# Package: nimblewomble (via r-universe)

May 13, 2026

**Type** Package

**Title** Bayesian Wombling using Nimble

**Version** 0.1.0

**Description** Performs Bayesian Wombling using Nimble. For more details on wombling please see, <doi:10.1198/016214506000000041> and <doi:10.1080/01621459.2023.2177166>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** nimble, ggplot2, MBA, metR, ggspatial, sf, terra, sp, coda, methods

**RoxygenNote** 7.3.2

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libglpk-dev make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

**Repository** https://arh926.r-universe.dev

**Date/Publication** 2025-04-05 16:21:43 UTC

**RemoteUrl** https://github.com/arh926/nimblewomble

**RemoteRef** HEAD

**RemoteSha** 1292bafa0eecfc6b73ca1252d9ece10f70206f0d

## Contents

---

`curvatures_gaussian` *Posterior samples of rates of change (gradients and curvatures) for the Matern kernel with $\nu \to \infty$ producing the squared exponential kernel.*

---

## Description

For internal use only.

## Usage

```
curvatures_gaussian(dists.1, dists.2, dists.3, z, phi, sigma2)
```

## Arguments

| | |
|---|---|
| `dists.1` | distance matrix generated from coordinates |
| `dists.2` | distance of grid from coordinates |
| `dists.3` | delta = coordinate - grid |
| `z` | posterior samples of $Z(s)$ |
| `phi` | posterior samples of $\phi$ |
| `sigma2` | posterior samples of $\sigma^2$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::sprates()
 CG = compileNimble(curvatures_gaussian)
 sprates = CG(dists.1 = distM,
              dists.2 = dist.2,
              dists.3 = dist.3,
              z = z,
              phi = phi,
              sigma2 = sigma2)

## End(Not run)
```

---

`curvatures_matern2` *Posterior samples of rates of change (gradients and curvatures) for the Matern kernel with* $\nu = 5/2$

---

## Description

For internal use only.

## Usage

```
curvatures_matern2(dists.1, dists.2, dists.3, z, phi, sigma2)
```

## Arguments

| | |
|---|---|
| `dists.1` | distance matrix generated from coordinates |
| `dists.2` | distance of grid from coordinates |
| `dists.3` | delta = coordinate - grid |
| `z` | posterior samples of $Z(s)$ |
| `phi` | posterior samples of $\phi$ |
| `sigma2` | posterior samples of $\sigma^2$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::sprates()
 CM2 = compileNimble(curvatures_matern2)
 sprates = CM2(dists.1 = distM,
               dists.2 = dist.2,
               dists.3 = dist.3,
               z = z,
               phi = phi,
               sigma2 = sigma2)

## End(Not run)
```

---

gamma_int *Incomplete Gamma Function*

---

## Description

For internal use only. Use integration as a limit of a sum to numerically compute the incomplete gamma integral

## Usage

```
gamma_int(x, a, b)
```

## Arguments

| | |
|---|---|
| x | gamma quantile |
| a | shape parameter |
| b | scale parameter |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage in nimblewomble::wombling_matern1(...) or,
# nimblewomble::wombling_matern2(...)
gamma_int(x = 1, a = 1, b = 1/sqrt(3))

## End(Not run)
```

`gamma1.mcov1` *Cross-covariance terms for the posterior distribution of wombling measures for Matern* $\nu = 3/2$.

## Description

For internal use only. Performs one-dimensional quadrature using integral as a limit of a sum.

## Usage

```
gamma1.mcov1(coords, t, u, s0, phi)
```

## Arguments

| | |
|---|---|
| `coords` | coordinates |
| `t` | value of t |
| `u` | vector of u |
| `s0` | starting point on curve $s_0$ |
| `phi` | posterior sample of $\phi$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
######################
# Internal use only #
######################
# Example usage inside nimblewomble::wombling_matern1(...)
gamma1.mcov1(coords = coords[1:ncoords, 1:2], t = tvec[j],
             u = umat[j, 1:2], s0 = curve[j, 1:2], phi = phi[i])

## End(Not run)
```

`gamma1n2.gauss` *Cross-covariance terms for the posterior distribution of wombling measures for Matern $\nu \to \infty$, the squared exponential kernel.*

## Description

For internal use only. Performs one-dimensional quadrature using integral as a limit of a sum.

