Lista 2 de exercícios

Exercícios do Livro

3.3.3

The correctness of the formula $u_k = (1 + h\lambda)^k u_0$ can be shown by induction. It holds for u_1, since $u_1 = u_0 + h(\lambda u_0)$. If $u_k = (1+h\lambda)^k u_0$ for some $k \in \mathbb{N}$, thenk $u_{k+1} = u_k + h(\lambda u_k) = u_k(1+h\lambda) = (1+h\lambda)^{k+1} u_0$.

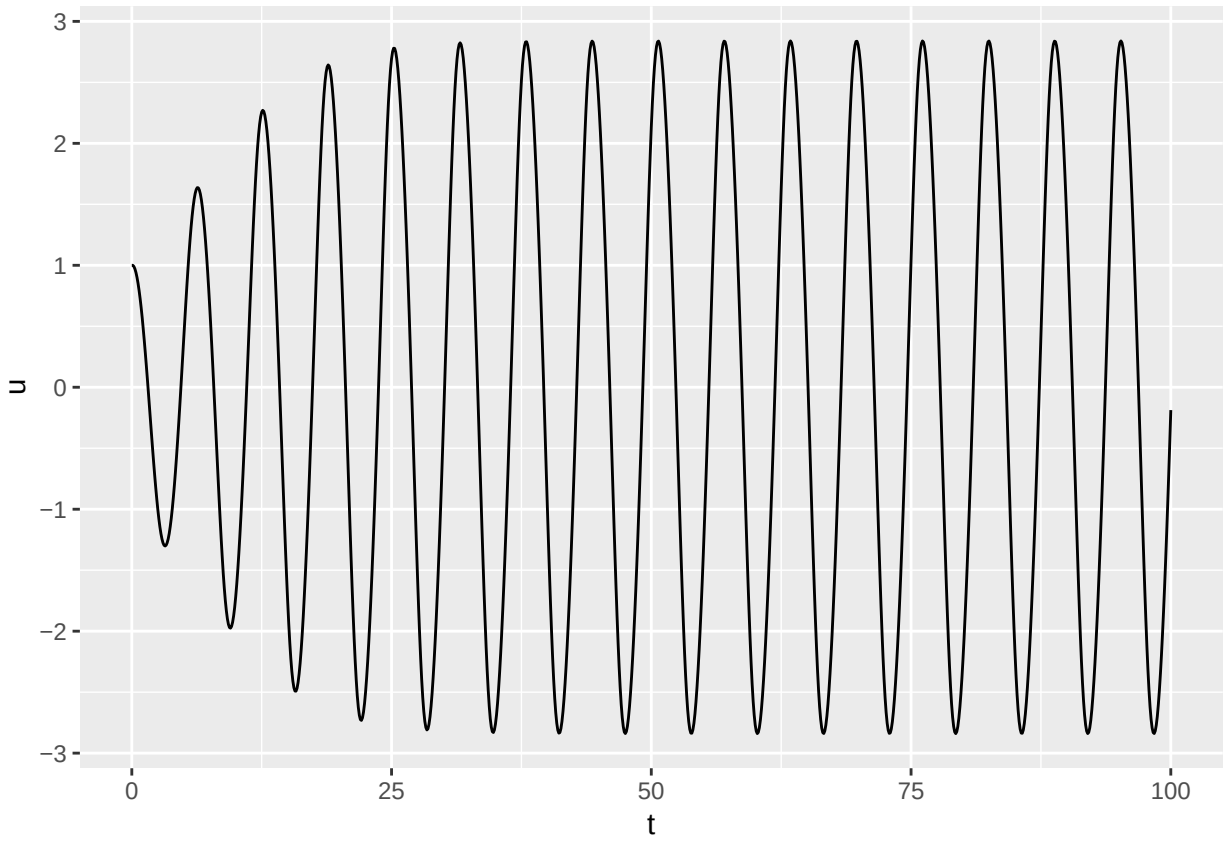
Implementações em linguagem de programação da aula 4:

Implementação do Forward Euler

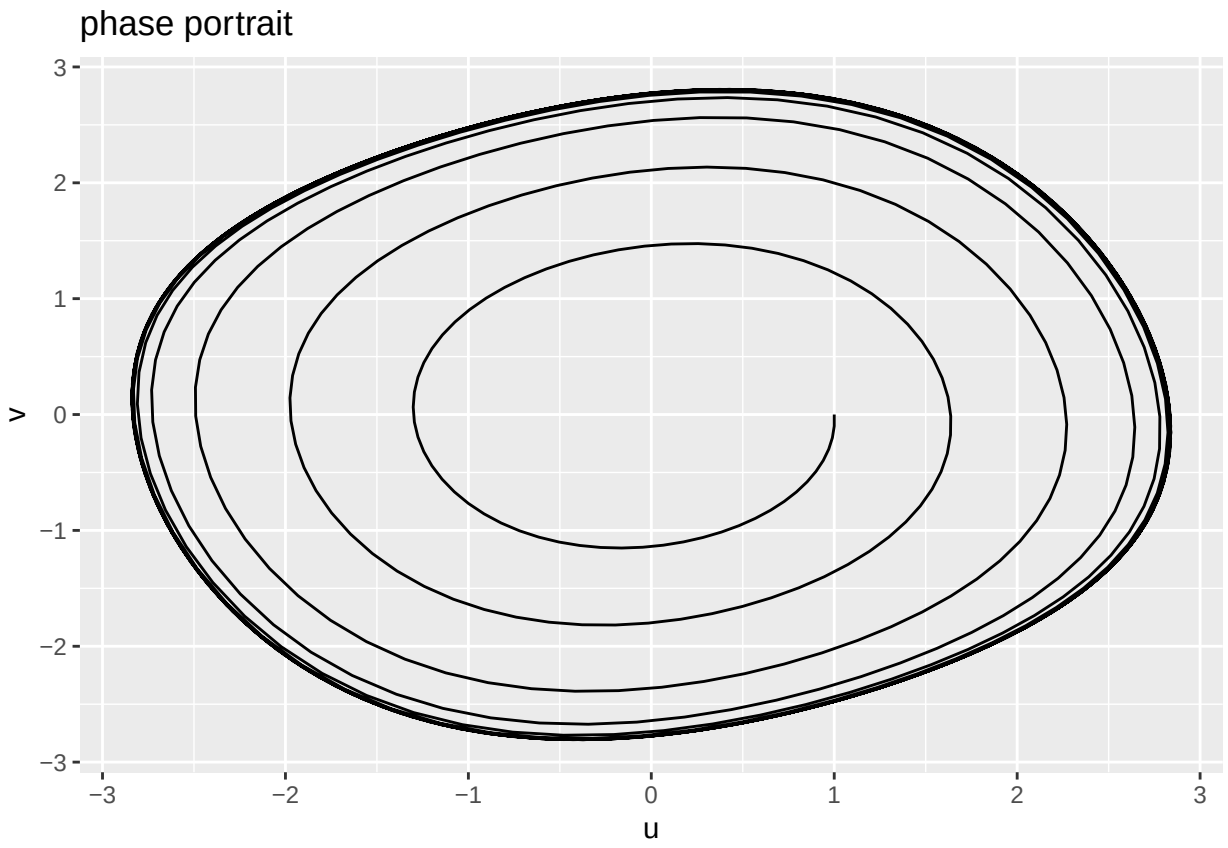
```
library(tidyverse)

euler = function(f, y0, a, b, h) {
  ts = seq(a, b, by = h)

  y = matrix(0, nrow = length(ts), ncol = length(y0))

  y[1,] = y0
  for (i in 1:(length(ts) - 1)) {
    y[i + 1,] = y[i,] + h*f(ts[i], y[i,])
  }

  m = cbind(ts, y)
  colnames(m) = c("t", if (is.null(names(y0))) paste0("y", 1:length(y0)) else names(y0))
  m
}
```

