Package ‘FieldSim’
March 3, 2015

Type Package
Title Random Fields (and Bridges) Simulations
Version 3.2.1
Date 2015-03-01
Author Alexandre Brouste <Alexandre.Brouste@univ-lemans.fr>, Sophie Lambert-Lacroix <Sophie.Lambert@imag.fr>
Maintainer Alexandre Brouste <Alexandre.Brouste@univ-lemans.fr>
Description Tools for random fields and bridges simulations.
License GPL (>= 3)
LazyLoad yes
Depends R (>= 2.0.0), methods, rgl, RColorBrewer
URL http://cran.r-project.org/web/packages/FieldSim/
NeedsCompilation no
Repository CRAN
Date/Publication 2015-03-03 09:11:01

R topics documented:

fieldsim .................................................. 2
manifold-class .......................................... 3
midpoint .................................................. 4
plot ....................................................... 5
plot-methods ............................................. 6
process-class .......................................... 6
process-method ......................................... 7
quadvar .................................................. 8
setAtlas ................................................ 9
setManifold ............................................ 11
setProcess ............................................. 12
setValues .............................................. 14
show-methods .......................................... 15

Index 16
Simulate manifold indexed Gaussian field by the Fieldsim method

Description

The function `fieldsim` yields simulation of sample path of a manifold indexed Gaussian field (or bridge) following the procedure described in Brouste et al. (2007, 2010, 2014).

Usage

`fieldsim(process, Ne, nbNeighbor)`

Arguments

- `process` an S4 object process
- `Ne` a positive integer corresponding to the number of points to simulate with the accurate simulation step
- `nbNeighbor` a positive integer (between 2 and 32) corresponding to the number of neighbors to use in the second refined step of the algorithm.

Value

The function returns in the slot `values` of the object process the values of the process on the manifold atlas.

Author(s)


References


See Also

`process-class`, `setProcess`. 
**Examples**

```r
# Load FieldSim library
library(FieldSim)

# Fractional Brownian field on [0,1]^2
plane.fBm <- setProcess("fBm-plane", 0.7)
str(plane.fBm)
fieldsim(plane.fBm)
plot(plane.fBm)

# Sphere indexed fractional Brownian field
#sphere.fBm <- setProcess("fBm-sphere", 0.3)
#fieldsim(sphere.fBm)
#plot(sphere.fBm)

# Bridge associated to the Fractional Brownian field on [0,1]^2
#Gamma <- matrix(c(1,0,0,0,1,1,1,1,1,1/2,1/2,0.5),3,4)
#bridge.plane.fBm <- setProcess("bridge-fBm-plane", list(Gamma=Gamma,par=0.9))
#fieldsim(bridge.plane.fBm)
#plot(bridge.plane.fBm)

# Other examples can be found in the setProcess documentation.
```

---

**Description**

The manifold class is a class of the FieldSim package.

**Slots**

- `name`: is the name of the manifold (a character string);
- `atlas`: is the mesh (a matrix);
- `gridtype`: is the grid type (a character string) to plotting;
- `distance`: is the distance set on the manifold (a function);
- `origin`: is the origin fixed on the manifold (a matrix).

**Author(s)**

Alexandre Brouste
Description

The function midpoint yields simulation of sample path of a fractional Brownian field by the midpoint displacement method.

Usage

midpoint(process)

Arguments

process an object of class process (namely an FBm).

Details

The subspace $[0,1] \times [0,1]$ is discretized in a regular space discretization of size $(2^{nlevel} + 1)^2$. At each point of the grid, the fractional Brownian field is simulated using the midpoint displacement method described for example in Fournier et al. (1982).

Value

an object of class process with the simulated sample path in the corresponding slot values.

Author(s)


References


See Also

fieldsim.
plot

Examples

# load FieldSim library
library(FieldSim)

plane.fBm <- setProcess("fBm-plane", 0.9)
midpoint(plane.fBm)
plot(plane.fBm)

plot

Generic plotting of specific manifold indexed fractional Gaussian processes

Description

The function plots some of usual manifold indexed fractional Gaussian processes.