## Usage

```
gamma1n2.gauss(coords, t, u, s0, phi)
```

## Arguments

| | |
|---|---|
| `coords` | coordinates |
| `t` | value of t |
| `u` | vector of u |
| `s0` | starting point on curve $s_0$ |
| `phi` | posterior sample of $\phi$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
######################
# Internal use only #
######################
# Example usage inside nimblewomble::wombling_gaussian(...)
gamma1n2.gauss(coords = coords[1:ncoords, 1:2], t = tvec[j],
               u = umat[j, 1:2], s0 = curve[j, 1:2], phi = phi[i])

## End(Not run)
```

`gamma1n2.mcov2` *Cross-covariance terms for the posterior distribution of wombling measures for Matern $\nu = 5/2$, the squared exponential kernel.*

## Description

For internal use only. Performs one-dimensional quadrature using integral as a limit of a sum.

## Usage

```
gamma1n2.mcov2(coords, t, u, s0, phi)
```

## Arguments

| | |
|---|---|
| `coords` | coordinates |
| `t` | value of t |
| `u` | vector of u |
| `s0` | starting point on curve $s_0$ |
| `phi` | posterior sample of $\phi$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
######################
# Internal use only #
######################
# Example usage inside nimblewomble::wombling_matern2(...)
gamma1n2.mcov2(coords = coords[1:ncoords, 1:2], t = tvec[j],
               u = umat[j, 1:2], s0 = curve[j, 1:2], phi = phi[i])

## End(Not run)
```

## gaussian *Squared Exponential Covariance kernel*

### Description

Computes the Matern covariance matrix with fractal parameter $\nu \to \infty$. Has the option to compute $\Sigma_{d\times d} + \tau^2 I_d$.

### Usage

```
gaussian(dists, phi, sigma2, tau2)
```

### Arguments

| | |
|---|---|
| dists | distance matrix |
| phi | spatial range |
| sigma2 | spatial variance |
| tau2 | nugget variance |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

### Examples

```
## Not run:
######################
# Internal use only #
######################
# Used across multiple functions
# Example usage
gaussian(dists = dists[1:ncoords, 1:ncoords], phi = phi[i], sigma2 = 1, tau2 = 0)

## End(Not run)
```

## gp_fit *Fit a Gaussian process*

### Description

Fits a Gaussian process with the choice of three kernels. Uses ‘nimble‘ to generate posterior samples.

## Usage

```
gp_fit(
  coords = NULL,
  y = NULL,
  X = NULL,
  kernel = c("matern1", "matern2", "gaussian"),
  niter = NULL,
  nburn = NULL
)
```

## Arguments

| | |
|---|---|
| coords | spatial coordinats (supply as a matrix) |
| y | response |
| X | covariates (supply as a matrix without the intercept) |
| kernel | choice of kernel; must be one of "matern1", "matern2", "gaussian" |
| niter | number of iterations |
| nburn | burn-in |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
require(nimble)
require(nimblewomble)

set.seed(1)
# Generated Simulated Data
N = 1e2
tau = 1
coords = matrix(runif(2 * N, -10, 10), ncol = 2)
colnames(coords) = c("x", "y")
y = rnorm(N, mean = 20 * sin(sqrt(coords[, 1]^2  + coords[, 2]^2)), sd = tau)
# Posterior samples for theta
mc_sp = gp_fit(coords = coords, y = y, kernel = "matern2")
mc_sp$estimates

## End(Not run)
```

`gradients_matern1` *Posterior samples of rates of change (gradients) for the Matern kernel with $\nu = 3/2$*

## Description

For internal use only.

## Usage

