Package 'fptdApprox'

October 13, 2022

Encoding latin1

Type Package

Title Approximation of First-Passage-Time Densities for Diffusion Processes

Version 2.4

Date 2022-08-21

Author Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

Maintainer Juan J. Serrano-Pérez <jjserra@ugr.es>

Description Efficient approximation of first-passage-time densities for diffusion processes based on the First-Passage-Time Location (FPTL) function.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-08-21 19:00:02 UTC

R topics documented:

`fptdApprox-package` *Approximation of First-Passage-Time Densities for Diffusion Processes*

Description

Efficient approximation of first-passage-time (f.p.t.) densities for diffusion processes based on the First-Passage-Time Location (FPTL) function.

For a complete list of functions, use library(help="fptdApprox").

Details

Package:	fptdApprox
Type:	Package
Version:	2.4
Date:	2022-08-21
License:	GPL-2
LazyLoad:	yes

The **fptdApprox** package allows to approximate efficiently the f.p.t. density for a diffusion process through a continuous time-dependent boundary in the cases of conditioned and unconditioned f.p.t. problems.

For an unconditioned f.p.t. problem, a step by step study can be performed. First, the diffusion process under consideration must be defined by using the function `diffproc`. Then, the `FPTL` function will be used to calculate the FPTL function for the specified process and boundary. The information provided by the function is then extracted by the method `summary.fptl` and used to find the range of variation of the f.p.t. variable. Finally, such information is used by the function `Approx.cfpt.density` to obtain the approximation of the f.p.t. density.

In the general case (conditioned and unconditioned f.p.t. problems) the function `Approx.fpt.density` allows to obtain directly the approximate f.p.t. density.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

Maintainer: Juan J. Serrano-Pérez, <jjserra@ugr.es>

References

Buonocore, A., Nobile, A.G. and Ricciardi, L.M. (1987) A new integral equation for the evaluation of first-passage-time probability densities. *Adv. Appl. Probab.*, **19**, 784–800.

Román, P., Serrano, J. J., Torres, F. (2008) First-passage-time location function: Application to determine first-passage-time densities in diffusion processes. *Comput. Stat. Data Anal.*, **52**, 4132–4146.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

`Approx.cfpt.density` *Approximating First-Passage-Time Densities for Conditioned Problems*

Description

`Approx.cfpt.density` computes values of the approximate first-passage-time (f.p.t.) density, for a conditioned problem, from an object of class "summary.fptl" that contains the information provided by First-Passage-Time Location (FPTL) function.

Usage

```
Approx.cfpt.density(sfptl, variableStep = TRUE, from.t0 = FALSE,
                    to.T = FALSE, skip = TRUE, n = 250, p = 0.2,
                    alpha = 1, tol = 1e-03, it.max = 50000L)
```

Arguments

`sfptl`	an object of class "summary.fptl", a result of applying the `summary.fptl` method to an object of class "fptl".
`variableStep`	a logical value indicating whether a variable integration step is used.
`from.t0`	a logical value indicating whether the approximation should be calculated from the lower end of the interval considered, t_0, specified in the `sfptl` object.
`to.T`	a logical value indicating whether the approximation should be calculated to the upper end of the interval considered, T, specified in the `sfptl` object.
`skip`	a logical value indicating whether the intervals at which the FPTL function is near zero could be avoided.
`n`	Number of points used to determine the integration step in subintervals $[t_i^*, t_{max,i}^+]$ $i = 1, \ldots, m$, from interesting instants provided by the FPTL function.
`p`	Ratio of n used to determine the integration step in subintervals $[t_{max,i}^+, t_{i+1}^*]$, $i = 1, \ldots, m$, $[t_0, t_1^*]$ and $[t_{max,m}^+, T]$.
`alpha`	Parameter used to determine the integration step in subintervals $[t_{max,i}^+, t_{i+1}^*]$, $i = 1, \ldots, m$, $[t_0, t_1^*]$ and $[t_{max,m}^+, T]$, in order to reduce the computational cost of approximating the f.p.t. density function in those cases where $t_{i+1}^* - t_{max,i}^+ >> t_{max,i}^- - t_i^*$, for some i, $t_1^* - t_0 >> t_{max,1}^- - t_1^*$ or $T - t_{max,m}^+ >> t_{max,m}^- - t_m^*$, respectively.

`tol`	If the cumulative integral of the approximation is greater than or equal to 1 - tol the algorithm is stopped.
`it.max`	If the number of iterations required for the approximation process is greater than it.max, the function asks for permission to continue.

Details

For a diffusion process $\{X(t), t_0 \leq t \leq T\}$, the f.p.t. variable, conditioned to $X(t_0) = x_0$, through a continuous boundary $S(t)$ is defined as

$$T_{S(t),x_0} = \begin{cases} Inf\ \{t \geq t_0\ :\ X(t) > S(t) \mid X(t_0) = x_0\} & if\ x_0 < S(t_0) \\ Inf\ \{t \geq t_0\ :\ X(t) < S(t) \mid X(t_0) = x_0\} & if\ x_0 > S(t_0) \end{cases}.$$

Its density function is the solution to a Volterra integral equation of the second kind. The kernel of this equation depends on the infinitesimal moments of the process, the transition probability density function and the boundary.

Nevertheless, and apart from some particular processes and boundaries, closed-form solutions for the integral equation are not available. For this reason, in the cases without explicit solutions, numerical procedures are required. That is the situation considered here and the numerical procedure implemented by the `Approx.fpt.density` function is the one proposed by Buonocore et al. (1987), based on the composite trapezoid method.

The `Approx.cfpt.density` function computes efficiently the approximate f.p.t. density by using the information provided by the FPTL function contained in the `sfptl` object. See the function `summary.fptl` for details.

By default the function does not compute the approximate f.p.t. density from the time instant t_0, but from a more suitable time instant t_1^* provided by the FPTL function. It also uses a variable integration step.

The function makes an internal call to `Integration.Steps` function in order to determine the subintervals and integration steps to be used in the application of the numerical algorithm according to the `variableStep`, `from.t0`, `to.T`, `n`, `p` and `alpha` arguments.

In addition, if `skip = TRUE`, the function checks the approximate density value for each $t_{max,i}^+$, and, if it is almost 0, the application of the numerical algorithm in the subinterval $[t_{max,i}^+, t_{i+1}^*]$ is avoided, and then continued from instant t_{i+1}^* considering a zero value of the approximate density.

Similarly, if `to.T = FALSE`, the function checks the cumulative value of the integral for each $t_{max,i}^+$ provided by the FPTL function and, if it is greater than or equal to 1 - tol, the numerical algorithm is stopped. In any case, the algorithm is stopped in the final $t_{max,i}^+$, and if the cumulative value of the integral is less than 1 - tol the function issues a warning.

Value

The `Approx.cfpt.density` function computes and returns an object of class "fpt.density". It is a three-component list:

`x`	a sequence of suitable time instants in $[t_0,\ T]$ according to the arguments in the function call.
`y`	the approximate conditioned f.p.t. density function values on the x sequence.
`y.x0`	NULL (for consistency with the object of class "fpt.density" that produces the `Approx.fpt.density` function).

It also includes six additional attributes:

`Call`	the unevaluated function call, substituting each name in this call by its value when the latter has length 1.
`Steps`	matrix of subintervals and integration steps to consider for computing the approximate conditioned f.p.t. density.
`cumIntegral`	vector of the values of the cumulative integral of the approximation for each subinterval considered.
`skips`	a list that contains, for each subinterval, the value 1 if the application of the numerical algorithm has been avoided or integer(0) otherwise.
`CPUTime`	matrix of user and system times, by columns, required to approximate the density for each subinterval considered, by rows.
`summary.fptl`	the object used as `sfptl` argument in the function call.

