Package ‘rSPDE’

February 23, 2021

Type Package

Title Rational Approximations of Fractional Stochastic Partial Differential Equations

Version 0.6.2

Maintainer David Bolin <davidbolin@gmail.com>

Description Functions that compute rational approximations of fractional elliptic stochastic partial differential equations. The package also contains functions for common statistical usage of these approximations. The main reference for the methods is Bolin and Kirchner (2020) <doi:10.1080/10618600.2019.1665537>, which can be generated by the citation function in R.

Depends R (>= 3.2.0), Matrix

Imports stats

License GPL (>= 3)

URL https://github.com/davidbolin/rSPDE

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, INLA (>= 0.0-1468840039), testthat, rgdal

Additional_repositories https://inla.r-inla-download.org/R/testing

BugReports https://github.com/davidbolin/rSPDE/issues

VignetteBuilder knitr

NeedsCompilation no

Author David Bolin [cre, aut]

Repository CRAN

Date/Publication 2021-02-23 10:10:02 UTC

R topics documented:

`fractional.operators` *Rational approximations of fractional operators*

Description

`fractional.operators` is used for computing an approximation, which can be used for inference and simulation, of the fractional SPDE

$$L^{\beta}(\tau u(s)) = W.$$

Here L is a differential operator, $\beta > 0$ is the fractional power, τ is a positive scalar or vector that scales the variance of the solution u, and W is white noise.

Usage

```
fractional.operators(L, beta, C, scale.factor, m = 1, tau = 1)
```

Arguments

`L`	A finite element discretization of the operator L.
`beta`	The positive fractional power.
`C`	The mass matrix of the finite element discretization.
`scale.factor`	A constant c is a lower bound for the the smallest eigenvalue of the non-discretized operator L.
`m`	The order of the rational approximation, which needs to be a positive integer. The default value is 1. Higer values gives a more accurate approximation, which are more computationally expensive to use for inference. Currently, the largest value of m that is implemented is 4.
`tau`	The constant or vector that scales the variance of the solution. The default value is 1.

Details

The approximation is based on a rational approximation of the fractional operator, resulting in an approximate model on the form

$$P_l u(s) = P_r W,$$

where $P_j = p_j(L)$ are non-fractional operators defined in terms of polynomials p_j for $j = l, r$. The order of p_r is given by `m` and the order of p_l is $m + m_\beta$ where m_β is the integer part of β if $\beta > 1$ and $m_\beta = 1$ otherwise.

The discrete approximation can be written as $u = P_r x$ where $x \sim N(0, Q^{-1})$ and $Q = P_l^T C^{-1} P_l$. Note that the matrices P_r and Q may be be ill-conditioned for $m > 1$. In this case, the metehods in `operator.operations` should be used for operations involving the matrices, since these methods are more numerically stable.

Value

`fractional.operators` returns an object of class "rSPDEobj". This object contains the following quantities:

`Pl`	The operator P_l.
`Pr`	The operator P_r.
`C`	The mass lumped mass matrix.
`Ci`	The inverse of `C`.
`m`	The order of the rational approximation.
`beta`	The fractional power.
`type`	String indicating the type of approximation.
`Q`	The matrix `t(Pl)%*%solve(C,Pl)`.
`type`	String indicating the type of approximation.
`Pl.factors`	List with elements that can be used to assemble P_l.
`Pr.factors`	List with elements that can be used to assemble P_r.

Author(s)

David Bolin `<davidbolin@gmail.com>`

See Also

`matern.operators`, `spde.matern.operators`

Examples

```
#Compute rational approximation of a Gaussian process with a
#Matern covariance function on R
kappa <- 10
sigma <- 1
nu <- 0.8

#create mass and stiffness matrices for a FEM discretization
```

```
x <- seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)

#compute rational approximation of covariance function at 0.5
tau <- sqrt(gamma(nu) / (sigma^2 * kappa^(2*nu) * (4*pi)^(1/2) * gamma(nu+1/2)))
op <- fractional.operators(L = fem$G + kappa^2*fem$C, beta = (nu + 1/2)/2,
                           C=fem$C, scale.factor = kappa^2, tau = tau)

v = t(rSPDE.A1d(x,0.5))
c.approx = op$Pr %*% solve(op$Q, op$Pr %*% v)

#plot the result and compare with the true Matern covariance
plot(x, matern.covariance(abs(x - 0.5), kappa, nu, sigma), type = "l", ylab = "C(h)",
     xlab="h", main = "Matern covariance and rational approximations")
lines(x, c.approx, col = 2)
```