```
gradients_matern1(dists.1, dists.2, dists.3, z, phi, sigma2)
```

## Arguments

| | |
|---|---|
| `dists.1` | distance matrix generated from coordinates |
| `dists.2` | distance of grid from coordinates |
| `dists.3` | delta = coordinate - grid |
| `z` | posterior samples of $Z(s)$ |
| `phi` | posterior samples of $\phi$ |
| `sigma2` | posterior samples of $\sigma^2$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::sprates()
 GM1 = compileNimble(gradients_matern1)
 sprates = GM!(dists.1 = distM,
               dists.2 = dist.2,
               dists.3 = dist.3,
               z = z,
               phi = phi,
               sigma2 = sigma2)

## End(Not run)
```

## maternccov1 *Matern Covariance kernel with $\nu = 3/2$*

### Description

Computes the Matern covariance matrix with fractal parameter $\nu = 3/2$. Has the option to compute $\Sigma_{d\times d} + \tau^2 I_d$.

### Usage

```
maternccov1(dists, phi, sigma2, tau2)
```

### Arguments

| | |
|---|---|
| dists | distance matrix |
| phi | spatial range |
| sigma2 | spatial variance |
| tau2 | nugget variance |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

### Examples

```
## Not run:
######################
# Internal use only #
######################
# Used across multiple functions
# Example usage
maternccov1(dists = dists[1:ncoords, 1:ncoords], phi = phi[i], sigma2 = 1, tau2 = 0)

## End(Not run)
```

## maternccov2 *Matern Covariance kernel with $\nu = 5/2$*

### Description

Computes the Matern covariance matrix with fractal parameter $\nu = 5/2$. Has the option to compute $\Sigma_{d\times d} + \tau^2 I_d$.

### Usage

```
materncov2(dists, phi, sigma2, tau2)
```

### Arguments

| | |
|---|---|
| dists | distance matrix |
| phi | spatial range |
| sigma2 | spatial variance |
| tau2 | nugget variance |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

### Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Used across multiple functions
# Example usage
materncov2(dists = dists[1:ncoords, 1:ncoords], phi = phi[i], sigma2 = 1, tau2 = 0)

## End(Not run)
```

---

## pnorm_nimble *Computes the Cumulative Distribution Function for the standard Gaussian probability distribution*

---

### Description

For internal use only.

### Usage

```
pnorm_nimble(x)
```

### Arguments

| | |
|---|---|
| x | standard Gaussian quantile |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::wombling_gaussian()
pnorm_nimble(sqrt(2) * phi[i] * tvec[j]) - 1)

## End(Not run)
```

---

`significance` *Determines significance for posterior estimates*

---

## Description

For internal use only.

## Usage

```
significance(data_frame = NULL)
```

## Arguments

| | |
|---|---|
| `data_frame` | matrix consisting of median, lower and upper Confidence Interval |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::spwombling(...)
estimate.wm$sig = significance(estimate.wm)

## End(Not run)
```

sp_ggplot *Spatial Plot Function*

## Description

Spatial Plot Function

## Usage

```
sp_ggplot(
  data_frame = NULL,
  sp = FALSE,
  shape = NULL,
  legend.key.height = 0.7,
  legend.key.width = 0.4,
  text.size = 10,
  point.size = 0.7,
  clr.pt = "black",
  palette = "Spectral",
  extend = TRUE,
  title = NULL,
  bound.box = NULL
)
```

## Arguments

| | |
|---|---|
| data_frame | data frame consisting of coordinates and data |
| sp | logical parameter indicating whether to make a spatial plot |
| shape | if sp = TRUE shape file should be provided (should be an sf object) |
| legend.key.height | height of legend (defaults to .7) |
| legend.key.width | width of legend (defaults to .4) |
| text.size | size of legend text (defaults to 10) |
| point.size | size of points to be plotted (defaults to 0.7) |
| clr.pt | color of point to be plotted (defaults to black) |
| palette | (optional) color palette |
| extend | logical parameter indicating whether to extend the interpolation (defaults to TRUE) |
| title | title of the plot (defaults to NULL) |
| bound.box | bounding box for spatial maps (leave as NULL if not known) |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

### Examples