`x` is the vector result of the concatenation of the sequences of equally spaced values in the suitable subintervals determined by the `Integration.Steps` function.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

Buonocore, A., Nobile, A.G. and Ricciardi, L.M. (1987) A new integral equation for the evaluation of first-passage-time probability densities. *Adv. Appl. Probab.*, **19**, 784–800.

Román, P., Serrano, J. J., Torres, F. (2008) First-passage-time location function: Application to determine first-passage-time densities in diffusion processes. *Comput. Stat. Data Anal.*, **52**, 4132–4146.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

`summary.fptl` to locate the f.p.t. variable and create objects of class "summary.fptl".

`is.fpt.density` to test for objects of class "fpt.density".

print.fpt.density to show objects of class "fpt.density".

report.fpt.density to generate a report.

plot.fpt.density for graphical display.

FPTL to evaluate the FPTL function and create objects of class "fptl".

Examples

```
## Continuing the summary.fptl(.) example:

## Making an efficient approximation of the f.p.t. density
## (optimal variable integration steps and small computational cost)
yyy <- Approx.cfpt.density(yy)
yyy

zzz <- Approx.cfpt.density(zz)
zzz

## Making a less efficient approximation of the f.p.t. density
## (optimal fixed integration step but high computational cost related to
##  the efficient approximation)
## Not run:
yyy1 <- Approx.cfpt.density(yy, variableStep = FALSE, from.t0 = TRUE, to.T =
                         TRUE, skip = FALSE)
yyy1
## End(Not run)
```

Approx.fpt.density *Approximating First-Passage-Time Densities*

Description

Approx.fpt.density computes values of the approximate first-passage-time (f.p.t.) density, for conditioned and unconditioned problems.

For the unconditioned case, values of the approximate densities for f.p.t. problems conditioned to suitable values of the initial distribution are also calculated.

Usage

```
Approx.fpt.density(dp, t0, T, id, S, env = NULL, variableStep = TRUE,
                   from.t0 = FALSE, to.T = FALSE, r = 4000, zeroSlope = 0.01,
                   p0.tol = 8, k = 3, m = 100, n = 250, p = 0.2, alpha = 1,
                   skip = TRUE, tol = 0.001, it.max)
```

Arguments

dp	an object of class "diffproc" defining a family of diffusion processes.
t0, T	lower and upper limits of the considered time interval. Must be finite.
id	a numerical value specifying a fixed initial value of the process in the time instant specified in the t0 argument (for a conditioned f.p.t. problem), or a four-component list specifying the initial distribution of the process (for an unconditioned f.p.t. problem). In the last case the first component is the R mathematical expression of the density as a character string, the second one is a label to denote the distribution in the reports generated by the report method, the third one is the label to denote the distribution for LaTeX, and the four one is a label to denote the distribution in the graphs generated by the plot method.
S	numerical value of a constant boundary or character string with the mathematical expression of a time dependent boundary.
env	a named list of objects of numeric or character type specifying the values of names which occur in the mathematical expressions in objects dp and S, or of names which occur in the specification of the previous values. Defaults to NULL, interpreted as an empty list. It is copied into a temporary environment for evaluating the mathematical expressions in the dp object and S.
variableStep	a logical value indicating whether a variable integration step is used.
from.t0	a logical value indicating whether the approximation should be calculated from the lower end of the interval considered, t_0.
to.T	a logical value indicating whether the approximation should be calculated to the upper end of the interval considered, T.
r	number of points at which the FPTL function is evaluated.
zeroSlope	maximum slope required to consider that a growing function is constant.
p0.tol	controls where the First-Passage-Time Location function begins to increase significantly.
k	controls whether the First-Passage-Time Location function decreases very slowly.
m	Number of equally spaced values to select from the specified initial distribution in the case of an unconditioned f.p.t. problem.
n	Number of points used to determine the integration step in subintervals $[t_i^*, t_{max,i}^+]$ $i = 1, \ldots, m$, from interesting instants provided by the FPTL function.
p	Ratio of n used to determine the integration step in subintervals $[t_{max,i}^+, t_{i+1}^*]$, $i = 1, \ldots, m$, $[t_0, t_1^*]$ and $[t_{max,m}^+, T]$.
alpha	Parameter used to determine the integration step in subintervals $[t_{max,i}^+, t_{i+1}^*]$, $i = 1, \ldots, m$, $[t_0, t_1^*]$ and $[t_{max,m}^+, T]$, in order to reduce the computational cost of approximating the f.p.t. density function in those cases where $t_{i+1}^* - t_{max,i}^+ >> t_{max,i}^- - t_i^*$, for some i, $t_1^* - t_0 >> t_{max,1}^- - t_1^*$ or $T - t_{max,m}^+ >> t_{max,m}^- - t_m^*$, respectively.
skip	a logical value indicating whether the intervals at which the FPTL function is near zero could be avoided.
tol	If the cumulative integral of the approximation is greater than or equal to 1 - tol the algorithm is stopped.

`it.max` If the number of iterations required for the approximation process is greater than it.max, the function asks for permission to continue. Defaults to 50000 for a conditioned f.p.t. problem, and 1000000 for an unconditioned f.p.t. problem.

Details

For an unconditioned f.p.t. problem, this function allows to compute directly the approximate f.p.t. density from an object of class "diffproc". In contrast to the approximation of a conditioned f.p.t. density by using the `Approx.cfpt.density` function, in this case it is not necessary for users to make previous calls to the functions `FPTL` and `summary.FPTL`.

For a diffusion process $\{X(t), t_0 \leq t \leq T\}$ with non-degenerate initial distribution the unconditioned f.p.t. variable, through a continuous boundary $S(t)$, is defined as

$$T_{S(t),X(t_0)} = \begin{cases} Inf\ \{t \geq t_0\ :\ X(t) > S(t)\} & if\ X(t_0) < S(t_0) \\ Inf\ \{t \geq t_0\ :\ X(t) < S(t)\} & if\ X(t_0) > S(t_0) \end{cases},$$

Its density function can be obtained by means of the numerical integration, in the range of variation of $X(t_0)$, of the corresponding f.p.t. densities conditioned to values of $X(t_0)$, weighted by the initial density function.

For each conditioned problem related to an unconditioned f.p.t. problem, the `Approx.fpt.density` function makes internal calls to `FPTL` and `summary.fptl` functions, in order to localize each conditioned f.p.t. variable, according to the `zeroSlope`, `p0.tol` and `k` arguments. Then, the function makes a internall call to `Integration.Steps` function, in order to determine the suitable subintervals and integration steps to be used to approximate the unconditioned f.p.t. density and conditioned f.p.t. densities according to the `variableStep`, `from.t0`, `to.T`, `n`, `p` and `alpha` arguments.

From this information, a suitable sequence of time instants in $[t_0, T]$ for the approximation of the f.p.t. densities is available. For each time instant `t` in such sequence, the `Approx.fpt.density` function computes the value of the f.p.t. density conditioned to `m` values of the initial distribution (equally spaced in its range of variation), and then it computes the value of the unconditioned f.p.t density.

For the approximation of each conditioned f.p.t density, the numerical procedure proposed by Buonocore et al. (1987), based on the composite trapezoid method, has been implemented. This method has also been used in the numerical integration for obtaining the unconditioned f.p.t. density.

The mathematical expression of the boundary `S` should be a function of t and may include the argument `t0`, the name `x0`, to refer to an initial value of the process, and the parameters specified in the `env` argument. The function checks if the mathematical expression shows syntax errors and if R can compute its symbolic derivative with respect to t.

The `env` argument is a list of tagged values in `name = value` form with `name` other than x, t, y and s. To name the expression of a function $h(u)$ as a character string we can use `` `h(u)` = value `` if we

want to show its dependence on u, or `h = value` otherwise.

