Package ‘Sim.DiffProc’

December 29, 2010

Type Package

Title Simulation of Diffusion Processes

Version 1.0

Date 2010-12-28

Author Prof. BOUKHETALA Kamal <kboukhetala@usthb.dz>, Mr. GUIDOUM Arsalane <starsalane@gmail.com>

Maintainer GUIDOUM Arsalane <starsalane@gmail.com>

Depends R (>= 2.11.0), tcltk, tcltk2 , stats4 , xlsx

Description Simulation of diffusion processes; Simulation the numerical solution of stochastic differential equations and analysis of discrete-time approximations for stochastic differential equations (SDE) driven by Wiener processes,in financial and actuarial modeling and other areas of application.

License GPL

LazyLoad yes

R topics documented:

`Sim.DiffProc-package` *Simulation of Diffusion Processes.*

Description

Simulation of diffusion processes; Simulation the numerical solution of stochastic differential equations and analysis of discrete-time approximations for stochastic differential equations (SDE) driven by Wiener processes,in financial and actuarial modeling and other areas of application.

Details

Package:	Sim.DiffProc
Type:	Package
Version:	1.0
Date:	2010-12-28
License:	GPL
LazyLoad:	yes

Author(s)

BOUKHETALA Kamal <kboukhetala@usthb.dz>, GUIDOUM Arsalane <starsalane@gmail.com>.

Maintainer: GUIDOUM Arsalane <starsalane@gmail.com>

References

1. Franck Jedrzejewski. Modeles aleatoires et physique probabiliste, Springer, 2009.
2. Fima C Klebaner. Introduction to stochastic calculus with application (Second Edition), Imperial College Press (ICP), 2005.
3. LAWRENCE C.EVANS. An introduction to stochastic differential equations (Version 1.2), Department of Mathematics (UC BERKELEY).
4. Hui-Hsiung Kuo. Introduction to stochastic integration, Springer, 2006.
5. E.Allen. Modeling with Ito stochastic differential equations, Springer, 2007.
6. Peter E.Kloeden, Eckhard Platen, Numerical solution of stochastic differential equations, Springer, 1995.
7. Douglas Henderson, Peter Plaschko, Stochastic differential equations in science and engineering,World Scientific, 2006.
8. A.Greiner, W.Strittmatter,and J.Honerkamp, Numerical Integration of Stochastic Differential Equations, Journal of Statistical Physics, Vol. 51, Nos. 1/2, 1988.
9. YOSHIHIRO SAITO, TAKETOMO MITSUI, SIMULATION OF STOCHASTIC DIFFERENTIAL EQUATIONS, Ann.Inst.Statist.Math, Vol. 45, No.3,419-432 (1993).
10. FRANCOIS-ERIC RACICOT, RAYMOND THEORET, Finance computationnelle et gestion des risques, Presses de universite du Quebec, 2006.
11. Avner Friedman, Stochastic differential equations and applications, Volume 1, ACADEMIC PRESS, 1975.

Examples

```
demo(BM2D)
demo(BMEuler)
demo(sim.sde)
example(snssde)
```

ABM *Creating Arithmetic Brownian Motion Model*

Description

Simulation of the arithmetic brownian motion model.

Usage

```
ABM(N, t0, T, x0, theta, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`theta`	constant (`Coefficient of drift`).
`sigma`	constant positive (`Coefficient of diffusion`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The function `ABM` returns a trajectory of the Arithmetic Brownian motion starting at `x0` at time `t0`, than the Discretization `dt = (T-t0)/N`.

The stochastic differential equation of the Arithmetic Brownian motion is :

$$dX(t) = theta * dt + sigma * dW(t)$$

with `theta :drift coefficient` and `sigma :diffusion coefficient`,`W(t)` is Wiener process.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`ABMF` creating flow of the arithmetic brownian motion model.

Examples

```
## Arithmetic Brownian Motion Model
## dX(t) = 3 * dt + 2 * dW(t) ; x0 = 0 and t0 = 0
ABM(N=1000,t0=0,T=1,x0=0,theta=3,sigma=2)
## Output in Excel 2007
ABM(N=1000,t0=0,T=1,x0=0,theta=3,sigma=3,output=TRUE)
```

ABMF *Creating Flow of The Arithmetic Brownian Motion Model*

Description

Simulation flow of the arithmetic brownian motion model.

Usage

```
ABMF(N, M, t0, T, x0, theta, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`theta`	constant (`Coefficient of drift`).
`sigma`	constant positive (`Coefficient of diffusion`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The function `ABMF` returns a flow of the Arithmetic Brownian motion starting at `x0` at time `t0`, than the discretization `dt = (T-t0)/N`.

The stochastic differential equation of the Arithmetic Brownian motion is :

$$dX(t) = theta * dt + sigma * dW(t)$$

With `theta :drift coefficient` and `sigma :diffusion coefficient`, `W(t)` is Wiener process.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`ABM` creating the arithmetic brownian motion model.

Examples

```
## Flow of Arithmetic Brownian Motion Model
## dX(t) = 3 * dt + 2 * dW(t) ; x0 = 0 and t0 = 0
ABMF(N=1000,M=100,t0=0,T=1,x0=0,theta=3,sigma=2)
## Output in Excel 2007
ABMF(N=1000,M=100,t0=0,T=1,x0=0,theta=3,sigma=2,output=TRUE)
```

`Asys` *Evolution a Telegraphic Process in Time*

Description

Simulation the evolution of the telegraphic process (the availability of a system).

Usage

```
Asys(lambda, mu, t, T)
```

Arguments

`lambda`	the rate so that the system functions.
`mu`	the rate so that the system is broken down.
`t`	calculate the matrix of transition `p(t)` has at the time `t`.
`T`	final time of evolution the process `[0,T]`.

Details

Calculate the matrix of transition `p(t)` at time `t`, the space states of the telegraphic process is `(0,1)` with `0 : the system is broken down` and `1 : the system functions`, the initial distribution at time `t = 0` of the process is `p(t=0)=(1,0)` or `p(t=0)=(0,1)`.

Value

matrix `p(t)` at time `t`, and plot of evolution the process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`Telegproc` simulation a telegraphic process.

Examples

```
## evolution a telegraphic process in time [0 , 5]
## calculate the matrix of transition p(t = 10)
Asys(0.5,0.5,10,5)
```

`BB` *Creating Brownian Bridge Model*

Description

Simulation of brownian bridge model.

Usage

```
BB(N, t0, T, x0, y, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`y`	terminal value of the process at time `T`.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The function returns a trajectory of the brownian bridge starting at `x0` at time `t0` and ending at `y` at time `T`.

It is defined as :

$$Xt(t0, x0, T, y) = x0 + W(t - t0) - (t - t0/T - t0) * (W(T - t0) - y + x0)$$

This process is easily simulated using the simulated trajectory of the Wiener process `W(t)`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BBF` simulation flow of brownian bridge Model, `diffBridge` Diffusion Bridge Models, `BMN` simulation brownian motion by the Normal Distribution , `BMRW` simulation brownian motion by a Random Walk, `GBM` simulation geometric brownian motion, `ABM` simulation arithmetic brownian motion, `snssde` Simulation Numerical Solution of SDE.

Examples

```
##brownian bridge model
##starting at x0 =0 at time t0=0 and ending at y =3 at time T =1.
BB(N=1000,t0=0,T=1,x0=0,y=3)
## Output in Excel 2007
BB(N=1000,t0=0,T=1,x0=0,y=3,output=TRUE)
```

BBF *Creating Flow of Brownian Bridge Model*

Description

Simulation flow of brownian bridge model.

Usage

```
BBF(N, M, t0, T, x0, y, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
x0	initial value of the process at time t0.
y	terminal value of the process at time T.
output	if output = TRUE write a output to an Excel 2007.

Details

The function BBF returns a flow of the brownian bridge starting at x0 at time t0 and ending at y at time T.

It is defined as :

$$Xt(t0, x0, T, y) = x0 + W(t - t0) - (t - t0/T - t0) * (W(T - t0) - y + x0)$$

This process is easily simulated using the simulated trajectory of the Wiener process W(t).

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

BB simulation brownian bridge Model, diffBridge Diffusion Bridge Models, BMN simulation brownian motion by the Normal Distribution , BMRW simulation brownian motion by a Random Walk, GBM simulation geometric brownian motion, ABM simulation arithmetic brownian motion, snssde Simulation Numerical Solution of SDE.

Examples

```
## flow of brownian bridge model
## starting at x0 =1 at time t0=0 and ending at y = -2 at time T =1.
BBF(N=1000,M=100,t0=0,T=1,x0=1,y=-2)
## Output in Excel 2007
BBF(N=1000,M=100,t0=0,T=1,x0=1,y=-2,output=TRUE)
```

Besselp *Creating Bessel process (by Milstein Scheme)*

Description

Simulation Besselp process by milstein scheme.

Usage

```
Besselp(N, M, t0, T, x0, alpha, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
x0	initial value of the process at time t0.
alpha	constant positive alpha >=2.
output	if output = TRUE write a output to an Excel 2007.

Details

The stochastic differential equation of Bessel process is :

$$dX(t) = (alpha - 1)/(2 * X(t)) * dt + dW(t)$$

with (alpha-1)/(2*X(t)) :drift coefficient and 1 :diffusion coefficient, W(t) is Wiener process, and the discretization dt = (T-t0)/N.

Constraints: alpha >= 2 and x0 =! 0.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

CEV Constant Elasticity of Variance Models, CIR Cox-Ingersoll-Ross Models, CIRhy modified CIR and hyperbolic Process, CKLS Chan-Karolyi-Longstaff-Sanders Models, DWP Double-Well Potential Model, GBM Model of Black-Scholes, HWV Hull-White/Vasicek Models, INFSR Inverse of Feller s Square Root models, JDP Jacobi Diffusion Process, PDP Pearson Diffusions Process, ROU Radial Ornstein-Uhlenbeck Process, diffBridge Diffusion Bridge Models, snssde Simulation Numerical Solution of SDE.

Examples

```
## Bessel Process
## alpha = 4
## dX(t) = 3/(2*x) * dt + dW(t)
## One trajectorie
Besselp(N=1000,M=1,t0=0,T=100,x0=1,alpha=4,output=FALSE)
## flow of  Besselp Process
Besselp(N=1000,M=10,t0=0,T=100,x0=1,alpha=4,output=FALSE)
## Output in Excel 2007
Besselp(N=1000,M=10,t0=0,T=100,x0=1,alpha=4,output=TRUE)
```

BMcov *Empirical Covariance for Brownian Motion*

Description

Calculate empirical covariance of the Brownian Motion.

Usage

```
BMcov(N, M, T, C)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`T`	final time.
`C`	constant positive (`if C = 1 it is standard brownian motion`).

Details

The brownian motion is a process with increase independent of function the covariance `cov(BM) = C * min(t,s)`, If `t > s` than `cov(BM) = C * s` else `cov(BM) = C * t`.