```
require(nimblewomble)

set.seed(1)
# Generated Simulated Data
N = 1e2
tau = 1
coords = matrix(runif(2 * N, -10, 10), ncol = 2)
colnames(coords) = c("x", "y")
y = rnorm(N, mean = 20 * sin(sqrt(coords[, 1]^2  + coords[, 2]^2)), sd = tau)

sp_ggplot(data_frame = data.frame(coords, z = y))
```

---

## sprates *Posterior samples for rates of change*

---

### Description

Posterior samples for rates of change

### Usage

```
sprates(
  coords = NULL,
  grid = NULL,
  model = NULL,
  kernel = c("matern1", "matern2", "gaussian")
)
```

### Arguments

| | |
|---|---|
| coords | coordinates |
| grid | grid for sampling the rates of change |
| model | posterior samples of $Z(s)$, $\phi$, $\sigma^2$ |
| kernel | choice of kernel; must be one of "matern1", "matern2", "gaussian" |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
require(nimble)
require(nimblewomble)
require(patchwork)

set.seed(1)
# Generated Simulated Data
N = 1e2
tau = 1
coords = matrix(runif(2 * N, -10, 10), ncol = 2); colnames(coords) = c("x", "y")
y = rnorm(N, mean = 20 * sin(sqrt(coords[, 1]^2  + coords[, 2]^2)), sd = tau)

# Create equally spaced grid of points
xsplit = ysplit = seq(-10, 10, by = 1)[-c(1, 21)]
grid = as.matrix(expand.grid(xsplit, ysplit), ncol = 2)
colnames(grid) = c("x", "y")

####################################
# Process for True Rates of Change #
####################################
# Gradient along x
true_sx = round(20 * cos(sqrt(grid[,1]^2 + grid[,2]^2)) *
                grid[,1]/sqrt(grid[,1]^2 + grid[,2]^2), 3)
# Gradient along y
true_sy = round(20 * cos(sqrt(grid[,1]^2 + grid[,2]^2)) *
                grid[,2]/sqrt(grid[,1]^2 + grid[,2]^2), 3)
# Curvature along x
true_sxx = round(20 * cos(sqrt(grid[,1]^2 + grid[,2]^2))/
                  sqrt(grid[,1]^2 + grid[,2]^2) -
                 20 * cos(sqrt(grid[,1]^2 + grid[,2]^2)) *
                 grid[,1]^2/(grid[,1]^2 + grid[,2]^2)^(3/2) -
                 20 * sin(sqrt(grid[,1]^2 + grid[,2]^2)) *
                 grid[,1]^2/(grid[,1]^2 + grid[,2]^2), 3)
# Mixed Curvature
true_sxy = round(-20 * (cos(sqrt(grid[,1]^2 + grid[,2]^2)) -
                 sin(sqrt(grid[,1]^2 + grid[,2]^2))) * grid[,1]
                  * grid[,2]/(grid[,1]^2 + grid[,2]^2), 3)
# Curvature along y
true_syy = round(20 * cos(sqrt(grid[,1]^2 + grid[,2]^2))/
                  sqrt(grid[,1]^2 + grid[,2]^2) -
                 20 * cos(sqrt(grid[,1]^2 + grid[,2]^2)) *
                 grid[,2]^2/(grid[,1]^2 + grid[,2]^2)^(3/2) -
                 20 * sin(sqrt(grid[,1]^2 + grid[,2]^2)) *
                 grid[,2]^2/(grid[,1]^2 + grid[,2]^2), 3)
# Create the plots
p1 = sp_ggplot(data_frame = data.frame(coords, z = y))
p2 = sp_ggplot(data_frame = data.frame(grid[-which(is.nan(true_sx)),],
               z = true_sx[-which(is.nan(true_sx))]))
p3 = sp_ggplot(data_frame = data.frame(grid[-which(is.nan(true_sy)),],
               z = true_sy[-which(is.nan(true_sy))]))
p4 = sp_ggplot(data_frame = data.frame(grid[-which(is.nan(true_sxx)),],
```

```
                z = true_sxx[-which(is.nan(true_sxx))]))
p5 = sp_ggplot(data_frame = data.frame(grid[-which(is.nan(true_sxy)),],
                z = true_sxy[-which(is.nan(true_sxy))]))
p6 = sp_ggplot(data_frame = data.frame(grid[-which(is.nan(true_syy)),],
                z = true_syy[-which(is.nan(true_syy))]))

((p1 + p2 + p3)/(p4 + p5 + p6))

##########################
# Fit a Gaussian Process #
##########################
# Posterior samples for theta
mc_sp = gp_fit(coords = coords, y = y, kernel = "matern2")
# Posterior samples for Z(s) and beta
model = zbeta_samples(y = y, coords = coords,
                      model = mc_sp$mcmc,
                      kernel = "matern2")
###################
# Rates of Change #
###################
gradients = sprates(grid = grid,
                    coords = coords,
                    model = model,
                    kernel = "matern2")
p8 = sp_ggplot(data_frame = data.frame(grid,
                z = gradients$estimate.sx[,"50%"],
                sig = gradients$estimate.sx$sig))
p9 = sp_ggplot(data_frame = data.frame(grid,
                z = gradients$estimate.sy[,"50%"],
                sig = gradients$estimate.sy$sig))
p10 = sp_ggplot(data_frame = data.frame(grid,
                  z = gradients$estimate.sxx[,"50%"],
                 sig = gradients$estimate.sxx$sig))
p11 = sp_ggplot(data_frame = data.frame(grid,
                  z = gradients$estimate.sxy[,"50%"],
                  sig = gradients$estimate.sxy$sig))
p12 = sp_ggplot(data_frame = data.frame(grid,
                z = gradients$estimate.syy[,"50%"],
                sig = gradients$estimate.syy$sig))

# compare with true estimates
((p7 + p8 + p9)/(p10 + p11 + p12))

## End(Not run)
```