Equação de van der Pol

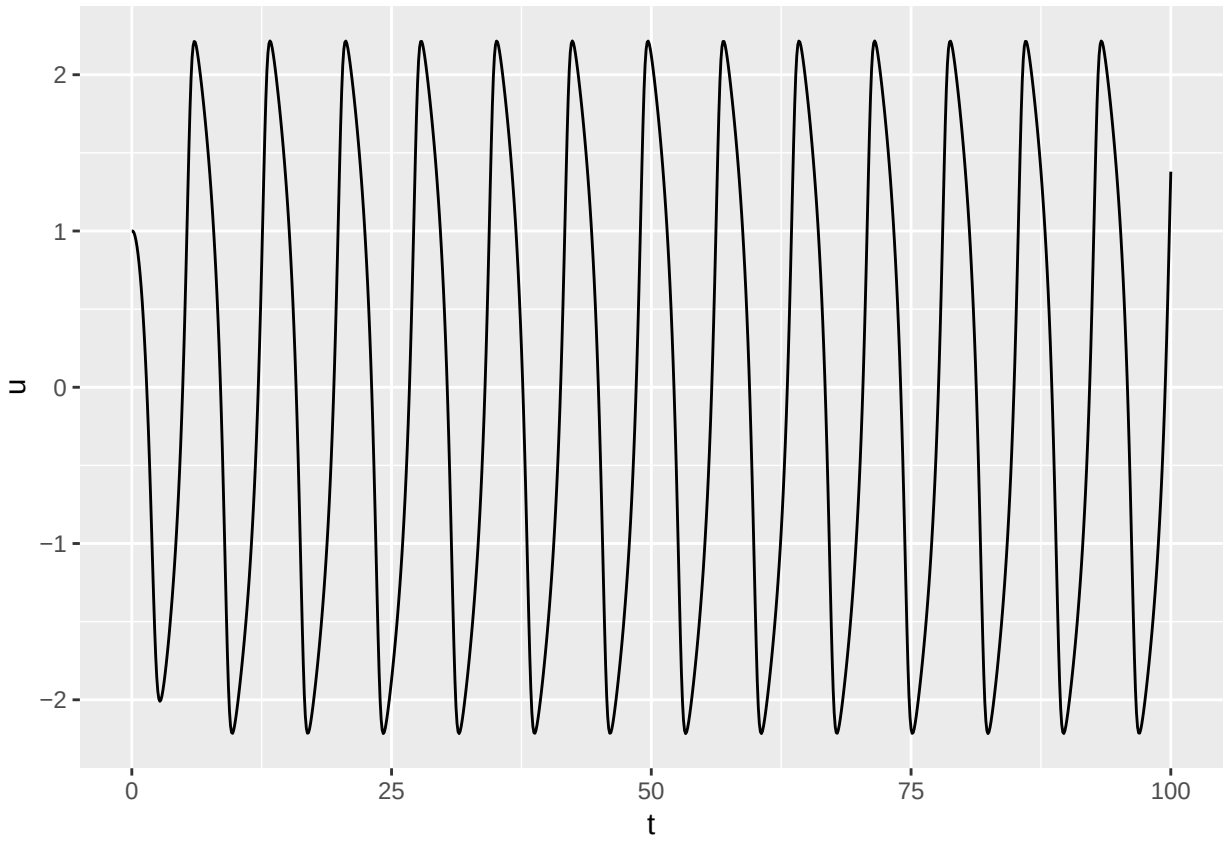
```
vanDerPol = function(epsilon) {
  function(t, y) c(y[2], -epsilon*(y[1]*y[1]-1)*y[2]-y[1])
}

tb = euler(vanDerPol(epsilon = 0.1), c("u" = 1, "v" = 0), 0, 100, h = 0.1) %>%
  as.data.frame

ggplot(tb, aes(x = t, y = u)) + geom_path()
```



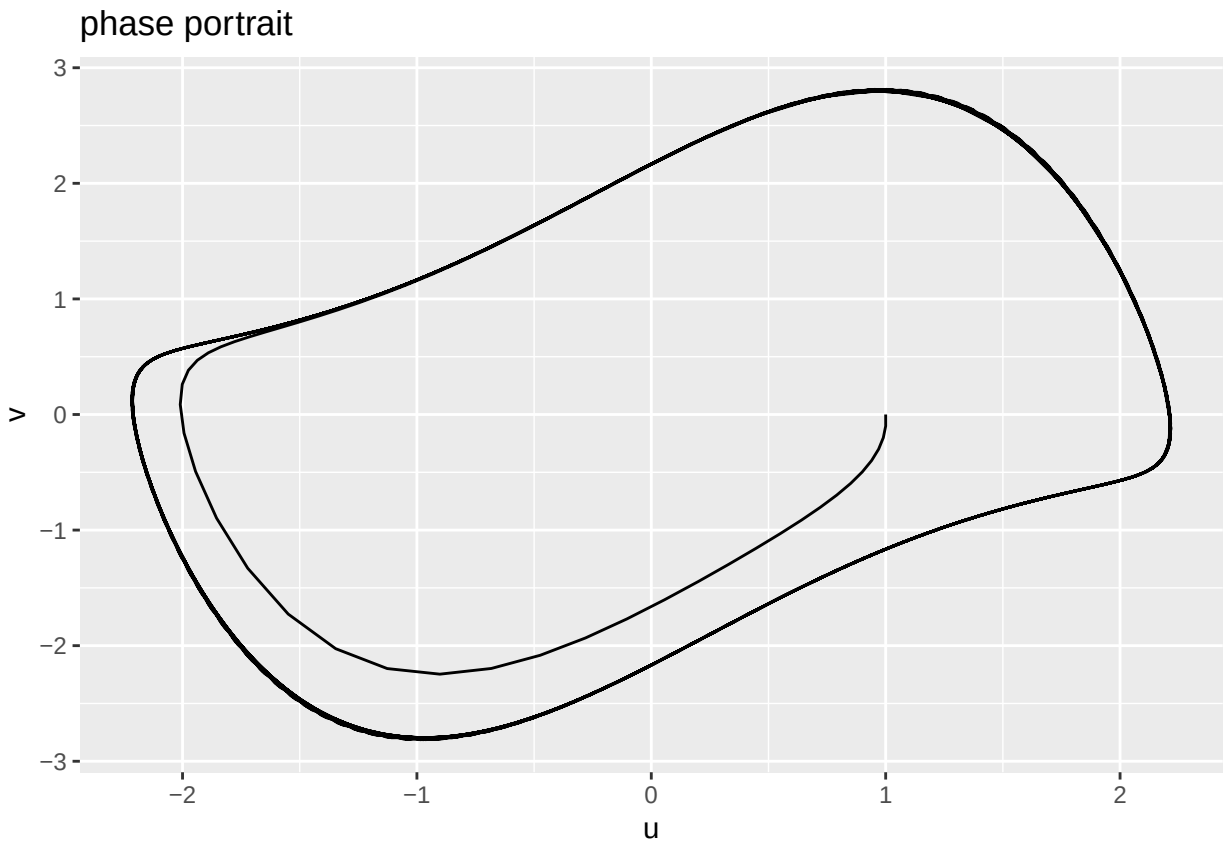

```
ggplot(tb, aes(x = u, y = v)) + geom_path() + ggtitle("phase portrait")
```



