

Thema_08_Les1

Sam Nelen

2022-04-25

1.0 Experiment

```
value <- list()
Y <- 0

for (I in 1:159){
  dY <- 10 - 0.1 * (Y + 10)
  Y = Y + dY
  value <- c(value, Y)
}
```

1.0 Antwoorden op de vragen

1. De parameters zijn de 10 die er altijd bij komt en de 10% die er vervolgens af gaat.
2. De variabele die steeds verranderd is het totale volume.
3. 0
4. $t = 0$ tot $t = 160$
5. Het evenwichts moment is bij Y is 90 omdat er dan door de 10% afname er 10 bijkomen en er 10 afgaan.
6. De variabele

2.0 Programmeer opdracht

```
if (!require("deSolve", quietly = TRUE))
  install.packages("deSolve")
library(deSolve)

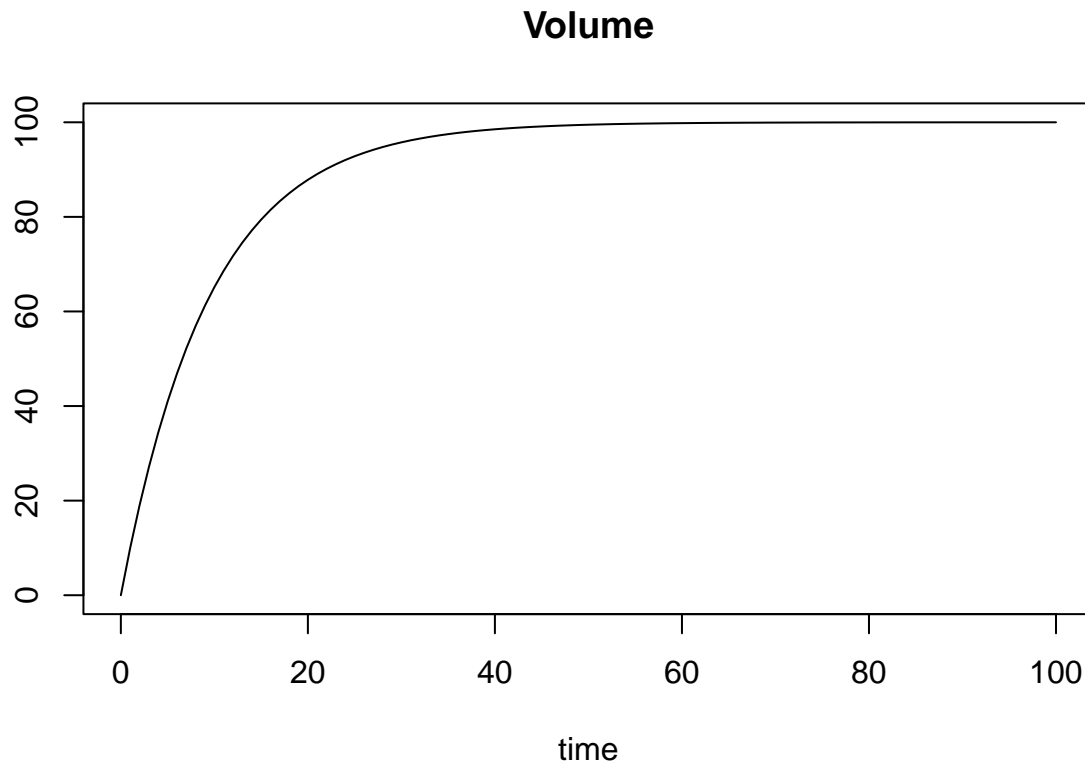
parameters <- c(addVolume = 10, pV = 0.1)

volume <- function(t, y, parms){
  with(as.list(c(parms)), {
    dY <- addVolume - pV * (y)
    return(list(c(dY)))
  })
}

state <- c(Volume = 0)
times <- seq(0, 100, by = 1)

out <- ode(times = times, y = state, parms = parameters, func = volume, method = "euler")
```

```
plot(out)
```



2.0 Antwoorden op de vragen

1. Het veranderen van de parameters zorgt ervoor dat het evenwichtsmoment later of eerder plaatsvind
2. Het veranderen van de initiële waarde zorgt er niet voor dat de evenwichts waarde veranderd, maar als die erboven zit dat die aftelt naar de evenwichts waarde in plaats van opteld er naar toe.
3. Het veranderen van de timeframe heeft geen invloed op de waardes die uit de berekening komen.
4. De evenwichts toestand ligt het aantal dat aan de 'addVolume' variabele word toegewezen hoger dan wanneer die variabele wel in de formule zit.

