

Package: adlaplaceHgp (via r-universe)

July 17, 2026

Type Package

Title Hierarchical Gaussian Processes for 'adlaplace'

Version 0.5.2

Description Hierarchical and random-slope integrated Wiener process model terms for use with the adlaplace Laplace approximation framework. Provides hiwp, rsiwp, rsiid, rsfpoly, and rsrpoly constructors plus conditional simulation utilities for IWP-style smooths.

License MPL-2.0

Depends R (>= 4.1.0), adlaplace (>= 0.5.0)

Imports methods, stats, Matrix

Suggests GET, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Collate '000.R' 'cond_simu.R' 'hiwp.R' 'hrpoly.R' 'rsfpoly.R' 'rsiid.R' 'rsiwp.R' 'rsrpoly.R'

NeedsCompilation no

Repository <https://eborgnine.r-universe.dev>

Date/Publication 2026-07-17 22:33:33 UTC

RemoteUrl <https://github.com/eborgnine/adlaplace>

RemoteRef main

RemoteSha 96d35f802cd1c076c717dc2284fafebab64dad52

RemoteSubdir adlaplaceHgp

Contents

cond_sim_iwp	2
cond_sim_iwp_at	3
cond_sim_iwp_inputs	4
hiwp-class	4
hrpoly-class	6
rsfpoly-class	8
rsiid-class	10
rsiwp	11
rsiwp-class	12
rsrpoly-class	13
Index	16

cond_sim_iwp	<i>Conditional simulation for IWP/HIWP terms</i>
--------------	--

Description

Draw conditional simulations of the Gaussian process model components, then summarize the resulting linear predictors and group-level effect curves.

Usage

```
cond_sim_iwp(
  fit,
  model_data,
  newx = NULL,
  n = 500,
  weights = NULL,
  probs = c(0.025, 0.5, 0.975),
  probs_envelope = c(0.1, 0.9)
)
```

Arguments

fit	Output of <code>log_lik_laplace(..., deriv = TRUE)</code> .
model_data	Output of <code>model_data()</code> .
newx	Optional list of prediction data frames, one per variable. When NULL, a default prediction grid is built from term knot ranges.
n	Number of conditional draws to simulate.
weights	Optional weights used to average group-level effects.
probs	Numeric vector of probabilities used when computing quantiles.
probs_envelope	Numeric vector of probabilities for envelopes.

Value

A list with components `sim`, `quantiles`, and `envelope`.

See Also

[cond_sim_iwp_at](#), [cond_sim_iwp_inputs](#)

`cond_sim_iwp_at`
Conditional simulation for IWP/HIWP terms (flat arguments)

Description

Draw conditional simulations of IWP/HIWP/RSIWP model components from explicit inputs. Prefer [cond_sim_iwp](#) when you have a Laplace fit and `model_data` objects; use this function for custom pipelines.

Usage

```
cond_sim_iwp_at(
  terms,
  random_info,
  beta,
  gamma_mode,
  half_H_inv,
  newx = NULL,
  n = 500,
  weights = NULL,
  probs = c(0.025, 0.5, 0.975),
  probs_envelope = c(0.1, 0.9)
)
```

Arguments

<code>terms</code>	Named list of model term objects from <code>model_data()</code> \$ <code>terms</code> .
<code>random_info</code>	Data frame from <code>model_data()</code> \$ <code>data</code> \$ <code>info</code> \$ <code>gamma</code> with columns <code>gamma_label</code> , <code>term</code> , <code>model</code> , and <code>by</code> .
<code>beta</code>	Named numeric vector of fixed-effect MLEs, or a data frame with columns <code>beta_label</code> and <code>mle</code> .
<code>gamma_mode</code>	Named numeric vector of random-effect modes at the Laplace inner optimum, or a data frame with columns <code>gamma_label</code> and <code>mode</code> .
<code>half_H_inv</code>	Matrix $H^{-1/2}$ for inner random effects; from laplace_half_H_inv (<code>laplace</code>).
<code>newx</code>	Optional list of prediction data frames, one per variable. When <code>NULL</code> , a default grid is built from term knot ranges.
<code>n</code>	Number of conditional draws.
<code>weights</code>	Optional group weights (named vector or list by variable).
<code>probs</code>	Quantile levels for summaries.
<code>probs_envelope</code>	Envelope probability levels (requires GET).