Value

contour of the empirical covariance for brownian motion.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMN` simulation brownian motion by the Normal Distribution , `BMRW` simulation brownian motion by a Random Walk, `BMinf` brownian motion property(Time tends towards the infinite), `BMIrt` brownian motion property(invariance by reversal of time), `BMscal` brownian motion property (invariance by scaling).

Examples

```
## empirical covariance of 200 trajectories brownian standard
BMcov(N=1000,M=200,T=1,C=1)
## empirical covariance of 200 trajectories brownian
BMcov(N=1000,M=200,T=1,C=4)
```

`BMinf` *Brownian Motion Property*

Description

Calculated the limit of standard brownian motion `limit(W(t)/t,0,T)`.

Usage

```
BMinf(N,T)
```

Arguments

`N`	size of process.
`T`	final time.

Details

Calculated the limit of standard brownian motion if the time tends towards the infinite,i.e the `limit(W(t)/t,0,T) = 0`.

Value

plot of `limit(W(t)/t)`.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMN` simulation brownian motion by the Normal Distribution , `BMRW` simulation brownian motion by a Random Walk, `BMIrt` brownian motion property(invariance by reversal of time), `BMscal` brownian motion property (invariance by scaling), `BMcov` empirical covariance for brownian motion.

Examples

```
BMinf(N=1000,T=10^5)
```

BMIrt *Brownian Motion Property (Invariance by reversal of time)*

Description

Brownian motion is invariance by reversal of time.

Usage

```
BMIrt(N, T)
```

Arguments

N	size of process.
T	final time.

Details

Brownian motion is invariance by reversal of time,i.e `W(t) = W(T-t) - W(T)`.

Value

plot of `W(T-t) - W(T)`.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

BMN simulation brownian motion by the Normal Distribution , BMRW simulation brownian motion by a Random Walk, BMinf Brownian Motion Property (time tends towards the infinite), BMscal brownian motion property (invariance by scaling), BMcov empirical covariance for brownian motion.

Examples

```
BMIrt(N=1000,T=1)
```

BMIto1 *Properties of the stochastic integral and Ito Process [1]*

Description

Simulation of the Ito `integral(W(s)dW(s),0,t)`.

Usage

```
BMIto1(N, T, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

However the Ito integral also has the peculiar property, amongst others, that :

$$integral(W(s)dW(s), 0, t) = 0.5 * (W(t)^2 - t)$$

from classical calculus for Ito integral with `w(0) = 0`.

The follows from the algebraic rearrangement :

$$integral(W(s)dW(s), 0, t) = sum(W(t) * (W(t + 1) - W(t)), 0, t)$$

Value

data frame(time,Ito,sum.Ito) and plot of the Ito integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMIto2` simulation of the Ito integral[2], `BMItoC` properties of the stochastic integral and Ito processes[3], `BMItoP` properties of the stochastic integral and Ito processes[4], `BMItoT` properties of the stochastic integral and Ito processes[5].

Examples

```
##
BMIto1(N=1000,T=1)
## Output in Excel 2007
BMIto1(N=1000,T=1,output=TRUE)
## comparison with BMIto2
system.time(BMIto1(N=10^4,T=1))
system.time(BMIto2(N=10^4,T=1))
```

`BMIto2` *Properties of the stochastic integral and Ito Process [2]*

Description

Simulation of the Ito `integral(W(s)dW(s),0,t)`.

Usage

```
BMIto2(N, T, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

However the Ito integral also has the peculiar property, amongst others, that :

$$integral(W(s)dW(s), 0, t) = 0.5 * (W(t)^2 - t)$$

from classical calculus for Ito integral with `w(0) = 0`.

The follows from the algebraic rearrangement :

$$integral(W(s)dW(s), 0, t) = sum(W(t) * (W(t + 1) - W(t)), 0, t)$$

Value

data frame(time,Ito,sum.Ito) and plot of the Ito integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMIto1` simulation of the Ito integral[1], `BMItoC` properties of the stochastic integral and Ito processes[3], `BMItoP` properties of the stochastic integral and Ito processes[4], `BMItoT` properties of the stochastic integral and Ito processes[5].

Examples

```
##
BMIto2(N=1000,T=1)
## Output in Excel 2007
BMIto2(N=1000,T=1,output=TRUE)
## comparison with BMIto1
system.time(BMIto2(N=10^4,T=1))
system.time(BMIto1(N=10^4,T=1))
```

BMItoC *Properties of the stochastic integral and Ito Process [3]*

Description

Simulation of the Ito `integral(alpha*dW(s),0,t)`.

Usage

```
BMItoC(N, T, alpha, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`alpha`	constant.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

However the Ito integral also has the peculiar property, amongst others, that :

$$integral(alpha * dW(s), 0, t) = alpha * W(t)$$

from classical calculus for Ito integral with `w(0) = 0`.

The follows from the algebraic rearrangement :

$$integral(alpha * dW(s), 0, t) = sum(alpha * (W(t + 1) - W(t)), 0, t)$$

Value

data frame(time,Ito,sum.Ito) and plot of the Ito integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMIto1` simulation of the Ito integral[1], `BMIto2` simulation of the Ito integral[2], `BMItoP` properties of the stochastic integral and Ito processes[4], `BMItoT` properties of the stochastic integral and Ito processes[5].

Examples

```
##
BMItoC(N=1000,T=1,alpha=2)
## Output in Excel 2007
BMItoC(N=1000,T=1,alpha=2,output=TRUE)
```

BMItoP *Properties of the stochastic integral and Ito Process [4]*

Description

Simulation of the Ito `integral(W(s)^n*dW(s),0,t)`.

Usage

```
BMItoP(N, T, power, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`power`	constant.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

However the Ito integral also has the peculiar property, amongst others, that :

$integral(W(s)^n * dW(s), 0, t) = W(t)^(n+1)/(n+1) - (n/2) * integral(W(s)^n - 1 * ds, 0, t)$

from classical calculus for Ito integral with `w(0) = 0`.

The follows from the algebraic rearrangement :

$$integral(W(s)^n * dW(s), 0, t) = sum(W(t)^n * (W(t+1) - W(t)), 0, t)$$

Value

data frame(time,Ito,sum.Ito) and plot of the Ito integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMIto1` simulation of the Ito integral[1], `BMIto2` simulation of the Ito integral[2], `BMItoC` properties of the stochastic integral and Ito processes[3], `BMItoT` properties of the stochastic integral and Ito processes[5].

Examples

```
## if power = 1
## integral(W(s) * dW(s),0,t) = W(t)^2/2 - 1/2 * t
BMItoP(N=1000,T=1,power =1)
## if power = 2
## integral(W(s)^2 * dW(s),0,t) = W(t)^3/3 - 2/2 * integral(W(s)*ds,0,t)
BMItoP(N=1000,T=1,power =2)
## Output in Excel 2007
BMItoP(N=1000,T=1,power =2,output=TRUE)
```

BMItoT *Properties of the stochastic integral and Ito Process [5]*

Description

Simulation of the Ito `integral(s*dW(s),0,t)`.

Usage

```
BMItoT(N, T, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

However the Ito integral also has the peculiar property, amongst others, that :

$$integral(s * dW(s), 0, t) = t * W(t) - integral(W(s) * ds, 0, t)$$

from classical calculus for Ito integral with `w(0) = 0`.

The follows from the algebraic rearrangement :

$$integral(s * dW(s), 0, t) = sum(t * (W(t + 1) - W(t)), 0, t)$$

Value

data frame(time,Ito,sum.Ito) and plot of the Ito integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMIto1` simulation of the Ito integral[1], `BMIto2` simulation of the Ito integral[2], `BMItoC` properties of the stochastic integral and Ito processes[3], `BMItoP` properties of the stochastic integral and Ito processes[4].

Examples

```
##
BMItoT(N=1000,T=1)
## Output in Excel 2007
BMItoT(N=1000,T=1,output=TRUE)
```

BMN *Creating Brownian Motion Model (by the Normal Distribution)*

Description

Simulation of the brownian motion model by the normal distribution.

Usage

```
BMN(N, t0, T, C, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`C`	constant positive (`if C = 1 it is standard brownian motion`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

Given a fixed time increment `dt = (T-t0)/N`, one can easily simulate a trajectory of the Wiener process in the time interval `[t0,T]`. Indeed, for `W(dt)` it holds true that `W(dt) = W(dt) - W(0) ~> N(0,dt) ~> sqrt(dt) * N(0,1)`,`N(0,1)` normal distribution.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMRW` simulation brownian motion by a random walk, `BMNF` simulation flow of brownian motion by the normal distribution, `BMRWF` simulation flow of brownian motion by a random walk, `BB` Simulation of brownian bridge model, `GBM` simulation geometric brownian motion Model.

Examples

```
##
BMN(N=1000,t0=0,T=1,C=1)
BMN(N=1000,t0=0,T=1,C=10)
## Output in Excel 2007
BMN(N=1000,t0=0,T=1,C=1,output=TRUE)
```

BMNF *Creating Flow of Brownian Motion (by the Normal Distribution)*

Description

Simulation flow of the brownian motion model by the normal distribution.

Usage

```
BMNF(N, M, t0, T, C, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
C	constant positive (if C = 1 it is standard brownian motion).
output	if output = TRUE write a output to an Excel 2007.

Details

Given a fixed time increment dt = (T-t0)/N, one can easily simulate a flow of the Wiener process in the time interval [t0,T]. Indeed, for W(dt) it holds true that W(dt) = W(dt) - W(0) ~> N(0,dt) ~> sqrt(dt) * N(0,1),N(0,1) normal distribution.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

BMRW simulation brownian motion by a random walk, BMN simulation of brownian motion by the normal distribution, BMRWF simulation flow of brownian motion by a random walk, BB Simulation of brownian bridge model, GBM simulation geometric brownian motion Model.

Examples

```
##
BMNF(N=1000,M=100,t0=0,T=1,C=1)
BMNF(N=1000,M=100,t0=0,T=1,C=10)
## Output in Excel 2007
BMNF(N=1000,M=100,t0=0,T=1,C=1,output=TRUE)
```

BMP *Brownian Motion Property (trajectories brownian lies between the two curves (+/-)2*sqrt(C*t))*

Description

trajectories Brownian lies between the two curves `(+/-)2*sqrt(C*t)`.

Usage

```
BMP(N, M, T, C)
```

Arguments

N	size of process.
M	number of trajectories.
T	final time.
C	constant positive (`if C = 1 it is standard brownian motion`).

Details

A flow of brownian motion lies between the two curves `(+/-)2*sqrt(C*t)`, `W(dt) - W(0) ~~> N(0,dt)`, `N(0,dt)` normal distribution.

Value

plot of the flow.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMscal` brownian motion property (invariance by scaling), `BMinf` brownian motion Property (time tends towards the infinite), `BMcov` empirical covariance for brownian motion, `BMIrt` brownian motion property(invariance by reversal of time).

Examples