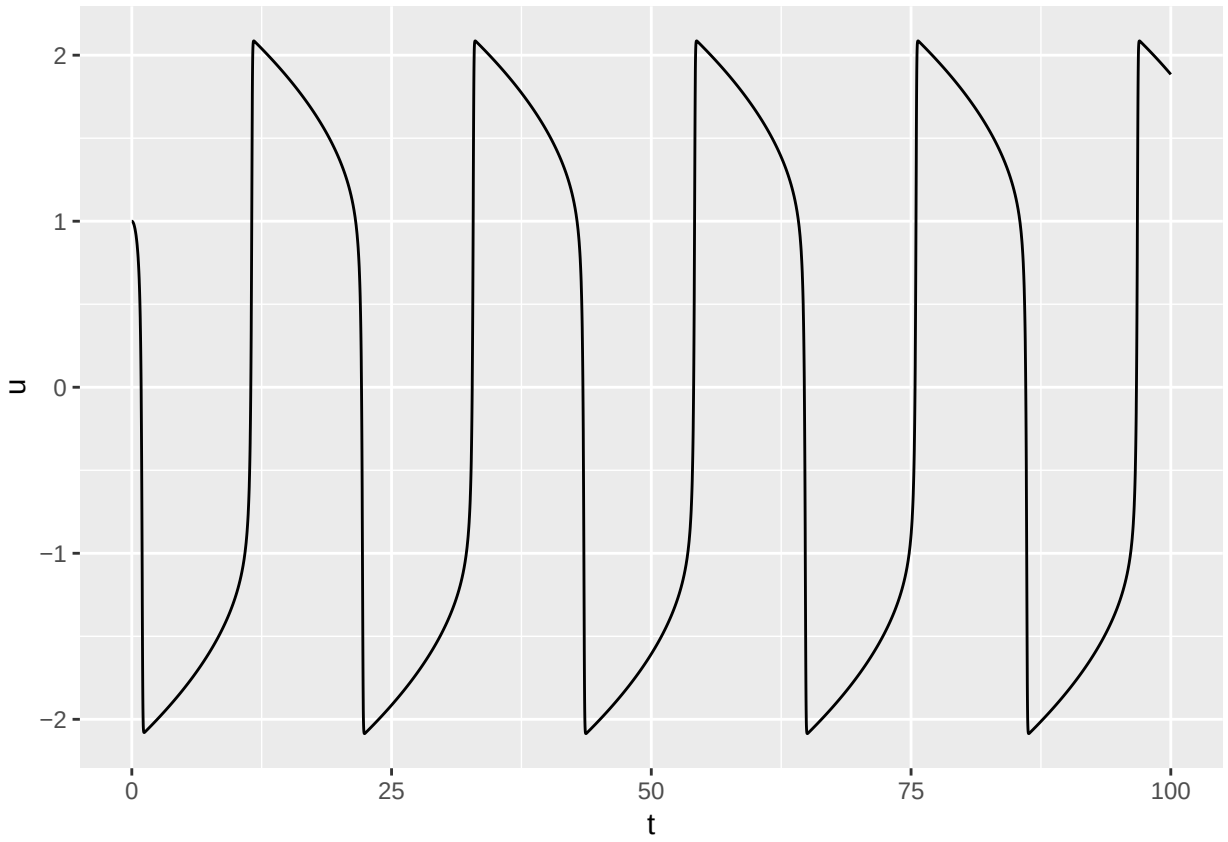

```
tb = euler(vanDerPol(epsilon = 1), c("u" = 1, "v" = 0), 0, 100, h = 0.1) %>%
  as.data.frame

ggplot(tb, aes(x = t, y = u)) + geom_path()
```



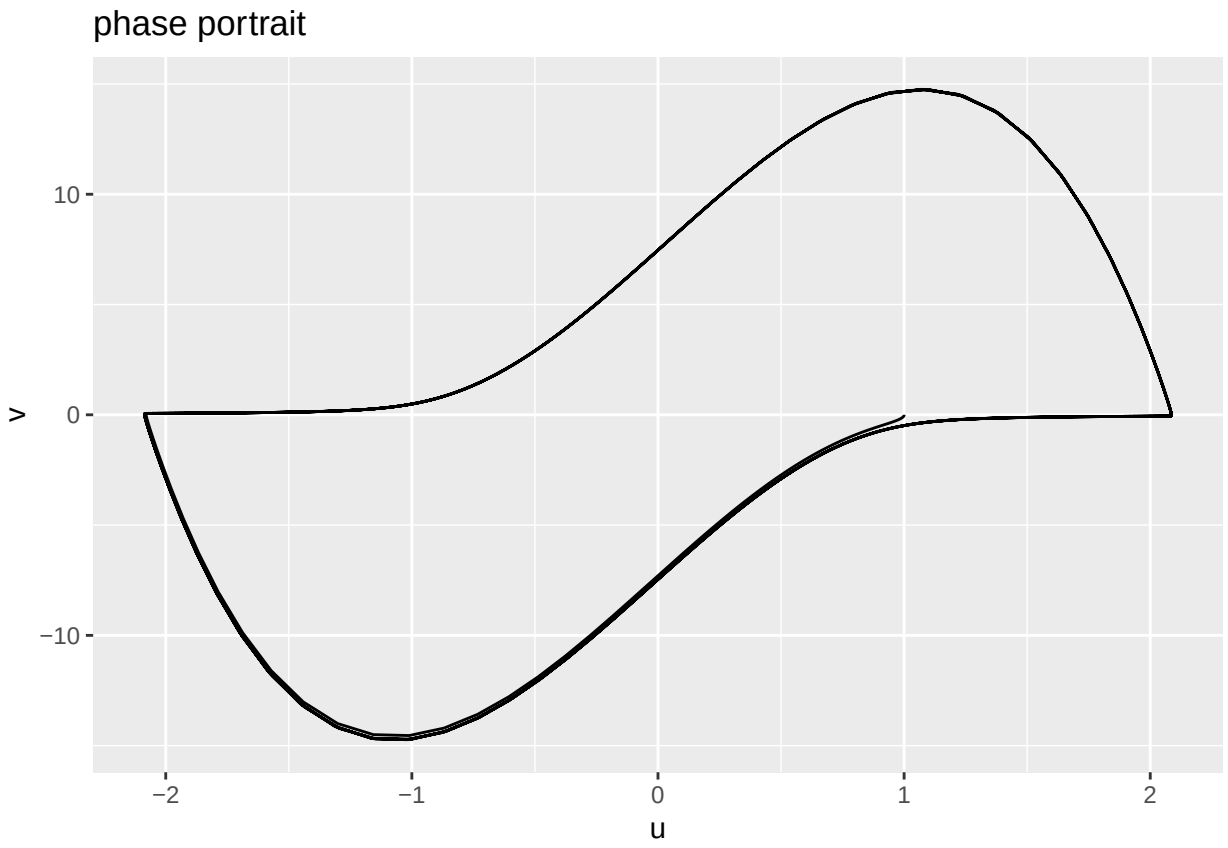

```
ggplot(tb, aes(x = u, y = v)) + geom_path() + ggtitle("phase portrait")
```