The `env` argument is copied into a temporary environment for evaluating the mathematical expressions in objects `dp` and `S`. R looks for the objects not found into this temporary environment in the parent.frame() environment.

By default the function does not compute the approximate f.p.t. density from the time instant t_0, but from a more suitable time instant provided by the First-Passage-Time Location (FPTL) function. It also uses a variable integration step.

If `skip = TRUE`, for each subinterval in which it could be possible to avoid calculating all the conditioned f.p.t. densities, the function checks the value of the approximate unconditioned f.p.t. density at the lower end of such subinterval. If it is almost 0, the approximate unconditioned f.p.t. density calculation is avoided in the subinterval, and it continues from the upper end of the subinterval considering a zero value of the approximate unconditioned f.p.t. density.

Similarly, if `to.T = FALSE`, the function checks the cumulative value of the integral for each upper end of the subintervals for which it is not possible to avoid the application of the numerical algorithm for each conditioned problem. If it is greater than or equal to 1 - `tol`, the approximation procedure is stopped. In any case, the procedure is stopped at the upper end of the last subinterval, and if the cumulative value of the integral is less than 1 - `tol` the function issues a warning.

Value

The `Approx.fpt.density` function computes and returns an object of class "fpt.density". This object is a three-component list:

`x`	a sequence of suitable time instants in $[t_0,\ T]$ according to the arguments in the function call.
`y`	the approximate f.p.t. density function values on the x sequence for the unconditioned or conditioned problem at hand.
`y.x0`	NULL for a conditioned f.p.t. problem or a matrix with the values, by columns, of the approximate f.p.t. densities conditioned to each considered value x_0 selected from the initial distribution for an unconditioned f.p.t. problem.

It also includes six additional attributes:

`Call`	the unevaluated function call, substituting each object of length 1 (referred by name in call) by its value.
`Steps`	matrix of subintervals and integration steps to consider for computing the approximate f.p.t. density.
`cumIntegral`	vector of the values of the cumulative integral of the approximation for each subinterval considered.
`skips`	a list that contains, for each subinterval, the indexes of the initial values for which the calculation of the approximate conditioned f.p.t. densities has been avoided.

CPUTime	matrix of user and system times, by columns, required to approximate the density for each subinterval considered, by rows.
summary.fptl	the object of class "summary.fptl" that results in the internal calls to the summary.fptl function which is used as sfptl argument in the internal call to the Integration.Steps function.

x is the vector result of the concatenation of the sequences of equally spaced values in the suitable subintervals determined by the Integration.Steps function.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

Buonocore, A., Nobile, A.G. and Ricciardi, L.M. (1987) A new integral equation for the evaluation of first-passage-time probability densities. *Adv. Appl. Probab.*, **19**, 784–800.

Román, P., Serrano, J. J., Torres, F. (2008) First-passage-time location function: Application to determine first-passage-time densities in diffusion processes. *Comput. Stat. Data Anal.*, **52**, 4132–4146.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

diffproc about creation of class "diffproc" objects.

is.fpt.density to test for objects of class "fpt.density".

print.fpt.density to show objects of class "fpt.density".

report.fpt.density to generate a report.

plot.fpt.density for graphical display.

FPTL to evaluate the FPTL function and create objects of class "fptl".

summary.fptl to locate the f.p.t. variable and create objects of class "summary.fptl".

Examples

```
## Continuing the diffproc(.) example:

## Making an efficient approximation of the f.p.t. density
## in the case of a conditioned f.p.t. problem. (optimal
## variable integration steps and small computational cost)
yyy.cp <- Approx.fpt.density(dp = Lognormal, t0 = 0, T = 18, id = 1, S =
```

```
                        "4.5 + 4*t^2 + 7*t*sqrt(t)*sin(6*sqrt(t))",
                        env = list(m = 0.48, sigma = 0.07))
yyy.cp

## Not run:
## Making a less efficient approximation of the f.p.t. density
## (optimal fixed integration step but high computational cost related to
##  the efficient approximation)
yyy1.cp <- Approx.fpt.density(dp = Lognormal, t0 = 0, T = 18, id = 1, S =
                        "4.5 + 4*t^2 + 7*t*sqrt(t)*sin(6*sqrt(t))",
                        env = list(m = 0.48, sigma = 0.07),
                        variableStep = FALSE, from.t0 = TRUE, to.T =
                        TRUE, skip = FALSE)
yyy1.cp

## Making an efficient approximation of the f.p.t. density
## in the case of an unconditioned f.p.t. problem.
yyy.ucp <- Approx.fpt.density(dp = Lognormal, t0 = 0, T = 18, id =
                list("dlnorm(x,-0.005,0.1)", "Lambda(-0.005,0.1)",
                "\\Lambda(-0.005,0.1)", "Lognormal(-0.005,0.1)"),
                S = "4.5 + 4*t^2 + 7*t*sqrt(t)*sin(6*sqrt(t))",
                env = list(m = 0.48, sigma = 0.07), m=25)
yyy.ucp
## End(Not run)
```

`diffproc` *Diffusion Processes*

Description

`diffproc` creates an object of class "diffproc" from the given set of values.

`as.diffproc` attempts to turn its argument into an object of class "diffproc".

`is.diffproc` tests if its argument is an object of class "diffproc".

`print` shows an object of class "diffproc".

Usage

```
diffproc(text)
as.diffproc(text)
is.diffproc(obj)

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

Arguments

`text`	a character vector or list of length four to be coerced, containing the infinitesimal mean, infinitesimal variance, transition probability density function and transition probability distribution function of a diffusion process.

obj	an R object to be tested.
x	an object of class "diffproc".
...	additional arguments potentially passed (currently none is considered).

Details

The main goal of the diffproc function is to validate the mathematical expressions in the given character string vector or list. If no errors happen, the function converts the input into an object of class "diffproc". Otherwise, the function reports the errors.

The mathematical expressions of the infinitesimal mean and variance should be expressions in x and t, $A_1(x,t)$ and $A_2(x,t)$, respectively.

The mathematical expressions of the transition probability density and distribution functions should be expressions in x, t, y and s, $f(x,t|y,s)$ and $F(x,t|y,s)$, respectively.

In addition, all mathematical expressions may depend on generic parameters and functions of t, or s and t, referred by name. To name the expression of a function $h(u)$ we can use `` `h(u)` = value `` if we want to show its dependence on u, or h otherwise.

The function checks if the mathematical expressions show syntax errors and if R can compute the symbolic derivative with respect to x for the infinitesimal variance and transition probability density function.

Value

If possible, this function returns an object of class "diffproc" that defines a family of diffusion processes. It is a four-component list:

mean	character of length 1 with the mathematical expression of the infinitesimal mean of the process.
var	character of length 1 with the mathematical expression of the infinitesimal variance of the process.
tpdf	character of length 1 with the mathematical expression of the transition probability density function of the process.
tpdF	character of length 1 with the mathematical expression of the transition probability distribution function of the process.

is.diffproc returns TRUE or FALSE depending on whether its argument is an object of class "diffproc" or not.

Note

The transition probability density functions specified by the dnorm function should be expressed in terms of the standard normal distribution, since only its derivative is considered in R.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

D to know more about the simple functions and expressions that support symbolic derivative in R, and in particular about the functions provided in R to evaluate probability density functions.

FPTL to locate the first-passage-time of a diffusion process through a continuous boundary.

Examples

```
## Creating a diffproc object for the lognormal diffusion process
x <- c("m*x", "sigma^2*x^2",
       "dnorm((log(x)-(log(y)+(m-sigma^2/2)*(t-s)))/(sigma*sqrt(t-s)),0,1)/
       (sigma*sqrt(t-s)*x)", "plnorm(x,log(y)+(m-sigma^2/2)*(t-s),
       sigma*sqrt(t-s))")

