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.

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` *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)

Alexandre Brouste (`http://perso.univ-lemans.fr/~abrouste/`) and Sophie Lambert-Lacroix (`http://membres-timc.imag.fr/Sophie.Lambert/`).

References

A. Brouste, J. Istas and S. Lambert-Lacroix (2007). On Gaussian random fields simulations. A. Brouste, J. Istas and S. Lambert-Lacroix (2010) On simulation of manifold indexed fractional Gaussian fields. A. Brouste, J. Istas and S. Lambert-Lacroix (2014) Fractional Gaussian bridges with the package `FieldSim`.

See Also

`process-class`, `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)

# 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.
```

`manifold-class` *Manifold class*

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

`midpoint` *Fractional Brownian field simulation by the midpoint displacement method*

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] x [0,1] is discretized in a regular space discretization of size $(2^{\texttt{nblevel}} + 1)^2$. 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)

Alexandre Brouste (`http://perso.univ-lemans.fr/~abrouste/`) and Sophie Lambert-Lacroix (`http://membres-timc.imag.fr/Sophie.Lambert/`).

References

A. Fournier, D. Fussel and L. Carpenter (1982) Computer rendering of stochastic model, Communication of the AMC, 25, 371-384.

H.O. Peitgen and D. Saupe (1998) The science of fractal images, Springer-Verlag.

R.F. Voss (1985) Random fractal forgeries. NATO ASI Series, F17, 805-835.

See Also

`fieldsim`.

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)

Alexandre Brouste (`http://perso.univ-lemans.fr/~abrouste/`) and Sophie Lambert-Lacroix (`http://membres-timc.imag.fr/Sophie.Lambert/`).

References

A. Brouste, J. Istas and S. Lambert-Lacroix (2010). On simulation of manifold indexed fractional Gaussian fields.

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)
```

---

plot-methods *plot methods*

Description

Specific plot method for process class

Author(s)

Alexandre Brouste

See Also

setProcess,manifold-class.

---

process-class *process 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, theta1=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`.

`process-method` *process method*

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).

Author(s)

Alexandre Brouste

See Also

setProcess,manifold-class.

---

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

Author(s)

Alexandre Brouste (http://perso.univ-lemans.fr/~abrouste/) and Sophie Lambert-Lacroix (http://membres-timc.imag.fr/Sophie.Lambert/).

References

J. Istas and G. Lang (1997). Quadratic variations and estimation of the local Holder index of a Gaussian process. Annales Institut Henri Poincare, 33,407-436.

See Also

fieldsim, setProcess, setValues.

Examples

```
# 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(xi){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

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

Arguments

`object`	a 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]x[-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-vizualisation 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)

Alexandre Brouste (http://perso.univ-lemans.fr/~abrouste/) and Sophie Lambert-Lacroix (http://membres-timc.imag.fr/Sophie.Lambert/).

References

A. Brouste, J. Istas and S. Lambert-Lacroix (2010). On simulation of manifold indexed fractional Gaussian fields.

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)

Alexandre Brouste (http://perso.univ-lemans.fr/~abrouste/) and Sophie Lambert-Lacroix (http://membres-timc.imag.fr/Sophie.Lambert/).

References

A. Brouste, J. Istas and S. Lambert-Lacroix (2010). On simulation of manifold indexed fractional Gaussian fields.

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)))}
```

```
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)
```

---

setProcess *Construct usual processes on manifolds*

Description

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

Usage

```
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`.

Author(s)

Alexandre Brouste (http://perso.univ-lemans.fr/~abrouste/) and Sophie Lambert-Lacroix (http://membres-timc.imag.fr/Sophie.Lambert/).

References

A. Brouste, J. Istas and S. Lambert-Lacroix (2010) On simulation of manifold indexed fractional Gaussian fields. A. Brouste, J. Istas and S. Lambert-Lacroix (2014) Fractional Gaussian bridges with the package FieldSim.

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
funcH<-function(xi){0.3+xi[1]*0.6}
plane.mBm<-setProcess("mBm-plane",funcH)
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.2pfBm<-setProcess("2pfBm-plane",list(H=0.7,K=0.5))
#str(plane.2pfBm)

# 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)

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

#user.sphere@parameter<-parameter
#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

```
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)

Alexandre Brouste (`http://perso.univ-lemans.fr/~abrouste/`) and Sophie Lambert-Lacroix (`http://membres-timc.imag.fr/Sophie.Lambert/`).

References

A. Brouste, J. Istas and S. Lambert-Lacroix (2015). Fractional Gaussian bridges with the package `FieldSim`.

See Also

fieldsim.

Examples

```
# 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 *show methods*

Description

Classical show and print methods available.

Author(s)

Alexandre Brouste

See Also

process-class,manifold-class.

Index