CRAN packages comparison

1 DiffusionRgqd

Uses the cumulant truncation procedure developed by Varughese (2013), whereby the transition density can be approximated over arbitrarily large transition horizons for a suitably general class of non-linear diffusion models.

Generalized quadratic diffusions (GQD) are the specific class of SDEs with quadratic drift and diffusion terms:

$$\begin{aligned} dX_t &= \mu(X_t, t)dt + \sigma(X_t, t)dW_t, \text{ where} \\ \mu(X_t, t) &= G_0(t) + G_1(t)X_t + G_2(t)X_t^2, \text{ and} \\ \sigma(X_t, t) &= Q_0(t) + Q_1(t)X_t + Q_2(t)X_t^2 \end{aligned}$$

For purposes of inference the drift and diffusion terms - and consequently the transitional density - are assumed to be dependent on a vector of parameters, θ. For example, an Ornstein-Uhlenbeck model with SDE:

$$dX_t = \theta_1(\theta_2 - X_t) + \sqrt{\theta_3^2}dW_t \tag{1}$$

```
G0=function(t){theta[1]*theta[2]}
G1=function(t){-theta[1]}
Q0=function(t){theta[3]*theta[3]}
```

1.1 Constant drift, diffusion SDE

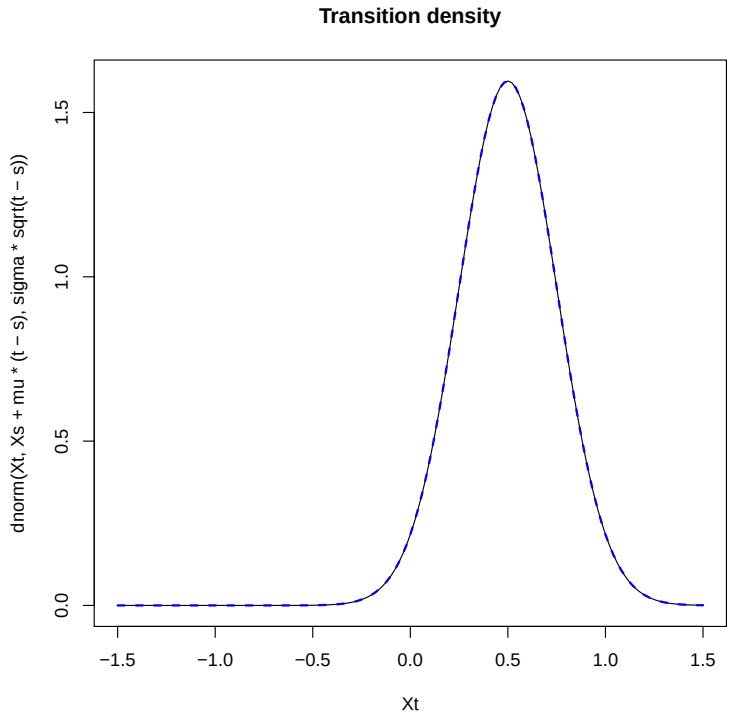
For a constant drift, diffusion SDE, with given initial condition X_s:

$$dX_t = \mu dt + \sigma dW_t \tag{2}$$

The distribution at time t of the process X_t is $\mathcal{N}(X_t, X_s + \mu(t-s), \sigma^2(t-s))$

```
library('DiffusionRgqd')

# Remove any existing coefficients
GQD.remove()

Xs <- 0                 # Initial state
Xt <- seq(-3/2,3/2,1/50)# Possible future states
s  <- 0                 # Starting time
t  <- 1                 # Final time
mu    <- 0.5            # Drift parameter
sigma <- 0.25           # Diffusion coefficient

# Define the model coefficients
G0 <- function(t){mu}
Q0 <- function(t){sigma^2}
```

```r
# Calculate the transitional density
BM <- GQD.density(Xs,Xt,s,t)

# Plot the transitional density
plot(dnorm(Xt, Xs+mu*(t-s), sigma*sqrt(t-s))~Xt, main = 'Transition density', type = 'l')
lines(BM$density[,100]~BM$Xt, col = 'blue', lty = 'dashed', lwd = 2)
```