Lognormal <- diffproc(x)
Lognormal

## Creating a diffproc object for the Ornstein Uhlenbeck diffusion process
x <- c("alpha*x + beta", "sigma^2", "dnorm((x-(y*exp(alpha*(t-s))-beta*
       (1-exp(alpha*(t-s)))/alpha))/(sigma*sqrt((exp(2*alpha*(t-s)) - 1)/
       (2*alpha))),0,1)/(sigma*sqrt((exp(2*alpha*(t-s)) - 1)/(2*alpha)))",
       "pnorm(x, y*exp(alpha*(t-s)) - beta*(1 - exp(alpha*(t-s)))/alpha,
       sigma*sqrt((exp(2*alpha*(t-s)) - 1)/(2*alpha)))")

OU <- diffproc(x)
OU

## Creating a diffproc object for the lognormal diffusion process with exogenous factors
x <- c("`h(t)`*x", "sigma^2*x^2", "dnorm((log(x)-(log(y)+`H(s,t)`-(sigma^2/2)*
       (t - s)))/(sigma*sqrt(t-s)),0,1)/(sigma*sqrt(t-s)*x)", "plnorm(x,log(y)+
       `H(s,t)`-(sigma^2/2)*(t-s),sigma*sqrt(t-s))")

LognormalFEx <- diffproc(x)
LognormalFEx

## Testing diffproc objects
is.diffproc(Lognormal)
is.diffproc(OU)
is.diffproc(LognormalFEx)
```

FPTL *First-Passage-Time Location Function*

Description

FPTL computes values of the First-Passage-Time Location (FPTL) function of a diffusion process for a continuous boundary.

is.fptl tests if its argument is an object of class "fptl".

print.fptl shows an object of class "fptl".

Usage

```
FPTL(dp, t0, T, x0, S, env = NULL, n = 4000)

is.fptl(obj)

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

Arguments

dp	an object of class "diffproc" defining a family of diffusion processes.
obj	an R object to be tested.
x	an object of class "fptl", a result of a call to this function.
t0, T	lower and upper limits of the considered time interval. Must be finite.
x0	fixed initial value of process in the time instant specified in the t0 argument.
S	numerical value of a constant boundary or character string with the mathematical expression of a time-dependent boundary.
env	a named list of objects of numeric or character type specifying the values of names which occur in the mathematical expressions in objects dp and S, or of names which occur in the specification of the previous values. Defaults to NULL, interpreted as an empty list.
n	number of points at which the FPTL function is evaluated.
...	additional arguments potentially passed (currently none is considered).

Details

The FPTL function for the problem of the first-passage-time of a diffusion process $\{X(t), t_0 \leq t \leq T\}$, conditioned to $X(t_0) = x_0$, through a continuous boundary $S(t)$ is defined as

$$FPTL(t) = \begin{cases} P[X(t) > S(t) \,|\, X(t_0) = x_0] = 1 - F(S(t), t \,|\, x_0, t_0) & if\ x_0 < S(t_0) \\ P[X(t) < S(t) \,|\, X(t_0) = x_0] = F(S(t), t \,|\, x_0, t_0) & if\ x_0 > S(t_0) \end{cases},$$

where $F(x, t|y, s)$ is the transition probability distribution function of the process.

Initially, the FPTL function is evaluated at a sequence of n equally spaced values from t0 to T. Then the FPTL function makes an internal call to the growth.intervals function in order to study the growth of the evaluation vector. Finally, the FPTL function is evaluated at a more adequate sequence of values from t0 to T according to the abovementioned study.

The mathematical expression of the boundary S should be a function of t and may include arguments t0, x0 and the parameters specified in the env argument. The FPTL function checks if the mathematical expression shows syntax errors and if R can compute its symbolic derivative with respect to t.

The env argument is a list of tagged values in name = value form with name other than x, t, y and s. To name the expression of a function $h(u)$ as a character string we can use `h(u)` = value if we want to show its dependence on u, or h = value otherwise.

The env argument is copied into a temporary environment for evaluating the mathematical expressions in objects dp and S. R looks for the objects not found into this temporary environment in the parent.frame() environment.

Value

The function FPTL computes and returns an object of class "fptl". It is a two-component list:

x	a sequence of n values from t0 to T.
y	the corresponding values of the FPTL function for the x sequence.

It also includes three additional attributes:

Call	the unevaluated function call, substituting each name in this call for its value when the latter has length 1.
dp	the object used as dp argument in the function call.
vars	NULL or a list containing the values of names in the function call for those names with values of length greater than 1.

is.fptl returns TRUE or FALSE depending on whether its argument is an object of class "fptl" or not.

Since n is usually large, the print.fptl function does not display an object of class "fptl" as a list, but in its 'basic' structure instead. However, each component can be displayed separately in the usual way.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

Román, P., Serrano, J. J., Torres, F. (2008) First-passage-time location function: Application to determine first-passage-time densities in diffusion processes. *Comput. Stat. Data Anal.*, **52**, 4132–4146.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

`diffproc` about creation of class "diffproc" objects.

`summary.fptl` for summaries and `plot.fptl` for graphical display.

Examples

```
## Continuing the diffproc(.) examples:

## Specifying a boundary
b <- "4.5 + 4*t^2 + 7*t*sqrt(t)*sin(6*sqrt(t))"

## Computing FPTL functions and creating objects of class fptl
y <- FPTL(dp = Lognormal, t0 = 0, T = 18, x0 = 1, S = b, env = list(m = 0.48,
          sigma = 0.07))
y

z <- FPTL(dp = LognormalFEx, t0 = 1, T = 10, x0 = 1, S = 15, env = list(sigma=0.1,
          `h(t)` = "t/4", `H(s,t)` = "(t^2-s^2)/8"))
z

## Testing fptl objects
is.fptl(y)
is.fptl(z)
```

`growth.intervals` *Studying the Growth of a Vector*

Description

For the vector of values resulting from the evaluation of a function, this function determines the positions of the values between which the vector grows.

Usage

```
growth.intervals(x, y, zeroSlope = 0.01)
```

Arguments

x	a vector of values.
y	the vector of the corresponding values of a function for the x vector.
zeroSlope	maximum slope (in degrees) required to consider that a growing function is constant.

Details

The growth.intervals function ignores the pairs of values between which the vector y starts and stops growing if the slope is less than zeroSlope.

Value

This function returns NULL if the vector y is not growing. Otherwise, the function computes a matrix of indexes with two columns. The first column contains the positions of the values from which the vector y starts growing and the second column those at which the vector y stops growing.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

Examples

```
u <- seq(0, 5, length = 200)
v <- sin(u)
w <- growth.intervals(u, v)
w

plot(u, v, type = "l", las = 1)
abline(v = u[as.vector(w)])

## Continuing the FPTL(.) examples:

growth.intervals(y$x, y$y)
growth.intervals(y$x, y$y, zeroSlope = 0.001)
```

Integration.Steps *Subintervals and Integration Steps To Approximate First-Passage-Time Densities*

Description

According to the First-Passage-Time Location (FPTL) function and the arguments in the function call, this function calculates suitable subintervals and integration steps in order to approximate the first-passage-time (f.p.t.) density.