`matern.covariance` *The Matern covariance function*

Description

`matern.covariance` evaluates the Matern covariance function

$$C(h) = \frac{\sigma^2}{2^(\nu - 1)\Gamma(\nu)}(\kappa h)^\nu K_\nu(\kappa h)$$

Usage

```
matern.covariance(h, kappa, nu, sigma)
```

Arguments

`h`	Distances to evaluate the covariance function at.
`kappa`	Range parameter.
`nu`	Shape parameter.
`sigma`	Standard deviation.

Value

A vector with the values C(h).

Examples

```
x = seq(from = 0, to = 1, length.out = 101)
plot(x, matern.covariance(abs(x - 0.5), kappa = 10, nu = 1/5, sigma = 1),
     type = "l", ylab = "C(h)", xlab = "h")
```

`matern.loglike` *Log-likelihood for a latent Gaussian Matern model using a rational SPDE approximation*

Description

This function evaluates the log-likelihood function for a Gaussian process with a Matern covariance function, that is observed under Gaussian measurement noise: $Y_i = u(s_i) + \epsilon_i$, where ϵ_i are iid mean-zero Gaussian variables. The latent model is approximated using a rational approximation of the fractional SPDE model corresponding to the Gaussian process.

Usage

```
matern.loglike(kappa, sigma, nu, sigma.e, Y, G, C, A, d = 2, m = 1)
```

Arguments

`kappa`	Range parameter of the latent process.
`sigma`	Standard deviation of the latent process.
`nu`	Shape parameter of the latent process.
`sigma.e`	The standard deviation of the measurement noise.
`Y`	The observations, either a vector or a matrix where the columns correspond to independent replicates of observations.
`G`	The stiffness matrix of a finite element discretization of the domain.
`C`	The mass matrix of a finite element discretization of the domain.
`A`	A matrix linking the measurement locations to the basis of the FEM approximation of the latent model.
`d`	The dimension of the domain. The default value is 2.
`m`	The order of the rational approximation, which needs to be a positive integer. The default value is 1.

Value

The log-likelihood value.

See Also

`spde.matern.loglike`, `rSPDE.loglike`, `matern.operators`.

Examples

```
#this example illustrates how the function can be used for maximum likelihood estimation
set.seed(123)
#Sample a Gaussian Matern process on R using a rational approximation
sigma = 1
nu = 0.8
kappa = 1
sigma.e = 0.3
n.rep = 10
n.obs = 100
n.x = 51

#create mass and stiffness matrices for a FEM discretization
x = seq(from = 0, to = 1, length.out = n.x)
fem <- rSPDE.fem1d(x)

#compute rational approximation
op <- matern.operators(kappa = kappa, sigma = sigma, nu = nu,
                       G = fem$G, C = fem$C, d = 1)

#Sample the model
u <- simulate(op, n.rep)

#Create some data
obs.loc <- runif(n = n.obs, min = 0, max = 1)
A <- rSPDE.A1d(x, obs.loc)
noise <- rnorm(n.obs*n.rep)
dim(noise) <- c(n.obs, n.rep)
Y = as.matrix(A%*%u + sigma.e*noise)

#define negative likelihood function for optimization using matern.loglike
mlik <- function(theta, Y, G, C, A){
return(-matern.loglike(exp(theta[1]), exp(theta[2]), exp(theta[3]), exp(theta[4]),
                       Y = Y, G = G, C = C, A = A, d = 1))
}

#The parameters can now be estimated by maximizing mlik with optim

#Choose some reasonable starting values depending on the size of the domain
theta0 = log(c(sqrt(8), sqrt(var(c(Y))), 0.9, 0.01))

#run estimation and display the results
theta <- optim(theta0, mlik, Y = Y, G = fem$G, C = fem$C, A = A)

print(data.frame(kappa = c(kappa,exp(theta$par[1])), sigma = c(sigma,exp(theta$par[2])),
                 nu = c(nu,exp(theta$par[3])), sigma.e = c(sigma.e,exp(theta$par[4])),
                 row.names = c("Truth","Estimates")))
```