Usage

plot(x, y, ...)

Arguments

x an object of class process;
y the type of the plot, possible choices are "default", "cloud" or "sun".
... Other plot arguments

Author(s)


References


See Also

fieldsim, setProcess.
Examples

# Load FieldSim library
library(FieldSim)

# Fractional Brownian field on [0,1]^2
plane.fBm<-setProcess("fBm-plane",0.7)
str(plane.fBm)
fieldsim(plane.fBm)
plot(plane.fBm)

# The "cloud" plotting
plot(plane.fBm,"cloud")

# The "sun" plotting
plot(plane.fBm,"sun")

# Sphere indexed fractional Brownian field
#sphere.fBm<-setProcess("fBm-sphere",0.3)
#fieldsim(sphere.fBm)
#plot(sphere.fBm)

Description

Specific plot method for process class

Author(s)

Alexandre Brouste

See Also

setProcess, manifold-class.

Description

The process class is a class of the FieldSim package.
Details

Several names for slot name are reserved for classical fractional Gaussian processes: "fBm" for fractional Brownian motion, "mBm" for multifractional Brownian motion, "2pfBm" for the standard bi-fractional Brownian motion, "stdfBm" for the space-time deformed fractional Brownian motion, "afBf" for anisotropic fractional Brownian field, "fBs" for fractional Brownian sheet and "bridge" for all kind of bridges.

The slot manifold contains an object of class manifold (see manifold-class).

The slot parameter that contains all the parameter associated to the covariance function of the process. Here are the classical parameter associated to classical processes. For instance, "fBm" has parameter numeric, "mBm" has parameter function, "2pfBm" has parameter list(H=numeric, K=numeric), "stdfBm" has parameter list(H=numeric, sigma=function, tau=function), "afBf" has parameter list(H=numeric, thetal=numeric, theta2=numeric), "fBs" has parameter vector and "bridge" has list(Gamma=matrix, R=function, Tp=..., par=list(...)).

Slots

- name: is the name of the manifold (a character string).
- values: the values of the simulated (or given) sample path of the process ()
- parameter: is the origin fixed on the manifold (a matrix)
- manifold: is the distance set on the manifold (a function).
- covf: is the mesh (a matrix).

Author(s)

Alexandre Brouste

See Also

setProcess, manifold-class.

Description

The process class is a class of the FieldSim package.

Slots

- name: is the name of the manifold (a character string).
- values: the values of the simulated (or given) sample path of the process ()
- parameter: is the origin fixed on the manifold (a matrix)
- manifold: is the distance set on the manifold (a function).
- covf: is the mesh (a matrix).
quadvar

Estimate the Hurst parameter of a plane indexed fractional Brownian field by the quadratic variations method

Description

The function `quadvar` yields the estimation of the Hurst parameter of a fractional Brownian field by the quadratic variations method in the plane case.

Usage

`quadvar(process, parameter)`

Arguments

- `process`: a S4 object process;
- `parameter`: parameter (in progress).

Details

The Hurst parameter of the fractal Brownian field is estimated by the procedure described in Istas and Lang (1997).

Value

`H` is a real in `[0, 1]` that represents the estimate of the Hurst parameter of the fractional Brownian field.

Author(s)


References


See Also

`fieldsim, setProcess, setValues`. 

Examples

```r
# load fieldsim library
library(fieldsim)

# Simulated Fractional Brownian field on [0,1]^2
plane.FBm <- setProcess("FBm-plane", 0.7)
fieldsim(plane.FBm)
quadvar(plane.FBm)

# Simulated Multifractional Brownian field on [0,1]^2
funcH <- function(x){0.3+xi[1]*0.6}
plane.mBm <- setProcess("mBm-plane", funcH)
fieldsim(plane.mBm)
quadvar(plane.mBm, parameter=list(point=c(0.5,0.5), h=0.2))
```

---

**setAtlas**

**Construct usual grids on some specific manifolds**

Description

The function `setAtlas` constructs usual grids on manifold.