3.0 Oefenen met de tutorial

```
parameters <- c(a = -8/3, b = -10, c = 28)

state <- c(X = 1, Y = 1, Z = 1)

Lorenz <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    # rate of change
    dX <- a*X + Y*Z
    dY <- b * (Y-Z)
    dZ <- -X*Y + c*Y - Z

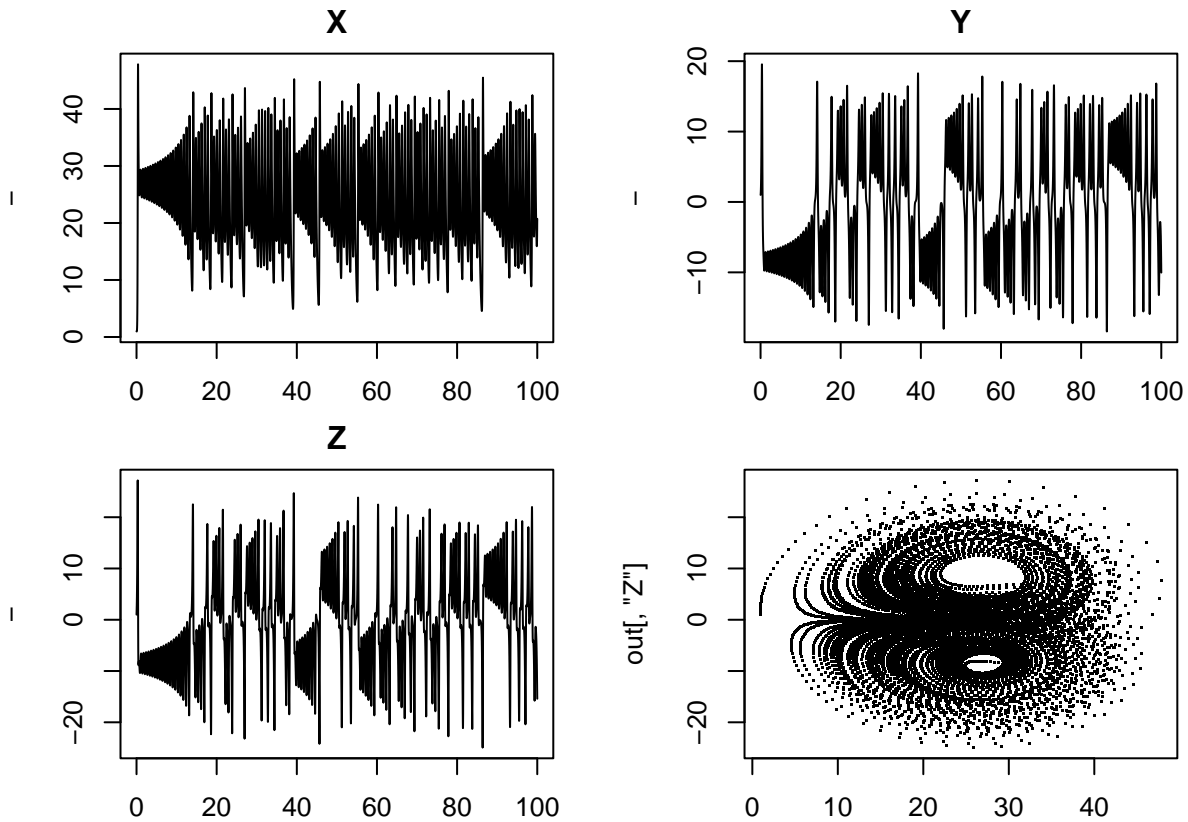
    # return the rate changes
    list(c(dX, dY, dZ))
  })
}

times <- seq(0, 100, by = 0.01)

out <- ode(y = state, times = times, func = Lorenz, parms = parameters)
#head(out)

# Plotting results
par(oma = c(0, 0, 1, 0), mar = c(2,4,2,2))
plot(out, xlab = "time", ylab = "-")
plot(out[, "X"], out[, "Z"], pch = ".")
mtext(outer = TRUE, side = 3, "Lorenz model")
```

