3nd Assignment: Continous Time Stochastic Modelling (SDEs)

David Ribberholt Ipsen (s164522)

November 2021

Part 1: Simulation and discretization of diffusion processes

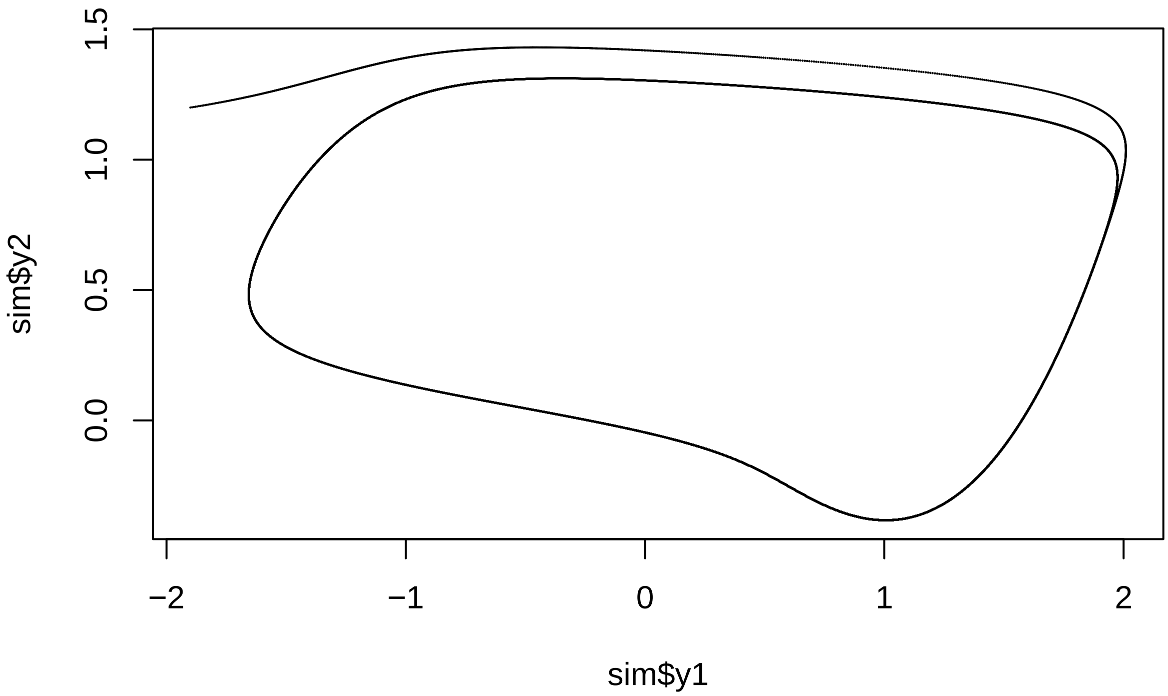
In the following exercise, the Euler-Marayama approximation to the Bonhoeffer-Van der Pol equations is simulated, e.g.:

$$Y_{n+1}^1 = Y_n^1 + \theta_3 \left(Y_n^1 + Y_n^2 - \frac{1}{3} \left(Y_n^1 \right)^3 + \theta_4 \right) \Delta + \sigma \Delta W_{n+1}^1$$
$$Y_{n+1}^2 = Y_n^2 - \frac{1}{\theta_3} \left(Y_n^1 + \theta_2 Y_n^2 - \theta_1 \right) \Delta$$