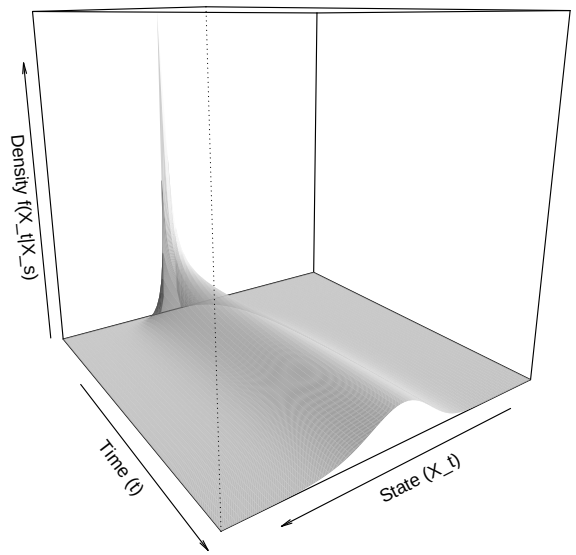
1.2 CIR process

Another example using the CIR process SDE:

$$dX_t = \theta_1(\theta_2 - X_t)dt + \theta_3\sqrt{X_t}dW_t \tag{3}$$

```r
GQD.remove()
a = 0.5; b = 5; sigma = 0.35; # Parameter values

G0 <- function(t){a*b}
G1 <- function(t){-a}
Q1 <- function(t){sigma^2}

states     <-  seq(1, 9, 1/10)# State values
initial    <-  6              # Starting value of the process
Tmax       <-  5              # Time horizon
Tstart     <-  1              # Time starts at 1
increment  <-  1/100          # Incremental time steps
```

```
# Generate the transitional density
M <- GQD.density(Xs = initial, Xt = states, s = Tstart, t = Tmax, delt = increment)

persp(x = M$Xt, y = M$time, z = M$density, col = 'white', xlab = 'State (X_t)',ylab
 = 'Time (t)', zlab = 'Density f(X_t|X_s)', border = NA, shade = 0.5, theta = 145)
```



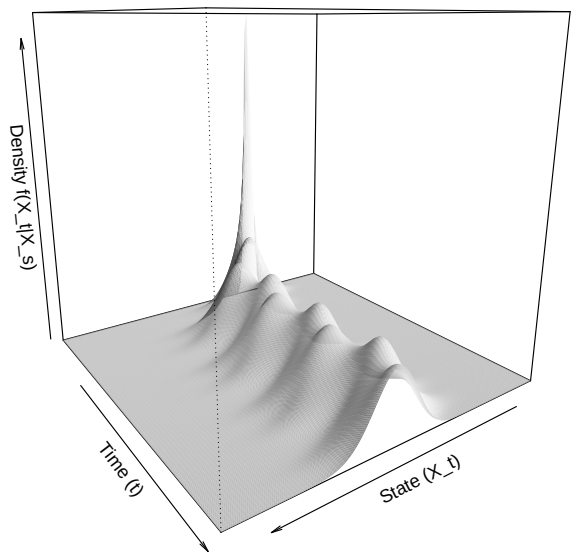
The GQD.density() returns (density, Xt, time, cumulants, moments, mesh).

1.3 Time dependent CIR process

Consider the time-inhomogeneous CIR process for a more complicated example

$$dX_t = 2(10 + \sin(2\pi(t - 0.5)) - X_t)dt + sqrt0.25(1 + 0.75\sin(4\pi t))X_t dW_t \tag{4}$$

```
library(DiffusionRgqd)
GQD.remove()
G0 <- function(t){2*(10+sin(2*pi*(t-0.5)))}
G1 <- function(t){-2}
Q1 <- function(t){0.25*(1+0.75*(sin(4*pi*t)))}

states    <- seq(5, 15, 1/10)
initial   <- 8
Tmax      <- 5
Tstart    <- 1
increment <- 1/100

M <- GQD.density(Xs = initial, Xt = states, s = Tstart, t = Tmax, delt = increment)
persp(x = M$Xt, y = M$time, z = M$density, col = 'white', xlab = 'State (X_t)', ylab = 'Time
```



