

Week 3 thema 8

Sam Nelen

2023-12-13

Inhoudsopgave

0.1	Introductie	1
1	Methode	4
1.1	Het software model	4
1.2	Model configuratie	5
2	Resultaten	6
2.1	Opdracht 1	6
2.2	Opdracht 2	8
3	Discussie en Conclusie	17
3.1	Discussie	17
3.2	Conclusie	17

0.1 Introductie

In dit onderzoek worden de waardes van het model vergeleken met de waardes van het fysieke onderzoek bij de ratten. Dit zorgt ervoor dat de werking van het model word bepaald en of deze goed werkt. Om dit te vergelijken worden met behulp van verschillende grafieken de resultaten van het model en het experiment vergeleken. Nu is dus de vraag of dit te behalen is met de verschillende grafieken en of dit een goed beeld geeft.

0.1.1 Doel

Met dit onderzoek word dus vooral gekeken naar de werking van het model ten opzichte van de resultaten uit de echte wereld. Dit doel word behaald door de verschillende waardes uit het model en het onderzoek te vergelijken. Door grafieken te maken worden ze het best vergeleken en is het ook op een overzichtelijke manier te zien. De verwachting zal zijn dat het model in grote lijnen overeen zal komen met het experiment. Bij het experiment is er natuurlijk altijd een mogelijkheid op uitschieters en meetfouten. Hierdoor kan het zijn dat op sommige punten de waardes niet overeen komen.

0.1.2 Theorie

In 1 wordt een biologisch model weergegeven van de werking van corticosteroiden. De eerste stap is via diffusie de corticosteroiden binnen krijgen in de cel. Deze corticosteroiden binden zich dan aan een receptor, waarna dit receptorcomplex de nucleus van een cel binnentreedt. Dit bind zich aan het target DNA waardoor de transcriptie van sommige onderdelen word verlaagd of verhoogd. Een van die onderdelen is een vermindering in de aanmaak van receptoren. Hierdoor word er op een bepaald moment een evenwicht bereikt. Dit is te zien in de resultaten die uit het model komen.

0.1.3 Functies

$$\begin{aligned}\frac{dmRNA_R}{dt} &= k_{s_Rm} * \left(1 - \frac{DR(N)}{IC_{50_Rm} + DR(N)}\right) - k_{d_Rm} * mRNA_R \\ \frac{dR}{dt} &= k_{s_R} * mRNA_R + R_f * k_{re} * DR(N) - k_{on} * D * R - k_{d_R} * R \\ \frac{dDR}{dt} &= k_{on} * D * R - k_T * DR \\ \frac{dDR(N)}{dt} &= k_T * DR - k_{re} * DR(N)\end{aligned}$$

```
data_MPL <- read.csv("MPL.csv", na.strings = "NA")
median_MPL_01 <- median(data_MPL$MPL_conc[data_MPL$dose==0.1], na.rm=TRUE)
median_MPL_03 <- median(data_MPL$MPL_conc[data_MPL$dose==0.3], na.rm=TRUE)
```

Voor de concentratie van methylprednisolon (MPL) word de mediaan van alle concentraties over de tijdstippen berekend, zoals hierboven in het code blok te zien is. Dit gebeurt voor elke doses apart waarbij de volgende waardes uitkomen: 14.59 ng/ml en 39.925 ng/ml. Deze waardes voor respectievelijk de 0.1 en 0.3 doses worden gebruikt bij het omzetten naar de juiste eenheden voor de concentratie MPL in het model.

```
medians <- aggregate(data_MPL[, c("MPL_conc", "mRNA", "Free_receptor")],
                     list(data_MPL$dose, data_MPL$time),
                     median, na.rm=TRUE)
names(medians)[1:2] <- c("dose", "time")
```

Verder wordt de dataset nog wat opgeschoond zodat die wat makkelijker in gebruik is. Op elk tijdstip in de dataset zijn namelijk vier metingen om de kans op meetfouten te verkleinen. Ook hier wordt daarom weer de mediaan per doses per tijdstip gepakt van de metingen, waarna deze worden samengevoegd in een nieuwe dataset. De manier waarop dit gebeurt word hierboven beschreven.

Hoofdstuk 1

Methode

1.1 Het software model

Het model word gesimuleerd met verschillende formules. Dit gebeurt doormiddel van de r (versie 4.3.0) programmeer taal. De gebruikte IDE is RStudio (versie 2023.03.0) met de deSolve (versie 1.35) library. Om door het model heen te lopen word er een functie opgezet waarbij alle wiskundige formules worden ingevoerd. Door de deSolve functie ode() aan te roepen kan er door de model functie heen gelopen worden. Aan deze deSolve functie worden alle parameters, de startwaarde en de looptijd meegegeven. Op deze manier worden de verschillende waarden op een bepaald tijdpunt berekend en is er in dit geval een biologisch proces gemodelleerd.

```
model <- function(t, y, parms){
  with(as.list(c(y, parms)),{
    dmRNA.R_dt <- ks_Rm * (1-(drn / (IC50_Rm + drn))) - kd_Rm * mRNA.R

    dR_dt <- ksr * mRNA.R + Rf * kre * drn - kon * D * R - kd_R * R

    dDR_dt <- kon * D * R - kT * dr

    dDRN_dt <- kT * dr - kre * drn

    return(list(c(dmRNA.R_dt, dR_dt, dDR_dt, dDRN_dt)))
  }
)

state <- c(mRNA.R = 4.74, R = 267, dr = 0, drn = 0)
times <- seq(0, 168, by = 1)
molar.mass <- 374.471

parameters <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329, kT = 0.63, kre = 0.57,
  Rf = 0.49, kd_R = 0.0572, kd_Rm = 0.612, ksr = 3.22,
  D = 20 * 1000 * (1/ molar.mass))

output.normal <- as.data.frame(ode(y = state, times = times, func = model,
  parms = parameters))
```

1.2 Model configuratie

De verschillende parameters en variabelen zijn op basis van een aantal experimenten met methylprednisolon in ratten bepaald. Deze parameters en variabelen zijn hieronder in tabel 1 en 2 te zien. De variabelen worden als startwaarden meegegeven aan het model. De parameters die tabel 1 te zien zijn, zijn de standaard waarden die voor sommige uitvoeringen worden aangepast om de functie weer te geven.