Lorenz model



3.0 Antwoorden op de vragen

1. De parameters (constantes) van deze vergelijking is de parameters r-variabele. Dus de a, b en c.
2. De initiële waarden van deze vergelijking is de state r-variabele. Dus de X, Y en Z.
3. In de tutorial hebben ze het over 100 dagen, wat hier te zien is aan de times variabele. Die loopt van 0 tot 100 in stapjes van 0.01.
4. De Lorenz vergelijkingen zijn drie differentiaalvergelijkingen die het ideale gedrag van de atmosfeer van de aarde omschrijven.

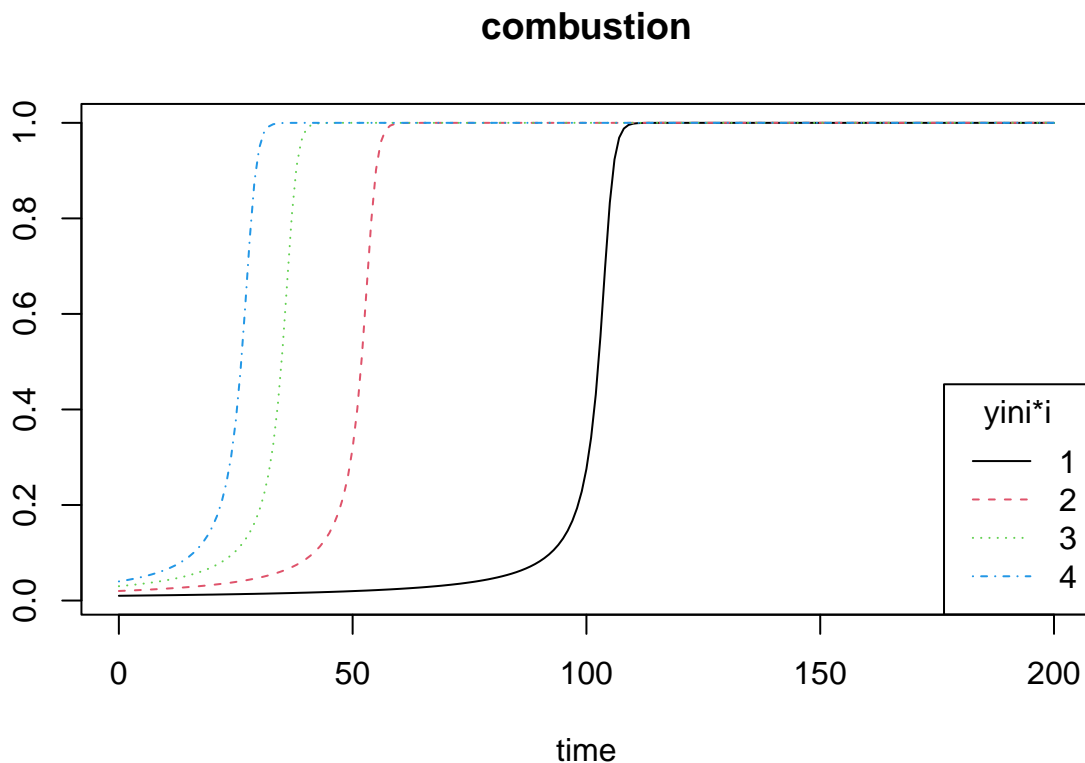
10. Plotting deSolve Objects

```
# 10.1. Plotting Multiple Scenario's
combustion <- function(t, y, parms){
  list(y^2 * (1-y) )
}

yini <- 0.01
times <- 0 : 200

out <- ode(times = times, y = yini, parms = 0, func = combustion)
out2 <- ode(times = times, y = yini*2, parms = 0, func = combustion)
out3 <- ode(times = times, y = yini*3, parms = 0, func = combustion)
out4 <- ode(times = times, y = yini*4, parms = 0, func = combustion)

plot(out, out2, out3, out4, main = "combustion")
legend("bottomright", lty = 1:4, col = 1:4, legend = 1:4, title = "yini*i")
```