1.4 Coupled SDEs - Prey predator model

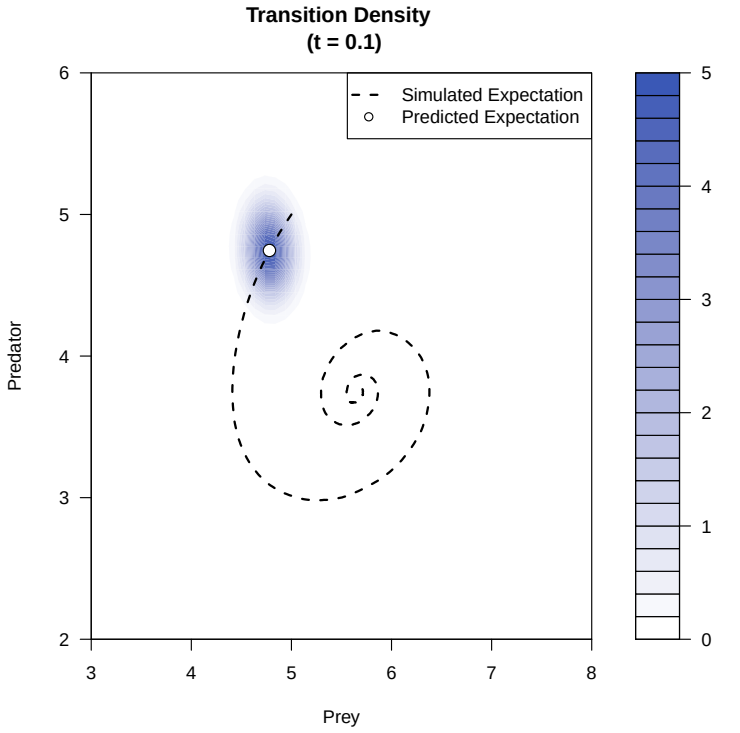
A model that is often used to illustrate non-linear dynamics in the analysis of ODEs is that of the Lotka-Volterra model. The equations are often used to describe the dynamics of two interacting populations wherein the population growth rate of the populations are mutually influenced by the current level of the opposing population. As such the model has been used to explain oscillatory behaviour in predator-prey relationships Hoppensteadt2006 where xtxt denotes the prey population and ytyt the predator population at time t. Continuing with the predator-prey metaphor, perhaps one deficiency of the model, one might argue, is the absence of random input and subsequent effects on population levels. Indeed, under the ODE formulation the predicted population behaviour (given fixed parameters) are completely deterministic. Another deficiency might be the absence of growth inhibiting factors such as disease or over-grazing. For these purposes we may define an example of a stochastic counterpart to the Lotka-Volterra equations as:

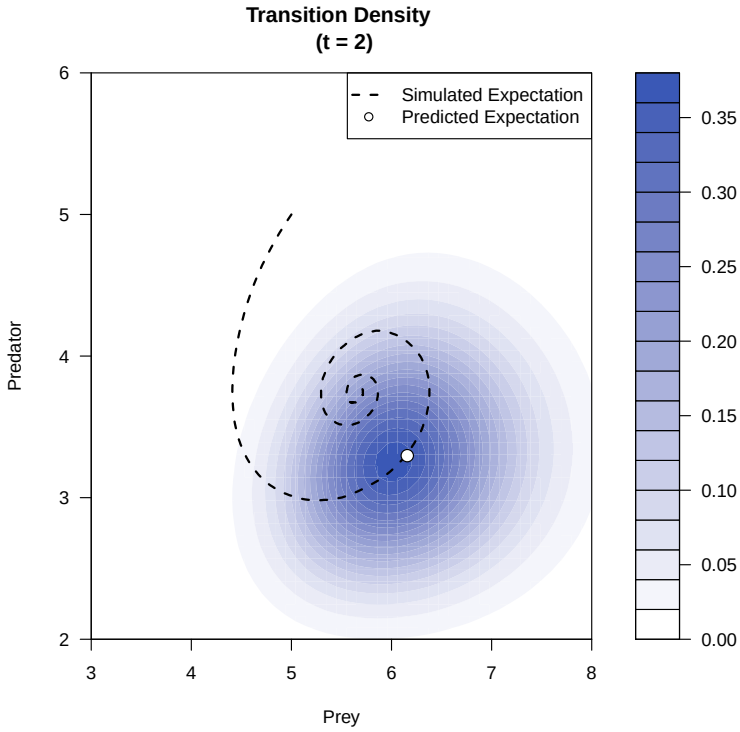
$$dX_t = (aX_t - bX_tY_t)dt + f\sqrt{X_t}dW_t^1$$
$$dY_t = (-cY_t + dX_tY_t - eY_t^2)dt + g\sqrt{Y_t}dW_t^2$$