```
##
BMP(N=1000,M=100,T=1,C=1)
BMP(N=1000,M=100,T=1,C=2)
BMP(N=1000,M=100,T=1,C=5)
BMP(N=1000,M=100,T=1,C=10)
```

BMRW *Creating Brownian Motion Model (by a Random Walk)*

Description

Simulation of the brownian motion model by a Random Walk.

Usage

```
BMRW(N, t0, T, C, output = FALSE)
```

Arguments

N	size of process.
t0	initial time.
T	final time.
C	constant positive (`if C = 1 it is standard brownian motion`).
output	if `output = TRUE` write a `output` to an Excel 2007.

Details

One characterization of the Brownian motion says that it can be seen as the limit of a random walk in the following sense.

Given a sequence of independent and identically distributed random variables `X1, X2, . . . , Xn`, taking only two values `+1` and `-1` with equal probability and considering the partial sum, `Sn = X1+ X2+ . . . + Xn`. then, as `n --> lnf`,`P(Sn/sqrt(N) < x) = P( W(t) < x)`.

Where `[x]` is the integer part of the real number `x`. Please note that this result is a refinement of the central limit theorem that, in our case, asserts that `Sn/sqrt(n) ~~> N(0,1)`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMN` simulation brownian motion by the normal distribution, `BMNF` simulation flow of brownian motion by the normal distribution, `BMRWF` simulation flow of brownian motion by a random walk, `BB` Simulation of brownian bridge model, `GBM` simulation geometric brownian motion Model.

Examples

```
##
BMRW(N=1000,t0=0,T=1,C=1)
BMRW(N=1000,t0=0,T=1,C=10)
## Output in Excel 2007
BMRW(N=1000,t0=0,T=1,C=1,output=TRUE)
```

BMRWF *Creating Flow of Brownian Motion (by a Random Walk)*

Description

Simulation flow of the brownian motion model by a Random Walk.

Usage

```
BMRWF(N, M, t0, T, C, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`C`	constant positive (`if C = 1 it is standard brownian motion`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

One characterization of the Brownian motion says that it can be seen as the limit of a random walk in the following sense.

Given a sequence of independent and identically distributed random variables `X1, X2, . . . , Xn`, taking only two values `+1` and `-1` with equal probability and considering the partial `sum, Sn = X1+ X2+ . . . + Xn.` then, as `n --> Inf`,`P(Sn/sqrt(N) < x) = P( W(t) < x).`

Where `[x]` is the integer part of the real number `x`. Please note that this result is a refinement of the central limit theorem that, in our case, asserts that `Sn/sqrt(n) ~~> N(0,1).`

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMN` simulation brownian motion by the normal distribution, `BMRW` simulation brownian motion by a random walk, `BB` Simulation of brownian bridge model, `GBM` simulation geometric brownian motion Model.

Examples

```
##
BMRWF(N=1000,M=100,t0=0,T=1,C=1)
BMRWF(N=1000,M=100,t0=0,T=1,C=10)
## Output in Excel 2007
BMRWF(N=1000,M=100,t0=0,T=1,C=1,output=TRUE)
```

`BMscal` *Brownian Motion Property (Invariance by scaling)*

Description

Brownian motion with different scales.

Usage

```
BMscal(N, T, S1, S2, S3, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`S1`	constant (`scale 1`).
`S2`	constant (`scale 2`).
`S3`	constant (`scale 3`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

Brownian motion is invariance by change the scales,i.e `W(t) = (1/S) * W(S^2 * t)`, `S` is scale.

Value

data.frame(w1,w2,w3) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMinf` brownian motion Property (time tends towards the infinite), `BMcov` empirical covariance for brownian motion, `BMIrt` brownian motion property(invariance by reversal of time).

Examples

```
##
BMscal(N=1000,T=10,S1=1,S2=1.1,S3=1.2)
## Output in Excel 2007
BMscal(N=1000,T=10,S1=1,S2=1.1,S3=1.2,output=TRUE)
```

`BMStra` *Stratonovitch Integral [1]*

Description

Simulation of the Stratonovitch `integral(W(s) o dW(s),0,t)`.

Usage

```
BMStra(N, T, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

Stratonovitch integral as defined :

$$integral(f(t)odW(s),0,t) = lim(sum(0.5*(f(t[i])+f(t[i+1]))*(W(t[i+1])-W(t[i]))))$$

calculus for Stratonovitch integral with `w(0) = 0`:

$$integral(W(s)odW(s),0,t) = 0.5*W(t)^2$$

The discretization `dt = T/N`, and `W(t)` is Wiener process.

Value

data frame(time,Stra) and plot of the Stratonovitch integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMStraC` Stratonovitch Integral [2], `BMStraP` Stratonovitch Integral [3], `BMStraT` Stratonovitch Integral [4].

Examples

```
##
BMStra(N=1000, T=1, output = FALSE)
## Output in Excel 2007
BMStra(N=1000, T=1, output = TRUE)
```

BMStraC *Stratonovitch Integral [2]*

Description

Simulation of the Stratonovitch `integral(alpha o dW(s),0,t)`.

Usage

```
BMStraC(N, T, alpha, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`alpha`	constant.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

Stratonovitch integral as defined :

$$integral(f(t)odW(s), 0, t) = lim(sum(0.5 * (f(t[i]) + f(t[i + 1])) * (W(t[i + 1]) - W(t[i]))))$$

calculus for Stratonovitch integral with `w(0) = 0`:

$$integral(alphaodW(s), 0, t) = alpha * W(t)$$

The discretization `dt = T/N`, and `W(t)` is Wiener process.

Value

data frame(time,Stra) and plot of the Stratonovitch integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMStra` Stratonovitch Integral [1], `BMStraP` Stratonovitch Integral [3], `BMStraT` Stratonovitch Integral [4].

Examples

```
##
BMStraC(N=1000, T=1, alpha = 2,output = FALSE)
## Output in Excel 2007
BMStraC(N=1000, T=1, alpha = 2,output = TRUE)
```

`BMStraP` *Stratonovitch Integral [3]*

Description

Simulation of the Stratonovitch `integral(W(s)^n o dW(s),0,t).`

Usage

```
BMStraP(N, T, power, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`power`	constant.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

Stratonovitch integral as defined :

$$integral(f(t)odW(s),0,t) = lim(sum(0.5*(f(t[i])+f(t[i+1]))*(W(t[i+1])-W(t[i]))))$$

calculus for Stratonovitch integral with `w(0) = 0`:

$$integral(W(s)^n odW(s),0,t) = lim(sum(0.5*(W(t[i])^(n-1)+W(t[i+1])^(n-1))*(W(t[i+1])^2-W(t[i])^2)))$$

The discretization `dt = T/N`, and `W(t)` is Wiener process.

Value

data frame(time,Stra) and plot of the Stratonovitch integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMStra` Stratonovitch Integral [1], `BMStraC` Stratonovitch Integral [2], `BMStraT` Stratonovitch Integral [4].

Examples

```
##
BMStraP(N=1000, T=1, power = 2,output = FALSE)
## Output in Excel 2007
BMStraP(N=1000, T=1, power = 2,output = TRUE)
```

BMStraT *Stratonovitch Integral [4]*

Description

Simulation of the Stratonovitch `integral(s o dW(s),0,t)`.

Usage

```
BMStraT(N, T, output = FALSE)
```

Arguments

`N`	size of process.
`T`	final time.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

Stratonovitch integral as defined :

$$integral(f(t)odW(s),0,t) = lim(sum(0.5*(f(t[i])+f(t[i+1]))*(W(t[i+1])-W(t[i]))))$$

calculus for Stratonovitch integral with `w(0) = 0`:

$$integral(sodW(s),0,t) = lim(sum(0.5*(t[i]*(W(t[i+1])-W(t[i]))+t[i+1]*(W(t[i+1])-W(t[i])))))$$

The discretization `dt = T/N`, and `W(t)` is Wiener process.

Value

data frame(time,Stra) and plot of the Stratonovitch integral.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`BMStra` Stratonovitch Integral [1], `BMStraC` Stratonovitch Integral [2], `BMStraC` Stratonovitch Integral [3].

Examples

```
BMStraT(N=1000, T=1,output = FALSE)
## Output in Excel 2007
BMStraT(N=1000, T=1,output = TRUE)
```

CEV *Creating Constant Elasticity of Variance (CEV) Models (by Milstein Scheme)*

Description

Simulation constant elasticity of variance models by milstein scheme.

Usage

```
CEV(N, M, t0, T, x0, mu, sigma, gamma, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`mu`	constant (`mu * X(t) :drift coefficient`).
`sigma`	constant positive (`sigma * X(t)^gamma :diffusion coefficient`).
`gamma`	constant positive (`sigma * X(t)^gamma :diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The Constant Elasticity of Variance `(CEV)` model also derives directly from the linear drift class, the discretization `dt = (T-t0)/N`.

The stochastic differential equation of `CEV` is :

$$dX(t) = mu * X(t) * dt + sigma * X(t)^g amma * dW(t)$$

with `mu * X(t) :drift coefficient` and `sigma * X(t)^gamma :diffusion coefficient`, `W(t)` is Wiener process.

This process is quite useful in modeling a skewed implied volatility. In particular,for `gamma < 1`, the skewness is negative, and for `gamma > 1` the skewness is positive. For `gamma = 1`, the `CEV` process is a particular version of the `geometric Brownian motion`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

CIR Cox-Ingersoll-Ross Models, CIRhy modified CIR and hyperbolic Process, CKLS Chan-Karolyi-Longstaff-Sanders Models, DWP Double-Well Potential Model, GBM Model of Black-Scholes, HWV Hull-White/Vasicek Models, INFSR Inverse of Feller s Square Root models, JDP Jacobi Diffusion Process, PDP Pearson Diffusions Process, ROU Radial Ornstein-Uhlenbeck Process, diffBridge Diffusion Bridge Models, snssde Simulation Numerical Solution of SDE.

Examples

```
## Constant Elasticity of Variance Models
## dX(t) = 0.3 *X(t) *dt + 2 * X(t)^1.2 * dW(t)
## One trajectorie
CEV(N=1000,M=1,t0=0,T=1,x0=0.1,mu=0.3,sigma=2,gamma=1.2)
## flow of CEV
CEV(N=1000,M=10,t0=0,T=1,x0=0.1,mu=0.3,sigma=2,gamma=1.2)
## Output in Excel 2007
CEV(N=1000,M=10,t0=0,T=1,x0=0.1,mu=0.3,sigma=2,gamma=1.2,output=TRUE)
```

CIR *Creating Cox-Ingersoll-Ross (CIR) Square Root Diffusion Models (by Milstein Scheme)*

Description

Simulation cox-ingersoll-ross models by milstein scheme.

Usage

```
CIR(N, M, t0, T, x0, theta, r, sigma, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
x0	initial value of the process at time t0.
theta	constant positive ((r - theta * X(t)) :drift coefficient).
r	constant positive ((r - theta * X(t)) :drift coefficient).
sigma	constant positive (sigma * sqrt(X(t)) :diffusion coefficient).
output	if output = TRUE write a output to an Excel 2007.