---

spwombling *Posterior samples for wombling measures*

---

## Description

Posterior samples for wombling measures

## Usage

```
spwombling(
  coords = NULL,
  curve = NULL,
  model = NULL,
  kernel = c("matern1", "matern2", "gaussian")
)
```

## Arguments

| | |
|---|---|
| coords | coordinates |
| curve | coordinates of the curve for wombling |
| model | posterior samples of $Z(s)$, $\phi$, $\sigma^2$ |
| kernel | choice of kernel; must be one of "matern1", "matern2", "gaussian" |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
require(nimble)
require(nimblewomble)

set.seed(1)
# Generated Simulated Data
N = 1e2
tau = 1
coords = matrix(runif(2 * N, -10, 10), ncol = 2); colnames(coords) = c("x", "y")
y = rnorm(N, mean = 20 * sin(sqrt(coords[, 1]^2  + coords[, 2]^2)), sd = tau)

# Create equally spaced grid of points
xsplit = ysplit = seq(-10, 10, by = 1)[-c(1, 21)]
grid = as.matrix(expand.grid(xsplit, ysplit), ncol = 2)
colnames(grid) = c("x", "y")

##########################
# Fit a Gaussian Process #
##########################
# Posterior samples for theta
mc_sp = gp_fit(coords = coords, y = y, kernel = "matern2")
# Posterior samples for Z(s) and beta
model = zbeta_samples(y = y, coords = coords,
                      model = mc_sp$mcmc,
                      kernel = "matern2")
############
# Wombling #
############
```

```
# Pick any curve (contour) of your choice
# curve = your contour
tvec = sapply(1:(nrow(curve) - 1), function(x) sqrt(sum((curve[(x + 1),] - curve[x,])^2)))
umat = as.matrix(t(sapply(1:(nrow(curve) - 1), function(x) (curve[(x + 1),] - curve[x,])))/tvec)

wm = spwombling(coords = coords,
                curve = curve,
                model = model,
                kernel = "matern2")

