Parameter estimation of a generalized Logisitc SDE with the ME method

1.0

Generated by Doxygen 1.9.1

Chapter 1

Todo List

Subprogram mod_random_number_generator::boxmuller ()

: Try other random-number generators, such like mersene an others

Subprogram mod_random_number_generator::normalvar ()

: Implement this furnction such that returns a realization path of the standard Brownian motion

Chapter 2

Modules Index

2.1 Modules List

Here is a list of all modules with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Module Documentation

4.1 mod_milstein_solver Module Reference

This module compute the coeffcients of the underlyng generalized logistc SDE.

Functions/Subroutines

- subroutine get_brownian_increment (delta, brownian_start, brownian_end)

  Simulate an increment of the related Browninar path.
- subroutine get_milstein_iteration (delta, current_state, current_drift, current_diffusion, diffusion_derivative, sigma, x_next)

  Computes a iteration of the Milstein method.

4.1.1 Detailed Description

This module compute the coeffcients of the underlyng generalized logistc SDE.

Author

E. Lince-Gomez, F. Baltazar-Larios, S. Diaz-Infante

$$
\begin{aligned}
dX(t) &= f(X(t))dt + g(X(t))dB(t) \\
f(x) &:= \alpha x\left[1 - \left(\frac{x}{K}\right)^m\right] \\
g(x) &:= \sigma x, \qquad t > t_0 \\
x_0 &= X(0), \quad x_0 \in [0,1].
\end{aligned}
$$

See also

the manuscript (...) for more details.

(details)

4.1.2 Function/Subroutine Documentation

4.1.2.1 get_brownian_increment()

```
subroutine mod_milstein_solver::get_brownian_increment (
            real(real32), intent(in) delta,
            real(real32), intent(in) brownian_start,
            real(real32), intent(out) brownian_end )
```

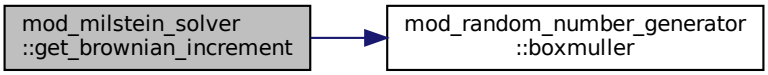
Simulate an increment of the related Browninar path.

This can be done by simulating the whole path

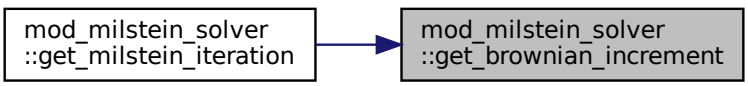
Parameters

in	*delta*	determininstic time step size
in	*brownian_start*	current observation of the Brownian process
in	*brownian_end*	observation at current time plus step-size

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.2 get_milstein_iteration()

```
subroutine mod_milstein_solver::get_milstein_iteration (
             real(real32), intent(in) delta,
             real(real32), intent(in) current_state,
             real(real32), intent(in) current_drift,
             real(real32), intent(in) current_diffusion,
             real(real32), intent(in) diffusion_derivative,
             real(real32), intent(in) sigma,
             real(real32), intent(out) x_next )
```

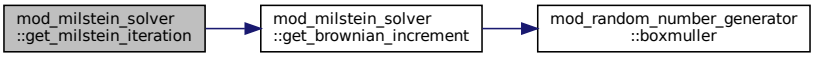
Computes a iteration of the Milstein method.

Defintion taken from the Kloedens book

Parameters

in	*delta*	determininstic time step-size
in	*brownian_start*	current observation of the Brownian process the Brownian increment.

Here is the call graph for this function:



4.2 mod_random_number_generator Module Reference

This module implements the Box-Muller algorithm to generate random variables with standard Gaussian distribution from a uniform ditributed random variable. This module enclose tree functions.

Functions/Subroutines

- real function unif (ix)

  Returns a random variables with uniform distribution using the standar gfortran random number generator, the returned value is a real 32.
- real(real32) function boxmuller ()

  Implementation of the Box-muller algorithm to genrate Gaussian random variables.
- real(real32) function normalvar ()

  Returns a number with Gaussian distribution using the Box-Muller algortihm.

4.2.1 Detailed Description

This module implements the Box-Muller algorithm to generate random variables with standard Gaussian distribution from a uniform ditributed random variable. This module enclose tree functions.

Author

E. Lince-Gomez, F. Baltazar-Larios, S. Diaz-Infante

See also

Kloeden & Platen 1992

4.2.2 Function/Subroutine Documentation

4.2.2.1 boxmuller()

```
real(real32) function mod_random_number_generator::boxmuller
```

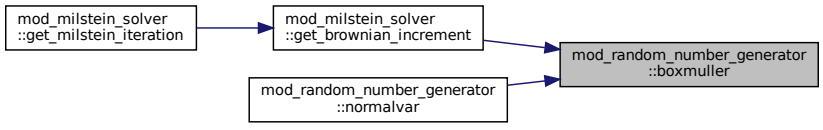
Implementation of the Box-muller algorithm to genrate Gaussian random variables.