Details

Another interesting family of parametric models is that of the Cox-Ingersoll-Ross process. This model was introduced by Feller as a model for population growth and became quite popular in finance after Cox, Ingersoll, and Ross proposed it to model short-term interest rates. It was recently adopted to model nitrous oxide emission from soil by Pedersen and to model the evolutionary rate variation across sites in molecular evolution.

The discretization `dt = (T-t0)/N`, and the stochastic differential equation of CIR is :

$$dX(t) = (r - theta * X(t)) * dt + sigma * sqrt(X(t)) * dW(t)$$

With `(r - theta *X(t)) :drift coefficient` and `sigma*sqrt(X(t)) :diffusion coefficient`, `W(t)` is Wiener process.

Constraints: `2*r > sigma^2`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`CEV` Constant Elasticity of Variance Models, `CIRhy` modified CIR and hyperbolic Process, `CKLS` Chan-Karolyi-Longstaff-Sanders Models, `DWP` Double-Well Potential Model, `GBM` Model of Black-Scholes, `HWV` Hull-White/Vasicek Models, `INFSR` Inverse of Feller s Square Root models, `JDP` Jacobi Diffusion Process, `PDP` Pearson Diffusions Process, `ROU` Radial Ornstein-Uhlenbeck Process, `diffBridge` Diffusion Bridge Models, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Cox-Ingersoll-Ross Models
## dX(t) = (0.1 - 0.2 *X(t)) *dt + 0.05 * sqrt(X(t)) * dW(t)
## One trajectorie
CIR(N=1000,M=1,t0=0,T=1,x0=0.2,theta=0.2,r=0.1,sigma=0.05)
## flow of CIR
CIR(N=1000,M=10,t0=0,T=1,x0=0.2,theta=0.2,r=0.1,sigma=0.05)
## Output in Excel 2007
CIR(N=1000,M=10,t0=0,T=1,x0=0.2,theta=0.2,r=0.1,sigma=0.05,output=TRUE)
```

`CIRhy` *Creating The modified CIR and hyperbolic Process (by Milstein Scheme)*

Description

Simulation the modified CIR and hyperbolic process by milstein scheme.

Usage

```
CIRhy(N, M, t0, T, x0, r, sigma, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
x0	initial value of the process at time t0.
r	constant (-r * X(t) :drift coefficient).
sigma	constant positive (sigma * sqrt(1+X(t)^2) :diffusion coefficient).
output	if output = TRUE write a output to an Excel 2007.

Details

The stochastic differential equation of the modified CIR is :

$$dX(t) = -r * X(t) * dt + sigma * sqrt(1 + X(t)^2) * dW(t)$$

With -r*X(t) :drift coefficient and sigma*sqrt(1+X(t)^2) :diffusion coefficient, W(t) is Wiener process, the discretization dt = (T-t0)/N.

Constraints: r + (sigma^2)/2 > 0 (this is needed to make the process positive recurrent).

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

CEV Constant Elasticity of Variance Models, CIR Cox-Ingersoll-Ross Models , CKLS Chan-Karolyi-Longstaff-Sanders Models, DWP Double-Well Potential Model, GBM Model of Black-Scholes, HWV Hull-White/Vasicek Models, INFSR Inverse of Feller s Square Root models, JDP Jacobi Diffusion Process, PDP Pearson Diffusions Process, ROU Radial Ornstein-Uhlenbeck Process, diffBridge Diffusion Bridge Models, snssde Simulation Numerical Solution of SDE.

Examples

```
## The modified CIR and hyperbolic Process
## dX(t) = - 0.3 *X(t) *dt + 0.9 * sqrt(1+X(t)^2) * dW(t)
## One trajectorie
CIRhy(N=1000,M=1,T=1,t0=0,x0=1,r=0.3,sigma=0.9)
## flow of CIRhy
CIRhy(N=1000,M=10,T=1,t0=0,x0=1,r=0.3,sigma=0.9)
## Output in Excel 2007
CIRhy(N=1000,M=10,T=1,t0=0,x0=1,r=0.3,sigma=0.9,output=TRUE)
```

CKLS *Creating The Chan-Karolyi-Longstaff-Sanders (CKLS) family of models (by Milstein Scheme)*

Description

Simulation the chan-karolyi-longstaff-sanders models by milstein scheme.

Usage

```
CKLS(N, M, t0, T, x0, r, theta, sigma, gamma, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`r`	constant (`(r + theta *X(t)) :drift coefficient`).
`theta`	constant (`(r + theta *X(t)) :drift coefficient`).
`sigma`	constant positive (`sigma * X(t)^gamma :diffusion coefficient`).
`gamma`	constant positive (`sigma * X(t)^gamma :diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The Chan-Karolyi-Longstaff-Sanders `(CKLS)` family of models is a class of parametric stochastic differential equations widely used in many finance applications, in particular to model interest rates or asset prices.

The CKLS process solves the stochastic differential equation :

$$dX(t) = (r + theta * X(t)) * dt + sigma * X(t)^gamma * dW(t)$$

With `(r + theta * X(t)) :drift coefficient` and `sigma* X(t)^gamma :diffusion coefficient`, `W(t)` is Wiener process, the discretization `dt = (T-t0)/N`.

This `CKLS` model is a further extension of the Cox-Ingersoll-Ross model and hence embeds all previous models.

The `CKLS` model does not admit an explicit transition density unless `r = 0` or `gamma = 0.5`. It takes values in `(0, + lnf)` if `r`,`theta > 0`, and `gamma > 0.5`. In all cases, `sigma` is assumed to be positive.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

CEV Constant Elasticity of Variance Models, CIR Cox-Ingersoll-Ross Models, CIRhy modified CIR and hyperbolic Process, DWP Double-Well Potential Model, GBM Model of Black-Scholes, HWV Hull-White/Vasicek Models, INFSR Inverse of Feller s Square Root models, JDP Jacobi Diffusion Process, PDP Pearson Diffusions Process, ROU Radial Ornstein-Uhlenbeck Process, diffBridge Diffusion Bridge Models, snssde Simulation Numerical Solution of SDE.

Examples

```
## Chan-Karolyi-Longstaff-Sanders Models
## dX(t) = (0.3 + 0.01 *X(t)) *dt + 0.1 * X(t)^0.2 * dW(t)
## One trajectorie
CKLS(N=1000,M=1,T=1,t0=0,x0=1,r=0.3,theta=0.01,sigma=0.1,gamma=   0.2)
## flow of CKLS
CKLS(N=1000,M=10,T=1,t0=0,x0=1,r=0.3,theta=0.01,sigma=0.1,gamma=0.2)
## Output in Excel 2007
CKLS(N=1000,M=10,T=1,t0=0,x0=1,r=0.3,theta=0.01,sigma=0.1,gamma=0.2,output=TRUE)
```

---

diffBridge *Creating Diffusion Bridge Models (by Euler Scheme)*

Description

Simulation of diffusion bridge models by euler scheme.

Usage

```
diffBridge(N, t0, T, x, y, drift, diffusion, Output = FALSE)
```

Arguments

N	size of process.
t0	initial time.
T	final time.
x	initial value of the process at time t0.
y	terminal value of the process at time T.
drift	drift coefficient: an expression of two variables t and x.
diffusion	diffusion coefficient: an expression of two variables t and x.
Output	if Output = TRUE write a Output to an Excel 2007.

Details

The function diffBridge returns a trajectory of the diffusion bridge starting at x at time t0 and ending at y at time T, the discretization dt = (T-t0)/N.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

CEV Constant Elasticity of Variance Models, CKLS Chan-Karolyi-Longstaff-Sanders Models, CIR Cox-Ingersoll-Ross Models, CIRhy modified CIR and hyperbolic Process, DWP Double-Well Potential Model, GBM Model of Black-Scholes, HWV Hull-White/Vasicek Models, INFSR Inverse of Feller s Square Root models, JDP Jacobi Diffusion Process, PDP Pearson Diffusions Process, ROU Radial Ornstein-Uhlenbeck Process, snssde Simulation Numerical Solution of SDE.

Examples

```
## example 1 : Ornstein-Uhlenbeck Bridge Model (x0=1,t0=0,y=3,T=1)
drift     <- expression( (3*(2-x)) )
diffusion <- expression( (2) )
diffBridge(N=1000,t0=0,T=1,x=1,y=3,drift,diffusion)
## Output in Excel 2007
diffBridge(N=1000,t0=0,T=1,x=1,y=3,drift,diffusion,Output
=TRUE)

## example 2 : Brownian Bridge Model (x0=0,t0=0,y=1,T=1)
diffBridge(N=1000,t0=0,T=1,x=0,y=1,drift=expression((0)),
diffusion=expression((1)))

## example 3 : Geometric Brownian Bridge Model (x0=1,t0=1,y=3,T=3)
drift     <- expression( (3*x) )
diffusion <- expression( (2*x) )
diffBridge(N=1000,t0=1,T=3,x=1,y=3,drift,diffusion)

## example 4 : sde\ dX(t)=(0.03*t*X(t)-X(t)^3)*dt+0.1*dW(t) (x0=0,t0=0,y=2,T=100)
drift     <- expression( (0.03*t*x-x^3) )
diffusion <- expression( (0.1) )
diffBridge(N=1000,t0=0,T=100,x=0,y=2,drift,diffusion)
```

DWP — *Creating Double-Well Potential Model (by Milstein Scheme)*

Description

Simulation double-well potential model by milstein scheme.

Usage

```
DWP(N, M, t0, T, x0, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.

`T`	final time.
`x0`	initial value of the process at time `t0`.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

This model is interesting because of the fact that its density has a bimodal shape.

The process satisfies the stochastic differential equation :

$$dX(t) = (X(t) - X(t)^3) * dt + dW(t)$$

With `(X(t) - X(t)^3) :drift coefficient` and `1 is diffusion coefficient`, `W(t)` is Wiener process,and the discretization `dt = (T-t0)/N`.

This model is challenging in the sense that the Milstein approximation.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`CEV` Constant Elasticity of Variance Models, `CIR` Cox-Ingersoll-Ross Models, `CIRhy` modified CIR and hyperbolic Process, `CKLS` Chan-Karolyi-Longstaff-Sanders Models, `GBM` Model of Black-Scholes, `HWV` Hull-White/Vasicek Models, `INFSR` Inverse of Feller s Square Root models, `JDP` Jacobi Diffusion Process, `PDP` Pearson Diffusions Process, `ROU` Radial Ornstein-Uhlenbeck Process, `diffBridge` Diffusion Bridge Models, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Double-Well Potential Model
## dX(t) = (X(t) - X(t)^3) * dt + dW(t)
## One trajectorie
DWP(N=1000,M=1,T=1,t0=0,x0=1)
## flow of DWP
DWP(N=1000,M=10,T=1,t0=0,x0=1,output=TRUE)
```

`GBM` *Creating Geometric Brownian Motion (GBM) Models*

Description

Simulation geometric brownian motion or Black-Scholes models.

Usage

```
GBM(N, t0, T, x0, theta, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0` (`x0 > 0`).
`theta`	constant (`theta is the constant interest rateand` and `theta * X(t) :drift coefficient`).
`sigma`	constant positive (`sigma is volatility of risky activities` and `sigma * X(t):diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

This process is sometimes called the Black-Scholes-Merton model after its introduction in the finance context to model asset prices.