# Total wombling measure for gradient
colSums(wm$estimate.wm.1[,-4]); colSums(wm$estimate.wm.1[,-4])/sum(tvec)
# Total wombling measure for curvature
colSums(wm$estimate.wm.2[,-4]); colSums(wm$estimate.wm.2[,-4])/sum(tvec)


# Color code points based on significance
col.pts.1 = sapply(wm$estimate.wm.1$sig, function(x){
  if(x == 1) return("green")
  else if(x == -1) return("cyan")
  else return(NA)
  })

  col.pts.2 = sapply(wm$estimate.wm.2$sig, function(x){
  if(x == 1) return("green")
  else if(x == -1) return("cyan")
  else return(NA)
  })

p13 = sp_ggplot(data_frame = data.frame(coords, y))
p14 = p13 + geom_path(curve, mapping = aes(x, y), linewidth = 2)
p15 = p13 + geom_path(curve, mapping = aes(x, y), linewidth = 2) +
geom_path(curve, mapping = aes(x, y),
          colour = c(col.pts.1, NA), linewidth = 1, na.rm = TRUE)
p16 = p13 + geom_path(curve, mapping = aes(x, y), linewidth = 2) +
            geom_path(curve, mapping = aes(x, y),
            colour = c(col.pts.2, NA), linewidth = 1, na.rm = TRUE)

p14 + (p15/p16)

###############
# True Values #
###############
truth = matrix(0, nrow = nrow(curve) - 1, ncol = 2)
rule = seq(0, 1, by = 0.01)

for(i in 1:(nrow(curve) - 1)){
  u.perp = c(umat[i, 2], - umat[i, 1])
  s0 = curve[i,]

truth.lsegment = sapply(rule * tvec[i], function(x){
  s.t = s0 + x * umat[i,]
  true_sx = 20 * cos(sqrt(s.t[1]^2 + s.t[2]^2)) * s.t[1]/
```

```
            sqrt(s.t[1]^2 + s.t[2]^2)
  true_sy = 20 * cos(sqrt(s.t[1]^2 + s.t[2]^2)) * s.t[2]/
            sqrt(s.t[1]^2 + s.t[2]^2)
  true_sx * u.perp[1] + true_sy * u.perp[2]
  })

truth[i, 1] = sum(truth.lsegment * (tvec[i]/101))

truth.lsegment = sapply(rule * tvec[i], function(x){
  s.t = s0 + x * umat[i,]
  true_sxx = 20 * cos(sqrt(s.t[1]^2 + s.t[2]^2))/sqrt(s.t[1]^2 + s.t[2]^2) -
    20 * cos(sqrt(s.t[1]^2 + s.t[2]^2)) *
            s.t[1]^2/(s.t[1]^2 + s.t[2]^2)^(3/2) -
    20 * sin(sqrt(s.t[1]^2 + s.t[2]^2)) * s.t[1]^2/(s.t[1]^2 + s.t[2]^2)
  true_sxy = -20 * (cos(sqrt(s.t[1]^2 + s.t[2]^2)) -
                sin(sqrt(s.t[1]^2 + s.t[2]^2))) *
                      s.t[1] * s.t[2]/(s.t[1]^2 + s.t[2]^2)
  true_syy = 20 * cos(sqrt(s.t[1]^2 + s.t[2]^2))/sqrt(s.t[1]^2 + s.t[2]^2) -
    20 * cos(sqrt(s.t[1]^2 + s.t[2]^2)) *
               s.t[2]^2/(s.t[1]^2 + s.t[2]^2)^(3/2) -
    20 * sin(sqrt(s.t[1]^2 + s.t[2]^2)) * s.t[2]^2/(s.t[1]^2 + s.t[2]^2)
  true_sxx * u.perp[1]^2 + 2 * true_sxy * u.perp[1] * u.perp[2] +
                true_syy * u.perp[2]^2
  })
  truth[i, 2] = sum(truth.lsegment * (tvec[i]/101))
}
true.total = colSums(truth); true.total
true.avg.total = true.total/sum(tvec); true.avg.total