Tabel 1.1: Parameter Waardes

Parameter	Value	Unit
k_{s_Rm}	2.90	fmol/g liver/h
IC_{50_Rm}	26.2	fmol/mg protein
k_{on}	0.00329	L/nmol/h
k_T	0.63	1 / h
k_{re}	0.57	1 / h
R_f	0.49	
k_{d_R}	0.0572	1 / h
k_{d_Rm}	0.612	
k_{s_R}	3.22	
$D0.1$	38.9616285	nmol/L
$D0.3$	106.6170678	nmol/L

Tabel 1.2: Variabele Waardes

Variabele	Value	Unit
R_{m0}	4.74	fmol / g liver
R_0	267	fmol/mg protein
DR	0	fmol/mg protein
$DR(N)$	0	fmol/mg protein

Hoofdstuk 2

Resultaten

2.1 Opdracht 1

```
parameters["D"] <- median_MPL_01 * 1000 * (1/ molar.mass)
Volume.D0.1 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

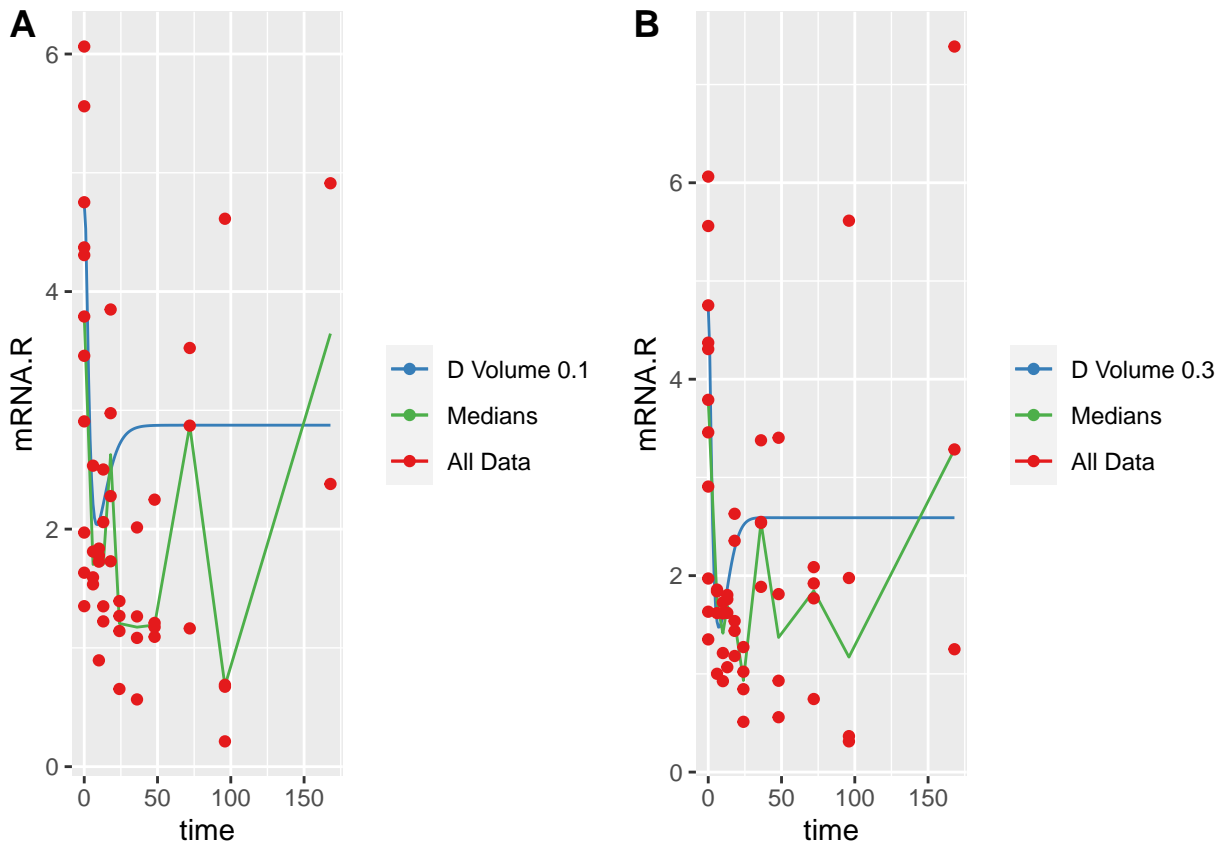
parameters["D"] <- median_MPL_03 * 1000 * (1/ molar.mass)
Volume.D0.3 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

#code to generate figures with title, subscripts, legenda etc
plot0.1 <- ggplot(data = Volume.D0.1, aes(x=time)) +
  geom_line(aes(y=mRNA.R, color = "D Volume 0.1")) +
  geom_line(data = subset(medians, dose %in% c(0.0, 0.1)), aes(x=time, y=mRNA,
                                                             color = "Medians")) +
  geom_point(data = subset(data_MPL, dose %in% c(0.0, 0.1)), aes(x=time, y=mRNA,
                                                             color = "All Data")) +
  scale_color_brewer("",
                    breaks = c("D Volume 0.1", "Medians", "All Data"),
                    palette = "Set1")

plot0.3 <- ggplot(data = Volume.D0.3, aes(x=time)) +
  geom_line(aes(y=mRNA.R, color = "D Volume 0.3")) +
  geom_line(data = subset(medians, dose %in% c(0.0, 0.3)), aes(x=time, y=mRNA,
                                                             color = "Medians")) +
  geom_point(data = subset(data_MPL, dose %in% c(0.0, 0.3)), aes(x=time, y=mRNA,
                                                             color = "All Data")) +
  scale_color_brewer("",
                    breaks = c("D Volume 0.3", "Medians", "All Data"),
                    palette = "Set1")

plot_grid(plot0.1, plot0.3, labels = "AUTO")

## Warning: Removed 5 rows containing missing values (`geom_point()`).
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



Figuur 2.1: Het verschil tussen het model, de medianen en alle waarden uit het onderzoek