`matern.operators` *Rational approximations of stationary Gaussian Matern random fields*

Description

`matern.operators` is used for computing a rational SPDE approximation of a stationary Gaussian random fields on R^d with a Matern covariance function

$$C(h) = \frac{\sigma^2}{2^(\nu - 1)\Gamma(\nu)}(\kappa h)^\nu K_\nu(\kappa h)$$

Usage

```
matern.operators(kappa, sigma, nu, G, C, d = NULL, m = 1)
```

Arguments

`kappa`	Range parameter of the covariance function.
`sigma`	Standard deviation of the covariance function.
`nu`	Shape parameter of the covariance function.
`G`	The stiffness matrix of a finite element discretization of the domain of interest.
`C`	The mass matrix of a finite element discretization of the domain of interest.
`d`	The dimension of the domain.
`m`	The order of the rational approximation, which needs to be a positive integer. The default value is 1.

Details

The approximation is based on a rational approximation of the fractional operator $(\kappa^2 - \Delta)^\beta$, where $\beta = (\nu + d/2)/2$. This results in an approximate model of the form

$$P_l u(s) = P_r W,$$

where $P_j = p_j(L)$ are non-fractional operators defined in terms of polynomials p_j for $j = l, r$. The order of p_r is given by `m` and the order of p_l is $m + m_\beta$ where m_β is the integer part of β if $\beta > 1$ and $m_\beta = 1$ otherwise.

The discrete approximation can be written as $u = P_r x$ where $x \sim N(0, Q^{-1})$ and $Q = P_l^T C^{-1} P_l$. Note that the matrices P_r and Q may be be ill-conditioned for $m > 1$. In this case, the metehods in `operator.operations` should be used for operations involving the matrices, since these methods are more numerically stable.

Value

`matern.operators` returns an object of class "rSPDEobj". This object contains the quantities listed in the output of `fractional.operators` as well as the parameters of the covariance functoin.

Author(s)

David Bolin `<davidbolin@gmail.com>`

See Also

`fractional.operators`, `spde.matern.operators`

Examples

```
#Compute rational approximation of a Gaussian process with a
#Matern covariance function on R
kappa <- 10
sigma <- 1
nu <- 0.8

#create mass and stiffness matrices for a FEM discretization
x <- seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)

#compute rational approximation of covariance function at 0.5
op <- matern.operators(kappa = kappa, sigma = sigma, nu = nu,
                       G = fem$G, C = fem$C, d = 1)

v = t(rSPDE.A1d(x,0.5))
c.approx = op$Pr %*% solve(op$Q, op$Pr %*% v)

#plot the result and compare with the true Matern covariance
plot(x, matern.covariance(abs(x - 0.5), kappa, nu, sigma), type = "l", ylab = "C(h)",
     xlab="h", main = "Matern covariance and rational approximation")
lines(x,c.approx,col=2)
```

`operator.operations` *Operations with the Pr and Pl operators*

Description

Functions for multiplying and solving with the P_r and P_l operators as well as the latent precision matrix $Q = P_l C^{-1} P_l$ and covariance matrix $\Sigma = P_r Q^{-1} P_r^T$. These operations are done without first assembling P_r, P_l in order to avoid numerical problems caused by ill-conditioned matrices.

Usage

```
Pr.mult(obj, v, transpose = FALSE)

Pr.solve(obj, v, transpose = FALSE)

Pl.mult(obj, v, transpose = FALSE)

Pl.solve(obj, v, transpose = FALSE)

Q.mult(obj, v)

Q.solve(obj, v)

Qsqrt.mult(obj, v, transpose = FALSE)
```

```
Qsqrt.solve(obj, v, transpose = FALSE)

Sigma.mult(obj, v)

Sigma.solve(obj, v)
```

Arguments

obj	rSPDE object
v	vector to apply the operation to
transpose	set to TRUE if the operation should be performed with the transposed object

Details

`Pl.mult`, `Pr.mult`, and `Q.mult` multiplies the vector with the respective object. Changing `mult` to `solve` in the function names multiplies the vector with the inverse of the object. `Qsqrt.mult` and `Qsqrt.solve` performs the operations with the square-root type object $Q_r = C^{-1/2}P_l$ defined so that $Q = Q_r^T Q_r$.

