot(x_naive, asp = 1, pch = 19, cex = .5)

# Uniformly sampled figure eight in 2-space with Gaussian noise
x <- sample_lemniscate_gerono(720, 0.1)
plot(x, asp = 1, pch = 19, cex = .5)
```

---

noise *Add noise to a sample*

Description

This function adds Gaussian noise to coordinate data contained in a matrix. It is called by samplers to introduce noise when `sd` is passed a positive value.

Usage

```
add_noise(x, sd = 0)
```

Arguments

`x`	A matrix of row coordinates.
`sd`	Standard deviation of (independent multivariate) Gaussian noise.

---

real-projective-planes *Sample (with noise) from real projective planes*

Description

These functions generate uniform samples from real projective planes in 4-dimensional space, optionally with noise.

Usage

```
sample_projective_plane(n, sd = 0)
```

Arguments

n	Number of observations.
sd	Standard deviation of (independent multivariate) Gaussian noise.

Details

The real projective plane only embeds into a Euclidean space of dimension at least 4. This embedding is adapted from Wikipedia. The uniform sample is generated through a rejection sampling process as described by Diaconis, Holmes, and Shahshahani (2013).

References

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

Examples

```
set.seed(22764L)

# real projective plane embedding in 4-space
x <- sample_projective_plane(120)
pairs(x, asp = 1, pch = 19, cex = .5)
```

---

rejection-samplers *Custom uniform rejection samplers*

Description

These functions create rejection samplers, and uniform manifold samplers based on them, using user-provided parameterization and Jacobian functions.

Usage

```
make_rejection_sampler(
  parameterization,
  jacobian,
  min_params,
  max_params,
  max_jacobian
)
```

Arguments

`parameterization`	A function that takes parameter vector arguments and returns a matrix of coordinates.
`jacobian`	A function that takes parameter vector arguments and returns a vector of Jacobian determinants.
`min_params, max_params`	(Optionally named) vectors of minimum and maximum values of the parameters, used for uniform sampling.
`max_jacobian`	An (ideally sharp) upper bound on the Jacobian determinant.

Details

The rejection sampling technique of Diaconis, Holmes, and Shahshahani (2013) uses a parameterized embedding from a parameter space to a coordinate space and relies on a formula for its jacobian determinant. The `parameterization` must be a function that takes vector arguments of equal length and returns a coordinate matrix of the same number of rows. The `jacobian` must be a function that takes the same arguments and returns a scalar value. The parameters must range from their respective minima `min_params` to their respective maxima `max_params`. `max_jacobian` must be provided, though it may be larger than the theoretical maximum of the jacobian determinant.

References

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

Examples

```
set.seed(47569L)

# parameterization and Jacobian for Klein bottle tube embedding
klein_parameterization <- function(theta, phi) {
  cbind(
    w = (1 + .5 * cos(theta)) * cos(phi),
    x = (1 + .5 * cos(theta)) * sin(phi),
    y = .5 * sin(theta) * cos(phi/2),
    z = .5 * sin(theta) * sin(phi/2)
  )
}
klein_jacobian <- function(theta, phi) {
  unname(.5 * sqrt((1 + .5 * cos(theta)) ^ 2 + (.5 * .5 * sin(theta)) ^ 2))
}
# custom sampler based on these functions
klein_sampler <- make_rejection_sampler(
  klein_parameterization,
  klein_jacobian,
  max_params = c(theta = 2*pi, phi = 2*pi),
  max_jacobian = klein_jacobian(cbind(theta = 0))
)
```

```
# compare custom sampler to `sample_klein_tube()`
pairs(klein_sampler(n = 360), asp = 1, pch = 19, cex = .5)
pairs(sample_klein_tube(n = 360, ar = 2), asp = 1, pch = 19, cex = .5)
```

`sphere` *Sample (with noise) from a sphere*

Description

These functions generate uniform samples from a sphere of radius 1 and dimension 2 in 3-space, or in arbitrary dimension in 1-higher-dimensional space, optionally with noise.

Usage