Parameters

in	*seed*	initial value for the uniform random generator

Todo : Try other random-number generators, such like mersene an others

Here is the caller graph for this function:



4.2.2.2 normalvar()

```
real(real32) function mod_random_number_generator::normalvar
```

Returns a number with Gaussian distribution using the Box-Muller algortihm.

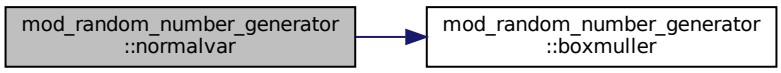
This function can be ommited in the case of a step forward implelemtation

Parameters

in	*seed*	a int32 with the initial value for the randim uniform generator

Todo : Implement this furnction such that returns a realization path of the standard Brownian motion

Here is the call graph for this function:



4.2.2.3 unif()

```
real function mod_random_number_generator::unif (
             integer(int32), intent(in) ix )
```

Returns a random variables with uniform distribution using the standar gfortran random number generator, the returned value is a real 32.

Parameters

in	*ix*	Dummy paramer for the seed initialization for the random generator

4.3 mod_stochastic_logistic_model Module Reference

This module compute the coeffcients of the underlyng generalized logistc SDE.

Functions/Subroutines

- subroutine compute_drift (alpha, m, x, y)
  <*BRIEF_DESCRIPTION*>
- subroutine compute_diffusion (sigma, x, y)
  <*BRIEF_DESCRIPTION*>

4.3.1 Detailed Description

This module compute the coeffcients of the underlyng generalized logistc SDE.

Author

E. Lince-Gomez, F. Baltazar-Larios, S. Diaz-Infante

$$\begin{aligned} dX(t) &= f(X(t))dt + g(X(t))dB(t) \\ f(x) &:= \alpha x\left[1 - \left(\frac{x}{K}\right)^m\right] \\ g(x) &:= \sigma x, \quad t > t_0 \\ x_0 &= X(0), \quad x_0 \in [0,1]. \end{aligned}$$

See also

the manuscript (...) for more details.

Version

1.0

4.3.2 Function/Subroutine Documentation

4.3.2.1 compute_diffusion()

```
subroutine mod_stochastic_logistic_model::compute_diffusion (
            real(real32), intent(in) sigma,
            real(real32), intent(in) x,
            real(real32), intent(out) y )
```

<BRIEF_DESCRIPTION>

<DETAILED_DESCRIPTION>

Parameters

[<in	or out or inout>] <PARAM1>
[<in	or out or inout>] <PARAM2>

4.3.2.2 compute_drift()

```
subroutine mod_stochastic_logistic_model::compute_drift (
            real(real32), intent(in) alpha,
            real(real32), intent(in) m,
            real(real32), intent(in) x,
            real(real32), intent(out) y )
```

<BRIEF_DESCRIPTION>

<DETAILED_DESCRIPTION>

Parameters

[<in	or out or inout>] <PARAM1>
[<in	or out or inout>] <PARAM2>

Chapter 5

File Documentation

5.1 src/mod_milstein_solver.f95 File Reference

Modules

- module mod_milstein_solver

  This module compute the coeffcients of the underlyng generalized logistc SDE.

Functions/Subroutines

- subroutine mod_milstein_solver::get_brownian_increment (delta, brownian_start, brownian_end)

  Simulate an increment of the related Browninar path.
- subroutine mod_milstein_solver::get_milstein_iteration (delta, current_state, current_drift, current_diffusion, diffusion_derivative, sigma, x_next)

  Computes a iteration of the Milstein method.

5.2 src/mod_milstein_solver.f95.d File Reference

5.3 src/mod_random_number_generator.f95 File Reference

Modules

- module mod_random_number_generator

  This module implements the Box-Muller algorithm to generate random variables with standard Gaussian distribution from a uniform ditributed random variable. This module enclose tree functions.

Functions/Subroutines

- real function mod_random_number_generator::unif (ix)

  Returns a random variables with uniform distribution using the standar gfortran random number generator, the returned value is a real 32.
- real(real32) function mod_random_number_generator::boxmuller ()

  Implementation of the Box-muller algorithm to genrate Gaussian random variables.
- real(real32) function mod_random_number_generator::normalvar ()

  Returns a number with Gaussian distribution using the Box-Muller algortihm.

5.4 src/mod_random_number_generator.f95.d File Reference

5.5 src/mod_stochastic_logistic_model.f95 File Reference

Modules

- module mod_stochastic_logistic_model

  This module compute the coeffcients of the underlyng generalized logistc SDE.

Functions/Subroutines

- subroutine mod_stochastic_logistic_model::compute_drift (alpha, m, x, y)

  <BRIEF_DESCRIPTION>
- subroutine mod_stochastic_logistic_model::compute_diffusion (sigma, x, y)

  <BRIEF_DESCRIPTION>

5.6 src/mod_stochastic_logistic_model.f95.d File Reference

Index