

ExtendedOrnsteinUhlenbeckProcess(3)

QuantLib

ExtendedOrnsteinUhlenbeckProcess(3)

**NAME**

ExtendedOrnsteinUhlenbeckProcess – Extended Ornstein-Uhlenbeck process class.

**SYNOPSIS**

```
#include
<ql/experimental/processes/extendedornsteinuhlenbeckprocess.hpp>
Inherits StochasticProcess1D.
```

**Public Types**

```
enum Discretization { MidPoint, Trapezodial, GaussLobatto }
```

**Public Member Functions**

```
ExtendedOrnsteinUhlenbeckProcess (Real speed, Volatility sigma, Real x0, const boost::function<  
    Real(Real)> &b, Discretization discretization=MidPoint, Real intEps=1e-4)
```

**StochasticProcess interface**

**Real** **x0** () const

returns the initial value of the state variable

**Real** **speed** () const

**Real** **volatility** () const

**Real** **drift** (**Time** t, **Real** x) const

returns the drift part of the equation, i.e. \$ (t, x\_t) \$

**Real** **diffusion** (**Time** t, **Real** x) const

returns the diffusion part of the equation, i.e. \$ ma(t, x\_t) \$

**Real** **expectation** (**Time** t0, **Real** x0, **Time** dt) const

**Real** **stdDeviation** (**Time** t0, **Real** x0, **Time** dt) const

**Real** **variance** (**Time** t0, **Real** x0, **Time** dt) const

**Additional Inherited Members****Detailed Description**

Extended Ornstein-Uhlenbeck process class.

This class describes the Ornstein-Uhlenbeck process governed by  $dx = a(b(t) - x_t) dt + ma dW_t$ . ]

**Member Function Documentation**

**Real** **expectation** (**Time** t0, **Real** x0, **Time** dt) const [virtual]

returns the expectation \$ E(x\_{t\_0 + \text{process after a time interval}}) \$ given discretization. This method can be overridden in derived classes which want to hard-code a particular discretization.

Reimplemented from **StochasticProcess1D**.

**Real** **stdDeviation** (**Time** t0, **Real** x0, **Time** dt) const [virtual]

returns the standard deviation \$ S(x\_{t\_0 + \text{process after a time interval}}) \$ given discretization. This method can be overridden in derived classes which want to hard-code a particular discretization.

Reimplemented from **StochasticProcess1D**.

**Real** **variance** (**Time** t0, **Real** x0, **Time** dt) const [virtual]

returns the variance \$ V(x\_{t\_0 + \text{process after a time interval}}) \$ given discretization. This method can be overridden in derived classes which want to hard-code a particular discretization.

Reimplemented from **StochasticProcess1D**.

**Author**

Generated automatically by Doxygen for QuantLib from the source code.