Value

List with simulated curves, quantiles, and envelopes.

cond_sim_iwp_inputs	<i>Build inputs for conditional IWP simulation</i>
---------------------	--

Description

Extracts flat arguments for `cond_sim_iwp_at` from the output of `log_lik_laplace` and `model_data`.

Usage

```
cond_sim_iwp_inputs(laplace, model_data)
```

Arguments

laplace	Output of <code>log_lik_laplace(..., deriv = TRUE)</code> .
model_data	Output of <code>model_data()</code> .

Value

A list with terms, random_info, beta, gamma_mode, and half_H_inv.

hiwp-class	<i>Hierarchical Integrated Wiener Process Model Term</i>
------------	--

Description

Creates and manages hierarchical integrated Wiener process (HIWP) model terms.

Usage

```
hiwp(
  x,
  p = 2,
  ref_value = 0,
  knots,
  by,
  init = NULL,
  lower = NULL,
  upper = NULL,
  parscale = NULL,
  boundary_is_random = TRUE,
  include_poly = TRUE,
  include_global = TRUE
```

```

)

## S4 method for signature 'hiwp'
design(term, data)

## S4 method for signature 'hiwp'
precision(term, data)

## S4 method for signature 'hiwp'
theta_info(term)

## S4 method for signature 'hiwp'
beta_info(term, data)

## S4 method for signature 'hiwp'
random_info(term, data)

```

Arguments

x	Variable name
p	Order of the integrated Wiener process (default: 2)
ref_value	Reference value for the basis
knots	Vector of knot locations
by	Grouping variable for hierarchical structure
init	Initial values for theta parameters
lower	Lower bounds for theta parameters
upper	Upper bounds for theta parameters
parscale	Parameter scales for optimization
boundary_is_random	Whether boundary should be treated as random
include_poly	Whether to include polynomial terms
include_global	Whether to include global component
term	An hiwp term object.
data	A data frame containing the variables used in the term.

Details

Hierarchical Integrated Wiener Process Model Term

Value

A list containing hiwp term object and optionally related terms

A design matrix for the HIWP term.

A precision matrix for the HIWP term.

A data frame containing theta parameter information for the HIWP term.

NULL (HIWP terms don't have beta parameters).

A data frame containing random effects information for the HIWP term.

Methods

The following methods are available for hiwp objects:

`design(term, data)` Creates design matrix for HIWP term

`precision(term, data)` Creates precision matrix for HIWP term

`theta_info(term)` Extracts theta parameter information

`beta_info(term, data)` Extracts beta parameter information

`random_info(term, data)` Extracts random effects information

Examples

```
# Example usage:
# knots <- seq(0, 1, length.out = 5)
# hiwp_term <- hiwp(x = "age", knots = knots, by = "group")
```

hrpoly-class

Hierarchical Random Polynomial Term Constructor

Description

Creates and manages hierarchical random polynomial model terms.

Creates a hierarchical random polynomial model term.

Usage

```
hrpoly(
  x,
  p = 1,
  ref_value = 0,
  by,
  init = NULL,
  lower = NULL,
  upper = NULL,
  parscale = NULL
)

## S4 method for signature 'hrpoly'
design(term, data)

## S4 method for signature 'hrpoly'
precision(term, data)
```

```
## S4 method for signature 'hrpoly'
theta_info(term)

## S4 method for signature 'hrpoly'
beta_info(term, data)

## S4 method for signature 'hrpoly'
random_info(term, data)
```

Arguments

x	Variable name.
p	Polynomial degree (default: 1).
ref_value	Reference value for the polynomial.
by	Grouping variable for hierarchical structure.
init	Initial value for theta parameter.
lower	Lower bound for theta parameter.
upper	Upper bound for theta parameter.
parscale	Parameter scale for optimization.
term	An hrpoly term object.
data	A data frame containing the variables used in the term.