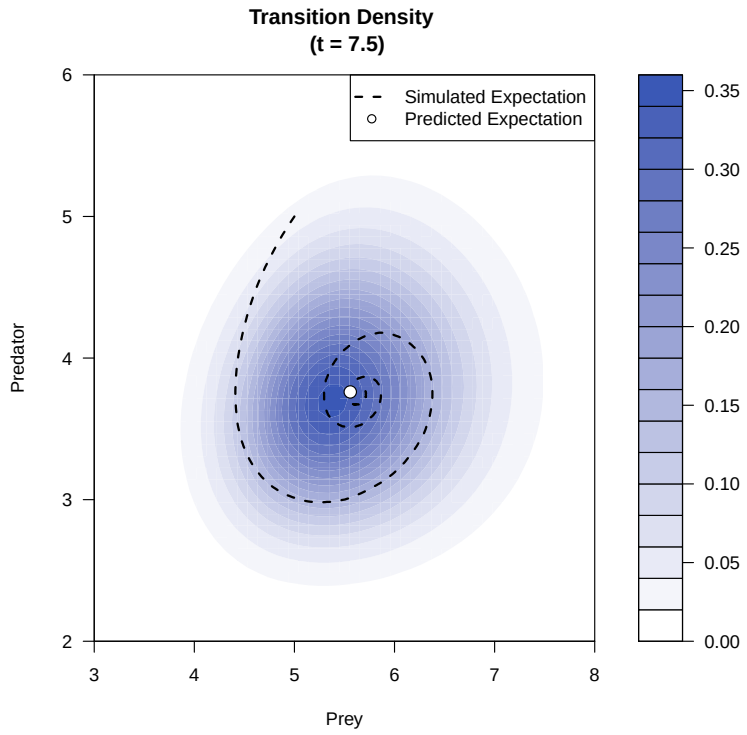
```
library(DiffusionRgqd)
# Remove any existing coefficients:
GQD.remove()

## [1] "Removed :  G0 G1 Q1"

# Define the X dimesnion coefficients:
a10 <- function(t){1.5}
```

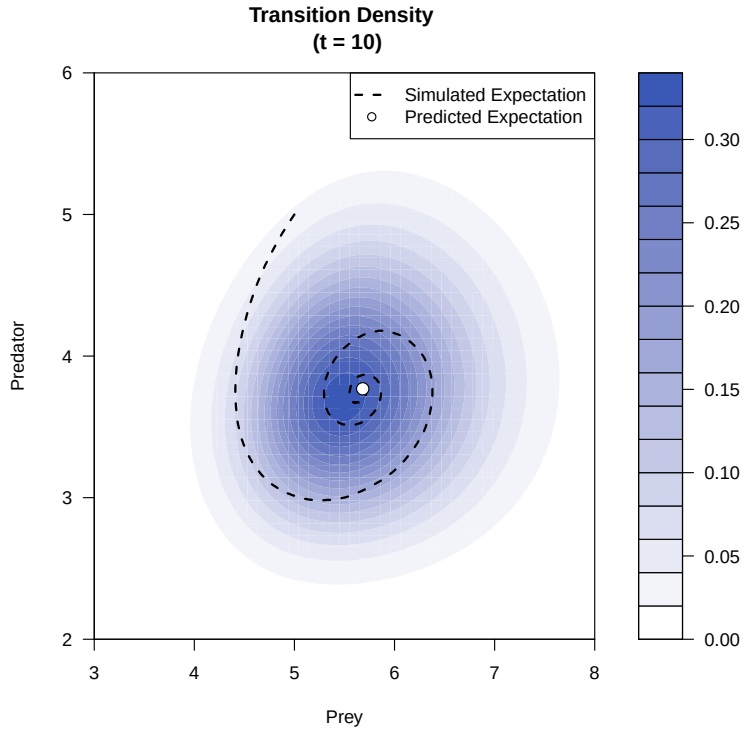
```r
a11 <- function(t){-0.4}
c10 <- function(t){0.05}
# Define the Y dimension coefficients:
b01 <- function(t){-1.5}
b11 <- function(t){0.4}
b02 <- function(t){-0.2}
f01 <- function(t){0.1}
# Approximate the transition density
res <- BiGQD.density(Xs = 5, Ys = 5, Xt = seq(3, 8, length = 50), Yt = seq(2, 6, length = 50)

##
##  ================================================================
##                     GENERALIZED QUADRATIC DIFFUSON
##  ================================================================
##  ______________________ Drift Coefficients ________________________
##  a10 : 1.5
##  a11 : -0.4
##  ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
##  b01 : -1.5
##  b02 : -0.2
##  b11 : 0.4
##  ____________________ Diffusion Coefficients ______________________
##  c10 : 0.05
##  ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
##  ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
##  ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
##  f01 : 0.1
## ================================================================
```

```
## Loading required package:  methods
```