The process is the solution to the stochastic differential equation :

$$dX(t) = theta * X(t) * dt + sigma * X(t) * dW(t)$$

With `theta * X(t) :drift coefficient` and `sigma * X(t) : diffusion coefficient`, `W(t)` is Wiener process, the discretization `dt = (T-t0)/N`.

`sigma > 0`, the parameter `theta` is interpreted as the constant interest rate and `sigma` as the volatility of risky activities.

The explicit solution is :

$$X(t) = x0 * exp((theta - 0.5 * sigma^2) * t + sigma * W(t))$$

The conditional density function is `log-normal`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`GBMF` Flow of Geometric Brownian Motion, `PEBS` Parametric Estimation of Model Black-Scholes, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Black-Scholes Models
## dX(t) = 4 * X(t) * dt + 2 * X(t) *dW(t)
GBM(N=1000,T=1,t0=0,x0=1,theta=4,sigma=2)
## Output in Excel 2007
GBM(N=1000,T=1,t0=0,x0=1,theta=4,sigma=2,output=TRUE)
```

GBMF *Creating Flow of Geometric Brownian Motion Models*

Description

Simulation flow of geometric brownian motion or Black-Scholes models.

Usage

```
GBMF(N, M, t0, T, x0, theta, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0` (`x0 > 0`).
`theta`	constant (`theta is the constant interest rateand` and `theta * X(t) :drift coefficient`).
`sigma`	constant positive (`sigma is volatility of risky activities` and `sigma * X(t):diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

This process is sometimes called the Black-Scholes-Merton model after its introduction in the finance context to model asset prices.

The process is the solution to the stochastic differential equation :

$$dX(t) = theta * X(t) * dt + sigma * X(t) * dW(t)$$

With `theta * X(t) :drift coefficient` and `sigma * X(t) : diffusion coefficient`, `W(t)` is Wiener process, the discretization `dt = (T-t0)/N`.

`sigma > 0`, the parameter `theta` is interpreted as the constant interest rate and `sigma` as the volatility of risky activities.

The explicit solution is :

$$X(t) = x0 * exp((theta - 0.5 * sigma^2) * t + sigma * W(t))$$

The conditional density function is `log-normal`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

GBM Geometric Brownian Motion, PEBS Parametric Estimation of Model Black-Scholes, snssde Simulation Numerical Solution of SDE.

Examples

```
## Flow of Black-Scholes Models
## dX(t) = 4 * X(t) * dt + 2 * X(t) *dW(t)
GBMF(N=1000,M=10,T=1,t0=0,x0=1,theta=4,sigma=2)
## Output in Excel 2007
GBMF(N=1000,M=10,T=1,t0=0,x0=1,theta=4,sigma=2,output=TRUE)
```

HWV — *Creating Hull-White/Vasicek (HWV) Gaussian Diffusion Models*

Description

Simulation the Hull-White/Vasicek or gaussian diffusion models.

Usage

```
HWV(N, t0, T, x0, theta, r, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`theta`	constant (`theta is the long-run equilibrium value of the process` and `r*(theta -X(t)) :drift coefficient`).
`r`	constant positive (`r is speed of reversion` and `r*(theta -X(t)):drift coefficient`).
`sigma`	constant positive (`sigma (volatility) :diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The Hull-White/Vasicek `(HWV)` short rate class derives directly from SDE with mean-reverting drift:

$$dX(t) = r * (theta - X(t)) * dt + sigma * dW(t)$$

With `r *(theta- X(t)) :drift coefficient` and `sigma : diffusion coefficient`, `W(t)` is Wiener process, the discretization `dt = (T-t0)/N`.

The process is also ergodic, and its invariant law is the `Gaussian density`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

HWVF Flow of Gaussian Diffusion Models, PEOUG Parametric Estimation of Hull-White/Vasicek Models, snssde Simulation Numerical Solution of SDE.

Examples

```
## Hull-White/Vasicek Models
## dX(t) = 4 * (2.5 - X(t)) * dt + 1 *dW(t)
HWV(N=1000,t0=0,T=1,x0=10,theta=2.5,r=4,sigma=1)
## Output in Excel 2007
HWV(N=1000,t0=0,T=1,x0=10,theta=2.5,r=4,sigma=1,output=TRUE)
## if theta = 0 than "OU" = "HWV"
## dX(t) = 4 * ( 0 - X(t)) * dt + 1 *dW(t)
system.time(OU(N=10^4,t0=0,T=1,x0=10,r=4,sigma=1))
system.time(HWV(N=10^4,t0=0,T=1,x0=10,theta=0,r=4,sigma=1))
```

HWVF — *Creating Flow of Hull-White/Vasicek (HWV) Gaussian Diffusion Models*

Description

Simulation flow of the Hull-White/Vasicek or gaussian diffusion models.

Usage

```
HWVF(N, M, t0, T, x0, theta, r, sigma, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
x0	initial value of the process at time t0.
theta	constant (theta is the long-run equilibrium value of the process and r*(theta -X(t)) :drift coefficient).
r	constant positive (r is speed of reversion and r*(theta -X(t)):drift coefficient).
sigma	constant positive (sigma (volatility) :diffusion coefficient).
output	if output = TRUE write a output to an Excel 2007.

Details

The Hull-White/Vasicek (HWV) short rate class derives directly from SDE with mean-reverting drift:

$$dX(t) = r * (theta - X(t)) * dt + sigma * dW(t)$$

With `r *(theta- X(t)) :drift coefficient` and `sigma : diffusion coefficient`, `W(t)` is Wiener process, the discretization `dt = (T-t0)/N`.

The process is also ergodic, and its invariant law is the `Gaussian density`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`HWV` Hull-White/Vasicek Models, `PEOUG` Parametric Estimation of Hull-White/Vasicek Models, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## flow of Hull-White/Vasicek Models
## dX(t) = 4 * (2.5 - X(t)) * dt + 1 *dW(t)
HWVF(N=1000,M=100,t0=0,T=1,x0=10,theta=2.5,r=4,sigma=1)
## Output in Excel 2007
HWVF(N=1000,M=100,t0=0,T=1,x0=10,theta=2.5,r=4,sigma=1,output=TRUE)
## if theta = 0 than "FOU" = "HWVF"
## dX(t) = 4 * ( 0 - X(t)) * dt + 1 *dW(t)
system.time(HWVF(N=1000,M=100,t0=0,T=1,x0=10,theta=0,r=4,sigma=1))
system.time(FOU(N=1000,M=100,t0=0,T=1,x0=10,r=4,sigma=1))
```

Hyproc *Creating The Hyperbolic Process (by Milstein Scheme)*

Description

Simulation hyperbolic process by milstein scheme.

Usage

```
Hyproc(N, M, t0, T, x0, theta, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
x0	initial value of the process at time t0.
theta	constant positive.
output	if output = TRUE write a output to an Excel 2007.

Details

A process X satisfying :

$$dX(t) = (-theta * X(t)/sqrt(1 + X(t)^2)) * dt + dW(t)$$

With (-theta*X(t)/sqrt(1+X(t)^2)) :drift coefficient and 1 :diffusion coefficient, W(t) is Wiener process, discretization dt = (T-t0)/N.

Constraints: theta > 0.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

Hyprocg General Hyperbolic Diffusion, CIRhy modified CIR and hyperbolic Process, snssde Simulation Numerical Solution of SDE.

Examples

```
## Hyperbolic Process
## dX(t) = (-2*X(t)/sqrt(1+X(t)^2)) *dt + dW(t)
## One trajectorie
Hyproc(N=1000,M=1,T=100,t0=0,x0=3,theta=2)
## flow of Hyproc
Hyproc(N=1000,M=10,T=100,t0=0,x0=3,theta=2)
## Output in Excel 2007
Hyproc(N=1000,M=10,T=100,t0=0,x0=3,theta=2,output=TRUE)
```

Hyprocg *Creating The General Hyperbolic Diffusion (by Milstein Scheme)*

Description

Simulation the general hyperbolic diffusion by milstein scheme.

Usage

```
Hyprocg(N, M, t0, T, x0, beta, gamma, theta, mu, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`beta`	constant (`0.5*sigma^2*(beta-(gamma*X(t))/sqrt(theta^2+(X(t)-mu)^2)):drift coefficient`).
`gamma`	constant positive (`0.5*sigma^2*(beta-(gamma*X(t))/sqrt(theta^2+(X(t)-mu)^2)):drift coefficient`).
`theta`	constant positive (`0.5*sigma^2*(beta-(gamma*X(t))/sqrt(theta^2+(X(t)-mu)^2)):drift coefficient`).
`mu`	constant (`0.5*sigma^2*(beta-(gamma*X(t))/sqrt(theta^2+(X(t)-mu)^2)):drift coefficient`).
`sigma`	constant positive (`sigma :diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

A process `X` satisfying :

$$dX(t) = (0.5 * sigma^2 * (beta - (gamma * X(t))/sqrt(theta^2 + (X(t) - mu)^2)) * dt + dW(t)$$

With `(0.5*sigma^2*(beta-(gamma*X(t))/sqrt(theta^2+(X(t)-mu)^2)):drift coefficient` and `sigma :diffusion coefficient`, `W(t)` is Wiener process, discretization `dt = (T-t0)/N`.

The parameters `gamma > 0` and `0 <= abs(beta) < gamma` determine the shape of the distribution, and `theta >= 0`, and `mu` are, respectively, the scale and location parameters of the distribution.

Constraints: `gamma > 0` , `0 <= abs(beta) < gamma` , `theta >= 0` , `sigma > 0`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`Hyproc` Hyperbolic Process, `CIRhy` modified CIR and hyperbolic Process, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Hyperbolic Process
## dX(t) = 0.5 * (2)^2*(0.25-(0.5*X(t))/sqrt(2^2+(X(t)-1)^2)) *dt + 2* dW(t)
## One trajectorie
Hyprocg(N=1000,M=1,T=100,t0=0,x0=-10,beta=0.25,gamma=0.5,theta=2,mu=1,sigma=2)
## flow of Hyprocg
Hyprocg(N=1000,M=10,T=100,t0=0,x0=-10,beta=0.25,gamma=0.5,theta=2,mu=1,sigma=2)
## Output in Excel 2007
Hyprocg(N=1000,M=10,T=100,t0=0,x0=-10,beta=0.25,gamma=0.5,theta=2,mu=1,sigma=2,
output=TRUE)
```

`INFSR` *Creating Ahn and Gao model or Inverse of Feller Square Root Models (by Milstein Scheme)*

Description

Simulation the inverse of feller square root model by milstein scheme.