```
sample_2hemisphere(n, bins = 1L, sd = 0)

sample_2sphere(n, bins = 1L, sd = 0)

sample_sphere(n, dim = 1, sd = 0)
```

Arguments

`n`	Number of observations.
`bins`	Number of intervals per dimension to stratify by. Default set to 1, which generates a uniform sample.
`sd`	Standard deviation of (independent multivariate) Gaussian noise.
`dim`	Dimension of the sphere.

Details

The function `sample_sphere()` is adapted from `sphereUnif()` in the **TDA** package. It uses `stats::rnorm()` to sample from a multivariate Gaussian and normalizes the resulting coordinates.

The function `sample_2hemisphere()` uses an area-preserving parameterization of the upper hemisphere, and `sample_2sphere()` uses two of these samples, one reflected over the horizontal plane, to produce a sample from a sphere. The parametrization was derived through the method for sampling 2-manifolds as described by Arvo (2001).

References

J Arvo (2001) Stratified Sampling of 2-Manifolds. *SIGRAPH 2001 (State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis)*, Course Notes, Vol. 29. https://www.cs.princeton.edu/courses/archive/fall04/cos526/papers/course29sig01.pdf

Examples

```
set.seed(50253L)

# 1-sphere in 2-space
x <- sample_sphere(120, dim = 1, sd = .1)
pairs(x, asp = 1, pch = 19, cex = .5)

# 2-sphere in 3-space
x <- sample_sphere(120, dim = 2, sd = .1)
pairs(x, asp = 1, pch = 19, cex = .5)

# 3-sphere in 4-space
x <- sample_sphere(120, dim = 3, sd = .1)
pairs(x, asp = 1, pch = 19, cex = .5)

# 4-sphere in 5-space
x <- sample_sphere(120, dim = 4, sd = .1)
pairs(x, asp = 1, pch = 19, cex = .5)
```

---

stratified-samplers *Stratified sample of any unit dimensional space*

Description

These functions generate stratified samples of any dimension including the unit line segment in 1-dimensional space, the unit square in 2-space, the unit cube in 3-space.

Usage

```
sample_strat_segment(n, bins)

sample_strat_square(n, bins)

sample_strat_cube(n, bins)

sample_stratify(n, bins, dim)
```

Arguments

n	Number of observations.
bins	Number of intervals per dimension for the stratification.
dim	Dimensional space of sample.

Details

(Details.)

Examples

```
set.seed(28522L)

#Stratified sample in 1-dimension with 10 intervals
values <- sample_strat_segment(13, 10)
x <- cbind(values, rep(0, 13))
plot(x, asp = 1, pch = 19, cex = .5, xlab = 'x', ylab = '')
segments(x0 = seq(0, 1, .1), y0 = -1, y1 = 1)

#Stratified sample of a unit square with 100 cells
x <- sample_strat_square(110, 10)
plot(x, asp = 1, pch = 19, cex = .5, xlab = 'x', ylab = 'y')
segments(x0 = seq(0, 1, .1), y0 = 0, y1 = 1)
segments(y0 = seq(0, 1, .1), x0 = 0, x1 = 1)

#Stratified sample of a unit cube with 27 cells
x <- sample_strat_cube(27, 3)
#Bird's eye view of the cube
plot(x[, c(1, 2)], asp = 1, pch = 19, cex = .5, xlab = 'x', ylab = 'y')
segments(x0 = seq(0, 1, 1/3), y0 = 0, y1 = 1)
segments(y0 = seq(0, 1, 1/3), x0 = 0, x1 = 1)
#Side view of the cube
plot(x[,c(2,3)], asp = 1, pch = 19, cex = .5, xlab = 'y', ylab = 'z')
segments(x0 = seq(0,1,1/3),y0 = 0, y1 = 1)
segments(y0 = seq(0,1,1/3),x0 = 0, x1 = 1)

#All of the same illustrations, but only using sample_stratify()

#Stratified sample in 1-dimension with 10 intervals
values <- sample_stratify(13,10,1)
x <- cbind(values,rep(0,13))
plot(x, asp = 1, pch = 19, cex = .5, xlab = 'x', ylab = '')
segments(x0 = seq(0,1,.1),y0 = -1, y1 = 1)