Usage

```
Integration.Steps(sfptl, variableStep = TRUE, from.t0 = FALSE,
                  to.T = FALSE, n = 250, p = 0.2, alpha = 1)
```

Arguments

`sfptl`	an object of class "summary.fptl".
`variableStep`	a logical value indicating whether a variable integration step is used.
`from.t0`	a logical value indicating whether the approximation should be calculated from the lower end of the interval considered, t_0, specified in the object used as `sfptl` argument.
`to.T`	a logical value indicating whether the approximation should be calculated to the upper end of the interval considered, T, specified in the object used as `sfptl` argument.
`n`	Number of points used to determine the integration step in subintervals $[t_i^*, t_{max,i}^+]$, $i = 1, \ldots, m$, from interesting instants provided by the FPTL function.
`p`	Ratio of n used to determine the integration step in subintervals $[t_{max,i}^+, t_{i+1}^*]$, $i = 1, \ldots, m$, $[t_0, t_1^*]$ and $[t_{max,m}^+, T]$.
`alpha`	Parameter used to determine the integration step in subintervals $[t_{max,i}^+, t_{i+1}^*]$, $i = 1, \ldots, m$, $[t_0, t_1^*]$ and $[t_{max,m}^+, T]$, in order to reduce the computational cost of approximating the f.p.t. density function in those cases where $t_{i+1}^* - t_{max,i}^+ >> t_{max,i}^- - t_i^*$, for some i, $t_1^* - t_0 >> t_{max,1}^- - t_1^*$ or $T - t_{max,m}^+ >> t_{max,m}^- - t_m^*$, respectively.

Details

Based on the information provided by the FPTL function contained in the `sfptl` object, this function computes and returns suitable subintervals and integration steps in order to approximate the density function of the f.p.t. variable according to the other arguments in the function call.

When the `sfptl` object is of length greater than 1, it comes from an unconditioned f.p.t. problem. Each component is associated with the same f.p.t. problem conditioned on different values of the initial distribution (equally spaced in the range of the distribution). Let $x_{0,j}$, $j = 1, \ldots, N$, these values. For each initial value $x_{0,j}$ let $t_{i,j}^*$, $t_{max,i,j}^-$ and $t_{max,i,j}^+$, $i = 1, \ldots\ m_j$, the interesting time instants provided by the FPTL function and stored in the `instants` component of the j-th list in the `sfptl` object. Then, the time instants $\{t_{i,j}, i = 1, 2, \ldots, 2m_j\}$, where

$$t_{i,j} = \begin{cases} t_{(i+1)/2,\,j}^* & for\ i\ odd \\ t_{max,\,i/2,\,j}^+ & for\ i\ even \end{cases},$$

provide a suitable partition of interval $[t_0, T]$ to approximate the f.p.t density for the fixed value $x_{0,j}$ of the initial distribution.

If the `sfptl` object is of length 1, it comes from a conditioned f.p.t. problem. In this case we denote the interesting time instants provided by the FPTL function and stored in the `sfptl` object by $t_{i,1}^*$,

$t^-_{max,i,1}$ and $t^+_{max,i,1}$.

In what follows, $\lceil x \rceil$ is the smallest integer not less than x.

For each list in the `sfptl` object the function computes

$$h_{i,j} = \frac{t^+_{max,i,j} - t^*_{i,j}}{n_{i,j}}, i = 1, \ldots, m_j,$$

where

$$n_{i,j} = \lceil n\, k_{i,j} \rceil$$

and

$$k_{i,j} = \frac{t^+_{max,i,j} - t^*_{i,j}}{t^-_{max,i,j} - t^*_{i,j}}\ .$$

If `variableStep` = `TRUE`, for $0 < p$ and $\alpha \leq 1$, also computes

- $h^*_{i,j} = \frac{t^*_{i+1,j} - t^+_{max,i,j}}{n^*_{i,j}}, i = 1, \ldots, m_j - 1$, where

$$n^*_{i,j} = \begin{cases} \lceil n\, p\, k^*_{i,j} \rceil & if\ k^*_{i,j} \leq 1 \\ \lceil n\, p\, {k^*_{i,j}}^{\alpha} \rceil & if\ k^*_{i,j} > 1 \end{cases}$$

  with

$$k^*_{i,j} = \frac{t^*_{i+1,j} - t^+_{max,i,j}}{t^-_{max,i,j} - t^*_{i,j}}\ .$$

  If $h^*_{i,j} < h_{i,j}$, we then set $t^+_{max,i,j}$ equal to $t^*_{i+1,j}$ and $h_{i,j}$ is recalculated.

- $h^*_{0,j} = \frac{t^*_{1,j} - t_0}{n^*_{0,j}}$, where

$$n^*_{0,j} = \begin{cases} \lceil n\, p\, k^*_{0,j} \rceil & if\ k^*_{0,j} \leq 1 \\ \lceil n\, p\, {k^*_{0,j}}^{\alpha} \rceil & if\ k^*_{0,j} > 1 \end{cases}$$

  with

$$k^*_{0,j} = \frac{t^*_{1,j} - t_0}{t^-_{max,1,j} - t^*_{1,j}}\ ,$$

  when the `sfptl` object is of length 1 and `from.t0` = `TRUE`, or the `sfptl` object is of length greater than 1.

  If $h^*_{0,j} < h_{1,j}$, we then set $t^*_{1,j}$ equal to t_0 and $h_{1,j}$ is recalculated.

- $h^*_{m_j,j} = \frac{T - t^+_{max,m_j,j}}{n^*_{m_j,j}}$, where

$$n^*_{m_j,j} = \begin{cases} \lceil n\, p\, k^*_{m_j,j} \rceil & if\ k^*_{m_j,j} \leq 1 \\ \lceil n\, p\, {k^*_{m_j,j}}^{\alpha} \rceil & if\ k^*_{m_j,j} > 1 \end{cases}$$

with

$$k^*_{m_j,j} = \frac{T - t^+_{max,m_j,j}}{t^-_{max,m_j,j} - t^*_{m_j,j}} \ ,$$

when the `sfptl` object is of length 1 and `to.T = TRUE`, or the `sfptl` object is of length greater than 1.

If $h^*_{m_j,j} < h_{m_j,j}$, we then set $t^+_{max,m_j,j}$ equal to T and $h_{m_j,j}$ is recalculated.

$p \geq 0.1$ and $0.75 \leq \alpha \leq 1$ are recommended; otherwise, some integration steps can be excessively large.

If the `sfptl` object is of length 1 (conditioned f.p.t. problem), the suitable subintervals and integration steps that the function provides are:

- If `variableStep = TRUE`,
  - $h_{i,1}$ in subintervals $[t^*_{i,1}, t^+_{max,i,1}]$, $i = 1, \ldots, m_1$.
  - $h^*_{i,1}$ in subintervals $[t^+_{max,i,1}, t^*_{i+1,1}]$, $i = 1, \ldots, m_1 - 1$. In these subintervals is possible to avoid applying the numerical algorithm to approximate the f.p.t. density provided that the value of the approximate density at the time instant $t^+_{max,i,1}$ is almost 0.
  - $h^*_{0,1}$ in subinterval $[t_0, t^*_{1,1}]$, if `from.t0 = TRUE`.
  - $h^*_{m_1,1}$ in subinterval $[t^+_{max,m_1,1}, T]$, if `to.T = TRUE`.

- If `variableStep = FALSE` the function computes

$$h = min\left\{h_{i,1} \ , \ i = 1, \ldots, m_1\right\} \ .$$

  Then
  - If `from.t0 = FALSE` and `to.T = FALSE`, h is readjusted to exactly split the interval $[t^*_{1,1}, t^+_{max,m_1,1}]$.
  - If `from.t0 = TRUE` and `to.T = FALSE`, h is readjusted to exactly split the interval $[t_0, t^+_{max,m_1,1}]$.
  - If `from.t0 = FALSE` and `to.T = TRUE`, h is readjusted to exactly split the interval $[t^*_{1,1}, T]$.
  - If `from.t0 = TRUE` and `to.T = TRUE`, h is readjusted to exactly split the interval $[t_0, T]$.

  h is a suitable fixed integration step in subintervals $[t^*_{i,1}, t^+_{max,i,1}]$, $i = 1, \ldots, m_1$, and $[t^+_{max,i,1}$, $t^*_{i+1,1}]$, $i = 1, \ldots, m_1 - 1$; in subintervals $[t_0, t^*_{1,1}]$ if `from.t0 = TRUE`, and in $[t^+_{max,m_1,1}, T]$ if `to.T = TRUE`. The endpoints of such subintervals are readjusted according to this integration step.