Usage

```r
setAtlas(object, gridtype, Ng)
```

Arguments

- `object`: an object of class process or manifold;
- `gridtype`: the type of the grid, possible choice "regular", "random" or "visualization";
- `Ng`: parameter of the size of the grid, see details.

Details

We list here the different implemented grids. For manifold name="plane" we have the gridtype="regular" grid (with the parameter Ng returns a regular grid on [0,1]x[0,1] of size Ng*Ng), the gridtype="random" grid (uniform random choice of the both coordinates on [0,1], grid of size Ng*Ng) and the gridtype="visualization" grid, of size (2^Ng+1)*(2^Ng+1) composed of regular refinments.

For manifold name="sphere", we have the following grids: there isn't exist gridtype="regular" grid for a sphere, but a gridtype="random" grid (uniform density sample on the sphere of size Ng*Ng) and a gridtype="visualization" grid (sphere-visualization grid on the sphere of size 6*Ng*Ng, union of the 6 domains centered around one of the 6 triply orthogonal poles, each domain are composed of the heights on the sphere (when they exists) corresponding to the regular mesh [-3/4,3/4]*[-3/4,3/4] of the others two cartesian coordinates).

Finally, for manifold name="hyperboloid" we have: no gridtype="regular" grid on the hyperboloid, but a gridtype="random" grid (uniform density sample on the sphere of size Ng*Ng) and a gridtype="visualization" grid (hyperboloid-visualization grid of size Ng*Ng, a domain
of composed of the height of the hyperboloid corresponding to the regular mesh [-3,3]x[-3,3] of the other two cartesian coordinates)

Author(s)

References

See Also
fieldsim.

Examples
# Load FieldSim library
library(FieldSim)

# Example on the line manifold
line<-setManifold("line")
str(line)

setAtlas(line,"regular",200)
str(line)

#Example on the fractional Brownian motion
line.fBm<-setProcess("fBm-line",0.7)
str(line.fBm)

setAtlas(line.fBm,"regular",200)
str(line.fBm)

setAtlas(line.fBm,"random",100)
str(line.fBm)

setAtlas(line.fBm,"finer",9)
str(line.fBm)

setAtlas(line.fBm,"visualization",9)
str(line.fBm)
setManifold

Set a S4 manifold object

Description

The function sets an object of class manifold.

Usage

setManifold(name, atlas, gridtype, distance, origin)

Arguments

name
  name of the manifold (type character);

atlas
  atlas of the manifold (type matrix);

gridtype
  is the grid type (a character string) to plotting;

distance
  distance on the manifold (type function);

origin
  origin of the manifold (type matrix).

Value

An object of class manifold with the 5 slots name, atlas, gridtype, distance and origin.

Author(s)


References


See Also

setProcess.

Examples

# Load FieldSim library
library(FieldSim)

# Example 1: User manifold
name1<-"plane1"
mesh<-seq(from=0,to=1,length=16)
atlas1<-rbind(rep(mesh,each=16),rep(mesh,16))
d1<-function(xi,xj){return(sqrt(t(xi-xj)%*%(xi-xj)))}
**Description**

The function `setProcess` constructs usual processes on a specific manifold.

**Usage**

```r
setProcess(name, parameter, values, manifold, covf)
```

**Arguments**

- `name` the name of the process (see details);
- `parameter` the parameters of the process (see details);
- `values` the values of the simulated (or given) sample path of the process;
- `manifold` the manifold of which the process is defined;
- `covf` the autocovariance function of the process.

**Details**

We list here the different usual process.