#Stratified sample of a unit square with 100 cells
x <- sample_stratify(110,10, 2)
plot(x, asp = 1, pch = 19, cex = .5, xlab = 'x', ylab = 'y')
segments(x0 = seq(0,1,.1),y0 = 0, y1 = 1)
segments(y0 = seq(0,1,.1),x0 = 0, x1 = 1)

#Stratified sample of a unit cube with 27 cells
x <- sample_stratify(27,3, 3)
#Bird's eye view of the cube
plot(x[,c(1,2)], asp = 1, pch = 19, cex = .5, xlab = 'x', ylab = 'y')
segments(x0 = seq(0,1,1/3),y0 = 0, y1 = 1)
segments(y0 = seq(0,1,1/3),x0 = 0, x1 = 1)
#Side view of the cube
plot(x[,c(2,3)], asp = 1, pch = 19, cex = .5, xlab = 'y', ylab = 'z')
segments(x0 = seq(0,1,1/3),y0 = 0, y1 = 1)
segments(y0 = seq(0,1,1/3),x0 = 0, x1 = 1)

#Stratified sample of a unit 4-cube with 81 cells
```

```
x <- sample_stratify(81, 3, 4)
#One view of the cube
plot(x[,c(1,2)], asp = 1, pch = 19, cex = .5, xlab = 'x', ylab = 'y')
segments(x0 = seq(0,1,1/3),y0 = 0, y1 = 1)
segments(y0 = seq(0,1,1/3),x0 = 0, x1 = 1)
#Another view of the cube
plot(x[,c(2,3)], asp = 1, pch = 19, cex = .5, xlab = 'y', ylab = 'z')
segments(x0 = seq(0,1,1/3),y0 = 0, y1 = 1)
segments(y0 = seq(0,1,1/3),x0 = 0, x1 = 1)
```

---

tdaunif **tdaunif***: Uniform manifold samplers for topological data analysis*

Description

Generate uniform random samples from embedded manifolds, optionally with noise.

Details

This package assembles functions that generate samples of points uniformly from the surfaces of embedded manifolds. An *embedding* is a one-to-one continuous map $f : M \to X$ from a manifold M to a Euclidean coordinate space X, and each function relies on a *parameterization* of M given by a continuous bijective function $p : S \to f(M)$ that may identify some points of s (boundary or interior) to produce the topology of M. (This means that the inverse of p may not be continuous.)

Sampling points P uniformly from S and mapping the sample to $f(M)$ may produce a non-uniform sample $p(P)$ due to differences in the local sampling rate per unit interior (length, area, volume, etc.), quantified as the Jacobian (higher-order derivative) of p. **tdaunif** uses two techniques to correct for this:

- The more numerical (brute-force) technique is to compute the Jacobian on the parameter space and oversample locally at a rate proportional to the Jacobian. This oversampling is done via rejection sampling as illustrated by Diaconis, Holmes, and Shahshahani (2013).
- The more analytic technique is to invert the Jacobian symbolically in order to define an interior-preserving parameterization $q : S \to f(M)$, as illustrated for 2-manifolds by Arvo (2001). Sampling P uniformly on S then produces a uniform sample $q(P)$ on $f(M)$. The interior-preserving map also enables stratified sampling on the manifold via stratification of the parameter space.

Multivariate Gaussian noise in the coordinate space can be added to any sample.

Author(s)

Jason Cory Brunson

Brandon Demkowicz

Sanmati Choudhary

References

J Arvo (2001) Stratified Sampling of 2-Manifolds. *SIGRAPH 2001 (State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis)*, Course Notes, Vol. 29. https://www.cs.princeton.edu/courses/archive/fall04/cos526/papers/course29sig01.pdf

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

`tori` *Sample (with noise) from tori*

Description

These functions generate uniform samples from configurations of tori of primary radius 1 in 3-dimensional space, optionally with noise.

Usage