1a - Without stochasticity

Initially, I will simulate the deterministic, ODE version of the above system, e.g. where $\sigma = 0$. I decided to increase the length of the simulation to $0 \leq t \leq T = 1000$ with a discretization step of $\Delta = 2^{-9}$. I will use the parameter values as stated in the problem description.

```r
# Let's define a function to simulate the system
SDEsim = function(y1_init = -1.9, y2_init=1.2, delta = 2^(-9),
                  theta1=0.7, theta2=0.8, theta3=3.0, theta4=-0.34,
                  sigma=0, T=200/delta){
    # Init
  D = list()
  y1 = rep(NaN, T); y1[1] = y1_init;
  y2 = rep(NaN, T); y2[1] = y2_init;

    # Run the system
  for (i in 1:(T-1)){
    y1[i+1] = y1[i] + theta3*(y1[i] + y2[i] - 1/3*y1[i]^3 + theta4)*delta + sigma * rnorm(1,0,sqrt(delta
    y2[i+1] = y2[i] - 1/theta3 * (y1[i] + theta2*y2[i] - theta1) * delta
  }
    # Save and return list
  D$y1 = y1
  D$y2 = y2
  return(D)
}
sim = SDEsim(sigma=0)
plot(sim$y1, sim$y2, cex=0.01)
```

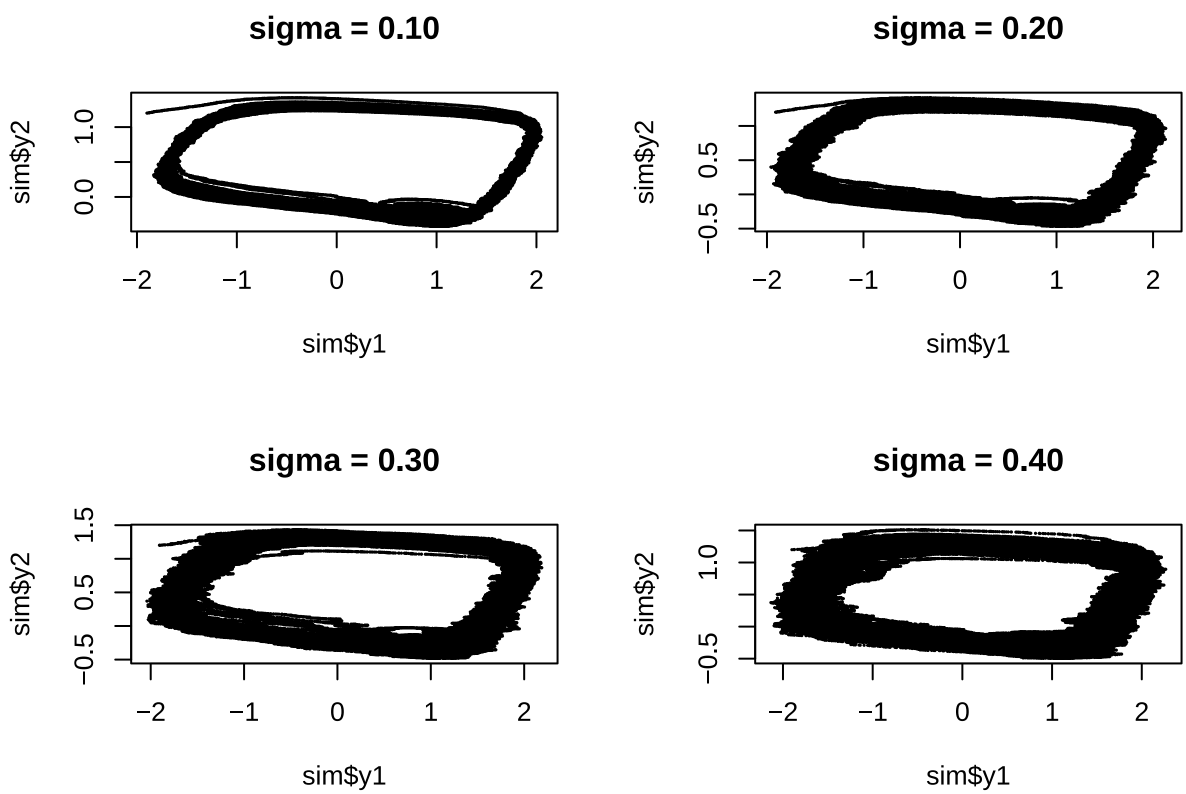


The system clearly enters a *limit cycle* after a bit of time.

Introducing stochasticity

Let's simulate the system with the following diffusion sizes: $\sigma \in \{0.1, 0.2, 0.3, 0.4\}$.