Usage

```
INFSR(N, M, t0, T, x0, theta, r, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`theta`	constant (`X(t)*(theta-(sigma^3-theta*r)*X(t)) :drift coefficient`).
`r`	constant (`X(t)*(theta-(sigma^3-theta*r)*X(t)) :drift coefficient`).
`sigma`	constant positive (`sigma * X(t)^(3/2) :diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

A process X satisfying :

$$dX(t) = X(t) * (theta - (sigma^3 - theta * r) * X(t)) * dt + sigma * X(t)^(3/2) * dW(t)$$

With `X(t)*(theta-(sigma^3-theta*r)*X(t)) :drift coefficient` and `sigma * X(t)^(3/2) :diffusion coefficient`, `W(t)` is Wiener process, discretization `dt = (T-t0)/N`.

The conditional distribution of this process is related to that of the Cox-Ingersoll-Ross `(CIR)` model.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`CEV` Constant Elasticity of Variance Models, `CIR` Cox-Ingersoll-Ross Models, `CIRhy` modified CIR and hyperbolic Process, `CKLS` Chan-Karolyi-Longstaff-Sanders Models, `DWP` Double-Well Potential Model, `GBM` Model of Black-Scholes, `HWV` Hull-White/Vasicek Models, `JDP` Jacobi Diffusion Process, `PDP` Pearson Diffusions Process, `ROU` Radial Ornstein-Uhlenbeck Process, `diffBridge` Diffusion Bridge Models, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Inverse of Feller Square Root Models
## dX(t) = X(t)*(0.5-(1^3-0.5*0.5)*X(t)) * dt + 1 * X(t)^(3/2) * dW(t)
## One trajectorie
INFSR(N=1000,M=1,T=50,t0=0,x0=0.5,theta=0.5,r=0.5,sigma=1)
## flow of IFSR
INFSR(N=1000,M=10,T=50,t0=0,x0=0.5,theta=0.5,r=0.5,sigma=1)
## Output in Excel 2007
INFSR(N=1000,M=10,T=50,t0=0,x0=0.5,theta=0.5,r=0.5,sigma=1,output=TRUE)
```

`JDP` *Creating The Jacobi Diffusion Process (by Milstein Scheme)*

Description

Simulation the jacobi diffusion process by milstein scheme.

Usage

```
JDP(N, M, t0, T, x0, theta, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`theta`	constant positive.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The Jacobi diffusion process is the solution to the stochastic differential equation :

$$dX(t) = -theta * (X(t) - 0.5) * dt + sqrt(theta * X(t) * (1 - X(t))) * dW(t)$$

With `-theta * (X(t) - 0.5) :drift coefficient` and `sqrt( theta*X(t)*(1-X(t))) :diffusion coefficient`, `W(t)` is Wiener process, discretization `dt = (T-t0)/N`.

For `theta > 0`. It has an invariant distribution that is `uniform on [0,1]`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`CEV` Constant Elasticity of Variance Models, `CIR` Cox-Ingersoll-Ross Models, `CIRhy` modified CIR and hyperbolic Process, `CKLS` Chan-Karolyi-Longstaff-Sanders Models, `DWP` Double-Well Potential Model, `GBM` Model of Black-Scholes, `HWV` Hull-White/Vasicek Models, `INFSR` Inverse of Feller s Square Root models, `PDP` Pearson Diffusions Process, `ROU` Radial Ornstein-Uhlenbeck Process, `diffBridge` Diffusion Bridge Models, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Jacobi Diffusion Process
## dX(t) = -0.05 * (X(t)-0.5)* dt + sqrt(0.05*X(t)*(1-X(t))) * dW(t),
## One trajectorie
JDP(N=1000,M=1,T=100,t0=0,x0=0,theta=0.05)
## flow of JDP
JDP(N=1000,M=5,T=100,t0=0,x0=0,theta=0.05)
## Output in Excel 2007
JDP(N=1000,M=5,T=100,t0=0,x0=0,theta=0.05,output=TRUE)
```

`MartExp` *Creating The Exponential Martingales Process*

Description

Simulation the exponential martingales.

Usage

```
MartExp(N, t0, T, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`sigma`	constant positive (`sigma is volatility`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

That is to say `W(t)` a Brownian movement the following processes are continuous martingales :

1. `X(t) = W(t)^2 - t.`
2. `Y(t) = exp( integral(f(s)dW(s),0,t)- 0.5 * integral(f(s)^2 ds,0,t) ).`

Value

data.frame(time,x,y) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

Examples

```
## Exponential Martingales Process
MartExp(N=1000,t0=0,T=1,sigma=2)
## Output in Excel 2007
MartExp(N=1000,t0=0,T=1,sigma=2,output=TRUE)
```

OU *Creating Ornstein-Uhlenbeck Process*

Description

Simulation the ornstein-uhlenbeck or Hull-White/Vasicek model.

Usage

```
OU(N, t0, T, x0, r, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`r`	constant positive (`r is speed of reversion` and `-r * X(t) :drift coefficient`).
`sigma`	constant positive (`sigma (volatility) :diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The Ornstein-Uhlenbeck or Vasicek process is the unique solution to the following stochastic differential equation :

$$dX(t) = -r * X(t) * dt + sigma * dW(t)$$

With `-r * X(t) :drift coefficient` and `sigma : diffusion coefficient`, `W(t)` is Wiener process, the discretization `dt = (T-t0)/N`.

Please note that the process is stationary only if `r > 0`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`OUF` Flow of Ornstein-Uhlenbeck Process, `PEOU` Parametric Estimation of Ornstein-Uhlenbeck Model, `PEOUexp` Explicit Estimators of Ornstein-Uhlenbeck Model, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Ornstein-Uhlenbeck Process
## dX(t) = -2 * X(t) * dt + 1 *dW(t)
OU(N=1000,t0=0,T=10,x0=10,r=2,sigma=1)
## Output in Excel 2007
OU(N=1000,t0=0,T=10,x0=10,r=2,sigma=1,output=TRUE)
```

OUF — *Creating Flow of Ornstein-Uhlenbeck Process*

Description

Simulation flow of ornstein-uhlenbeck or Hull-White/Vasicek model.

Usage

```
OUF(N, M, t0, T, x0, r, sigma, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`r`	constant positive (`r is speed of reversion` and `-r * X(t) :drift coefficient`).
`sigma`	constant positive (`sigma (volatility) :diffusion coefficient`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

The Ornstein-Uhlenbeck or Vasicek process is the unique solution to the following stochastic differential equation :

$$dX(t) = -r * X(t) * dt + sigma * dW(t)$$

With `-r * X(t) :drift coefficient` and `sigma : diffusion coefficient`, `W(t)` is Wiener process, the discretization `dt = (T-t0)/N`.

Please note that the process is stationary only if `r > 0`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`OU` Ornstein-Uhlenbeck Process, `PEOU` Parametric Estimation of Ornstein-Uhlenbeck Model, `PEOUexp` Explicit Estimators of Ornstein-Uhlenbeck Model, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## Flow of Ornstein-Uhlenbeck Process
## dX(t) = -2 * X(t) * dt + 1 *dW(t)
OUF(N=1000,M=100,t0=0,T=1,x0=10,r=2,sigma=1)
## Output in Excel 2007
OUF(N=1000,M=100,t0=0,T=1,x0=10,r=2,sigma=1,output=TRUE)
```

PDP — *Creating Pearson Diffusions Process (by Milstein Scheme)*

Description

Simulation the pearson diffusions process by milstein scheme.

Usage

```
PDP(N, M, t0, T, x0, theta, mu, a, b, c, output = FALSE)
```

Arguments

`N`	size of process.
`M`	number of trajectories.
`t0`	initial time.
`T`	final time.
`x0`	initial value of the process at time `t0`.
`theta`	constant positive.
`mu`	constant.
`a`	constant.
`b`	constant.
`c`	constant.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Details

A class that further generalizes the Ornstein-Uhlenbeck and Cox-Ingersoll-Ross processes is the class of Pearson diffusion, the pearson diffusions process is the solution to the stochastic differential equation :

$$dX(t) = -theta * (X(t) - mu) * dt + sqrt(2 * theta * (a * X(t)^2 + b * X(t) + c)) * dW(t)$$

With `-theta *(X(t)-mu) :drift coefficient` and `sqrt( 2*theta*(a*X(t)^2 + b *X(t)+ c)) :diffusion coefficient`, `W(t)` is Wiener process, discretization `dt = (T-t0)/N`.

With `theta > 0` and `a`, `b`, and `c` such that the diffusion coefficient is well-defined i.e., the square root can be extracted for all the values of the state space of `X(t)`.

1. When the `diffusion coefficient = sqrt(2*theta*c)` i.e, `(a=0,b=0)`, we recover the Ornstein-Uhlenbeck process.
2. For `diffusion coefficient = sqrt(2*theta*X(t))` and `0 < mu <= 1` i.e, `(a=0,b=1,c=0)`, we obtain the Cox-Ingersoll-Ross process, and if `mu > 1` the invariant distribution is a `Gamma law` with scale parameter `1` and shape parameter `mu`.
3. For `a > 0` and `diffusion coefficient = sqrt(2*theta*a*(X(t)^2+1))` i.e, `(b=0,c=a)`, the invariant distribution always exists on the real line, and for `mu = 0` the invariant distribution is a scaled `t distribution` with `v=(1+a^(-1))` degrees of freedom and scale parameter `v^(-0.5)`, while for `mu =! 0` the distribution is a form of skewed `t distribution` that is called Pearson type `IV` distribution.
4. For `a > 0`, `mu > 0`, and `diffusion coefficient = sqrt(2*theta*a*X(t)^2)` i.e, `(b=0,c=0)`, the distribution is defined on the positive half line and it is an `inverse Gamma distribution` with shape parameter `1 + a^-1` and scale parameter `a/mu`.
5. For `a > 0`, `mu >= a`, and `diffusion coefficient = sqrt(2*theta*a*X(t)*(X(t)+1))` i.e, `(b=a,c=0)`, the invariant distribution is the scaled `F distribution` with `(2*mu)/a` and `(2/a)+2` degrees of freedom and scale parameter `mu / (a+1)`. For `0 < mu < 1`, some reflecting conditions on the boundaries are also needed.
6. If `a < 0` and `mu > 0` are such that `min(mu,1-mu) >= -a` and `diffusion coefficient = sqrt(2*theta*a*X(t)*(X(t)-1))` i.e, `(b=-a,c=0)`, the invariant distribution exists on the interval `[0,1]` and is a `Beta distribution` with parameters `-mu/a` and `(mu-1)/a`.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`CEV` Constant Elasticity of Variance Models, `CIR` Cox-Ingersoll-Ross Models, `CIRhy` modified CIR and hyperbolic Process, `CKLS` Chan-Karolyi-Longstaff-Sanders Models, `DWP` Double-Well Potential Model, `GBM` Model of Black-Scholes, `HWV` Hull-White/Vasicek Models, `INFSR` Inverse of Feller s Square Root models, `JDP` Jacobi Diffusion Process, `ROU` Radial Ornstein-Uhlenbeck Process, `diffBridge` Diffusion Bridge Models, `snssde` Simulation Numerical Solution of SDE.

Examples