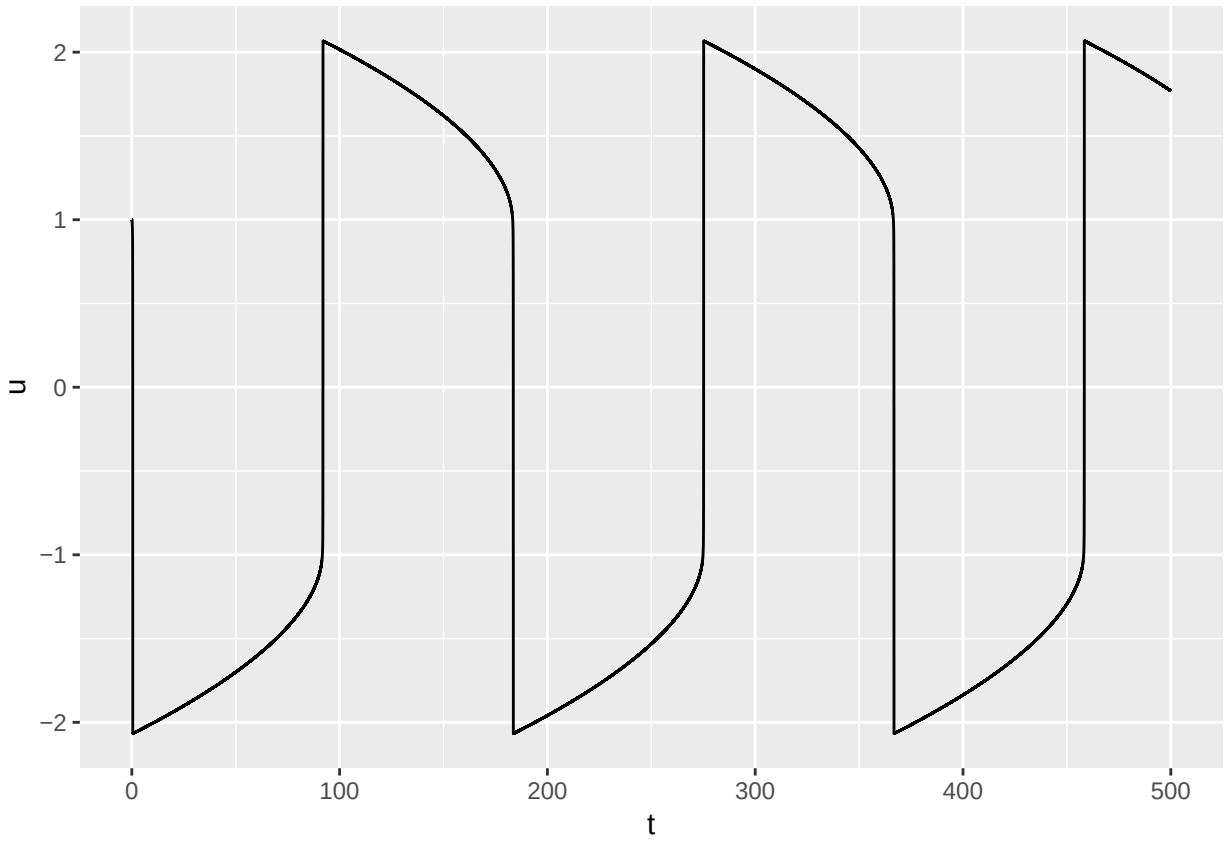

```
tb = euler(vanDerPol(epsilon = 10), c("u" = 1, "v" = 0), 0, 100, h = 0.01) %>%
  as.data.frame

ggplot(tb, aes(x = t, y = u)) + geom_path()
```



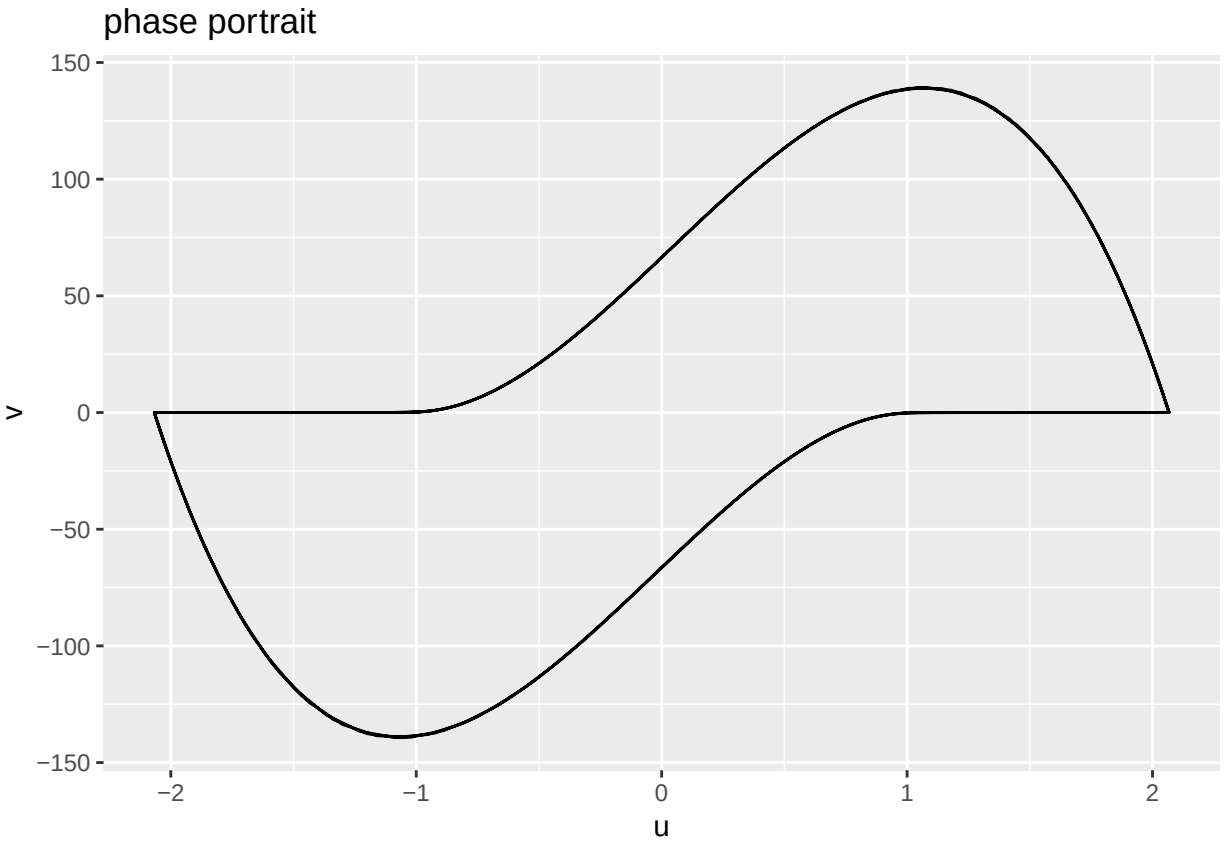

```
ggplot(tb, aes(x = u, y = v)) + geom_path() + ggtitle("phase portrait")
```




```
tb = euler(vanDerPol(epsilon = 100), c("u" = 1, "v" = 0), 0, 500, h = 0.001) %>%
  as.data.frame

ggplot(tb, aes(x = t, y = u)) + geom_path()
```