```
sample_torus_tube(n, ar = 2, sd = 0)

sample_tori_interlocked(n, ar = 2, sd = 0)

sample_torus_flat(n, ar = 1, sd = 0)
```

Arguments

`n`	Number of observations.
`ar`	Aspect ratio for tube torus (ratio of major and minor radii) or flat torus (ratio of scale factors).
`sd`	Standard deviation of (independent multivariate) Gaussian noise.

Details

The function `sample_torus_tube()` uses the tubular parameterization into 3-dimensional space documented at MathWorld.

The function `sample_torus_flat()` uses a flat parameterization (having zero Gaussian curvature) into 4-dimensional space, as presented on Wikipedia.

The function `sample_tori_interlocked()` samples from two tubular tori interlocked in the same way as `sample_circles_interlocked()`.

All uniform samples are generated through a rejection sampling process as described by Diaconis, Holmes, and Shahshahani (2013).

References

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

Examples

```
set.seed(33183L)

# torus tube embedding in 3-space
x <- sample_torus_tube(120, sd = .05)
pairs(x, asp = 1, pch = 19, cex = .5)

# torus flat embedding in 4-space
x <- sample_torus_flat(120, sd = .05)
pairs(x, asp = 1, pch = 19, cex = .5)
```

---

trefoil *Sample (with noise) from trefoil knot*

Description

These functions generate uniform samples from trefoil knot in 3-dimensional space, optionally with noise.

Usage

```
sample_trefoil(n, sd = 0)
```

Arguments

n	Number of observations.
sd	Standard deviation of (independent multivariate) Gaussian noise.

Details

The trefoil knot is the simplest nontrivial knot and contains three unique crossings in three dimensional space. This uniform sample is generated by rejection sampling. This process allows for simulation of random samples from the trefoil knot distribution, by using random samples from a more convenient distribution. It applies rejection/acceptance criterion such that the samples that are accepted are to be distributed as if they were from the target distribution. The uniform sample is generated through a rejection sampling process as described by Diaconis, Holmes, and Shahshahani (2013).

References

P Diaconis, S Holmes, and M Shahshahani (2013) Sampling from a Manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, 102–125. doi: 10.1214/12IMSCOLL1006

Examples

```
set.seed(73398L)

# Uniformly sampled trefoil knot in 3-space
x <- sample_trefoil(180)
pairs(x, asp = 1, pch = 19, cex = .5, col = "#00000077")

# Uniformly sampled trefoil knot in 3-space, with Gaussian noise
x <- sample_trefoil(180, sd = .1)
pairs(x, asp = 1, pch = 19, cex = .5, col = "#00000077")
```

---

triangles *Sample (with noise) from planar triangles*

Description

This function generates uniform and stratified samples from configurations of planar triangles in 2-dimensional space, optionally with noise.

Usage

```
sample_triangle_planar(n, triangle, bins = 1, sd = 0)
```

Arguments

n	Number of observations.
triangle	The (x,y) coordinates of the vertices of a triangle, formatted in a 2x3 matrix
bins	Number of intervals per dimension to stratify by. Default set to 1, which generates a uniform sample.
sd	Standard deviation of (independent multivariate) Gaussian noise.

Details

The sample is generated by an area-preserving parameterization of the planar triangle. This parametrization was derived through the method for sampling 2-manifolds as described by Arvo (2001).

References

J Arvo (2001) Stratified Sampling of 2-Manifolds. *SIGRAPH 2001 (State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis)*, Course Notes, Vol. 29. https://www.cs.princeton.edu/courses/archive/fall04/cos526/papers/course29sig01.pdf

Examples

```
set.seed(23004L)

#Uniformly sampled equilateral planar triangle in 2-space
equilateral_triangle <- cbind(c(0,0), c(0.5,sqrt(3)/2), c(1,0))
x <-  sample_triangle_planar(10000, equilateral_triangle)
plot(x, asp = 1, pch = 19, cex = .25)

#Stratified sample of equilateral planar triangle in 2-space with 100 bins
equilateral_triangle <- cbind(c(0,0), c(0.5,sqrt(3)/2), c(1,0))
x <-  sample_triangle_planar(10000, equilateral_triangle, bins = 100)
plot(x, asp = 1, pch = 19, cex = .25)

#Uniformly sampled equilateral planar triangle in 2-space with Gaussian noise
equilateral_triangle <- cbind(c(0,0), c(0.5,sqrt(3)/2), c(1,0))
x <-  sample_triangle_planar(10000, equilateral_triangle, sd = .1)
plot(x, asp = 1, pch = 19, cex = .25)
```

Index