## End(Not run)
```

---

`wombling_gaussian` *Posterior samples for wombling measures for the squared exponential kernel*

---

## Description

For internal use only.

## Usage

```
wombling_gaussian(coords, curve, dists, tvec, umat, z, phi, sigma2)
```

## Arguments

| | |
|---|---|
| `coords` | coordinates |
| `curve` | curve coordinates |
| `dists` | distance matrix |
| `tvec` | vector of t's |

| | |
|---|---|
| umat | matrix of u's |
| z | posterior samples of $Z(s)$ |
| phi | posterior samples of $\phi$ |
| sigma2 | posterior samples of $\sigma^2$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::spwombling(...)
WG = compileNimble(wombling_gaussian)
wmeasure = WG(coords = coords,
              curve = curve,
              dists = distM,
              tvec = tvec,
              umat = umat,
              z = z,
              phi = phi,
              sigma2 = sigma2)

## End(Not run)
```

---

wombling_matern1 *Posterior samples for wombling measures from the Matern kernel with* $\nu = 3/2$

---

## Description

For internal use only.

## Usage

```
wombling_matern1(coords, curve, dists, tvec, umat, z, phi, sigma2)
```

## Arguments

| | |
|---|---|
| coords | coordinates |
| curve | curve coordinates |
| dists | distance matrix |
| tvec | vector of t's |

| | |
|---|---|
| umat | matrix of u's |
| z | posterior samples of $Z(s)$ |
| phi | posterior samples of $\phi$ |
| sigma2 | posterior samples of $\sigma^2$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::spwombling(...)
WM1 = compileNimble(wombling_matern1)
wmeasure = WM1(coords = coords,
               curve = curve,
               dists = distM,
               tvec = tvec,
               umat = umat,
               z = z,
               phi = phi,
               sigma2 = sigma2)

## End(Not run)
```

---

# wombling_matern2 *Posterior samples for wombling measures from the Matern kernel with $\nu = 5/2$*

---

## Description

For internal use only.

## Usage

```
wombling_matern2(coords, curve, dists, tvec, umat, z, phi, sigma2)
```

## Arguments

| | |
|---|---|
| coords | coordinates |
| curve | curve coordinates |
| dists | distance matrix |
| tvec | vector of t's |

| | |
|---|---|
| umat | matrix of u's |
| z | posterior samples of $Z(s)$ |
| phi | posterior samples of $\phi$ |
| sigma2 | posterior samples of $\sigma^2$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::spwombling(...)
WM2 = compileNimble(wombling_matern2)
wmeasure = WM2(coords = coords,
               curve = curve,
               dists = distM,
               tvec = tvec,
               umat = umat,
               z = z,
               phi = phi,
               sigma2 = sigma2)

## End(Not run)
```

---

# zbeta_gaussian *Posterior samples of spatial effects and intercept for the squared exponential kernel*

---

## Description

For internal use only.

## Usage

```
zbeta_gaussian(y, dists, phi, sigma2, tau2)
```

## Arguments

| | |
|---|---|
| y | response |
| dists | distance matrix derived from coordinates |
| phi | posterior samples of $\phi$ |
| sigma2 | posterior samples of $\sigma^2$ |
| tau2 | posterior samples of $\tau^2$ |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

### Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::zbeta_samples(...)
zbG = compileNimble(zbeta_gaussian)
zb.samples = zbG(y = y, dists = dists, phi = phi, sigma2 = sigma2,
                 tau2 = tau2)

## End(Not run)
```

---

## zbeta_matern1 — *Posterior samples of spatial effects and intercept for the Matern kernel with $nu = 3/2$*

---

### Description

For internal use only.

### Usage

```
zbeta_matern1(y, dists, phi, sigma2, tau2)
```

### Arguments

| | |
|---|---|
| y | response |
| dists | distance matrix derived from coordinates |
| phi | posterior samples of $\phi$ |
| sigma2 | posterior samples of $\sigma^2$ |
| tau2 | posterior samples of $\tau^2$ |

### Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::zbeta_samples(...)
zbM1 = compileNimble(zbeta_matern1)
zb.samples = zbM1(y = y, dists = dists, phi = phi, sigma2 = sigma2,
                  tau2 = tau2)

## End(Not run)
```

---

`zbeta_matern2` *Posterior samples of spatial effects and intercept for the Matern kernel with $\nu = 5/2$*

---

## Description

For internal use only.