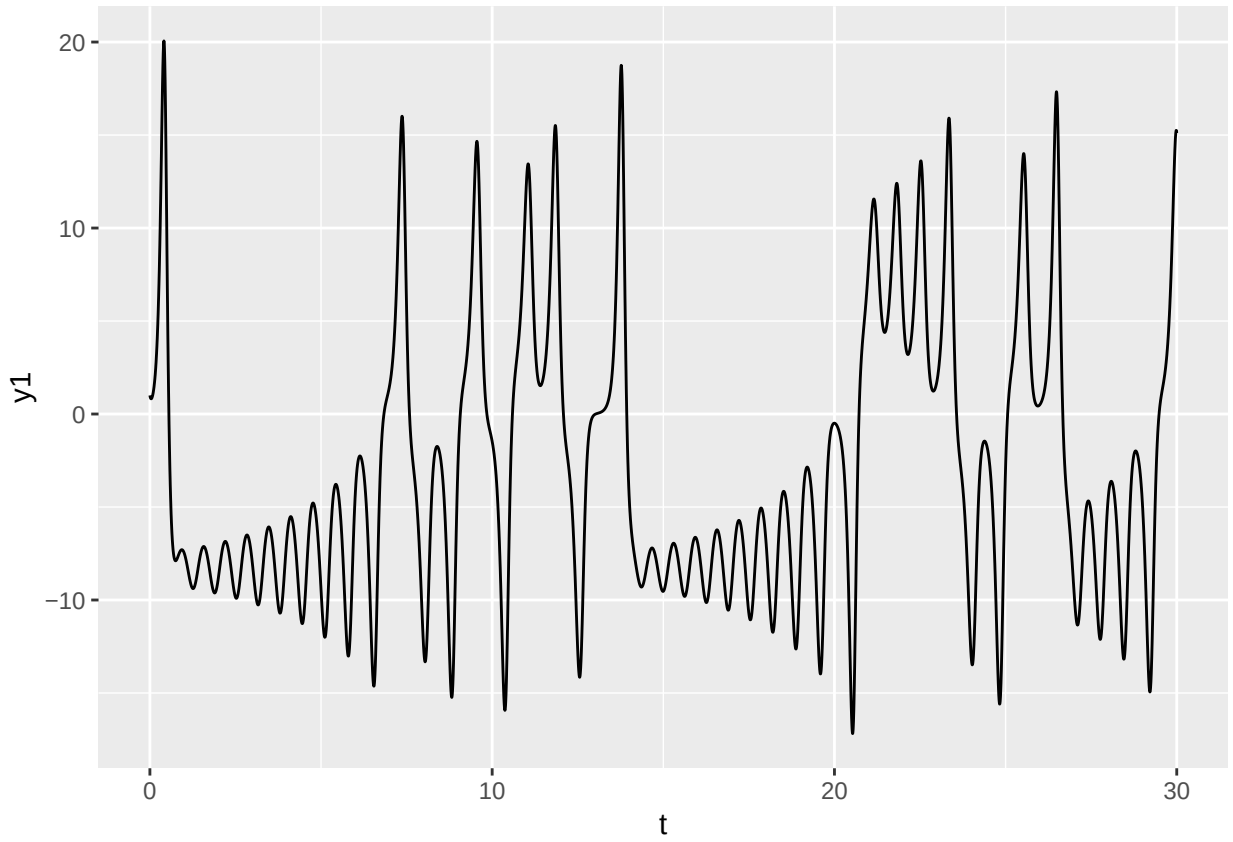



```
ggplot(tb, aes(x = u, y = v)) + geom_path() + ggtitle("phase portrait")
```