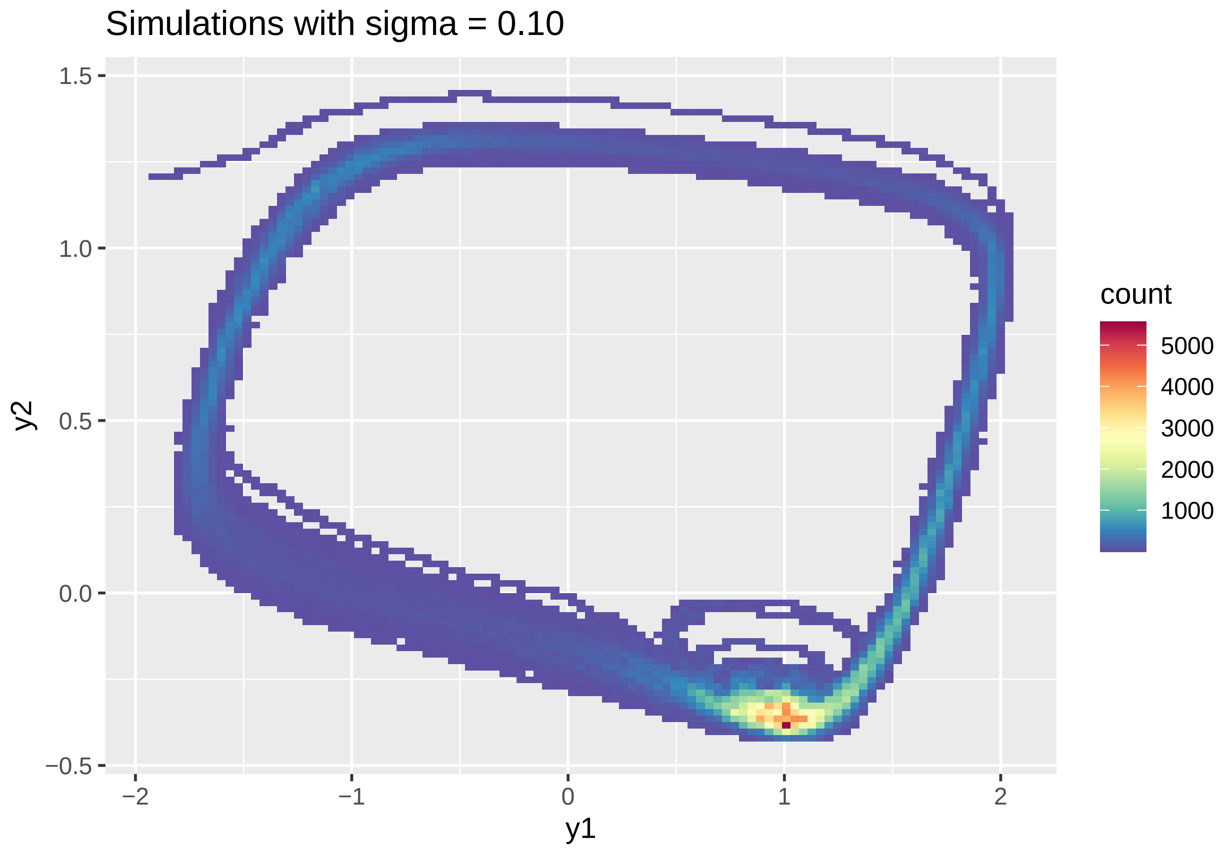
```
par(mfrow=c(2,2))
for (sd in c(0.1, 0.2, 0.3, 0.4)){
  sim = SDEsim(sigma=sd)
  plot(sim$y1, sim$y2, cex=0.1, main=sprintf("sigma = %.2f", sd))
}
```



Interestingly, there seems to a occur a bifurcation of another limit cycles with small perturbations in the system at $\sigma = 0.1$. However, with increasing diffusion size the system seems to generally circle around the ODE limit cycle. It's quite hard to see, so instead let's visualize the density of the trajectories using a heatmap.