Value

A vector with the values of the operation

`predict.rSPDEobj` *Prediction of a fractional SPDE using a rational SPDE approximation*

Description

The function is used for computing kriging predictions based on data $Y_i = u(s_i) + \epsilon_i$, where ϵ is mean-zero Gaussian measurement noise and $u(s)$ is defined by a fractional SPDE $L^\beta u(s) = W$, where W is Gaussian white noise.

Usage

```
## S3 method for class 'rSPDEobj'
predict(object, A, Aprd, Y, sigma.e, compute.variances = FALSE, ...)
```

Arguments

object	The rational SPDE approximation, computed using `fractional.operators`, `matern.operators`, or `spde.matern.operators`.
A	A matrix linking the measurement locations to the basis of the FEM approximation of the latent model.
Aprd	A matrix linking the prediction locations to the basis of the FEM approximation of the latent model.
Y	A vector with the observed data, can also be a matrix where the columns are observations of independent replicates of u.

`sigma.e` The standard deviation of the Gaussian measurement noise. Put to zero if the model does not have measurement noise.

`compute.variances` Set to also TRUE to compute the kriging variances.

`...` further arguments passed to or from other methods.

Value

A list with elements

`mean` The kriging predictor (the posterior mean of u|Y).

`variance` The posterior variances (if computed).

Examples

```
#Sample a Gaussian Matern process on R using a rational approximation
kappa <- 10
sigma <- 1
nu <- 0.8
sigma.e <- 0.3

#create mass and stiffness matrices for a FEM discretization
x <- seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)

#compute rational approximation
op <- matern.operators(kappa = kappa, sigma = sigma,
                       nu = nu, G=fem$G, C = fem$C, d = 1)

#Sample the model
u <- simulate(op)

#Create some data
obs.loc <- runif(n = 10, min = 0, max = 1)
A <- rSPDE.A1d(x, obs.loc)
Y <- as.vector(A%*%u + sigma.e*rnorm(10))

#compute kriging predictions at the FEM grid
A.krig <- rSPDE.A1d(x, x)
u.krig <- predict(op, A = A, Aprd = A.krig, Y = Y, sigma.e = sigma.e,
                  compute.variances= TRUE)

plot(obs.loc, Y, ylab = "u(x)", xlab = "x", main = "Data and prediction",
     ylim = c(min(u.krig$mean - 2*sqrt(u.krig$variance)),
              max(u.krig$mean + 2*sqrt(u.krig$variance))))
lines(x, u.krig$mean)
lines(x, u.krig$mean + 2*sqrt(u.krig$variance), col = 2)
lines(x, u.krig$mean - 2*sqrt(u.krig$variance), col = 2)
```

require.nowarnings — *Warnings free loading of add-on packages*

Description

Turn off all warnings for require(), to allow clean completion of examples that require unavailable Suggested packages.

Usage

```
require.nowarnings(package, lib.loc = NULL, character.only = FALSE)
```

Arguments

`package`	The name of a package, given as a character string.
`lib.loc`	a character vector describing the location of R library trees to search through, or `NULL`. The default value of `NULL` corresponds to all libraries currently known to `.libPaths()`. Non-existent library trees are silently ignored.
`character.only`	a logical indicating whether package can be assumed to be a character string.

Details

`require(package)` acts the same as `require(package,quietly = TRUE)` but with warnings turned off. In particular, no warning or error is given if the package is unavailable. Most cases should use `requireNamespace(package,quietly = TRUE)` instead, which doesn't produce warnings.

Value

`require.nowarnings` returns (invisibly) `TRUE` if it succeeds, otherwise `FALSE`

See Also

`require`

Examples

```
## This should produce no output:
if (require.nowarnings(nonexistent)) {
  message("Package loaded successfully")
}
```

`rSPDE` *Rational approximations of fractional SPDEs.*

Description

`rSPDE` is used for approximating fractional elliptic SPDEs

$$L^\beta u(s) = W$$

, where L is a differential operator and $\beta > 0$ is a general fractional power.