## Usage

```
zbeta_matern2(y, dists, phi, sigma2, tau2)
```

## Arguments

| | |
|---|---|
| `y` | response |
| `dists` | distance matrix derived from coordinates |
| `phi` | posterior samples of $\phi$ |
| `sigma2` | posterior samples of $\sigma^2$ |
| `tau2` | posterior samples of $\tau^2$ |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::zbeta_samples(...)
zbM2 = compileNimble(zbeta_matern2)
zb.samples = zbM2(y = y, dists = dists, phi = phi, sigma2 = sigma2,
                  tau2 = tau2)

## End(Not run)
```

zbeta_samples *Posterior samples of spatial effects and intercept for Matern with $nu = 3/2$*

## Description

For internal use only.

## Usage

```
zbeta_samples(
  coords = NULL,
  y = NULL,
  X = NULL,
  model = NULL,
  kernel = c("matern1", "matern2", "gaussian")
)
```

## Arguments

| | |
|---|---|
| coords | coordinates |
| y | response |
| X | covariates (supply as a matrix without intercept) |
| model | matrix of posterior samples of $\phi$, $\sigma^2$ and $\tau^2$ |
| kernel | choice of kernel; must be one of "matern1", "matern2", "gaussian" |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
require(nimble)
require(nimblewomble)

set.seed(1)
# Generated Simulated Data
N = 1e2
tau = 1
coords = matrix(runif(2 * N, -10, 10), ncol = 2); colnames(coords) = c("x", "y")
y = rnorm(N, mean = 20 * sin(sqrt(coords[, 1]^2  + coords[, 2]^2)), sd = tau)

# Posterior samples for theta
mc_sp = gp_fit(coords = coords, y = y, kernel = "matern2")
# Posterior samples for Z(s) and beta
```

```
model = zbeta_samples(y = y, coords = coords,
                      model = mc_sp$mcmc,
                      kernel = "matern2")
estimates = t(round(apply(model, 2, quantile, probs = c(0.5, 0.025, 0.975)), 3))
yfit = estimates[paste("z", 1:N, sep = ""), "50%"] + estimates["b0", "50%"]
ylow = estimates[paste("z", 1:N, sep = ""), "2.5%"] + estimates["b0", "2.5%"]
yhigh = estimates[paste("z", 1:N, sep = ""), "97.5%"] + estimates["b0", "97.5%"]
fit_frame = cbind(true = y, est = yfit, `2.5%` = ylow, `97.5%` = yhigh)
fit_frame$sig = significance(data_frame = data.frame(fit_frame[,-1]))

# Plot
sp_ggplot(data_frame = data.frame(coords, z = yfit, sig = fit_frame$sig))

## End(Not run)
```

---

`zXbeta` *Posterior samples of spatial effects and intercept for all kernels in the presence of covariates*

---

## Description

For internal use only.

## Usage

```
zXbeta(y, X, beta)
```

## Arguments

| | |
|---|---|
| `y` | response |
| `X` | covariates (supply as a matrix without intercept) |
| `beta` | posterior samples of $\beta$ (supply as a matrix) |

## Author(s)

Aritra Halder <aritra.halder@drexel.edu>,
Sudipto Banerjee <sudipto@ucla.edu>

## Examples

```
## Not run:
#####################
# Internal use only #
#####################
# Example usage inside of nimblewomble::zbeta_samples(...)
zXb = nimble::compileNimble(zXbeta)
zb.samples = zXb(y = y, X = X, beta = beta)

## End(Not run)
```

# Index