If the `sfptl` object is a list of length greater than 1 (unconditioned f.p.t problem), a common partition of the interval $[t_0, T]$ is calculated from the suitable partitions of this interval for each fixed value of the initial distribution.

Let, in unified form, $H_{r,j}$, $r = 1, \ldots, 2m_j + 1$, the suitable integration steps (calculated for each j in similar manner to the case of the `sfptl` object is of length 1) in subintervals $I_{r,j} = [t_{r-1,j}, t_{r,j}]$, with $t_{0,j} = t_0$ and $t_{2m_j+1,j} = T$, $j = 1, \ldots, N$. Then, the ordered values of all time instants in the suitable partitions, $t_{(1)}, \ldots, t_{(M)}$, provide a common suitable partition of the interval $[t_0, T]$ in

subintervals $[t_{(i-1)}, t_{(i)}]$, $i = 1, \ldots, M+1$, where $t_{(0)} = t_0$ and $t_{(M+1)} = T$.

For this partition, the function computes

- $H_i = min\{H_{r,j} : j = 1, \ldots, N,\ and\ [t_{(i-1)}, t_{(i)}] \subseteq I_{r,j}\},\ i = 2, \ldots, M.$
- $H_1 = min\{H_{r,j} : j = 1, \ldots, N,\ and\ [t_0, t_{(1)}] \subseteq I_{r,j}\}$, if `from.t0 = TRUE`.
- $H_{M+1} = min\{H_{r,j} : j = 1, \ldots, N,\ and\ [t_{(M)}, T] \subseteq I_{r,j}\}$, if `to.T = TRUE`.

Thus,

- If `variableStep = TRUE`, the suitable subintervals and integrations steps that the function provides are
  - H_i in subintervals $[t_{(i-1)}, t_{(i)}]$, $i = 2, \ldots, M$.
  - H_1 in subinterval $[t_0, t_{(1)}]$, if `from.t0 = TRUE`.
  - H_{M+1} in subinterval $[t_{(M)}, T]$, if `to.T = TRUE`.

  Each integration step is readjusted to exactly split the corresponding subinterval.

- If `variableStep = FALSE`, a suitable fixed integration step for any subinterval $[t_{(i-1)}, t_{(i)}]$ is

$$h = min\{H_i : i = 1, \ldots, M+1\}.$$

  In this case it is not possible to avoid applying the approximation algorithm in $[t_{(i-1)}, t_{(i)}]\ \forall\, i = 1, \ldots, M+1$.

  Then
  - If `from.t0 = FALSE` and `to.T = FALSE`, h is readjusted to exactly split the interval $[t_{(1)}, t_{(M)}]$.
  - If `from.t0 = TRUE` and `to.T = FALSE`, h is readjusted to exactly split the interval $[t_0, t_{(M)}]$.
  - If `from.t0 = FALSE` and `to.T = TRUE`, h is readjusted to exactly split the interval $[t_{(1)}, T]$.
  - If `from.t0 = TRUE` and `to.T = TRUE`, h is readjusted to exactly split the interval $[t_0, T]$.

  h is a suitable fixed integration step in subintervals $[t_{(i-1)}, t_{(i)}]$, $i = 2, \ldots, M$, in subintervals $[t_0, t_{(1)}]$ if `from.t0 = TRUE`, and in $[t_{(M)}, T]$ if `to.T = TRUE`. The endpoints of such subintervals are readjusted according to this integration step.

Value

A two-component list:

`H`	A matrix of subintervals and integrations steps that we must consider in order to approximate the f.p.t. density according to the information contained in the `sfptl` object and the arguments in the function call.
`skip`	A list of logical vectors indicating, for each subinterval, the values of the initial distribution for which we must check whether it is possible to avoid applying the numerical algorithm.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

Román, P., Serrano, J. J., Torres, F. (2008) First-passage-time location function: Application to determine first-passage-time densities in diffusion processes. *Comput. Stat. Data Anal.*, **52**, 4132–4146.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

`Approx.fpt.density` to approximate f.p.t. densities from objects of class "summary.fptl" and create objects of class "fpt.density".

`summary.fptl` to locate the f.p.t. variable and create objects of class "summary.fptl" from objects of class "fptl".

`FPTL` to evaluate the FPTL function and create objects of class "fptl".

Examples

```
## Continuing the summary.fptl(.) example:

Integration.Steps(yy)
Integration.Steps(yy, from.t0 = TRUE)
Integration.Steps(yy, to.T = TRUE, n = 100, p = 0.25)

Integration.Steps(zz)
```

---

is.fpt.density *Testing for objects of class "fpt.density"*

Description

`is.fpt.density` tests if its argument is an object of class "fpt.density".

Usage

```
is.fpt.density(obj)
```

Arguments

`obj`	an R object to be tested.

Value

`is.fpt.density` returns `TRUE` or `FALSE` depending on whether its argument is an object of class "fpt.density" or not.

An object of class "fpt.density" is a three-component list:

`x`	a sequence of suitable time instants in $[t_0,\ T]$ according to the arguments in the function call.
`y`	the approximate f.p.t. density function values on the x sequence for the unconditioned or conditioned problem at hand.
`y.x0`	NULL for a conditioned f.p.t. problem or a matrix with the values, by columns, of the approximate f.p.t. densities conditioned to each considered value x_0 of the initial ditribution for an unconditioned f.p.t. problem.

It also includes six additional attributes. For more details, see the values of `Approx.cfpt.density` and `Approx.fpt.density` functions.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

Examples

```
## Testing fpt.density objects

## Continuing the Approx.cfpt.density example:
is.fpt.density(yyy)

## Continuing the Approx.fpt.density example:
is.fpt.density(yyy.cp)
## Not run:
is.fpt.density(yyy.ucp)
## End(Not run)
```

`plot.fpt.density` *Plotting Method for fpt.density Objects*

Description

This function creates a plot of the approximate density function for the conditioned or unconditioned f.p.t. problem at hand. Optionally, it displays the information provided by the First-Passage-Time Location (FPTL) function for a conditioned problem. Moreover, for an unconditioned problem creates an additional plot with all the approximate f.p.t. densities conditioned to each value x_0 selected from the initial ditribution.

Usage

```
## S3 method for class 'fpt.density'
plot(x, from.t0, to.T, dp.legend = TRUE, dp.legend.cex = 1,
     ylab = TRUE, growth.points = FALSE, instants = FALSE, ...)
```

Arguments

`x`	an object of class "fpt.density", a result of a call to `Approx.cfpt.density` or `Approx.fpt.density` functions.
`from.t0`	a logical value indicating whether the approximation should be plotted from the lower end of the interval considered, t_0, specified in the `x` object.
`to.T`	a logical value indicating whether the approximation should be plotted to the upper end of the interval considered, T, specified in the `x` object.
`dp.legend`	logical. If TRUE, adds a legend to the plot for identifying the diffusion process and boundary.
`dp.legend.cex`	the magnification to be used for legend relative to the current setting of `cex`.
`ylab`	logical. If TRUE, adds a title for the y axis.
`growth.points`	logical. If TRUE, for a conditioned f.p.t. problem adds one or more vertical lines and labels to the plot of the approximate density function in order to identify the time instants from which the FPTL function starts growing.
`instants`	logical. If TRUE, for a conditioned f.p.t. problem adds vertical lines and labels to the plot of the approximate density function in order to identify the other points of interest provided by the FPTL function.
`...`	graphical parameters to set before generating the plot, see `par`.