Value

An hrpoly term object.

A design matrix for the hrpoly term.

A precision matrix for the hrpoly term.

A data frame containing theta parameter information for the hrpoly term.

NULL (hrpoly terms don't have beta parameters).

A data frame containing random effects information for the hrpoly term.

Methods

The following methods are available for hrpoly objects:

```
design(term, data) Creates design matrix for hrpoly term
precision(term, data) Creates precision matrix for hrpoly term
theta_info(term) Extracts theta parameter information
beta_info(term, data) Extracts beta parameter information
random_info(term, data) Extracts random effects information
```

Examples

```
# Create an hrpoly term for variable 'age' with grouping by 'site'
term <- hrpoly(x = "age", p = 2, ref_value = 0, by = "site")
str(term)
# Create sample data
set.seed(42)
dat <- data.frame(
  age = rep(seq(0, 80, by = 10), each = 3),
  site = rep(c("A", "B", "C"), times = 9)
)
by_group(term@by@term, dat)
# Compute the design matrix
design_mat <- design(term, dat)
print(design_mat)
```

rsfpoly-class

Random Slope Fixed Polynomial Model Term

Description

Creates a fixed polynomial model term for random slope models.

Usage

```
rsfpoly(
  x,
  mult,
  p = 2,
  ref_value = 0,
  ref_mult = 0,
  init = NULL,
  lower = NULL,
  upper = NULL,
  parscale = NULL
)

## S4 method for signature 'rsfpoly'
design(term, data)

## S4 method for signature 'rsfpoly'
precision(term, data)

## S4 method for signature 'rsfpoly'
theta_info(term)

## S4 method for signature 'rsfpoly'
beta_info(term, data)
```



```
## S4 method for signature 'rsfpoly'
random_info(term, data)
```

Arguments

x	Variable name.
mult	Variable to multiply the polynomial by.
p	Polynomial degree (default: 2).
ref_value	Reference value for the polynomial.
ref_mult	Reference value for the covariate.
init	Initial values for beta parameters.
lower	Lower bounds for beta parameters.
upper	Upper bounds for beta parameters.
parscale	Parameter scales for optimization.
term	A rsfpoly term object
data	A data frame containing the term variables

Value

A rsfpoly term object.

A design matrix for the random slope fixed polynomial term, or NULL if p.order is 0

NULL (fixed effects don't have precision matrices)

NULL (fixed effects don't have theta parameters)

A data frame containing beta parameter information for the random slope fixed polynomial term

NULL (fixed effects don't have random effects information)

Functions

- `design(rsfpoly)`: Creates design matrix for rsfpoly term
- `precision(rsfpoly)`: Creates precision matrix for rsfpoly term
- `theta_info(rsfpoly)`: Extracts theta parameter information for rsfpoly term
- `beta_info(rsfpoly)`: Extracts beta parameter information for rsfpoly term
- `random_info(rsfpoly)`: Extracts random effects information for rsfpoly term

Methods

The following methods are available for rsfpoly objects:

`design(term, data)` Creates design matrix for rsfpoly term

`precision(term, data)` Creates precision matrix for rsfpoly term

`theta_info(term)` Extracts theta parameter information

`beta_info(term, data)` Extracts beta parameter information

`random_info(term, data)` Extracts random effects information

rsiid-class	<i>independent Random Slope</i>
-------------	---------------------------------

Description

Creates an iid random slope model term

Usage

```
rsiid(
  x,
  mult,
  ref_mult = 0,
  init = NULL,
  lower = NULL,
  upper = NULL,
  parscale = NULL
)

## S4 method for signature 'rsiid'
design(term, data)

## S4 method for signature 'rsiid'
precision(term, data)

## S4 method for signature 'rsiid'
theta_info(term)

## S4 method for signature 'rsiid'
beta_info(term, data)

## S4 method for signature 'rsiid'
random_info(term, data)
```

Arguments

x	Variable name.
mult	Variable to multiply the polynomial by.
ref_mult	Reference value for the covariate.
init	Initial values for beta parameters.
lower	Lower bounds for beta parameters.
upper	Upper bounds for beta parameters.
parscale	Parameter scales for optimization.
term	A rsiid term object
data	A data frame containing the term variables

Value

A rsiid term object.