Transition Density
(t = 0.1)
Simulated Expectation
Predicted Expectation
Predator
Prey
6
5
4
3
2
3
4
5
6
7
8
5
4
3
2
1
0


Transition Density
(t = 2)
Simulated Expectation
Predicted Expectation
Predator
Prey
6
5
4
3
2
3
4
5
6
7
8
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00



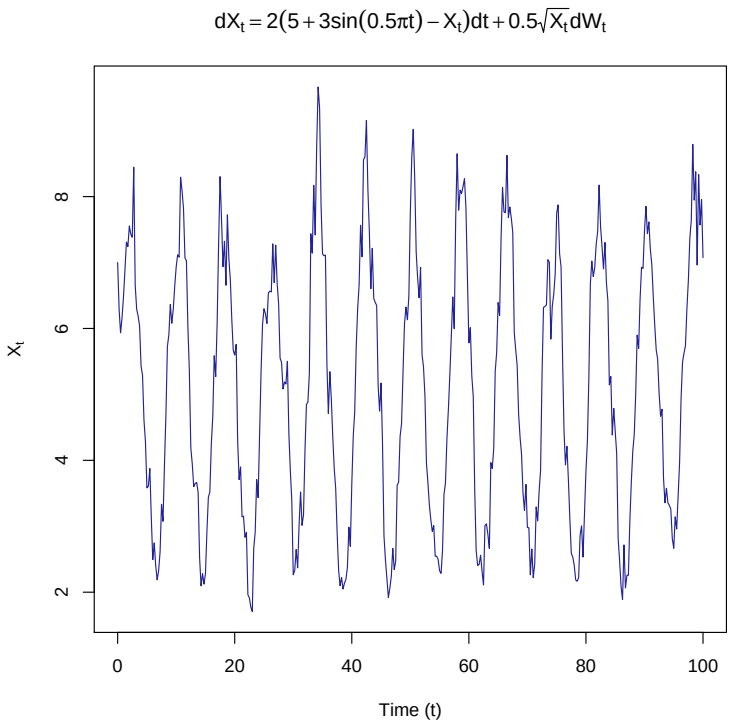


1.5 Inference on Diffusion Processes

```
library("DiffusionRgqd")
data(SDEsim1)
attach(SDEsim1)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     Xt
## The following object is masked from SDEsim3:
##
##     time
```

```
par(mfrow=c(1,1))
expr1=expression(dX[t]==2*(5+3*sin(0.5*pi*t)-X[t])*dt+0.5*sqrt(X[t])*dW[t])
plot(SDEsim1$Xt~SDEsim1$time, type = 'l', col = '#222299', xlab = 'Time (t)', ylab = expressi
```




```
GQD.remove()

G0 <- function(t){theta[1]*(theta[2]+theta[3]*sin(0.25*pi*t))}
G1 <- function(t){-theta[1]}
Q1 <- function(t){theta[4]*theta[4]}

theta   <- c(1, 10, 1, 1)               # Starting values for the chain
sds     <- c(0.25, 0.25, 0.2, 0.05)/1.5 # Std devs for proposal distributions
mesh    <- 10                           # Number of mesh points
updates <- 110000                       # Perform 110000 updates
burns   <- 10000                        # Burn 10000 updates

# Run the MCMC procedure for the model defined above:
model_1 <- GQD.mcmc(SDEsim1$Xt, SDEsim1$time, mesh, theta, sds, updates, burns)
```

The estimates are given by

```
GQD.estimates(model_1, thin = 100, burns = 10000, corrmat = TRUE)
```

2 pomp: statistical inference for partially-observed Markov processes

3 Robfilter

4 Sim.DiffProc Package - FitSDE

5 HPloglik

6 abctools