Details

The approximation is based on a rational approximation of the fractional operator, and allows for computationally efficient inference and simulation.

The main function for computing the rational operators is `fractional.operators`, and the following simplified interfaces are available

- `matern.operators` Compuation of operators for random fields with stationary Matern covariance functions
- `spde.matern.operators` Compuation of operators for random fields with defined as solutions to a possibly non-stationary Matern-type SPDE model.

Basic statistical operations such as likelihood evaluations (see `rSPDE.loglike`) and kriging predictions (see `predict.rSPDEobj`) using the fractional approximations are also implemented.

For illustration purposes, the package contains a simple FEM implementation for models on R. For spatial models, the FEM implementation in the `R-INLA` package is recommended.

For a more detailed introduction to the package, see the rSPDE Vignette.

`rSPDE.A1d` *Observation matrix for finite element discretization on R*

Description

A finite element discretization on R can be written as $u(s) = \sum_i^n u_i \varphi_i(s)$ where $\varphi_i(s)$ is a piecewise linear "hat function" centered at location x_i. This function computes an $m \times n$ matrix A that links the basis function in the expansion to specified locations $s = (s_1, \ldots, s_m)$ in the domain through $A_ij = \varphi_j(s_i)$.

Usage

```
rSPDE.A1d(x, loc)
```

Arguments

`x`	The locations of the nodes in the FEM discretization.
`loc`	The locations $(s_1, \ldots, s_m)$

Value

The sparse matrix `A`.

Author(s)

David Bolin `<davidbolin@gmail.com>`

See Also

`rSPDE.fem1d`

Examples

```
#create mass and stiffness matrices for a FEM discretization on [0,1]
x = seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)

#create the observation matrix for some locations in the domain
obs.loc <- runif(n = 10, min = 0, max = 1)
A <- rSPDE.A1d(x, obs.loc)
```

`rSPDE.fem1d` *Finite element calculations for problems on R*

Description

This function computes mass and stiffness matrices for a FEM approximation on R, assuming Neumann boundary conditions. These matrices are needed when discretizing the operators in rational approximations.

Usage

```
rSPDE.fem1d(x)
```

Arguments

`x`	Locations of the nodes in the FEM approximation.

Value

The function returns a list with the following elements

`G`	The stiffness matrix.
`C`	The mass matrix.

Author(s)

David Bolin <davidbolin@gmail.com>

See Also

rSPDE.A1d

Examples

```
#create mass and stiffness matrices for a FEM discretization on [0,1]
x = seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)
```

---

rSPDE.loglike *Log-likelihood function for latent Gaussian fractional SPDE model*

Description

This function evaluates the log-likelihood function for a fractional SPDE model $L^\beta u(s) = W$ that is observed under Gaussian measurement noise: $Y_i = u(s_i) + \epsilon_i$, where ϵ_i are iid mean-zero Gaussian variables.

Usage

```
rSPDE.loglike(obj, Y, A, sigma.e)
```

Arguments

obj	The rational SPDE approximation, computed using fractional.operators, matern.operators, or spde.matern.operators.
Y	The observations, either a vector or a matrix where the columns correspond to independent replicates of observations.
A	An observation matrix that links the measurement location to the finite elemen basis.
sigma.e	The standard deviation of the measurement noise.

Value

The log-likelihood value.

Note

This example below shows how the function can be used to evaluate the likelihood of a latent Matern model. Se matern.loglike for an example of how this can be used for maximum likelihood estimation.

See Also

matern.loglike, spde.matern.loglike

Examples

```
#Sample a Gaussian Matern process on R using a rational approximation
kappa = 10
sigma = 1
nu = 0.8
sigma.e = 0.3

#create mass and stiffness matrices for a FEM discretization
x = seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)

#compute rational approximation
op <- matern.operators(kappa = kappa, sigma = sigma, nu = nu,
                       G = fem$G, C = fem$C, d = 1)

#Sample the model
u <- simulate(op)

#Create some data
obs.loc <- runif(n = 10, min = 0, max = 1)
A <- rSPDE.A1d(x, obs.loc)
Y = as.vector(A%*%u + sigma.e*rnorm(10))

#compute log-likelihood of the data
lik1 <- rSPDE.loglike(op, Y, A, sigma.e)
cat(lik1)
```

---

simulate *Simulation of a fractional SPDE using a rational SPDE approximation*

Description

The function samples a Gaussian random field based on a pre-computed rational SPDE approximation.