1b

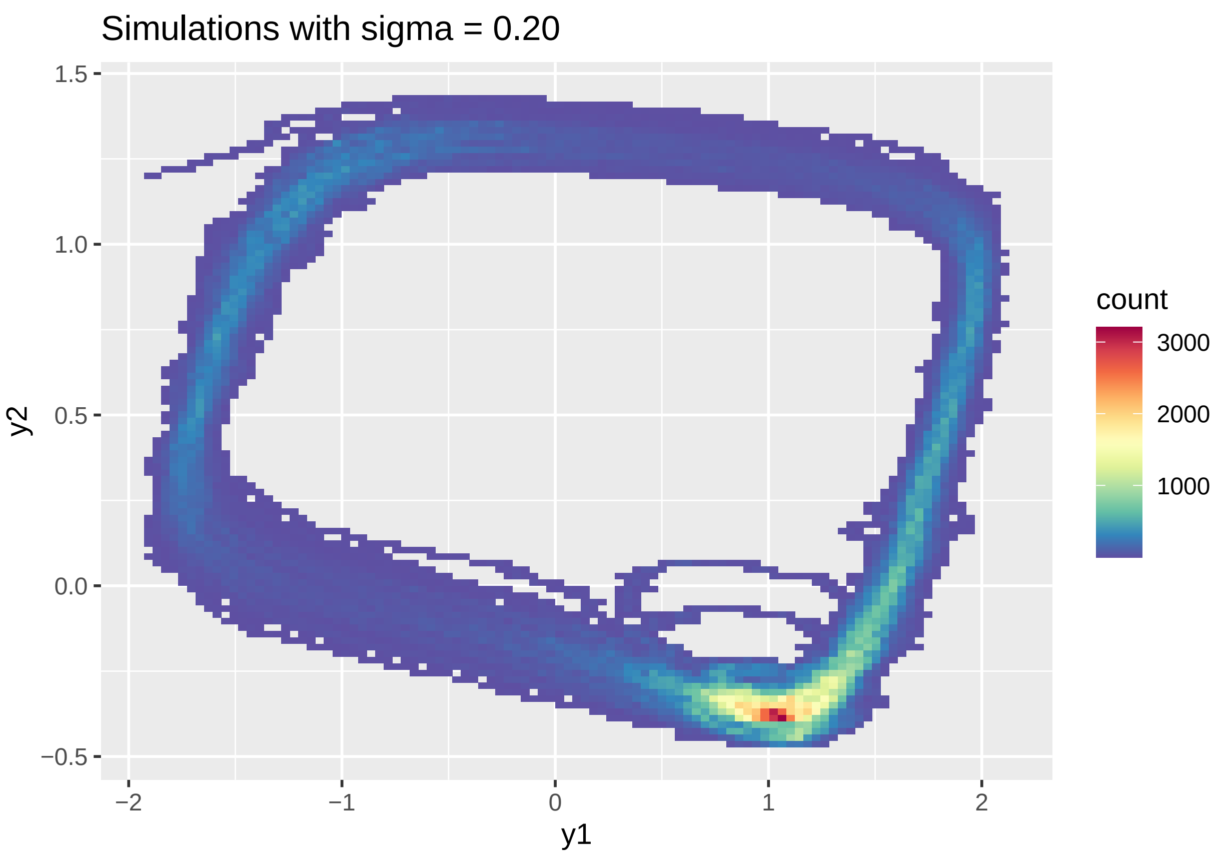
```
sim = data.frame(SDEsim(sigma=0.1))

# Ref: https://www.r-bloggers.com/2014/09/5-ways-to-do-2d-histograms-in-r/
#install.packages("ggplot2")
library(ggplot2)

# Default call (as object)
p <- ggplot(sim, aes(y1,y2))

# Add colouring and change bins
library(RColorBrewer)
rf <- colorRampPalette(rev(brewer.pal(11,'Spectral')))
r <- rf(32)
h3 <- p + stat_bin2d(bins=100) + scale_fill_gradientn(colours=r) + ggtitle(sprintf("Simulations with si
h3
```