**Value**

an object of class `process`. 

```
origin1<-rbind(0,0)
manifold1<-setManifold(name=name1, atlas=atlas1, distance=d1, origin=origin1)
str(manifold1)

#Example 2: The "line" manifold
line<-setManifold("line")
str(line)

#Example 3: The "plane" manifold
plane<-setManifold("plane")
str(plane)

#Example 4: The "sphere" manifold
sphere<-setManifold("sphere")
str(sphere)

#Example 5: The "hyperboloid" manifold
hyper<-setManifold("hyperboloid")
str(hyper)
```
Author(s)

References

See Also
fieldsim

Examples

# Load FieldSim library
library(FieldSim)

# Fractional Brownian field on [0,1]^2
plane.fBm <- setProcess("fBm-plane", 0.7)
str(plane.fBm)

# Multifractional Brownian field on [0,1]^2
func <- function(xi){0.3+xi[1]*0.6}
plane.mBm <- setProcess("mBm-plane", func)
str(plane.mBm)

# Fractional Brownian sheet on [0,1]^2
#plane.fBs <- setProcess("fBs-plane", c(0.9,0.3))
#str(plane.fBs)

# Anisotropic fractional Brownian field on [0,1]^2
#plane.afBf <- setProcess("afBf-plane", list(H=0.7, theta1=pi/6, theta2=pi/3))
#str(plane.afBf)

# Bifractional fractional Brownian field on [0,1]^2
#plane.zpBm <- setProcess("zpBm-plane", list(H=0.7, K=0.5))
#str(plane.zpBm)

# Spherical fractional Brownian field
#sphere.fBm <- setProcess("fBm-sphere", 0.3)
#str(sphere.fBm)

# Fractional Brownian field on the hyperboloid
#hyper.fBm <- setProcess("fBm-hyperboloid", 0.7)
#str(hyper.fBm)

# Bridge associated to the Fractional Brownian field on [0,1]^2
#Gamma <- matrix(c(1,0,0,0,1,1,1,1,1,1/2,1/2,0.5),3,4)
#bridge.plane.fBm <- setProcess("bridge-fBm-plane", list(Gamma=Gamma, par=0.9))
#str(bridge.plane.fBm)
# User defined process (see Brouste et al. 2010)

#sphere<-setManifold("sphere")
#user.sphere<-setProcess(name="user",manifold=sphere)

#param<-0.7
#acov<-function(xi,xj){exp(-#user.sphere@manifold@distance(xi,xj)^2*#user.sphere@parameter))}

#user.sphere@parameter<-param
#user.sphere@covf<-acov

#fieldsim(user.sphere)
#plot(user.sphere)

---

**setValues**

Set the values of an object of class `process`

### Description

The function `setValues` set the values of the process.

### Usage

```r
setValues(process, values)
```

### Arguments

- **process** an S4 object process;
- **values** the values of the process on the atlas.

### Details

Statistical tools developed in the Fieldsim package allows real dataset inserted in the model process. Consequently the user can set the values of the process. Parameter will be forgotten using statistical command. This values will be erase with the use of fieldsim.

### Author(s)


### References

Show methods

See Also

fieldsim.

Examples

```r
# Load FieldSim library
library(FieldSim)

# Dataset (to do)
plane.fBm <- setProcess("fBm-plane", 0.6)
fieldsim(plane.fBm)
sample <- plane.fBm@values

plane.fBm.2 <- setProcess("fBm-plane", 0.7)
setValues(plane.fBm.2, sample)
```

show-methods

Description

Classical show and print methods available.

Author(s)

Alexandre Brouste

See Also

process-class, manifold-class.
Index

**Topic classes**

- manifold-class, 3
- plot-methods, 6
- process-class, 6
- process-method, 7
- show-methods, 15

fieldsim, 2, 4, 5, 8, 10, 13, 15

initialize, process-method
  (process-method), 7
initialize, manifold-method
  (manifold-class), 3

manifold-class, 3, 7
midpoint, 4

plot, 5
plot, manifold-method (plot-methods), 6
plot, process-method (plot-methods), 6
plot-methods, 6
print, manifold-method (show-methods), 15
print-methods (show-methods), 15
process (process-class), 6
process-class, 6
process-method, 7

quadvar, 8

setAtlas, 9
setAtlas, manifold-method (setAtlas), 9
setAtlas, process-method (setAtlas), 9
setManifold, 11
setProcess, 2, 5–8, 11, 12
setValues, 8, 14
setValues, process-method (setValues), 14
show, manifold-method (show-methods), 15
show-methods, 15