```
# ggplot(data = Volume.D0.1, aes(x=time, y=R)) +
#   geom_line() +
#   geom_line(data = subset(medians, dose %in% c(0.0, 0.1)), aes(x=time, y=Free_receptor),
#             color="red") +
#   geom_point(data = subset(data_MPL, dose %in% c(0.0, 0.1)), aes(x=time, y=Free_receptor))
#
# ggplot(data = Volume.D0.3, aes(x=time, y=R)) +
#   geom_line() +
#   geom_line(data = subset(medians, dose %in% c(0.0, 0.3)), aes(x=time, y=Free_receptor),
#             color="red") +
#   geom_point(data = subset(data_MPL, dose %in% c(0.0, 0.3)), aes(x=time, y=Free_receptor))
```

Zoals is te zien in figuur 2.1 geeft de groene lijn de mediaan aan van de gemeten resultaten van het fysieke onderzoek. Dit geeft een beter beeld van de verschillende metingen omdat er op elk tijdstip 4 metingen zijn gedaan. Door deze dan weer te geven met de mediaan worden uitschieters en bijvoorbeeld meetfouten er beter uit gehaald. Dit geeft dan weer een beter beeld van de verkregen resultaten. Als er dan verder gekeken wordt naar het verschil in doseringen is er een groot verschil te zien. Bij een lagere dosering van het medicijn is duidelijk te zien dat de verlaging in vrije receptoren langzamer en minder groot is. Wat ook vrij logisch is aangezien er minder methylprednisolon aanwezig is om aan de receptoren te binden.

Verder lijkt het model goed overeen te komen op de meeste punten met de resultaten van het onderzoek. Op sommige tijdstippen en bij bijvoorbeeld receptor mRNA lijken de waarden wat af te wijken van elkaar. Dit kan aan meerdere dingen gelegen hebben. Zo kunnen er bijvoorbeeld wat meetfouten zijn geweest, maar kunnen het ook dingen zijn die moeilijk in een model zijn weer te geven. In het echt kan natuurlijk een deel van het medicijn nooit binden aan de receptor en meteen afgevoerd worden. Deze dingen zijn natuurlijk moeilijk weer te geven in een model en kan de verschillen in sommige resultaten verklaren. Maar over het algemeen lijken het model en de onderzoeksresultaten goed overeen te komen.

2.2 Opdracht 2

```
# Adjusted the model to reflect the assignments question.
model_adjusted <- function(t, y, parms){
  with(as.list(c(y, parms)),{
    dmRNA.R_dt <- ks_Rm * (1-(drn * 0 / (IC50_Rm + drn * 0))) - kd_Rm * mRNA.R

    dR_dt <- ksr * mRNA.R + Rf * kre * drn - kon * D * R - kd_R * R

    dDR_dt <- kon * D * R - kT * dr

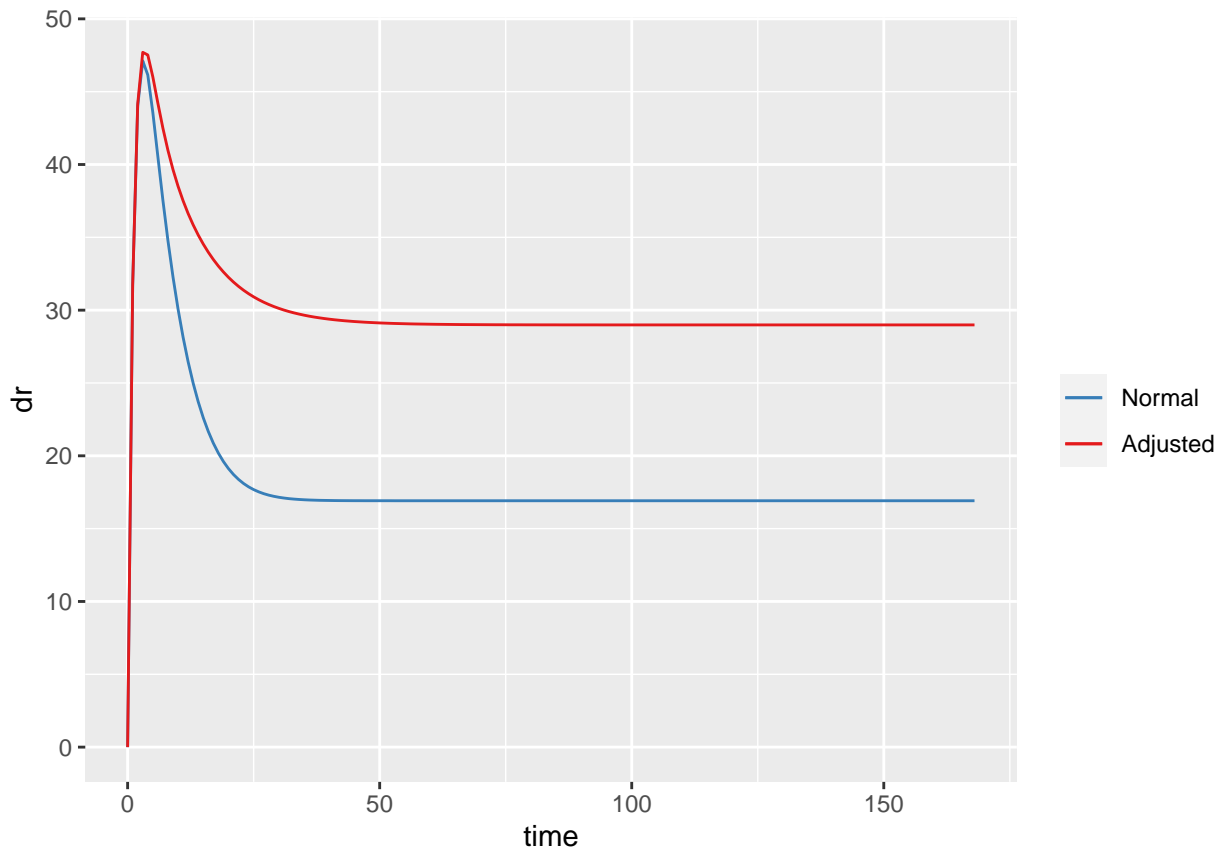
    dDRN_dt <- kT * dr - kre * drn

    return(list(c(dmRNA.R_dt, dR_dt, dDR_dt, dDRN_dt)))
  }
)
}

# Set the parameters to base value and run the simulation.
parameters <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329, kT = 0.63, kre = 0.57,
               Rf = 0.49, kd_R = 0.0572, kd_Rm = 0.612, ksr = 3.22,
               D = 20 * 1000 * (1/ molar.mass))
output.adjusted <- as.data.frame(ode(y = state, times = times, func = model_adjusted,
                                   parms = parameters))
```

