

# Package ‘FracKrigingR’

May 7, 2026

**Type** Package

**Title** Spatial Multivariate Data Modeling

**Version** 1.0.0

**Copyright** Vilnius University Institute of Data Science and Digital Technologies

**Author** Neringa Urbonaite [aut, cre],  
Leonidas Sakalauskas [aut]

**Maintainer** Neringa Urbonaite <neringa.urbonaite@mif.vu.lt>

**Description** Aim is to provide fractional Brownian vector field generation algorithm, Hurst parameter estimation method and fractional kriging model for multivariate data modeling.

**License** GPL-2

**Encoding** UTF-8

**URL** <https://github.com/NidaGreen/FracKriging>

**Imports** psych, clusterGeneration, graphics, stats

**Suggests** knitr, gstat, sp, rmarkdown, raster

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-11-08 08:40:08 UTC

## Contents

FracField . . . . .	2
FracKrig . . . . .	3
FracMatrix . . . . .	4
MaxLikelihood . . . . .	5
<b>Index</b>	<b>6</b>

---

FracField	<i>FracField</i>
-----------	------------------

---

**Description**

Generates fractional Brownian vector field data

**Usage**

```
FracField(K, m, H, X)
```

**Arguments**

K	number of observations
m	number of criteria
H	Hurst parameter (a real in interval [0,1))
X	Coordinates

**Value**

Returns a fractional Brownian vector field matrix.

**Examples**

```
# Load FracKrigingR library
library(FracKrigingR)
# generate Coordinates
p=2; K=10;
X<-matrix(0,ncol=p, nrow=K)
for(j in 1:p){
  for(i in 1:K){
    X[i,j] = rnorm(1, 0, 1)
  }
}
# generate fractional Brownian vector field
H = 0.5; m = 3
FracField(K,m,H,X)
```

---

FracKrig	<i>FracKrig</i>
----------	-----------------

---

**Description**

Performs extrapolation for spatial multivariate data

**Usage**

```
FracKrig(X, Z, Xnew, H)
```

**Arguments**

X	Coordinates
Z	observations
Xnew	Coordinates of points where the prognosis should be made
H	Hurst parameter (a real in interval [0,1))

**Value**

Returns a matrix of fractional kriging prognosis.

**Examples**

```
library(sp)
library(gstat)
data(meuse)
xy<-cbind(meuse$x,meuse$y)
X<-xy[1:50,]
min_max_norm <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
dat<-cbind(meuse[3],meuse[4],meuse[5])
data<-dat[51:100,]
zz1 <- as.data.frame(lapply(dat, normalize))
data1=as.data.frame(lapply(as.data.frame(data), normalize))
Z<-as.matrix(zz1[1:50,])
library(FracKrigingR)
K<-50
#Hurst parameter estimation
H<-0.2
Xnew<-xy[51:100,]
results<- FracKrig(X,Z,Xnew,H)
denormalize <- function(x, bottom, top){
  (top - bottom) * x + bottom
```

```

    }
  z1 = denormalize(
    results[,1], top = max(data[,1]), bottom = min(data[,1])
  )
  z2 = denormalize(
    results[,2], top = max(data[,2]), bottom = min(data[,2])
  )
  z3 = denormalize(
    results[,3], top = max(data[,3]), bottom = min(data[,3])
  )
  RMSE<-function(z,prognosis){
    rmse<-sqrt(((1/(length(z))))*sum((z-prognosis)^2))
    rmse
  }
  Cd<-RMSE(data[,1],z1)
  Cu<-RMSE(data[,2],z2)
  Pb<-RMSE(data[,3],z3)
  Cd
  Cu
  Pb

```

---

 FracMatrix

*FracMatrix*


---

## Description

Fractional distance matrix

## Usage

```
FracMatrix(H, K, X)
```

## Arguments

H	Hurst parameter (a real in interval [0,1))
K	number of observations
X	Coordinates

## Value

Returns a fractional distance matrix, which depends on the Hurst parameter.

## Examples

```

# Load FracKrigingR library
library(FracKrigingR)
#Fractional Brownian vector field
K = 10; H = 0.5; p = 2

```

```
#Generate coordinates
X<-matrix(0,ncol=p, nrow=K)
for(j in 1:p){
  for(i in 1:K){
    X[i,j] = rnorm(1, 0, 1)
  }
}
FracMatrix(H, K, X)
```

---

MaxLikelihood

*MaxLikelihood*


---

### Description

Maximum likelihood method for Hurst parameter estimation of multivariate data

### Usage

```
MaxLikelihood(X, Z)
```

### Arguments

X	Coordinates
Z	Observations

### Value

Returns the estimate of the Hurst parameter (a real in  $[0,1)$ ) and a graph indicating the minimized maximum likelihood function with the Hurst parameter.

### Examples

```
# Load FracKrigingR library
library(FracKrigingR)
# generate Coordinates
p<-2; K<-20;
X<-matrix(0,ncol=p, nrow=K)
for(j in 1:p){
  for(i in 1:K){
    X[i,j] = rnorm(1, 0, 1)
  }
}
# generate fractional Brownian vector field
H <- 0.8; m <- 3
Z<-FracField(K,m,H,X)
# Hurst parameter estimation
MaxLikelihood(X,Z)
```

# Index

FracField, 2

FracKrig, 3

FracMatrix, 4

MaxLikelihood, 5