The classes are demonstrated below.

```
> library(stocins)
```

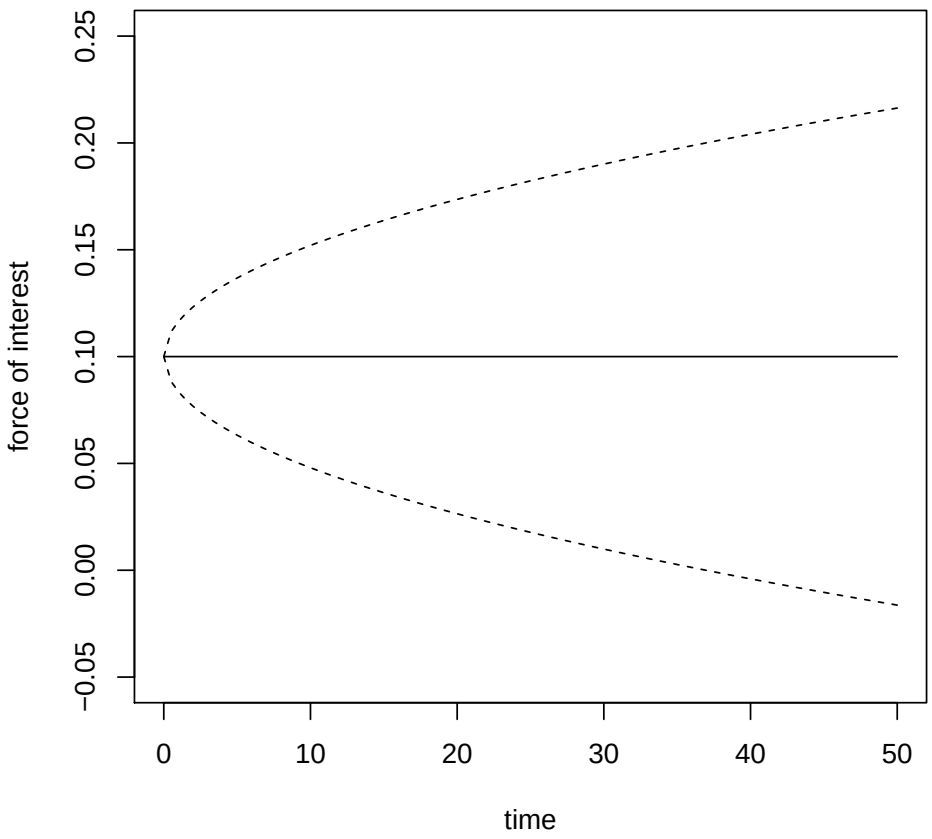
1 Interest Rate Models

1.1 Wiener Process

```
> wienermodel = iratemodel(list(delta0 = 0.05, sigma = 0.01),
+                          "gbm")
> pv.ev(10, wienermodel)

[1] 0.6167242

> pv.var(10, wienermodel)

[1] 0.01289196

> pv.cov(5,10,wienermodel)

[1] 0.005039818
```

A plot is shown below.

```
> wienermodel = iratemodel(list(delta0 = 0.10, sigma = 0.01),
+                          "gbm")
> plot(function(t) delta.ev(t, wienermodel), 0, 50, col = 'black',
+      ylim = c(-0.05, 0.25), xlab = "time", ylab = "force of interest")
> plot(function(t) delta.ev(t, wienermodel) -
+        1.645 * sqrt(delta.var(t, wienermodel)), 0, 50,
+      add = TRUE, lty = 2)
> plot(function(t) delta.ev(t, wienermodel) +
+        1.645 * sqrt(delta.var(t, wienermodel)), 0, 50,
+      add = TRUE, lty = 2)
```



1.2 Ornstein-Uhlenbeck Process

```
> oumodel = iratemodel(list(delta0 = 0.08, delta = 0.05,
+                           alpha = 0.1, sigma = 0.01), "ou")
> pv.ev(10, oumodel)
```

```
[1] 0.5059934
```

```
> pv.var(10, oumodel)
```

```
[1] 0.004340003
```

```
> pv.cov(5, 10, oumodel)
```

```
[1] 0.002095713
```

A plot is shown below.

```
> oumodel = iratemodel(list(delta0 = 0.1, delta = 0.06,
+                           alpha = 0.1, sigma = 0.01), "ou")
```

```
> plot(function(t) delta.ev(t, oumodel), 0, 50, col = 'black',
+      ylim = c(0, 0.12), xlab = "time", ylab = "force of interest")
> plot(function(t) delta.ev(t, oumodel) -
+        1.645 * sqrt(delta.var(t, oumodel)), 0, 50,
+      add = TRUE, lty = 2)
> plot(function(t) delta.ev(t, oumodel) +
+        1.645 * sqrt(delta.var(t, oumodel)), 0, 50,
+      add = TRUE, lty = 2)
```