Details

If `from.t0` or `to.T` arguments are missing the function considers the corresponding arguments used in the call to `Approx.cfpt.density` or `Approx.fpt.density` functions, which in turn generated the `x` object.

If the approximate density functions were calculated from the lower end of the interval considered and `from.t0 = FALSE`, the approximate density functions should be plotted from the lower endlimit of the first subinterval in the attribute `Steps` of the `x` object.

If the approximate density functions were calculated to the upper end of the interval considered and `to.T = FALSE`, the approximate density functions should be plotted to the upper endlimit of last subinterval in the attribute `Steps` of the `x` object.

If `dp.legend = TRUE`, a legend is placed in the top inside of each plot frame.

Additional graphical arguments as `cex`, `lwd` and `ps` can be specified.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz

References

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

`FPTL` to know more about the FPTL function and objects of class "fptl".

`summary.fptl` to extract the information contained in objects of class "fptl" and create objects of class "summary.fptl".

`Approx.cfpt.density` to approximate the f.p.t. density from objects of class "summary.fptl" and create objects of class "fpt.density".

`Approx.fpt.density` to approximate the f.p.t. density from objects of class "dp" and create objects of class "fpt.density".

Examples

```
## Continuing the Approx.cfpt.density(.) example:
plot(yyy)
plot(yyy, from.t0 = TRUE)
plot(yyy, growth.points = TRUE)
plot(yyy, growth.points = TRUE, instants = TRUE)
plot(yyy, dp.legend = FALSE, growth.points = TRUE, instants = TRUE)
plot(yyy, cex = 1.25, cex.main = 1.15)
plot(yyy, cex = 1.25, cex.main = 1.15, dp.legend.cex = 0.8, growth.points = TRUE, instants = TRUE)

## Continuing the Approx.fpt.density example:
plot(yyy.cp)
## Not run:
plot(yyy.ucp)
## End(Not run)
```

plot.fptl *Plotting Method for fptl Objects*

Description

This function creates a plot of the First-Passage-Time Location (FPTL) function for a first-passage-time problem, displaying the information of interest contained in an object of class "fptl" and a corresponding object of class "summary.fptl".

Usage

```
## S3 method for class 'fptl'
plot(x, sfptl, from.t0 = TRUE, to.T = TRUE, dp.legend = TRUE,
     dp.legend.cex = 1, ylab = TRUE, growth.points = TRUE,
     instants = TRUE, ...)
```

Arguments

x	an object of class "fptl", a result of a call to FPTL.
sfptl	an object of class "summary.fptl", a result of applying the summary method to the x object.
from.t0	a logical value indicating whether the FPTL function should be plotted from the lower end of the interval considered, t_0, specified in the x object.
to.T	a logical value indicating whether the approximation should be plotted to the upper end of the interval considered, T, specified in the x object.
dp.legend	logical. If TRUE, adds a legend to the plot in order to identify the diffusion process and boundary used in the call to FPTL function which in turn generated the x object.
dp.legend.cex	the magnification to be used for legend relative to the current setting of cex.
ylab	logical. If TRUE, adds a title for the y axis.
growth.points	logical. If TRUE, adds one or more vertical lines and labels to the plot in order to identify the time instants from which the FPTL function starts growing.
instants	logical. If TRUE, draws and identify the other points of interest provided by the FPTL function.
...	graphical parameters to set before generating the plot, see par.

Details

If the sfptl object is missing, the function makes an internal call to the summary.fptl function in order to identify the points of interest provided by the FPTL function.

If the FPTL function shows at least a local maximum and from.t0 = FALSE, the FPTL function should be plotted from the first time instant from which the function starts growing.

If the FPTL function shows at least a local maximum and to.T = FALSE, the FPTL function should be plotted to the $t^{+}_{max,m}$ value related to the last local maximum $t_{max,m}$ if the function does not

decrease subsequently, or to the local minimum following the last local maximum $t_{max,m}$ if the function decreases subsequently.

If `dp.legend = TRUE`, a legend is placed in the top inside of the plot frame.

Additional graphical arguments as `cex`, `lwd` and `ps` can be specified.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

See Also

`FPTL` to know more about the FPTL function and objects of class "fptl".

`summary.fptl` for extract the information of interest in an object of class "fptl".

Examples

```
## Continuing the FPTL(.) example:

plot(y)
plot(y, cex.main = 1.4, growth.points = FALSE)
plot(y, cex.main = 1.4, growth.points = FALSE, instants = FALSE)
plot(y, cex.main = 1.4, dp.legend = FALSE, growth.points = FALSE, instants = FALSE)
plot(y, cex = 1.25, cex.main = 1.25)
plot(y, cex = 1.25, cex.main = 1.25, dp.legend.cex = 0.8)

plot(z)
plot(z, from.t0 = FALSE)
plot(z, to.T = FALSE)
plot(z, from.t0 = FALSE, to.T = FALSE)
```

`print.fpt.density` *Printing First-Passage-Time Densities*

Description

`print.fpt.density` shows an object of class "fpt.density".

Usage

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

Arguments

x	an object of class "fpt.density".
...	further arguments passed to print and format methods.

Value

Since the length of components of an object of class "fpt.density" is usually large, the print.fpt.density function does not display such object as a list, but in its 'basic' structure instead. However, each component can be displayed separately in the usual way.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

Examples

```
## Continuing the Approx.cfpt.density example:
yyy
print(yyy, digits=10)

## Continuing the Approx.fpt.density example:
yyy.cp
## Not run:
yyy.ucp
## End(Not run)
```

---

report *Writing a Report of an Object*

Description

report is a function used to generate a report of the results of various statistical methods. The function invokes particular 'methods' which depend on the 'class' of the first argument.

Usage

```
report(obj, ...)
```

Arguments

obj	an object for which a report is desired.
...	additional arguments affecting the report generated.

Details

The functions report.summary.fptl and report.fpt.density are examples of particular methods which report objects of class "summary.fptl" and "fpt.density".

Value

The form of the value returned by report depends on the class of its argument.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

See Also

report.summary.fptl

report.fpt.density

---

report.fpt.density *Writing a Report of a fpt.density Object*

Description

A report is generated with the information contained in an object of class "fpt.density".

Usage

```
## S3 method for class 'fpt.density'
report(obj, report.sfptl = FALSE, tex = FALSE, digits = 8, ...)
```

Arguments

obj	an object of class "fpt.density", a result of a call to Approx.cfpt.density or Approx.fpt.density functions.
report.sfptl	logical. If TRUE, also writes a report with the information contained in the object of class "summary.fptl" from which the obj object was created.
tex	logical, specifies whether to generate formatted LaTeX output (TRUE) or plain text file (FALSE).
digits	integer indicating the significant digits to be used.
...	additional arguments potentially passed (currently none is considered).

Details

The resulting report can be recicled by copy-and-pasting into a document (if `tex = FALSE`) or directly included into a LaTeX file (if `tex = TRUE`).

Value

Send the report to the command windows.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

`Approx.cfpt.density` to approximate the f.p.t. density from objects of class "summary.fptl" and to create objects of class "fpt.density".

`Approx.fpt.density` to approximate the f.p.t. density from objects of class "dp" and to create objects of class "fpt.density".

`is.fpt.density` to test whether an object is an object of class "fpt.density".

Examples

```
## Continuing the Approx.cfpt.density(.) example:
report(yyy, digits = 4)
report(yyy, report.sfptl = TRUE, digits = 4)
report(yyy, tex = TRUE, digits = 4)
report(yyy, report.sfptl = TRUE, tex = TRUE, digits = 4)

## Continuing the Approx.fpt.density(.) example:
report(yyy.cp)
## Not run:
report(yyy.ucp)
## End(Not run)
```

`report.summary.fptl` *Writing a Report of a summary.fptl Object*

Description

A report is generated with the information contained in an object of class "summary.fptl".