10 antwoorden op de vragen

1. In de combustion vergelijking zitten geen parameters.
2. De yini-r-variabele is de initiële waarde van de combustion functie.
3. Het tijdframe van dit experiment loopt van 0 tot 200, maar er wordt verder geen eenheid meegegeven.
- 4.

10.2. Plotting Output with Observations

```
obs <- subset(cc14data, animal == "A", c(time, ChamberConc))
names(obs) <- c("time", "CP")
head(obs)

##      time      CP
## 1 0.083 828.4376
## 2 0.167 779.6795
## 3 0.333 713.8045
## 4 0.500 672.0502
## 5 0.667 631.9522
## 6 0.833 600.6975

parms <- c(0.182, 4.0, 4.0, 0.08, 0.04, 0.74, 0.05, 0.15, 0.32, 16.17,
           281.48, 13.3, 16.17, 5.487, 153.8, 0.04321671,
           0.40272550, 951.46, 0.02, 1.0, 3.80000000)
yini <- c(AI = 21, AAM = 0, AT = 0, AF = 0, AL = 0, CLT = 0, AM = 0)

out <- ccl4model(times = seq(0, 6, by = 0.05), y = yini, parms = parms)

par2 <- parms
par2[1] <- 0.1
out2 <- ccl4model(times = seq(0, 6, by = 0.05), y = yini, parms = par2)

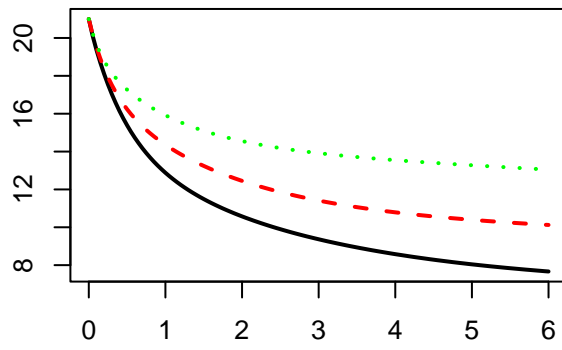
par3 <- parms
par3[1] <- 0.05
out3 <- ccl4model(times = seq(0, 6, by = 0.05), y = yini, parms = par3)

par(mar = c(2, 2, 3, 2))
plot(out, out2, out3, which = c("AI", "MASS", "CP"),
     col = c("black", "red", "green"), lwd = 2,
     obs = obs, obspar = list(pch = 18, col = "blue", cex = 1.2))
legend("topright", lty = c(1, 2, 3, NA), pch = c(NA, NA, NA, 18),
     col = c("black", "red", "green", "blue"), lwd = 2,
     legend = c("par1", "par2", "par3", "obs"))

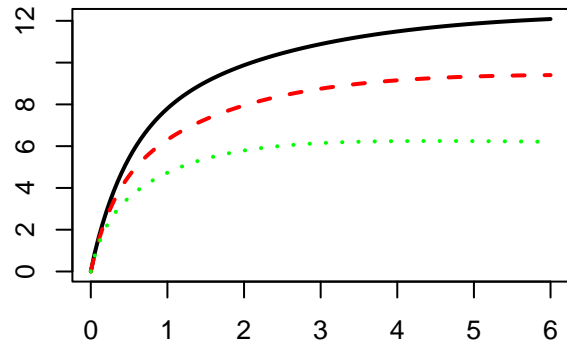
obs2 <- data.frame(time = 6, MASS = 12)

plot(out, out2, out3, lwd = 2,
     obs = list(obs, obs2),
     obspar = list(pch = c(16, 18), col = c("blue", "black"),
                   cex = c(1.2, 2))
)
```