```
## example 1
## theta = 5, mu = 10, (a=0,b=0,c=0.5)
## dX(t) = -5 *(X(t)-10)*dt + sqrt( 2*5*0.5)* dW(t)
PDP(N=1000,M=1,T=1,t0=0,x0=1,theta=5,mu=10,a=0,b=0,c=0.5)

## example 2
## theta = 0.1, mu = 0.25, (a=0,b=1,c=0)
## dX(t) = -0.1 *(X(t)-0.25)*dt + sqrt( 2*0.1*X(t))* dW(t)
PDP(N=1000,M=1,T=1,t0=0,x0=1,theta=0.1,mu=0.25,a=0,b=1,c=0)

## example 3
## theta = 0.1, mu = 1, (a=2,b=0,c=2)
```

```
## dX(t) = -0.1*(X(t)-1)*dt + sqrt( 2*0.1*(2*X(t)^2+2))* dW(t)
PDP(N=1000,M=1,T=1,t0=0,x0=1,theta=0.1,mu=1,a=2,b=0,c=2)

## example 4
## theta = 0.1, mu = 1, (a=2,b=0,c=0)
## dX(t) = -0.1*(X(t)-1)*dt + sqrt( 2*0.1*2*X(t)^2)* dW(t)
PDP(N=1000,M=1,T=1,t0=0,x0=1,theta=0.1,mu=1,a=2,b=0,c=0)

## example 5
## theta = 0.1, mu = 3, (a=2,b=2,c=0)
## dX(t) = -0.1*(X(t)-3)*dt + sqrt( 2*0.1*(2*X(t)^2+2*X(t)))* dW(t)
PDP(N=1000,M=1,T=1,t0=0,x0=0.1,theta=0.1,mu=3,a=2,b=2,c=0)

## example 6
## theta = 0.1, mu = 0.5, (a=-1,b=1,c=0)
## dX(t) = -0.1*(X(t)-0.5)*dt + sqrt( 2*0.1*(-X(t)^2+X(t)))* dW(t)
PDP(N=1000,M=1,T=1,t0=0,x0=0.1,theta=0.1,mu=0.5,a=-1,b=1,c=0)
```

PEABM *Parametric Estimation of Arithmetic Brownian Motion(Exact likelihood inference)*

Description

Parametric estimation of Arithmetic Brownian Motion

Usage

```
PEABM(X, delta, starts = list(theta, sigma), leve = 0.95)
```

Arguments

X	a numeric vector of the observed time-series values.
delta	the fraction of the sampling period between successive observations.
starts	named list. Initial values for optimizer.
leve	the confidence level required.

Details

This process solves the stochastic differential equation :

$$dX(t) = theta * dt + sigma * dW(t)$$

The conditional density `p(t,.|x)` is the density of a `Gaussian law` with `mean = x0 + theta * t` and `variance = sigma^2 * t`.

`R` has the `[dqpr]norm` functions to evaluate the density, the quantiles, and the cumulative distribution or generate pseudo random numbers from the normal distribution.

Value

coef	Coefficients extracted from the model.
AIC	A numeric value with the corresponding AIC.
vcov	A matrix of the estimated covariances between the parameter estimates in the linear or non-linear predictor of the model.
confint	A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

PEOU Parametric Estimation of Ornstein-Uhlenbeck Model, PEOUexp Explicit Estimators of Ornstein-Uhlenbeck Model, PEOUG Parametric Estimation of Hull-White/Vasicek Models, PEBS Parametric Estimation of model Black-Scholes.

Examples

```
## Parametric estimation of Arithmetic Brownian Motion.
## t0 = 0 ,T = 100
data(DATA3)
res <- PEABM(DATA3,delta=0.1,starts=list(theta=1,sigma=1),leve = 0.95)
res
ABMF(N=1000,M=10,t0=0,T=100,x0=DATA3[1],theta=res$coef[1],sigma=res$coef[2])
points(seq(0,100,length=length(DATA3)),DATA3,type="l",lwd=3,col="red")
```

PEBS *Parametric Estimation of Model Black-Scholes (Exact likelihood inference)*

Description

Parametric estimation of model Black-Scholes.

Usage

```
PEBS(X, delta, starts = list(theta, sigma), leve = 0.95)
```

Arguments

X	a numeric vector of the observed time-series values.
delta	the fraction of the sampling period between successive observations.
starts	named list. Initial values for optimizer.
leve	the confidence level required.

Details

The Black and Scholes, or geometric Brownian motion model solves the stochastic differential equation:

$$dX(t) = theta * X(t) * dt + sigma * X(t) * dW(t)$$

The conditional density function `p(t,.|x)` is `log-normal` with `mean = x * exp(theta*t)` and `variance = x^2 * exp(2*theta*t)*(exp(sigma^2 *t ) -1 )`.

`R` has the `[dqpr]lnorm` functions to evaluate the density, the quantiles, and the cumulative distribution or generate pseudo random numbers from the lognormal distribution.

Value

`coef`	Coefficients extracted from the model.
`AIC`	A numeric value with the corresponding AIC.
`vcov`	A matrix of the estimated covariances between the parameter estimates in the linear or non-linear predictor of the model.
`confint`	A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`PEABM` Parametric Estimation of Arithmetic Brownian Motion, `PEOU` Parametric Estimation of Ornstein-Uhlenbeck Model, `PEOUexp` Explicit Estimators of Ornstein-Uhlenbeck Model, `PEOUG` Parametric Estimation of Hull-White/Vasicek Models.

Examples

```
## Parametric estimation of model Black-Scholes.
## t0 = 0 ,T = 1
data(DATA2)
res <- PEBS(DATA2,delta=0.001,starts=list(theta=2,sigma=1))
res
GBMF(N=1000,M=10,T=1,t0=0,x0=DATA2[1],theta=res$coef[1],sigma=res$coef[2])
points(seq(0,1,length=length(DATA2)),DATA2,type="l",lwd=3,col="red")
```

PEOU — *Parametric Estimation of Ornstein-Uhlenbeck Model (Exact likelihood inference)*

Description

Parametric estimation of Ornstein-Uhlenbeck Model.

Usage

```
PEOU(X, delta, starts = list(r, sigma), leve = 0.95)
```

Arguments

X	a numeric vector of the observed time-series values.
delta	the fraction of the sampling period between successive observations.
starts	named list. Initial values for optimizer.
leve	the confidence level required.

Details

This process solves the stochastic differential equation :

$$dX(t) = -r * X(t) * dt + sigma * dW(t)$$

It is ergodic for `r > 0`. We have also shown its exact conditional and stationary densities. In particular, the conditional density `p(t,.|x)` is the density of a `Gaussian law` with `mean = x0 * exp(-r*t)` and `variance = ((sigma^2)/(2*r))*(1-exp(-2*r*t))`.

`R` has the `[dqpr]norm` functions to evaluate the density, the quantiles, and the cumulative distribution or generate pseudo random numbers from the normal distribution.

Value

coef	Coefficients extracted from the model.
AIC	A numeric value with the corresponding AIC.
vcov	A matrix of the estimated covariances between the parameter estimates in the linear or non-linear predictor of the model.
confint	A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

PEABM Parametric Estimation of Arithmetic Brownian Motion, PEOUexp Explicit Estimators of Ornstein-Uhlenbeck Model, PEOUG Parametric Estimation of Hull-White/Vasicek Models, PEBS Parametric Estimation of model Black-Scholes.

Examples

```
## Parametric estimation of Ornstein-Uhlenbeck Model.
## t0 = 0 ,T = 10
data(DATA1)
res <- PEOU(DATA1,delta=0.01,starts=list(r=2,sigma=1),leve = 0.90)
res
OUF(N=1000,M=10,t0=0,T=10,x0=DATA1[1],r=res$coef[1],sigma=res$coef[2])
points(seq(0,10,length=length(DATA1)),DATA1,type="l",lwd=3,col="red")
```

PEOUexp — *Parametric Estimation of Ornstein-Uhlenbeck Model (Explicit Estimators)*

Description

Explicit estimators of Ornstein-Uhlenbeck Model.

Usage

```
PEOUexp(X, delta)
```

Arguments

`X`	a numeric vector of the observed time-series values.
`delta`	the fraction of the sampling period between successive observations.

Details

This process solves the stochastic differential equation :

$$dX(t) = -r * X(t) * dt + sigma * dW(t)$$

It is ergodic for `r > 0`.

We have also shown its exact conditional and stationary densities. In particular, the conditional density `p(t,.|x)` is the density of a `Gaussian law` with `mean = x0 * exp(-r*t)` and `variance = ((sigma^2)/(2*r))*(1-exp(-2*r*t))`, the maximum likelihood estimator of `r` is available in explicit form and takes the form :

$$r = -(1/dt) * log(sum(X(t) * X(t-1))/sum(X(t-1)^2))$$

which is defined only if `sum(X(t)*X(t-1)) > 0`, this estimator is consistent and asymptotically Gaussian.

The maximum likelihood estimator of :

$$sigma^2 = (2 * r)/(N * (1 - exp(-2 * dt * r))) * sum(X(t) - X(t-1) * *exp(-dt * r))^2$$

Value

`r`	Estimator of speed of reversion.
`sigma`	Estimator of volatility.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`PEABM` Parametric Estimation of Arithmetic Brownian Motion, `PEOU` Parametric Estimation of Ornstein-Uhlenbeck Model, `PEOUG` Parametric Estimation of Hull-White/Vasicek Models, `PEBS` Parametric Estimation of model Black-Scholes.

Examples

```
## t0 = 0 ,T = 10
data(DATA1)
res <- PEOUexp(DATA1,delt=0.01)
res
OUF(N=1000,M=10,t0=0,T=10,x0=DATA1[1],r=res$r,sigma=res$sigma)
points(seq(0,10,length=length(DATA1)),DATA1,type="l",lwd=3,col="red")
```

PEOUG — *Parametric Estimation of Hull-White/Vasicek (HWV) Gaussian Diffusion Models(Exact likelihood inference)*

Description

Parametric estimation of Hull-White/Vasicek Model.

Usage

```
PEOUG(X, delta, starts = list(r, theta, sigma), leve = 0.95)
```

Arguments

`X`	a numeric vector of the observed time-series values.
`delta`	the fraction of the sampling period between successive observations.
`starts`	named list. Initial values for optimizer.
`leve`	the confidence level required.

Details

the Vasicek or Ornstein-Uhlenbeck model solves the stochastic differential equation :

$$dX(t) = r * (theta - X(t)) * dt + sigma * dW(t)$$

It is ergodic for `r > 0`. We have also shown its exact conditional and stationary densities. In particular, the conditional density `p(t,.|x)` is the density of a `Gaussian law` with `mean = theta+(x0-theta)*exp(-r*t)` and `variance = (sigma^2/(2*r))*(1-exp(-2*r*t))`.

`R` has the `[dqpr]norm` functions to evaluate the density, the quantiles, and the cumulative distribution or generate pseudo random numbers from the normal distribution.

Value

`coef`	Coefficients extracted from the model.
`AIC`	A numeric value with the corresponding AIC.
`vcov`	A matrix of the estimated covariances between the parameter estimates in the linear or non-linear predictor of the model.
`confint`	A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

PEABM Parametric Estimation of Arithmetic Brownian Motion, PEOUexp Explicit Estimators of Ornstein-Uhlenbeck Model, PEOU Parametric Estimation of Ornstein-Uhlenbeck Model, PEBS Parametric Estimation of model Black-Scholes.

Examples