A design matrix for the random slope polynomial term, or NULL if p.order is 0

A precision matrix for the random slope polynomial term

A data frame containing theta parameter information for the random slope term

NULL (random slope polynomial terms don't have beta parameters)

A data frame containing random effects information for the random slope polynomial term

Functions

- `design(rsiid)`: Creates design matrix for rsiid term
- `precision(rsiid)`: Creates precision matrix for rsiid term
- `theta_info(rsiid)`: Extracts theta parameter information for rsiid term
- `beta_info(rsiid)`: Extracts beta parameter information for rsiid term
- `random_info(rsiid)`: Extracts random effects information for rsiid term

Methods

The following methods are available for rsiid objects:

`design(term, data)` Creates design matrix for rsiid term

`precision(term, data)` Creates precision matrix for rsiid term

`theta_info(term)` Extracts theta parameter information

`beta_info(term, data)` Extracts beta parameter information

`random_info(term, data)` Extracts random effects information

rsiwp

Random Slope Integrated Wiener Process Term Constructor

Description

Creates a random slope integrated Wiener process (RSIWP) model term.

Usage

```
rsiwp(
  x,
  mult,
  p = 2,
  ref_value = 0,
  ref_mult = 0,
  knots,
  init = NULL,
```

```

    lower = NULL,
    upper = NULL,
    parscale = NULL,
    boundary_is_random = TRUE,
    include_poly = TRUE,
    include_linear = TRUE
  )

```

Arguments

x	Variable name.
mult	Variable to multiply the IWP by.
p	Order of the integrated Wiener process (default: 2).
ref_value	Reference value for the basis.
ref_mult	Reference value for the covariate.
knots	Vector of knot locations.
init	Initial values for theta parameters.
lower	Lower bounds for theta parameters.
upper	Upper bounds for theta parameters.
parscale	Parameter scales for optimization.
boundary_is_random	Whether boundary should be treated as random.
include_poly	Whether to include polynomial terms.
include_linear	Whether to include a linear (fixed) term in mult.

Value

A list containing the rsiwp term object and optionally polynomial terms.

rsiwp-class

Random Slope Integrated Wiener Process Term

Description

Creates and manages random slope integrated Wiener process (RSIWP) model terms.

Usage

```

## S4 method for signature 'rsiwp'
design(term, data)

## S4 method for signature 'rsiwp'
precision(term, data)

```

```
## S4 method for signature 'rsiwp'
theta_info(term)

## S4 method for signature 'rsiwp'
beta_info(term, data)

## S4 method for signature 'rsiwp'
random_info(term, data)
```

Arguments

term	An rsiwp term object
data	A data frame containing the term variable

Functions

- `design(rsiwp)`: Creates design matrix for RSIWP term
- `precision(rsiwp)`: Creates precision matrix for RSIWP term
- `theta_info(rsiwp)`: Extracts theta parameter information for RSIWP term
- `beta_info(rsiwp)`: Extracts beta parameter information for RSIWP term
- `random_info(rsiwp)`: Extracts random effects information for RSIWP term

Methods

The following methods are available for rsiwp objects:

```
design(term, data)  Creates design matrix for RSIWP term
precision(term, data)  Creates precision matrix for RSIWP term
theta_info(term)  Extracts theta parameter information
beta_info(term, data)  Extracts beta parameter information
random_info(term, data)  Extracts random effects information
```

rsrpoly-class

Random Polynomial Model for Random Slope

Description

Creates a random polynomial model term for use in a random slope model.