```
# Plot the output form the adjusted model in a ggplot line plot.
```

```
ggplot(data = output.normal, aes(x=time)) +  
  geom_line(aes(y = dr, color = "Normal")) +  
  geom_line(data = output.adjusted, aes(y=dr, color = "Adjusted")) +  
  scale_color_brewer("",  
    breaks = c("Normal", "Adjusted"),  
    palette = "Set1")
```



Figuur 2.2: Het verschil tussen het standaard modellen een model waar de regulatie van de glucocorticoidreceptor is weggefallen.

In dit scenario heeft het medicijn geen invloed op de synthese van het mRNA. Dit is goed te zien in figuur 2.2, met het aangepaste model komen de waarden van de dichtheid van het MPL-receptor complex een stuk hoger uit. Dit komt doordat de synthese van het mRNA gewoon doorgaat en niet wordt gestopt door een gebonden receptor. Hierdoor blijven er receptoren gemaakt worden waar MPL aan kan blijven binden. En zonder deze regulatie komt de dichtheid van MPL-receptor complexen op een hogere stabiele toestand uit dan met regulatie.

```
# Set start state to the base values.
```

```
state <- c(mRNA.R = 4.74, R = 267, dr = 0, drn = 0)
```

```
# Set the time course till the model reaches a steady state and run the model.
```

```
times <- seq(0, 50)
```

```
parameters["D"] <- 20 * 1000 * (1/ molar.mass)
```

```
D.steady <- as.data.frame(ode(y = state, times = times, func = model,  
  parms = parameters))
```

```
# Continue the model with the drug concentration at 0 and the last values from
```

```

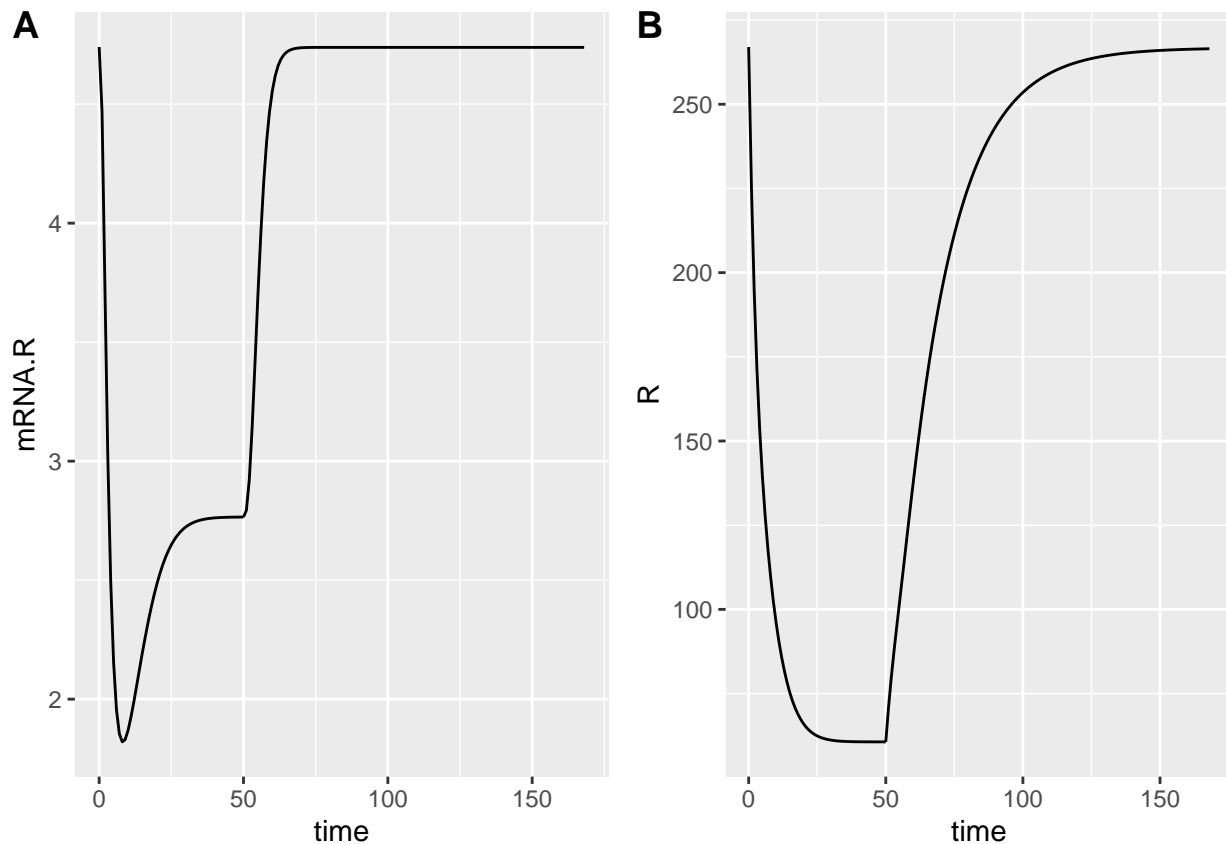
# the run above.
times <- seq(50, 168)
parameters["D"] <- 0
state <- unlist(tail(D.steady, n=1)[-1])
D.steady.second <- as.data.frame(ode(y = state, times = times,
                                     func = model, parms = parameters))

# Plot the data from the two model outputs to combine the data in one graph.
steady.mRNA <- ggplot(data = D.steady, aes(x = time)) +
  geom_line(aes(y = mRNA.R)) +
  geom_line(data = D.steady.second, aes(y = mRNA.R))

steady.R <- ggplot(data = D.steady, aes(x = time)) +
  geom_line(aes(y = R)) +
  geom_line(data = D.steady.second, aes(y = R))

plot_grid(steady.mRNA, steady.R, labels = "AUTO")

```



Figuur 2.3: De werking van het medicijn op de mRNA en receptor concentratie

Verder word er nog naar het model gekeken als op een punt de concentratie van het medicijn in een keer naar 0 gaat. Als punt is de evenwichtstoestand gekozen, op deze manier is er ook een duidelijk verschil te zien tussen de fases. Er wordt gekeken naar de mRNA en receptor concentraties in figuur 2.3. Dit zijn respectievelijk grafiek A en B in de figuur. Zoals goed te zien is schieten deze waarden weer terug omhoog zodra de concentratie van het medicijn op 0 wordt gezet. De reden hiervoor is dat er geen MPL-receptor complexen meer ontstaan die de synthese van mRNA stoppen. Hierdoor zullen de concentraties van de

receptor en mRNA weer terug omhoog schieten.