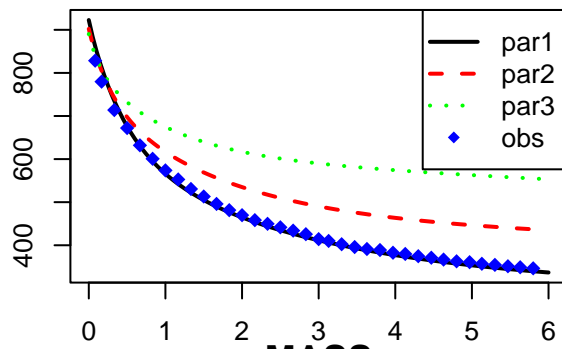
AI



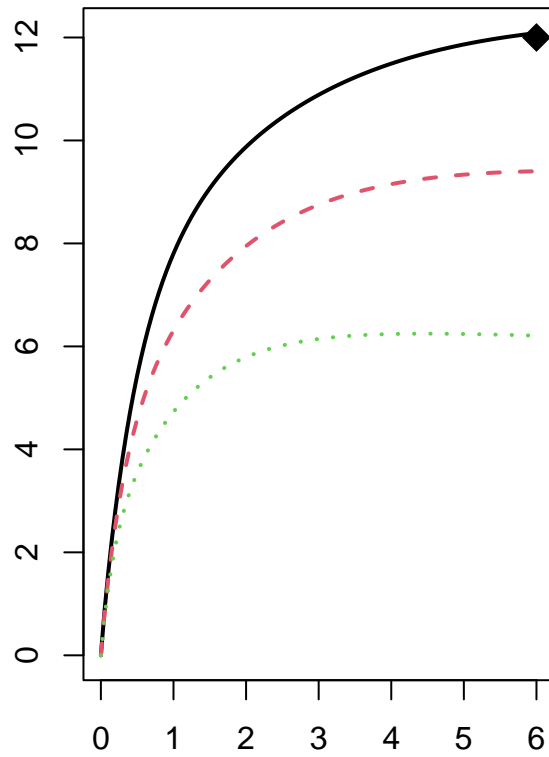
MASS



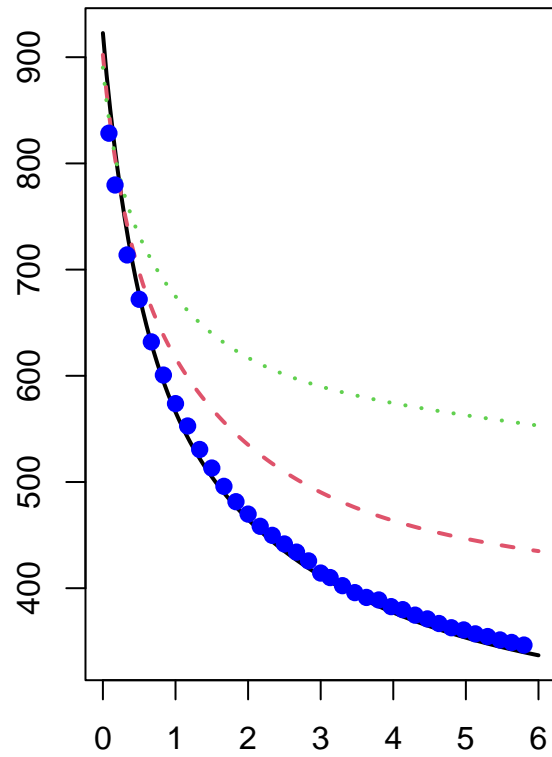
CP



MASS



CP



10.2 antwoorden op de vragen

1. De parameters in het ccl4model zijn de nummers in de parms r-variabele.
2. De initiële waarde van het ccl4model zijn de waarden in de yini r-variabele.
3. Het tijdframe loopt van 0 tot 6 in stappen van 0.05, hierbij is geen eenheid meegegeven.
4. Het ccl4model omschrijft het verloop van hoe ratten gedrogeerd worden met de giftige stof ccl4.

10.3. Plotting Summary Histograms

```
hist(out, col = grey(seq(0, 1, by = 0.1)), mfrow = c(3, 4))
```

