Package: polyhedralCubature (via r-universe)

March 12, 2025

Title Multiple Integration over Convex Polyhedra

Version 1.1.0

Description Evaluation of multiple integrals over convex polyhedra. This is useful when the bounds of the integrals are some linear combinations of the variables.

License GPL-3

URL <https://github.com/stla/polyhedralCubature>

BugReports <https://github.com/stla/polyhedralCubature/issues>

Imports gmp, magrittr, Matrix, ompr, qspray, rcdd, SimplicialCubature, spray, tessellation

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Stéphane Laurent [aut, cre]

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Date/Publication 2023-11-22 17:00:02 UTC

Additional_repositories <https://cranhaven.r-universe.dev>

Config/pak/sysreqs libfreetype6-dev libglu1-mesa-dev libgmp3-dev make libicu-dev libpng-dev libgl1-mesa-dev libssl-dev libx11-dev zlib1g-dev

Repository https://cranhaven.r-universe.dev

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/polyhedralCubature

RemoteSha f879b31949feb4316639342dbcf02535d0908979

RemoteSubdir polyhedralCubature

Contents

---

getAb   *Easily get the matrix* A *and the vector* b

Description

Get the matrix A and the vector b representing the linear inequalities with a user-friendly syntax.

Usage

```
getAb(model)
```

Arguments

model   a "MIP model"; see the example

Value

A list with the matrix A and the vector b for usage in [integrateOverPolyhedron](#).

Examples

```
library(ompr)
model <- MIPModel() %>%
  add_variable(x) %>% add_variable(y) %>% add_variable(z) %>%
  add_constraint(-5 <= x) %>% add_constraint(x <= 4) %>%
  add_constraint(-5 <= y) %>% add_constraint(y <= 3 - x) %>%
  add_constraint(-10 <= z) %>% add_constraint(z <= 6 - x - y)
getAb(model)
```

---

integrateOverPolyhedron
                            Multiple integral over a polyhedron

---

Description

Multiple integral over a convex polyhedron given by a set of linear inequalities. See the vignette for explanations and examples.

Usage

```
integrateOverPolyhedron(f, A, b)
```

Arguments

f	either a function, a **spray** polynomial, or a **qspray** polynomial; its number of variables must match the number of columns of the matrix A
A, b	matrix and vector defining the linear inequalities which must be in numeric mode or, for exactness, in character mode, with an integer or a fraction as each entry; if f is a **qspray** polynomial, then A and b will be converted to character mode if they are in numeric mode, with the function d2q

Value

There are three possible values: an output of adaptIntegrateSimplex if f is a function, an output of integrateSimplexPolynomial if f is a **spray** polynomial, or a character representing the value of the integral as a fraction if f is a **qspray** polynomial.

Examples

```
A <- rbind(
  c(-1, 0, 0), # -x
  c( 1, 0, 0), # x
  c( 0,-1, 0), # -y
  c( 1, 1, 0), # x+y
  c( 0, 0,-1), # -z
  c( 1, 1, 1)  # x+y+z
)
b <- c(
  5, 4,  # -5 < x < 4      <=> -x < 5  &  x < 4
  5, 3,  # -5 < y < 3-x    <=> -y < 5  &  x+y < 3
  10, 6  # -10 < z < 6-x-y <=> -z < 10 &  x+y+z < 6
)
f <- function(x, y, z) {
  x*y + 5*cos(z)
}
integrateOverPolyhedron(f, A, b)
```

---

pipe-operator	*Pipe operator*

Description

This is the 'magrittr' pipe operator. We import it in this package in order to help the user to construct the model argument of the getAb function.

Index