```
# Reset parameters, start states and the time to base values.
parameters <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329, kT = 0.63, kre = 0.57,
               Rf = 0.49, kd_R = 0.0572, kd_Rm = 0.612, ksr = 3.22,
               D = 20 * 1000 * (1/ molar.mass))
state <- c(mRNA.R = 4.74, R = 267, dr = 0, drn = 0)
times <- seq(0, 168, by = 1)

parameters["kon"] <- 0.00329/5
kon.divided5 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["kon"] <- 0.00329/2
kon.divided2 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["kon"] <- 0.00329*5
kon.times5 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["kon"] <- 0.00329*2
kon.times2 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["kon"] <- 0.00329

parameters["kre"] <- 0.57/5
kre.divided5 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["kre"] <- 0.57/2
kre.divided2 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["kre"] <- 0.57*5
kre.times5 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["kre"] <- 0.57*2
kre.times2 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

kon.mRNA.R <- ggplot(output.normal, aes(x = time)) + geom_line(aes(y = mRNA.R, color = "normal")) +
  geom_line(data = kon.divided5, aes(y = mRNA.R, color = "divided by 5")) +
  geom_line(data = kon.divided2, aes(y = mRNA.R, color = "divided by 2")) +
  geom_line(data = kon.times5, aes(y = mRNA.R, color = "times 5")) +
  geom_line(data = kon.times2, aes(y = mRNA.R, color = "times 2")) +
  scale_color_brewer("K_on",
                    breaks = c("normal", "divided by 5", "divided by 2", "times 5", "times 2"),
                    palette = "Set1")

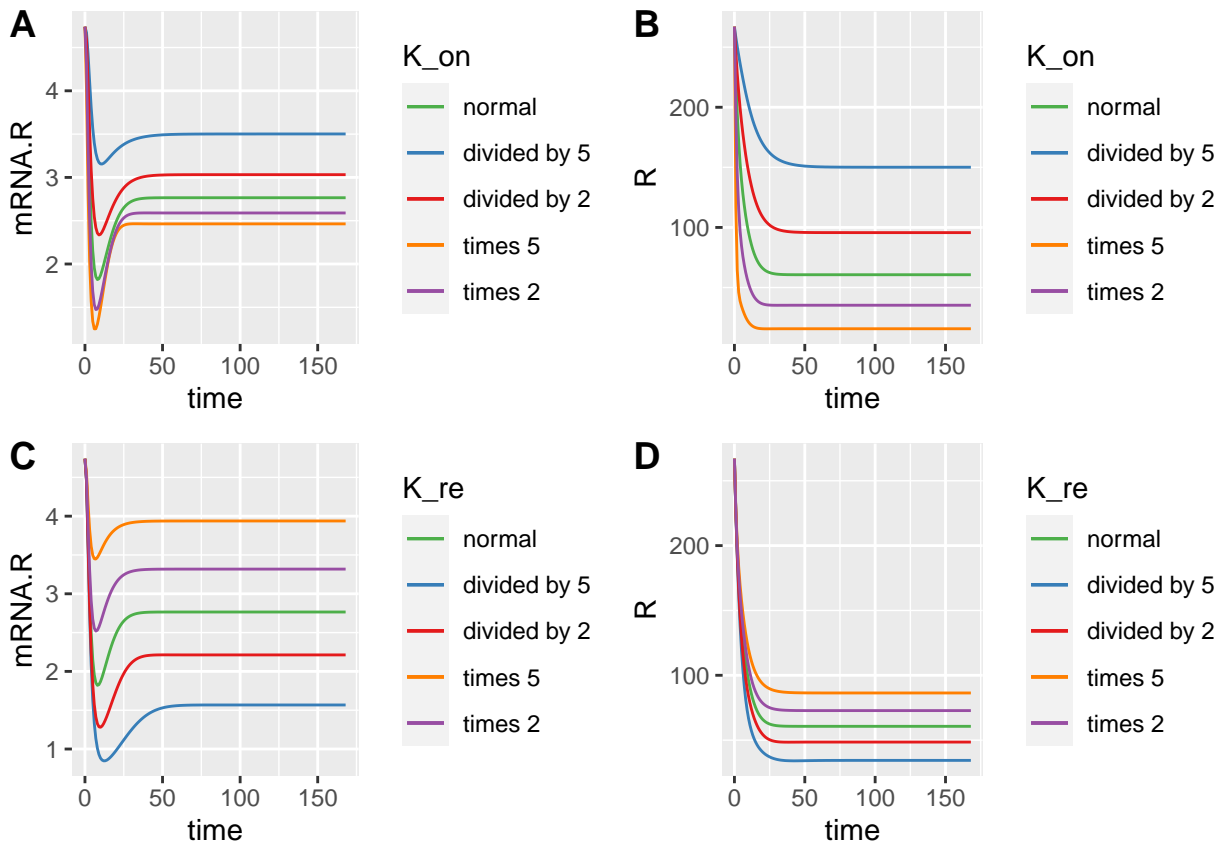
kon.R <- ggplot(output.normal, aes(x = time)) + geom_line(aes(y = R, color = "normal")) +
  geom_line(data = kon.divided5, aes(y = R, color = "divided by 5")) +
  geom_line(data = kon.divided2, aes(y = R, color = "divided by 2")) +
  geom_line(data = kon.times5, aes(y = R, color = "times 5")) +
  geom_line(data = kon.times2, aes(y = R, color = "times 2")) +
  scale_color_brewer("K_on",
                    breaks = c("normal", "divided by 5", "divided by 2", "times 5", "times 2"),
                    palette = "Set1")
```