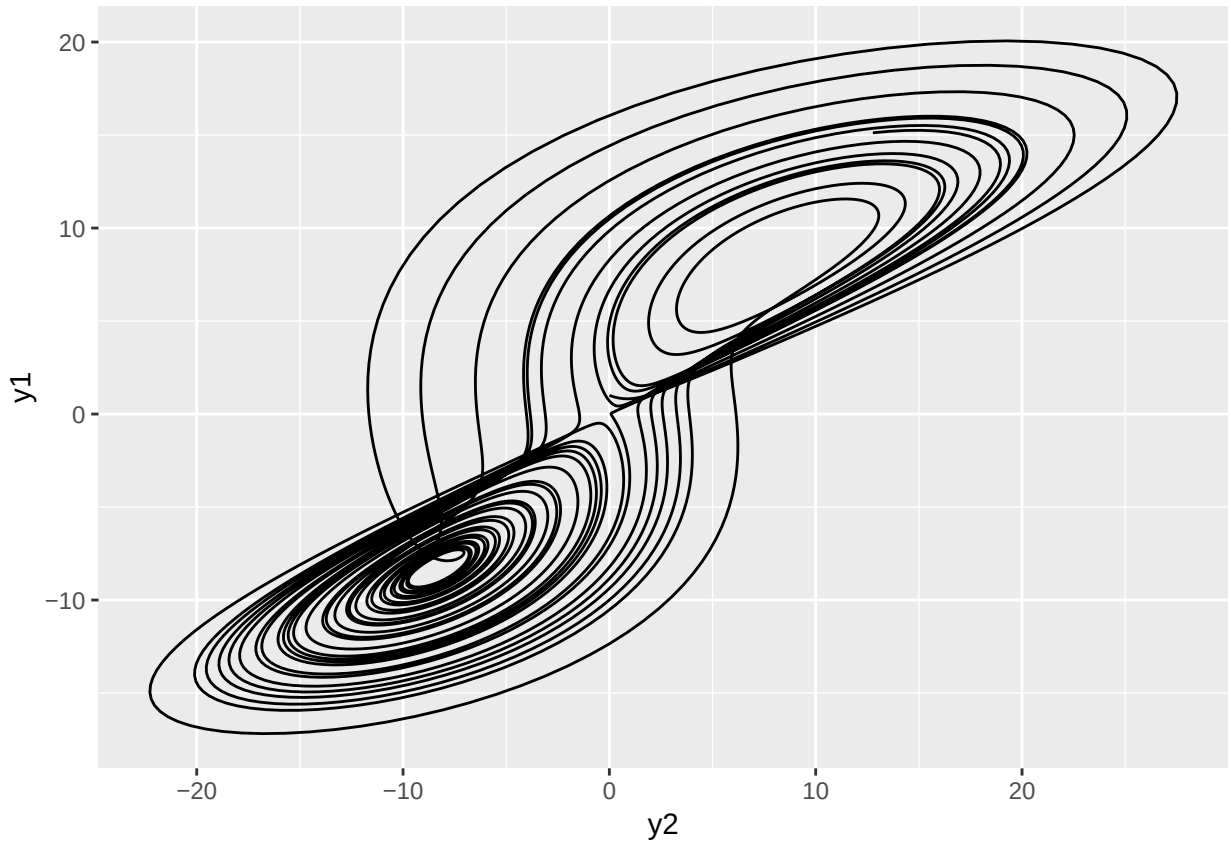


Equação de Lorenz

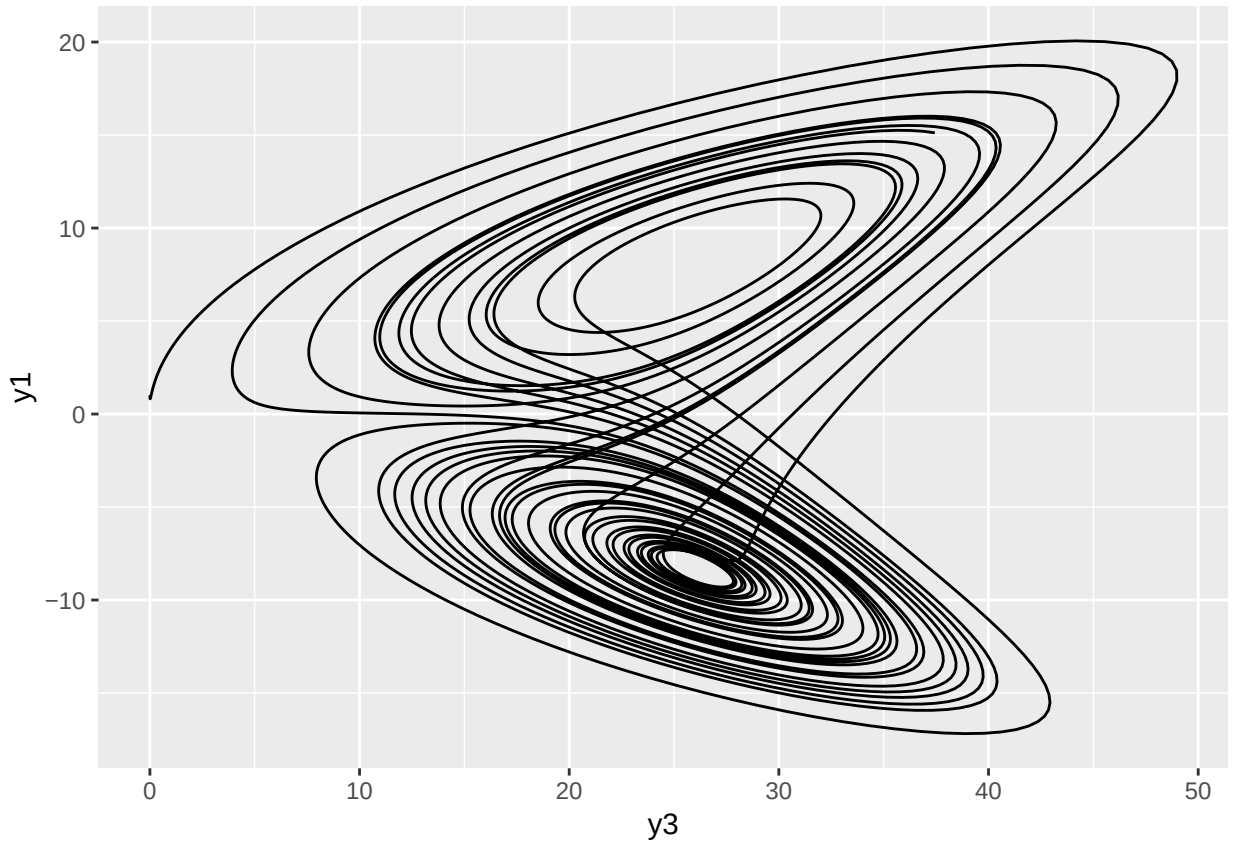
```
sigma = 10
rho = 27
beta = 8/3
tb = euler(function(t, y) c(sigma*(y[2]- y[1]),
                            y[1]*(rho - y[3]) - y[2],
                            y[1]*y[2] - beta*y[3]),
           c(1, 0, 0), 0, 30, 0.005) %>% as.data.frame()

ggplot(tb %>% as.data.frame(), aes(x = t, y = y1)) + geom_path()
```



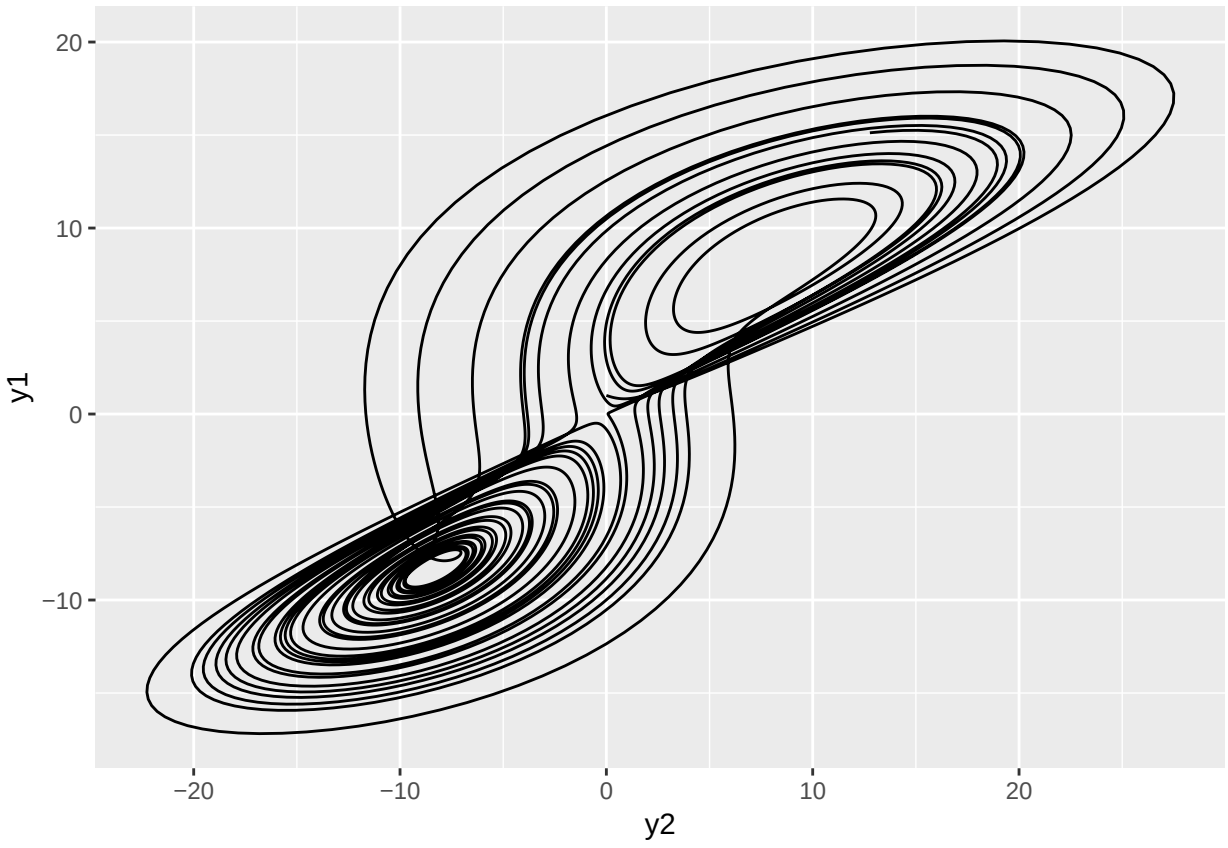

```
ggplot(tb %>% as.data.frame(), aes(x = y2, y = y1)) + geom_path()
```




```
ggplot(tb %>% as.data.frame(), aes(x = y3, y = y1)) + geom_path()
```




```
ggplot(tb %>% as.data.frame(), aes(x = y2, y = y1)) + geom_path()
```