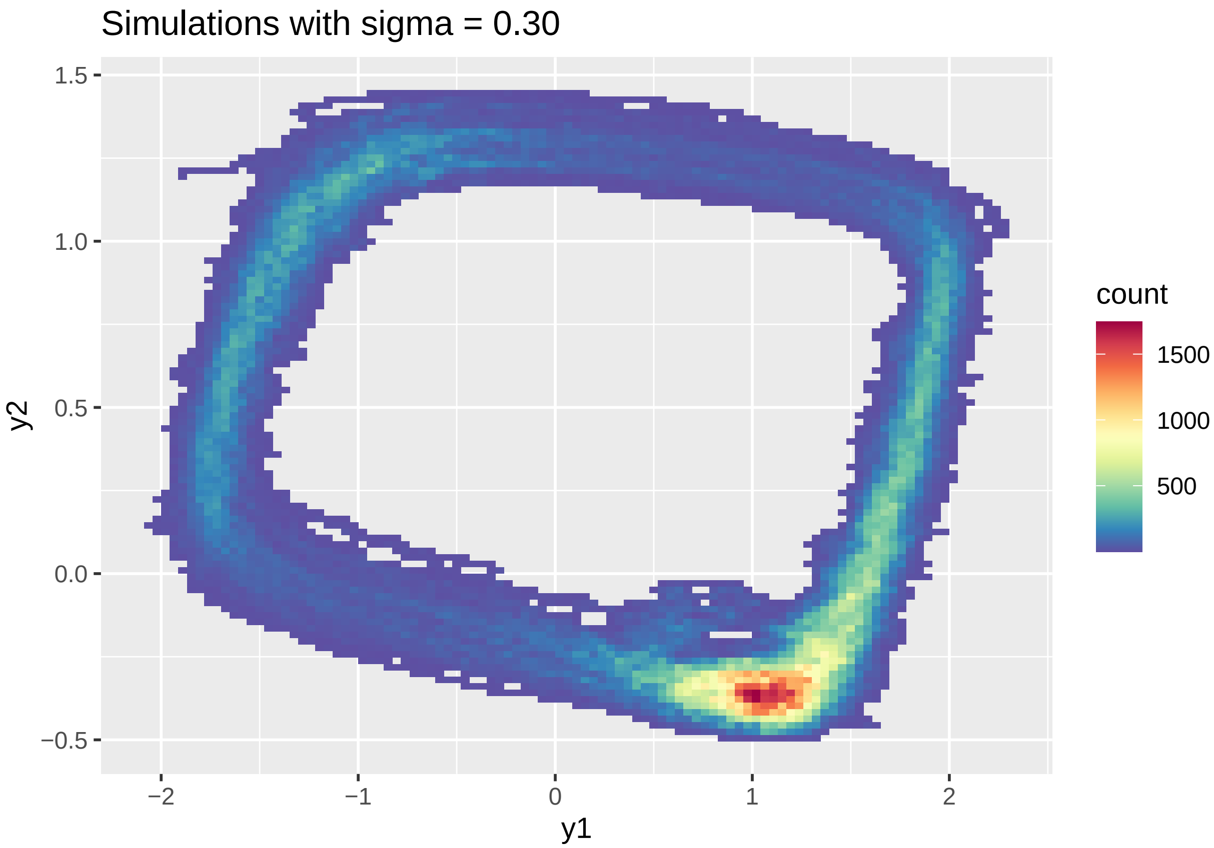
Here it is clearly seen that pertubations in the lower right corner results in new cycles and completely different trajectories, whereas pertubations in the other areas doesn't result in much different trajectories.