```
#plot_grid(kon.mRNA.R, kon.R, labels = "AUTO")

kre.mRNA.R <- ggplot(output.normal, aes(x = time)) + geom_line(aes(y = mRNA.R, color = "normal")) +
  geom_line(data = kre.divided5, aes(y = mRNA.R, color = "divided by 5")) +
  geom_line(data = kre.divided2, aes(y = mRNA.R, color = "divided by 2")) +
  geom_line(data = kre.times5, aes(y = mRNA.R, color = "times 5")) +
  geom_line(data = kre.times2, aes(y = mRNA.R, color = "times 2")) +
  scale_color_brewer("K_re" ,
    breaks = c("normal", "divided by 5", "divided by 2", "times 5", "times 2"),
    palette = "Set1")

kre.R <- ggplot(output.normal, aes(x = time)) + geom_line(aes(y = R, color = "normal")) +
  geom_line(data = kre.divided5, aes(y = R, color = "divided by 5")) +
  geom_line(data = kre.divided2, aes(y = R, color = "divided by 2")) +
  geom_line(data = kre.times5, aes(y = R, color = "times 5")) +
  geom_line(data = kre.times2, aes(y = R, color = "times 2")) +
  scale_color_brewer("K_re",
    breaks = c("normal", "divided by 5", "divided by 2", "times 5", "times 2"),
    palette = "Set1")

plot_grid(kon.mRNA.R, kon.R, kre.mRNA.R, kre.R, labels = "AUTO")
```



Figuur 2.4: De werking van verschillende corticosteroiden

In figuur 2.4 worden verschillende soorten corticosteroiden gebruikt voor de simulatie van het model. Dit wordt gedaan door de parameters k_{on} en k_{re} aan te passen. K_{on} is de snelheid van binden aan de

receptor en k_{re} is het loslaten van de receptor. Deze snelheden worden met 5 of 2 keer verhoogd en verlaagd. Het verschil in snelheden geeft een duidelijke verandering aan in de grafieken van figuur 2.4. Bij het veranderen van de snelheid van het binden aan de receptor zijn duidelijk andere evenwichtstoestanden te zien. Als de snelheid verlaagd komen deze hoger uit en met verlagen het tegenovergestelde. Dit komt doordat er minder of juist meer receptor complexen in de celkern komen. Bij het veranderen van de snelheid van het ontbinden gebeurt juist het omgekeerde. Daar blijven de corticosteroiden juist langer gebonden en zullen er bij een verlaging van de snelheid juist meer receptor complexen in de celkern komen. Respectievelijk geeft dit een hogere of lagere synthese van mRNA en dus meer receptoren.

```
# Reset parameters and start states to base values.
parameters <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329, kT = 0.63, kre = 0.57,
                Rf = 0.49, kd_R = 0.0572, kd_Rm = 0.612, ksr = 3.22,
                D = 20 * 1000 * (1/ molar.mass))
state <- c(mRNA.R = 4.74, R = 267, dr = 0, drn = 0)

# Set ksr to 0 which stands for the stopped synthesis in the model and run it.
parameters["ksr"] <- 0
ksr.0 <- as.data.frame(ode(y = state, times = times, func = model,
                          parms = parameters))

# Make all the graphs for every variable compared to the normal output
plot.ksr.mRNA_R <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = mRNA.R, color = "Normal")) +
  geom_line(data = ksr.0, aes(y = mRNA.R, color = "Ksr as 0")) +
  scale_color_brewer("",
                    breaks = c("Normal", "Ksr as 0"),
                    palette = "Set1")

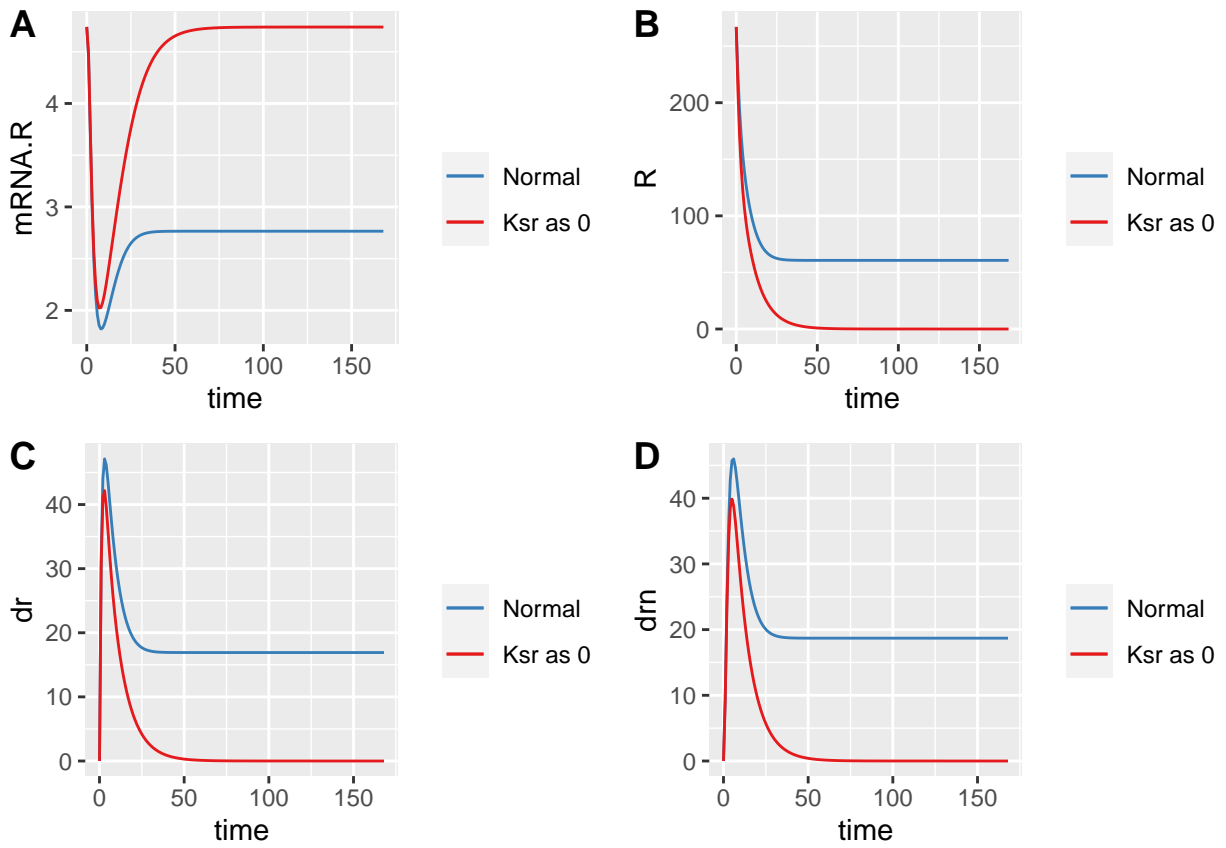
plot.ksr.R <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = R, color = "Normal")) +
  geom_line(data = ksr.0, aes(y = R, color = "Ksr as 0")) +
  scale_color_brewer("",
                    breaks = c("Normal", "Ksr as 0"),
                    palette = "Set1")