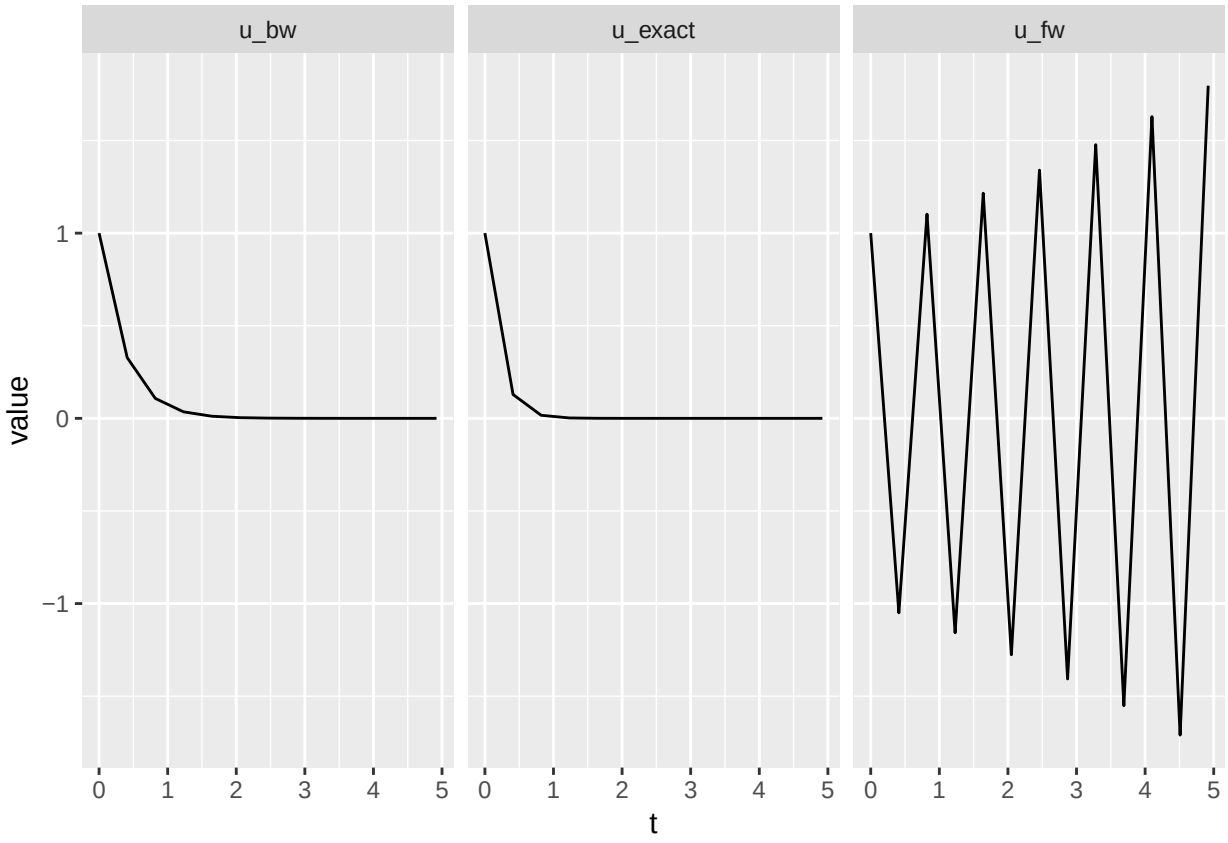
Euler avançado x Euler atrasado

```
eulerBw = function(y0, a, b, h) {
  ts = seq(a, b, by = h)

  y = matrix(0, nrow = length(ts), ncol = length(y0))

  y[1,] = y0
  for (i in 1:(length(ts) - 1)) {
    y[i + 1,] = y[i,]/(1 + 5*h)
  }

  m = cbind(ts, y)
  colnames(m) = c("t", if (is.null(names(y0))) paste0("y", 1:length(y0)) else names(y0))
  m
}

compare = function(h) {
  tb1 = euler(function(t, u) -5*u, c("u" = 1), 0, 5, h = h) %>%
  as.data.frame
  tb2 = eulerBw(c("u" = 1), 0, 5, h = h) %>% as.data.frame

  tb = cbind(tb1 %>% rename(u_fw = u), tb2 %>% select(u_bw = u)) %>%
    mutate(u_exact = exp(-5*t))
```

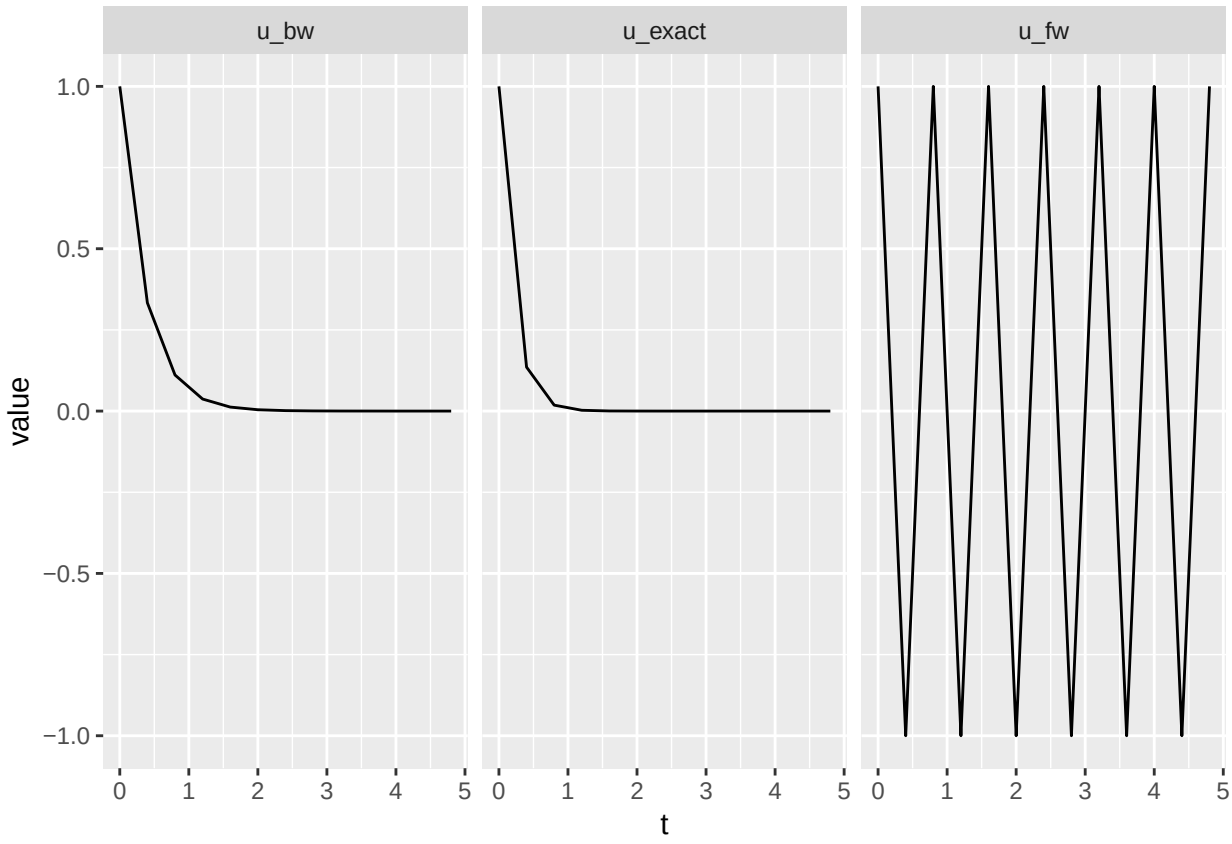
```
  print(tb %>%
    mutate(h = h, error_fw = abs(u_fw - u_exact), error_bw = abs(u_bw - u_exact)) %>%
    select(-starts_with("u")) %>%
    tail(1)
  )

  ggplot(tb %>%gather("method", "value", starts_with("u")), aes(x = t, y = value)) +
    geom_path() + facet_wrap(~ method)
}

map(c(0.41, 0.4, 0.3, 0.1), compare)
```

```
##       t    h error_fw     error_bw
## 13 4.92 0.41 1.795856 1.543107e-06
##      t   h error_fw     error_bw
## 13 4.8 0.4        1 1.881639e-06
##      t   h     error_fw    error_bw
## 17 4.8 0.3 1.525875e-05 4.29459e-07
##    t   h     error_fw     error_bw
## 51 5 0.1 1.388706e-11 1.554441e-09
```

