Package ‘waydown’

July 22, 2025

Type Package

Title Computation of Approximate Potentials for Weakly Non-Gradient Fields

Version 1.1.0

Author Pablo Rodríguez-Sánchez

Maintainer Pablo Rodríguez-Sánchez <pablo.rodriguez.sanchez@gmail.com>

Description Computation of approximate potentials for both gradient and non gradient fields. It is known from physics that only gradient fields, also known as conservative, have a well defined potential function. Here we present an algorithm, based on the classical Helmholtz decomposition, to obtain an approximate potential function for non gradient fields.
More information in
Rodríguez-Sánchez (2020) <doi:10.1371/journal.pcbi.1007788>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

Suggests testthat, knitr, rmarkdown, deSolve, dplyr, colorRamps, ggplot2, gridExtra, latticeExtra, bindrcpp

Imports numDeriv, Matrix

NeedsCompilation no

Repository CRAN

Date/Publication 2021-03-17 13:50:02 UTC

Contents

approxPot1D *Approximate potential in one dimension*

Description

Approximate potential in one dimension

Usage

```
approxPot1D(f, xs, V0 = "auto")
```

Arguments

f	One-dimensional representing the flow (right hand side of differential equation)
xs	Vector of positions to evaluate
V0	(Optional) Value of V at first element of xs. When default, the global minimum is assigned 0

Value

The potential estimated at each point in xs

Author(s)

Pablo Rodríguez-Sánchez (https://pabrod.github.io)

References

https://arxiv.org/abs/1903.05615

See Also

approxPot2D, deltaV

Examples

```
# Flow
f = function(x) { sin(x) }

# Sampling points
xs <- seq(0, 2*pi, length.out = 1e3)

# Approximated potential
Vs <- approxPot1D(f, xs)
```

`approxPot2D` *Approximate potential in two dimensions*

Description

Approximate potential in two dimensions

Usage

```
approxPot2D(f, xs, ys, V0 = "auto", mode = "mixed")
```

Arguments

`f`	Two-dimensional representing the flow (right hand side of differential equation)
`xs`	Vector xs positions to evaluate
`ys`	Vector of ys positions to evaluate
`V0`	(Optional) Value of V at first element of (xs,ys). When default, the global minimum is assigned 0
`mode`	(Optional) Integration mode. Options are horizontal (default), vertical and mixed

Value

The potential estimated at each point (xs, ys)

Author(s)

Pablo Rodríguez-Sánchez (`https://pabrod.github.io`)

References

`https://arxiv.org/abs/1903.05615`

See Also

`approxPot1D`, `deltaV`

Examples

```
# Flow
f = function(x) {c(-x[1]*(x[1]^2 - 1.1), -x[2]*(x[2]^2 - 1))}

# Sampling points
xs <- seq(-1.5, 1.5, length.out = 10)
ys <- seq(-1.5, 1.5, length.out = 15)

# Approximated potential
Vs <- approxPot2D(f, xs, ys, mode = 'horizontal')
```

deltaV *Approximate potential difference between two points*

Description

Approximate potential difference between two points

Usage

```
deltaV(f, x, x0, normType = "f")
```

Arguments

`f` Flow equations (right hand side of differential equation)

`x` Position where we want to know the approximate potential

`x0` Reference position (center of the Taylor expansion)

`normType` (default: 'f') Matrix norm used to compute the error

Value

A list containing the approximate potential difference between x and x0 and the estimated error

Author(s)

Pablo Rodríguez-Sánchez (https://pabrod.github.io)

References

https://arxiv.org/abs/1903.05615

See Also

approxPot1D, approxPot2D, norm

Examples

```
# One dimensional flow
f <- function(x) { cos(x) }

# Evaluation points
x0 <- 1
x1 <- 1.02

dV <- deltaV(f, x1, x0)

 # Two dimensional flow
f <- function(x) { c(
 -2*x[1]*x[2],
 -x[1]^2 - 1
)}

# Evaluation points
x0 <- matrix(c(1,2), ncol = 1)
x1 <- matrix(c(0.98,2.01), ncol = 1)

dV <- deltaV(f, x1, x0)
```

Index