Usage

```
simulate(object, nsim)

## S3 method for class 'rSPDEobj'
simulate(object, nsim = 1)
```

Arguments

object	The rational SPDE approximation, computed using fractional.operators, matern.operators, or spde.matern.operators.
nsim	The number of simulations.

Value

A matrix with the n samples as columns.

Examples

```
#Sample a Gaussian Matern process on R using a rational approximation
kappa <- 10
sigma <- 1
nu <- 0.8

#create mass and stiffness matrices for a FEM discretization
x <- seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)

#compute rational approximation
op <- matern.operators(kappa = kappa, sigma = sigma,
                       nu = nu, G=fem$G, C=fem$C, d = 1)

#Sample the model and plot the result
Y <- simulate(op)
plot(x, Y, type = "l", ylab = "u(x)", xlab = "x")
```

---

spde.matern.loglike *Log-likelihood for a latent Gaussian Matern SPDE model using a rational SPDE approximation*

Description

This function evaluates the log-likelihood function for observations of a Gaussian process defined as the solution to the SPDE

$$(\kappa(s) - \Delta)^{\beta}(\tau(s)u(s)) = W$$

,

Usage

```
spde.matern.loglike(kappa, tau, nu, sigma.e, Y, G, C, A, d = 2, m = 1)
```

Arguments

- `kappa` Vector with the, possibly spatially varying, range parameter evaluated at the locations of the mesh used for the finite element discretization of the SPDE.
- `tau` Vector with the, possibly spatially varying, precision parameter evaluated at the locations of the mesh used for the finite element discretization of the SPDE.
- `nu` Shape parameter of the covariance function, related to β through the equation $\beta = (\nu + d/2)/2$.
- `sigma.e` The standard deviation of the measurement noise.
- `Y` The observations, either a vector or a matrix where the columns correspond to independent replicates of observations.
- `G` The stiffness matrix of a finite element discretization of the domain.
- `C` The mass matrix of a finite element discretization of the domain.
- `A` A matrix linking the measurement locations to the basis of the FEM approximation of the latent model.
- `d` The dimension of the domain. The default value is 2.
- `m` The order of the rational approximation, which needs to be a positive integer. The default value is 1.

Details

The observations are assumed to be generated as $Y_i = u(s_i) + \epsilon_i$, where ϵ_i are iid mean-zero Gaussian variables. The latent model is approximated using a rational approximation of the fractional SPDE model.

Value

The log-likelihood value.

See Also

`matern.loglike`, `rSPDE.loglike`.

Examples

```
#this example illustrates how the function can be used for maximum likelihood estimation
set.seed(1)
#Sample a Gaussian Matern process on R using a rational approximation
sigma.e = 0.1
n.rep = 10
n.obs = 100
n.x = 51

#create mass and stiffness matrices for a FEM discretization
x = seq(from = 0, to = 1, length.out = n.x)
fem <- rSPDE.fem1d(x)

tau = rep(0.5,n.x)
```

```
nu = 0.8
kappa = rep(1,n.x)

#compute rational approximation
op <- spde.matern.operators(kappa = kappa, tau = tau, nu = nu,
                            G = fem$G, C = fem$C, d = 1)

#Sample the model
u <- simulate(op, n.rep)

#Create some data
obs.loc <- runif(n = n.obs, min = 0, max = 1)
A <- rSPDE.A1d(x, obs.loc)
noise <- rnorm(n.obs*n.rep)
dim(noise) <- c(n.obs, n.rep)
Y = as.matrix(A%*%u + sigma.e*noise)

#define negative likelihood function for optimization using matern.loglike
mlik <- function(theta, Y, G, C, A){
return(-spde.matern.loglike(rep(exp(theta[1]),n.x), rep(exp(theta[2]),n.x),
                            exp(theta[3]), exp(theta[4]),
                            Y = Y, G = G, C = C, A = A, d = 1))
}

#' #The parameters can now be estimated by maximizing mlik with optim

#Choose some reasonable starting values depending on the size of the domain
theta0 = log(c(sqrt(8), 1/sqrt(var(c(Y))), 0.9, 0.01))

#run estimation and display the results
theta <- optim(theta0, mlik, Y = Y, G = fem$G, C = fem$C, A = A)

print(data.frame(kappa = c(kappa[1],exp(theta$par[1])), tau = c(tau[1],exp(theta$par[2])),
                 nu = c(nu,exp(theta$par[3])), sigma.e = c(sigma.e,exp(theta$par[4])),
                 row.names = c("Truth","Estimates")))
```