Usage

```

rsrpoly(x, mult, p = 2, ref_value = 0, ref_mult = 0, sd = Inf)

## S4 method for signature 'rsrpoly'
design(term, data)

## S4 method for signature 'rsrpoly'
precision(term, data)

## S4 method for signature 'rsrpoly'
theta_info(term)

## S4 method for signature 'rsrpoly'
beta_info(term, data)

## S4 method for signature 'rsrpoly'
random_info(term, data)

```

Arguments

x	Variable name.
mult	Variable to multiply the polynomial by.
p	Polynomial degree (default: 2).
ref_value	Reference value for the polynomial.
ref_mult	Reference value for the covariate.
sd	Standard deviation for random effects.
term	A rsrpoly term object
data	A data frame containing the term variables

Value

A rsrpoly term object.

A design matrix for the random slope polynomial term, or NULL if p.order is 0

A precision matrix for the random slope polynomial term

NULL (random slope polynomial terms don't have theta parameters)

NULL (random slope polynomial terms don't have beta parameters)

A data frame containing random effects information for the random slope polynomial term

Functions

- `design(rsrpoly)`: Creates design matrix for rsrpoly term
- `precision(rsrpoly)`: Creates precision matrix for rsrpoly term
- `theta_info(rsrpoly)`: Extracts theta parameter information for rsrpoly term
- `beta_info(rsrpoly)`: Extracts beta parameter information for rsrpoly term
- `random_info(rsrpoly)`: Extracts random effects information for rsrpoly term

Methods

The following methods are available for *rsrpoly* objects:

`design(term, data)` Creates design matrix for *rsrpoly* term
`precision(term, data)` Creates precision matrix for *rsrpoly* term
`theta_info(term)` Extracts theta parameter information
`beta_info(term, data)` Extracts beta parameter information
`random_info(term, data)` Extracts random effects information

Index

beta_info,hiwp-method (hiwp-class), 4
beta_info,hrpoly-method (hrpoly-class),
6
beta_info,rsfpoly-method
(rsfpoly-class), 8
beta_info,rsiid-method (rsiid-class), 10
beta_info,rsiwp-method (rsiwp-class), 12
beta_info,rsrpoly-method
(rsrpoly-class), 13

cond_sim_iwp, 2, 3
cond_sim_iwp_at, 3, 3, 4
cond_sim_iwp_inputs, 3, 4

design,hiwp-method (hiwp-class), 4
design,hrpoly-method (hrpoly-class), 6
design,rsfpoly-method (rsfpoly-class), 8
design,rsiid-method (rsiid-class), 10
design,rsiwp-method (rsiwp-class), 12
design,rsrpoly-method (rsrpoly-class),
13

hiwp (hiwp-class), 4
hiwp-class, 4
hrpoly (hrpoly-class), 6
hrpoly-class, 6

laplace_half_H_inv, 3
log_lik_laplace, 2, 4

model_data, 2, 4

precision,hiwp-method (hiwp-class), 4
precision,hrpoly-method (hrpoly-class),
6
precision,rsfpoly-method
(rsfpoly-class), 8
precision,rsiid-method (rsiid-class), 10
precision,rsiwp-method (rsiwp-class), 12
precision,rsrpoly-method
(rsrpoly-class), 13

random_info,hiwp-method (hiwp-class), 4
random_info,hrpoly-method
(hrpoly-class), 6
random_info,rsfpoly-method
(rsfpoly-class), 8
random_info,rsiid-method (rsiid-class),
10
random_info,rsiwp-method (rsiwp-class),
12
random_info,rsrpoly-method
(rsrpoly-class), 13
rsfpoly (rsfpoly-class), 8
rsfpoly-class, 8
rsiid (rsiid-class), 10
rsiid-class, 10
rsiwp, 11
rsiwp-class, 12
rsrpoly (rsrpoly-class), 13
rsrpoly-class, 13

theta_info,hiwp-method (hiwp-class), 4
theta_info,hrpoly-method
(hrpoly-class), 6
theta_info,rsfpoly-method
(rsfpoly-class), 8
theta_info,rsiid-method (rsiid-class),
10
theta_info,rsiwp-method (rsiwp-class),
12
theta_info,rsrpoly-method
(rsrpoly-class), 13