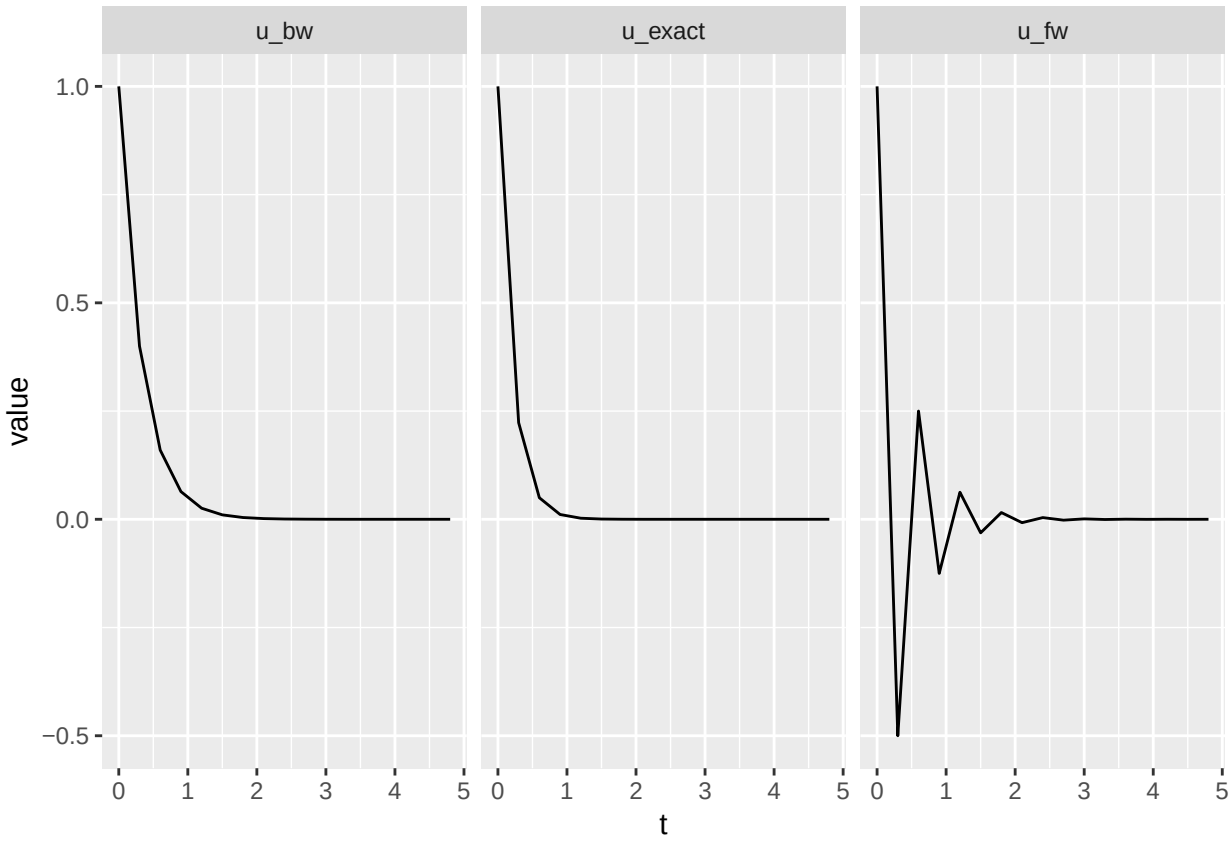
```
## [[1]]
```



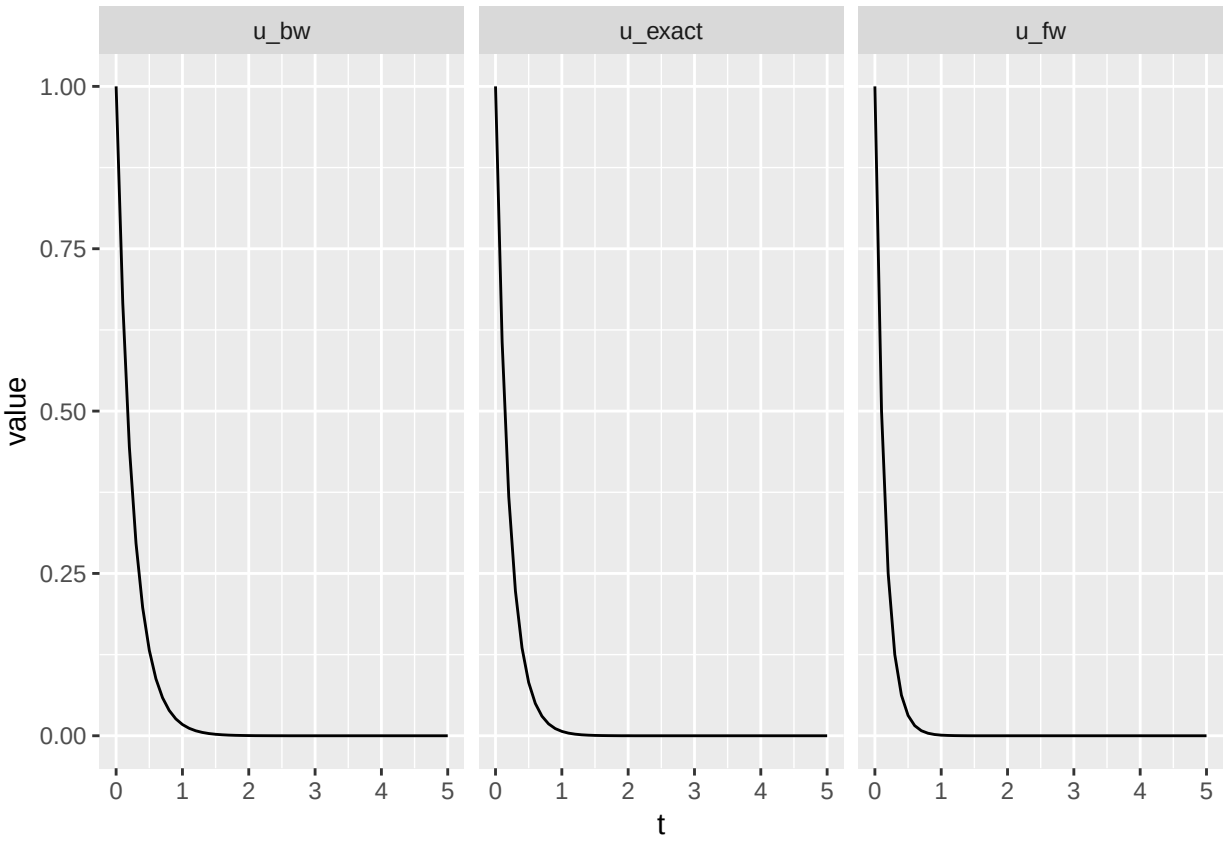

```
##
## [[2]]
```




```
##
## [[3]]
```




```
##
## [[4]]
```


u_bw
u_exact
u_fw
value
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5
t