`spde.matern.operators` *Rational approximations of non-stationary Gaussian SPDE Matern random fields*

Description

`spde.matern.operators` is used for computing a rational SPDE approximation of a Gaussian random fields on R^d defined as a solution to the SPDE

$$(\kappa(s) - \Delta)^\beta (\tau(s) u(s)) = W$$

Usage

```
spde.matern.operators(kappa, tau, nu, G, C, d, m = 1)
```

Arguments

`kappa`	Vector with the, possibly spatially varying, range parameter evaluated at the locations of the mesh used for the finite element discretization of the SPDE.
`tau`	Vector with the, possibly spatially varying, precision parameter evaluated at the locations of the mesh used for the finite element discretization of the SPDE.
`nu`	Shape parameter of the covariance function, related to β through the equation $\beta = (\nu + d/2)/2$.
`G`	The stiffness matrix of a finite element discretization of the domain of interest.
`C`	The mass matrix of a finite element discretization of the domain of interest.
`d`	The dimension of the domain.
`m`	The order of the rational approximation, which needs to be a positive integer. The default value is 1.

Details

The approximation is based on a rational approximation of the fractional operator $(\kappa(s)^2 - \Delta)^\beta$, where $\beta = (\nu + d/2)/2$. This results in an approximate model on the form

$$P_l u(s) = P_r W,$$

where $P_j = p_j(L)$ are non-fractional operators defined in terms of polynomials p_j for $j = l, r$. The order of p_r is given by `m` and the order of p_l is $m + m_\beta$ where m_β is the integer part of β if $\beta > 1$ and $m_\beta = 1$ otherwise.

The discrete approximation can be written as $u = P_r x$ where $x \sim N(0, Q^{-1})$ and $Q = P_l^T C^{-1} P_l$. Note that the matrices P_r and Q may be be ill-conditioned for $m > 1$. In this case, the metehods in `operator.operations` should be used for operations involving the matrices, since these methods are more numerically stable.

Value

`spde.matern.operators` returns an object of class "rSPDEobj. This object contains the quantities listed in the output of `fractional.operators` as well as the smoothness parameter ν.

Author(s)

David Bolin `<davidbolin@gmail.com>`

See Also

`fractional.operators`, `spde.matern.operators`

Examples

```
#Sample non-stationary Matern field on R
tau <- 1
nu <- 0.8

#create mass and stiffness matrices for a FEM discretization
```

```
x <- seq(from = 0, to = 1, length.out = 101)
fem <- rSPDE.fem1d(x)

 #define a non-stationary range parameter
 kappa <- seq(from = 2, to = 20, length.out = length(x))

#compute rational approximation
op <- spde.matern.operators(kappa = kappa, tau = tau, nu = nu,
                            G = fem$G, C = fem$C, d = 1)

#sample the field
u <- simulate(op)

#plot the sample
plot(x, u, type = "l", ylab = "u(s)", xlab = "s")
```

`summary.rSPDEobj` *Summarise excurobj objects*

Description

Summary method for class "rSPDEobj"

Usage

```
## S3 method for class 'rSPDEobj'
summary(object, ...)

## S3 method for class 'summary.rSPDEobj'
print(x, ...)

## S3 method for class 'rSPDEobj'
print(x, ...)
```

Arguments

`object`	an object of class "rSPDEobj", usually, a result of a call to `fractional.operators`, `matern.operators`, or `spde.matern.operators`.
`...`	further arguments passed to or from other methods.
`x`	an object of class "summary.rSPDEobj", usually, a result of a call to `summary.rSPDEobj`.

Index