plot.ksr.dr <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = dr, color = "Normal")) +
  geom_line(data = ksr.0, aes(y = dr, color = "Ksr as 0")) +
  scale_color_brewer("",
                    breaks = c("Normal", "Ksr as 0"),
                    palette = "Set1")

plot.ksr.drn <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = drn, color = "Normal")) +
  geom_line(data = ksr.0, aes(y = drn, color = "Ksr as 0")) +
  scale_color_brewer("",
                    breaks = c("Normal", "Ksr as 0"),
                    palette = "Set1")

#
plot_grid(plot.ksr.mRNA_R, plot.ksr.R, plot.ksr.dr, plot.ksr.drn, labels = "AUTO")
```

Met figuur 2.5 word het verloop van het model als er geen synthese van mRNA plaats vindt vergeleken met het normale verloop. In het begin lijken ze nog redelijk hetzelfde verloop te tonen wat voornamelijk zal komen door de meegegeven startwaardes aan het model. Naarmate het model dan verder loopt zullen er geen nieuwe



Figuur 2.5: Hier is het verloop van het model te zien als er geen synthese van het mRNA plaatsvindt.

receptors meer aangemaakt worden. Dit is ook te zien in het figuur 2.5 B te zien waarbij het aangepaste model uiteindelijk helemaal naar 0 gaat. Dit betekent dus dat er helemaal geen vrije receptor meer in het cytosol zit. Hetzelfde geldt voor figuur 2.5 C en D die laten zien dat het aantal MPL-receptor complexen helemaal naar 0 gaan. En ondertussen schiet het aantal mRNA weer helemaal terug naar de startwaarde omdat er geen receptor complexen zijn die de transcriptie remmen. Zoals te zien is in figuur 2.5 A.

```
# Reset parameters to base value.
parameters <- c(ks_Rm = 2.90, IC50_Rm = 26.2, kon = 0.00329, kT = 0.63, kre = 0.57,
                Rf = 0.49, kd_R = 0.0572, kd_Rm = 0.612, ksr = 3.22,
                D = 20 * 1000 * (1/ molar.mass))

# Adjusting the speed of mRNA production with the ks_Rm and kd_Rm parameters.
parameters["ks_Rm"] <- 2.9 / 5
parameters["kd_Rm"] <- 2.9 / 5 / 4.74
ksrm.divided5 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["ks_Rm"] <- 2.9 / 2
parameters["kd_Rm"] <- 2.9 / 2 / 4.74
ksrm.divided2 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["ks_Rm"] <- 2.9 * 5
parameters["kd_Rm"] <- 2.9 * 5 / 4.74
ksrm.times5 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

parameters["ks_Rm"] <- 2.9 * 2
parameters["kd_Rm"] <- 2.9 * 2 / 4.74
ksrm.times2 <- as.data.frame(ode(y = state, times = times, func = model,
                                parms = parameters))

plot.ks_Rm.mRNA_R <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = mRNA.R, color = "Normal")) +
  geom_line(data = ksrm.divided5, aes(y = mRNA.R, color = "divided by 5")) +
  geom_line(data = ksrm.divided2, aes(y = mRNA.R, color = "divided by 2")) +
  geom_line(data = ksrm.times5, aes(y = mRNA.R, color = "times 5")) +
  geom_line(data = ksrm.times2, aes(y = mRNA.R, color = "times 2")) +
  scale_color_brewer("ks_Rm & kd_RM",
                    breaks = c("Normal", "divided by 5", "divided by 2", "times 5", "times 2"),
                    palette = "Set1")

plot.ks_Rm.R <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = R, color = "Normal")) +
  geom_line(data = ksrm.divided5, aes(y = R, color = "divided by 5")) +
  geom_line(data = ksrm.divided2, aes(y = R, color = "divided by 2")) +
  geom_line(data = ksrm.times5, aes(y = R, color = "times 5")) +
  geom_line(data = ksrm.times2, aes(y = R, color = "times 2")) +
  scale_color_brewer("ks_Rm & kd_RM",
                    breaks = c("Normal", "divided by 5", "divided by 2", "times 5", "times 2"),
                    palette = "Set1")