Similarly for $\sigma = \{0.2, 0.3, 0.4\}$

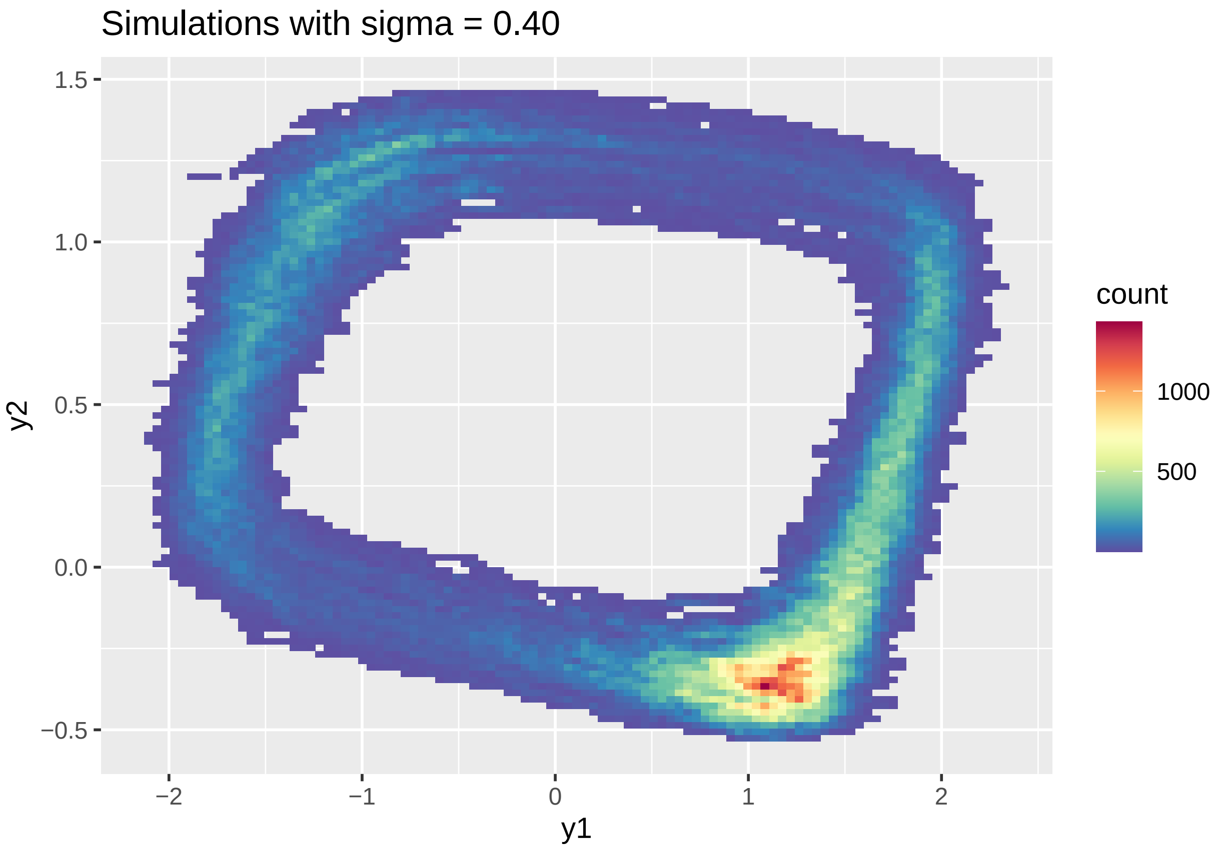
```
sim2 = data.frame(SDEsim(sigma=0.20))
sim3 = data.frame(SDEsim(sigma=0.30))
sim4 = data.frame(SDEsim(sigma=0.40))

# Default call (as object)
p2 <- ggplot(sim2, aes(y1,y2))
p3 <- ggplot(sim3, aes(y1,y2))
p4 <- ggplot(sim4, aes(y1,y2))

# Add colouring and change bins
library(RColorBrewer)
rf <- colorRampPalette(rev(brewer.pal(11,'Spectral')))
r <- rf(32)
(h2 <- p2 + stat_bin2d(bins=100) + scale_fill_gradientn(colours=r) + ggtitle(sprintf("Simulations with s
```




```
(h3 <- p3 + stat_bin2d(bins=100) + scale_fill_gradientn(colours=r) + ggtitle(sprintf("Simulations with s
```




```
(h4 <- p4 + stat_bin2d(bins=100) + scale_fill_gradientn(colours=r) + ggtitle(sprintf("Simulations with s
```



Clearly, with the increasing σ, the trajectories more often takes “the full round” as defined by the deterministic ODE.

END OF 3A