```
## example 1
## t0 = 0 ,T = 10
data(DATA1)
res <- PEOUG(DATA1,delta=0.01,starts=list(r=2,theta=0,sigma=1))
res
HWVF(N=1000,M=10,t0=0,T=10,x0=DATA1[1],r=res$coef[1],theta=res$coef[2],sigma=res$coef[3])
points(seq(0,10,length=length(DATA1)),DATA1,type="l",lwd=3,col="red")
```

ROU — *Creating Radial Ornstein-Uhlenbeck Process (by Milstein Scheme)*

Description

Simulation the radial ornstein-uhlenbeck process by milstein scheme.

Usage

```
ROU(N, M, t0, T, x0, theta, output = FALSE)
```

Arguments

N	size of process.
M	number of trajectories.
t0	initial time.
T	final time.
x0	initial value of the process at time t0.
theta	constant positive.
output	if output = TRUE write a output to an Excel 2007.

Details

The radial Ornstein-Uhlenbeck process is the solution to the stochastic differential equation :

$$dX(t) = (theta * X(t)^{-}(1) - X(t)) * dt + dW(t)$$

With (theta * X(t)^-1 - X(t)) :drift coefficient and 1 :diffusion coefficient, the discretization dt = (T-t0)/N, W(t) is Wiener process.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

CEV Constant Elasticity of Variance Models, CIR Cox-Ingersoll-Ross Models, CIRhy modified CIR and hyperbolic Process, CKLS Chan-Karolyi-Longstaff-Sanders Models, DWP Double-Well Potential Model, GBM Model of Black-Scholes, HWV Hull-White/Vasicek Models, INFSR Inverse of Feller s Square Root models, JDP Jacobi Diffusion Process, PDP Pearson Diffusions Process, diffBridge Diffusion Bridge Models, snssde Simulation Numerical Solution of SDE.

Examples

```
## Radial Ornstein-Uhlenbeck
## dX(t) = (0.05*X(t)^(-1) - X(t)) *dt + dW(t)
## One trajectorie
ROU(N=1000,M=1,T=1,t0=0,x0=1,theta=0.05)
## flow of POU
ROU(N=1000,M=10,T=1,t0=0,x0=1,theta=0.05)
## Output in Excel 2007
ROU(N=1000,M=10,T=1,t0=0,x0=1,theta=0.05,output=TRUE)
```

---

snssde *Simulation Numerical Solution of Stochastic Differential Equation*

Description

Different methods of simulation of solutions to stochastic differential equations.

Usage

```
snssde(N, M, T, t0, x0, Dt, drift, diffusion, Output = c(FALSE,
       TRUE),Methods = c("SchEuler","SchMilstein",
       "SchMilsteinS","SchTaylor","SchHeun","SchRK3"), ...)
```

Arguments

N	size of process.
M	number of trajectories.
T	final time.
t0	initial time.
x0	initial value of the process at time t0.
Dt	time step of the simulation (discretization).
drift	drift coefficient: an expression of two variables t and x.
diffusion	diffusion coefficient: an expression of two variables t and x.
Output	if Output = TRUE write a Output to an Excel 2007.
Methods	method of simulation ,see details.
...	

Details

The function `snssde` returns a trajectory of the process; i.e., `x0` and the new `N` simulated values if `M = 1`. For `M > 1`, an mts (multidimensional trajectories) is returned, which means that `M` independent trajectories are simulated. `Dt` the best discretization `Dt = (T-t0)/N`.

Simulation methods are usually based on discrete approximations of the continuous solution to a stochastic differential equation. The methods of approximation are classified according to their different properties. Mainly two criteria of optimality are used in the literature: the strong and the weak (orders of) convergence. The `methods` of simulation can be one among: `Euler Order 0.5`, `Milstein Order 1`, `Milstein Second-Order`, `Ito-Taylor Order 1.5`, `Heun Order 2`, `Runge-Kutta Order 3`.

Value

data.frame(time,x) and plot of process.

Note

- If `methods` is not specified, it is assumed to be the `Euler Scheme`.
- If `T` and `t0` specified, the best discretization `Dt = (T-t0)/N`.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`diffBridge` Creating Diffusion Bridge Models.

Examples

```
## example 1
## Hull-White/Vasicek Model
## T = 1 , t0 = 0 and N = 1000 ===> Dt = 0.001
drift     <- expression( (3*(2-x)) )
diffusion <- expression( (2) )
snssde(N=1000,M=1,T=1,t0=0,x0=10,Dt=0.001,
drift,diffusion,Output=TRUE)
## Multiple trajectories of the OU process by Euler Scheme
snssde(N=1000,M=10,T=1,t0=0,x0=10,Dt=0.001,
drift,diffusion,Output=FALSE)

## example 2
## Black-Scholes models
## T = 1 , t0 = 0 and N = 1000 ===> Dt = 0.001
drift     <- expression( (3*x) )
diffusion <- expression( (2*x) )
snssde(N=1000,M=1,T=1,t0=0,x0=10,Dt=0.001,drift,
diffusion,Output=FALSE,Methods="SchMilstein")
## Multiple trajectories of the BS process by Milstein Scheme
snssde(N=1000,M=10,T=1,t0=0,x0=10,Dt=0.001,drift,
diffusion,Output=FALSE,Methods="SchMilstein")

## example 3
## Constant Elasticity of Variance (CEV) Models
```

```
## T = 1 , t0 = 0 and N = 1000 ===> Dt = 0.001
## Multiple trajectories of the CEV process by Milstein Second Scheme
drift     <- expression( (0.3*x) )
diffusion <- expression( (0.2*x^0.75) )
snssde(N=1000,M=10,T=1,t0=0,x0=1,Dt=0.001,drift,
diffusion,Output=FALSE,Methods="SchMilsteinS")

## example 4
## sde\ dX(t)=(0.03*t*X(t)-X(t)^3)*dt+0.1*dW(t)
## Multiple trajectories of sde by Ito-Taylor Scheme
## T = 100 , t0 = 0 and N = 1000 ===> Dt = 0.1
drift     <- expression( (0.03*t*x-x^3) )
diffusion <- expression( (0.1) )
snssde(N=1000,M=20,T=100,t0=0,x0=0,Dt=0.1,drift,
diffusion,Output=FALSE,Methods="SchTaylor")

## example 5
## sde\ dX(t)=cos(t*x)*dt+sin(t*x)*dW(t) by Heun Scheme
drift     <- expression( (cos(t*x)) )
diffusion <- expression( (sin(t*x)) )
snssde(N=1000,M=1,T=100,t0=0,x0=0,Dt=0.1,drift,
diffusion,Output=FALSE,Methods="SchHeun")

## example 6
## sde\ dX(t)=exp(t)*dt+tan(t)*dW(t) by Runge-Kutta Scheme
drift     <- expression( (exp(t)) )
diffusion <- expression( (tan(t)) )
snssde(N=1000,M=1,T=1,t0=0,x0=1,Dt=0.001,drift,
diffusion,Output=FALSE,Methods="SchRK3")
```

`SRW` *Creating Random Walk*

Description

Simulation random walk.

Usage

```
SRW(N, t0, T, p, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`p`	probability of choosing `X = -1 or +1`.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

Stgamma Stochastic Process The Gamma Distribution, Stst Stochastic Process The Student Distribution, WNG White Noise Gaussian.

Examples

```
## Random Walk
SRW(N=1000,t0=0,T=1,p=0.5)
SRW(N=1000,t0=0,T=1,p=0.25)
SRW(N=1000,t0=0,T=1,p=0.75)
## Output in Excel 2007
SRW(N=1000,t0=0,T=1,p=0.5,output=TRUE)
```

Stgamma *Creating Stochastic Process The Gamma Distribution*

Description

Simulation stochastic process by a gamma distribution.

Usage

```
Stgamma(N, t0, T, alpha, beta, output = FALSE)
```

Arguments

N	size of process.
t0	initial time.
T	final time.
alpha	constant positive.
beta	an alternative way to specify the scale.
output	if output = TRUE write a output to an Excel 2007.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

SRW Creating Random Walk, Stst Stochastic Process The Student Distribution, WNG White Noise Gaussian.

Examples

```
## Stochastic Process The Gamma Distribution
Stgamma(N=1000,t0=0,T=5,alpha=1,beta=1)
## Output in Excel 2007
Stgamma(N=1000,t0=0,T=5,alpha=1,beta=1,output=TRUE)
```

Stst — *Creating Stochastic Process The Student Distribution*

Description

Simulation stochastic process by a Student distribution.

Usage

```
Stst(N, t0, T, n, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`n`	degrees of freedom (`> 0,non-integer`).
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`SRW` Creating Random Walk, `Stgamma` Stochastic Process The Gamma Distribution, `WNG` White Noise Gaussian.

Examples

```
## Stochastic Process The Student Distribution
Stst(N=1000,t0=0,T=1,n=2)
## Output in Excel 2007
Stst(N=1000,t0=0,T=1,n=2,output=TRUE)
```

Telegproc — *Realization a Telegraphic Process*

Description

Simulation a telegraphic process.

Usage

```
Telegproc(t0, x0, T, lambda, output = FALSE)
```

Arguments

`t0`	initial time.
`x0`	state initial `(x0 = -1 or +1)`.
`T`	final time of the simulation.
`lambda`	exponential distribution with rate `lambda`.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Author(s)

boukhetala Kamal, guidoum Arsalane.

See Also

`Asys` Evolution a Telegraphic Process.

Examples

```
## Simulation a telegraphic process
Telegproc(t0=0,x0=1,T=1,lambda=0.5)
## Output in Excel 2007
Telegproc(t0=0,x0=1,T=1,lambda=0.5,output=TRUE)
```

WNG — *Creating White Noise Gaussian*

Description

Simulation white noise gaussian.

Usage

```
WNG(N, t0, T, m, sigma2, output = FALSE)
```

Arguments

`N`	size of process.
`t0`	initial time.
`T`	final time.
`m`	mean.
`sigma2`	variance.
`output`	if `output = TRUE` write a `output` to an Excel 2007.

Value

data.frame(time,x) and plot of process.

Author(s)

boukhetala Kamal, guidoum Arsalane.

Examples

```
## White Noise Gaussian
WNG(N=1000,t0=0,T=1,m=0,sigma2=4)
## Output in Excel 2007
WNG(N=1000,t0=0,T=1,m=0,sigma2=4,output=TRUE)
```

Index