plot.ks_Rm.dr <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = dr, color = "Normal")) +
  geom_line(data = ksrm.divided5, aes(y = dr, color = "divided by 5")) +
  geom_line(data = ksrm.divided2, aes(y = dr, color = "divided by 2")) +
  geom_line(data = ksrm.times5, aes(y = dr, color = "times 5")) +
```



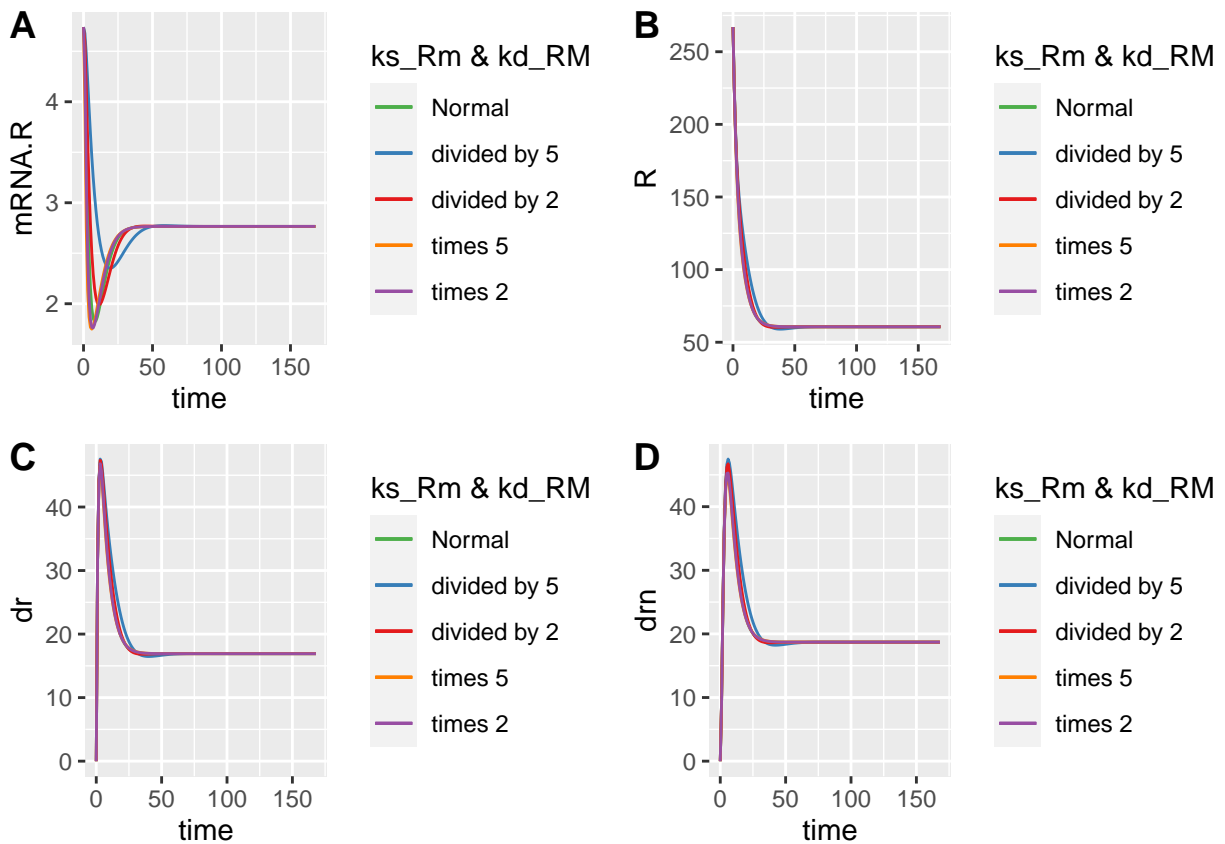
```

geom_line(data = ksrm.times2, aes(y = dr, color = "times 2")) +
scale_color_brewer("ks_Rm & kd_RM",
  breaks = c("Normal", "divided by 5", "divided by 2", "times 5", "times 2"),
  palette = "Set1")

plot.ks_Rm.drn <- ggplot(data = output.normal, aes(x = time)) +
  geom_line(aes(y = drn, color = "Normal")) +
  geom_line(data = ksrm.divided5, aes(y = drn, color = "divided by 5")) +
  geom_line(data = ksrm.divided2, aes(y = drn, color = "divided by 2")) +
  geom_line(data = ksrm.times5, aes(y = drn, color = "times 5")) +
  geom_line(data = ksrm.times2, aes(y = drn, color = "times 2")) +
  scale_color_brewer("ks_Rm & kd_RM",
    breaks = c("Normal", "divided by 5", "divided by 2", "times 5", "times 2"),
    palette = "Set1")

plot_grid(plot.ks_Rm.mRNA_R, plot.ks_Rm.R, plot.ks_Rm.dr, plot.ks_Rm.drn, labels = "AUTO")

```



Figuur 2.6: Het vertragen of versnellen van de mRNA productie vergeleken met normaal.

Als laatste is er gekeken of de snelheid van de mRNA productie nog enige invloed heeft op het model. En zoals te zien is in figuur 2.6 heeft het eigenlijk weinig tot geen invloed. Alleen bij de hoeveelheid receptor mRNA is er een verschil te zien wat ook begrijpelijk is als de snelheid aangepast wordt. Dit is te zien in figuur 2.6 A.

Hoofdstuk 3

Discussie en Conclusie

3.1 Discussie

Het grootste verschil in de waarden van het model met de metingen van het fysieke onderzoek zijn het aantal meetpunten en het verloop van de metingen. Bij het model word elk punt natuurlijk nauwkeurig berekend en is er geen kans voor toeval. Bij biologische processen en het meten daarvan gaat natuurlijk niet alles even soepel wat ook goed te zien is in figuur 2.1. De mediaan verspringt nogal eens tussen meetpunten, maar als daar doorheen wordt gekeken lijkt het wel redelijk gelijk aan het model. Er zijn verder ook geen verrassende metingen die in het oog springen. Wel hadden er meer metingen kunnen plaatsvinden. In dit onderzoek zijn er elke 6 uur vier metingen gedaan bij de twee dosissen, als dit al gehalveerd kan worden naar 3 uur is er een betere vergelijking met het model te maken. Het model word namelijk voor elk uur berekend.

3.2 Conclusie

Het doel van dit onderzoek was het vergelijken van het model met resultaten van een fysiek experiment. Op deze manier is er gekeken of het model ook klopt en of het geen verkeerde berekenen en dus waarden heeft. Hieruit is gebleken dat het model klopt en het goed overeenkomt met de waarden uit het experiment. Verder zijn er nog verschillende aanpassingen aan het model gedaan om hiervan de werking te zien. Denk hierbij aan het aanpassen of compleet weghalen van sommige variabelen of parameters. Bij een vervolg onderzoek zou er gekeken kunnen worden naar de precisie van het model, of naar de werking met andere corticosteroïden en of het daar ook mogelijk voor is om een model te maken.

Bibliografie

- [1] Soetaert, K., Petzoldt, T., and Woodrow Setzer, R.: *Solving differential equations in R: package deSolve*, J. Stat. Softw., 33, 1-25, 2010.