Usage

```
## S3 method for class 'summary.fptl'
report(obj, tex = FALSE, digits = 8, heading = TRUE, ...)
```

Arguments

`obj`	an object of class "summary.fptl", a result of a call to `summary.fptl` function or the attribute `summary.fptl` of the value (of class "fpt.density") of the `Approx.fpt.density` function (created through one or several successive internal calls to the `summary.fptl` function).
`tex`	logical, specifies whether to generate formatted LaTeX output (`TRUE`) or plain text file (`FALSE`).
`digits`	integer indicating the significant digits to be used.
`heading`	logical, specifies whether to include information about the diffusion process and the boundary (`TRUE`) or not (`FALSE`).
`...`	additional arguments potentially passed (currently none is considered).

Details

The resulting report can be recicled by copy-and-pasting into a document (if `tex = FALSE`) or directly included into a LaTeX file (if `tex = TRUE`).

Value

Send the report to the command windows.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

summary.fptl to create objects of class "summary.fptl" from objects of class "fptl".

Approx.fpt.density to create objects of class "fpt.density" with the attribute summary.fptl of class "summary.fptl"

is.summary.fptl to test whether an object is of class "summary.fptl".

Examples

```
## Continuing the summary.fptl(.) example:

report(yy, digits = 4)
report(yy, tex = TRUE, digits = 4)

report(zz)
report(zz, tex = TRUE)
```

---

summary.fptl *Locating a Conditioned First-Passage-Time Variable*

Description

summary.fptl summary method for class "fptl".

is.summary.fptl tests if its argument is an object of class "summary.fptl".

print.summary.fptl shows an object of class "summary.fptl".

Usage

```
## S3 method for class 'fptl'
summary(object, zeroSlope = 0.01, p0.tol = 8, k = 3, ...)

is.summary.fptl(obj)

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

Arguments

object	an object of class 'fptl', a result of a call to FPTL.
obj	an R object to be tested.
x	an object of class 'summary.fptl', a result of a call to summary.fptl.
zeroSlope	maximum slope required to consider that a growing function is constant.
p0.tol	controls where the First-Passage-Time Location function begins to increase significantly.
k	controls whether the First-Passage-Time Location function decreases very slowly.
...	other arguments passed to functions.

Details

The `summary.fptl` function extracts the information contained in `object` about the location of the variation range of a conditioned first-passage-time (f.p.t.) variable.

It makes an internal call to `growth.intervals` function in order to determine the time instants $t_i,\ i = 1, \ldots, m$, from which the First-Passage-Time Location (FPTL) function starts growing, and its local maximums $t_{max,i}$. For this, `zeroSlope` argument is considered.

If there is no growth subinterval, the execution of the function `summary.fptl` is stopped and an error is reported. Otherwise, for each of the subintervals $I_i = [t_i, t_{i+1}]$ the function determines:

- The first time instant $t_i^* \in [t_i, t_{max,i}]$ at which the function is bigger than or equal to

$$p_i^* = p_i + 10^{-p0.tol}(p_{max,i} - p_i)\ ,$$

  where $p_i = FPTL(t_i)$ and $p_{max,i} = FPTL(t_{max,i})$.

  $10^{-p0.tol}$ is the ratio of the global increase of the function in the growth subinterval $[t_i, t_{max,i}]$ that should be reached to consider that it begins to increase significantly.

- The first time instant $t_{max,i}^- \in [t_i, t_{max,i}]$ at which the FPTL function is bigger than or equal to

$$p_{max,i}^- = p_{max,i}\big(1 - 0.05(p_{max,i} - p_i)\big)\ .$$

- The last time instant $t_{max,i}^+ \in \big[t_{max,i},\ T_i\big]$ at which the FPTL function is bigger than or equal to

$$p_{max,i}^+ = max\left\{1 - (1 - p_{max,i}^2)^{(1+q)/2}, FPTL(T_i)\right\},$$

  where

$$T_i = min\big\{t_{max,i} + k\,(t_{max,i} - t_i^*)(1 - p_{max,i}),\ t_{i+1}\big\}$$

  and

$$q = \frac{p_{max,i} - p_i}{p_{max,i}}\ .$$

`print.summary.fptl` displays an object of class "summary.fptl" for immediate understanding of the information it contains.

Value

The `summary.fptl` function computes and returns an object of class "summary.fptl" and length 1.

An object of class "summary.fptl" is a list of length 1 for a conditioned f.p.t problem, or of the same length as the number of values selected from the non-degenerate initial distribution for an unconditioned f.p.t problem. Each component of the list is again a named list with two components:

`instants` a matrix whose columns correspond to t_i, t_i^*, $t_{max,i}^-$, $t_{max,i}$and $t_{max,i}^+$ values for each conditioned f.p.t problem.

`FPTLValues` the matrix of values of the FPTL function on `instants`.

It also includes four additional attributes:

`Call`	a list of the unevaluated calls to the `summary.fptl` function, substituting each name in these calls by its value when the latter has length 1.
`FPTLCall`	a list of the unevaluated calls to the `FPTL` function that resulted in the objects used as `object` argument in `Call`.
`dp`	the common object used as `dp` argument in the unevaluated calls to the `FPTL` function in `FPTLCall`.
`vars`	NULL or a list containing the common values of names in `FPTLCall` for those names with values of length greater than 1.

For an unconditioned f.p.t problem, the object includes the additional attribute `id` specifying the non-degenerate initial distribution.

The attribute "`summary.fptl`" of the value (of class "fpt.density") of the `Approx.fpt.density` function is an object of class `summary.fptl` of length 1 for a conditioned problem, and of length greather than 1 for an unconditioned problem. It is created from one or successive internal calls to the `summary.fptl` function.

`is.summary.fptl` returns `TRUE` or `FALSE` depending on whether its argument is an object of class "summary.fptl" or not.

Author(s)

Patricia Román-Román, Juan J. Serrano-Pérez and Francisco Torres-Ruiz.

References

Román, P., Serrano, J. J., Torres, F. (2008) First-passage-time location function: Application to determine first-passage-time densities in diffusion processes. *Comput. Stat. Data Anal.*, **52**, 4132–4146.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2012) An R package for an efficient approximation of first-passage-time densities for diffusion processes based on the FPTL function. *Applied Mathematics and Computation*, **218**, 8408–8428.

P. Román-Román, J.J. Serrano-Pérez, F. Torres-Ruiz. (2014) More general problems on first-passage times for diffusion processes: A new version of the fptdApprox R package. *Applied Mathematics and Computation*, **244**, 432–446.

See Also

`Approx.cfpt.density` to approximate densities of f.p.t. variables conditioned to a fixed initial value from objects of class "summary.fptl" and create objects of class "fpt.density".

`Approx.fpt.density` to approximate densities of conditioned or unconditioned f.p.t. variables and create objects of class "fpt.density" from objects of class "dp".

`FPTL` to evaluate the FPTL function and create objects of class "fptl".

`report.summary.fptl` to generate a report.

growth.intervals to study the growth of the vector of values resulting from the evaluation of a function.

Examples

```
## Continuing the FPTL(.) example:

## Summarizing an object of class fptl
yy <- summary(y)
yy
print(yy, digits=10)
yy1 <- summary(y, zeroSlope = 0.001)
yy1
yy2 <- summary(y, zeroSlope = 0.001, p0.tol = 10)
yy2

zz <- summary(z)
zz

## Testing summary.fptl objects
is.summary.fptl(yy)
is.summary.fptl(zz)
```

Index