Assignment 4

Separation of Time Scales: Rapid Equilibrium Approximation

Juan Carlos Villaseñor-Derbez

February 8, 2018

Defining the system

The system has the following form:

$$A \underset{k_2}{\overset{k_1}{\rightleftarrows}} B \underset{k_4}{\overset{k_3}{\rightleftarrows}} C$$

And the system of ODE's that define it, is given by:

$$\begin{aligned}
\frac{da}{dt} &= (-k_1 * a) + (k_2 * b) \\
\frac{da}{dt} &= (k_1 * a) + (k_4 * c) - (k_2 * b) - (k_3 * b) \\
\frac{da}{dt} &= (k_3 * b) - (k_4 * c)
\end{aligned} \tag{1}$$

With rate constants $k_1 = 0.05$, $k_2 = 0.005$, $k_3 = 0.7$, and $k_4 = 0.4$.

Numerical solution

```r
# Define a function
system <- function(t, values, params){
  # Extract parameters
  k1 <- params[1]
  k2 <- params[2]
  k3 <- params[3]
  k4 <- params[4]

  # Extract state variables
  a <- values[1]
  b <- values[2]
  c <- values[3]

  # Calculate variables
  dadt <- (-k1 * a) + (k2 * b)
  dbdt <- (k1 * a) + (k4 * c) - (k2 * b) - (k3 * b)
  dcdt <- (k3 * b) - (k4 * c)

  # Return results
  return(list(c(dadt, dbdt, dcdt)))
}
```

```r
# Define timesteps
t <- seq(0, 50)

# Define initial conditions
a <- 1.5
b <- 3
c <- 2
values <- c(a, b, c)

# Define parameters
k1 <- 0.05
k2 <- 0.005
k3 <- 0.7
k4 <- 0.4
params <- c(k1, k2, k3, k4)

# Numerical solution
full <- lsoda(y = values, times = t, func = system, parms = params) %>%
  as.data.frame() %>%
  magrittr::set_colnames(value = c("time", "A", "B", "C")) %>%
  gather(Letter, Value, -time) %>%
  mutate(Model = "Full")
```

Visualizing the transient and steady-state behavior of the system

```r
ggplot(full, aes(x = time, y = Value, color = Letter)) +
  geom_line(size = 1) +
  cowplot::theme_cowplot() +
  scale_color_brewer(palette = "Set1")
```

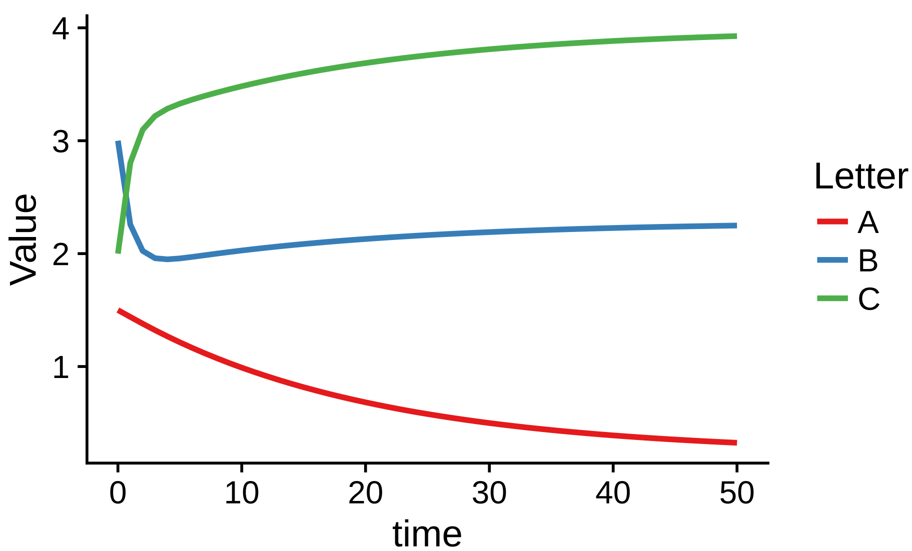

Figure 1: Transient and steady-state behavior of state variables A, B, and C

Separation of time-scales by the *rapid equilibrium assumption*

From the Eq 1 we can see that the main processes are between at A-B and B-C. However, they occur at different timescales, governed by the k_i coefficients.

For A-B, the timescale is given by:

$$\frac{1}{k_1 + k_2} = \frac{1}{0.05 + 0.005} = \frac{1}{0.055}$$

And the changes between B-C are given by:

$$\frac{1}{k_3 + k_4} = \frac{1}{0.7 + 0.4} = \frac{1}{1.1}$$

Since the conversions $B \underset{\leftarrow}{\rightarrow} C$ have a smaller time constant than the conversion $A \underset{\leftarrow}{\rightarrow} B$ (*i.e.* $\frac{1}{k_1+k_2} << \frac{1}{k_3+k_4}$), indicating that B and C reach an equilibrium faster than A. We can pool B and C together, and focus on how their pooled concentrations and A interact. From Eq 1, we have that

$$\frac{\tilde{c}(t)}{\tilde{b}(t)} = \frac{k_3}{k_4}$$

which in turn is

$$\tilde{c}(t) = \tilde{b}(t)\frac{k_3}{k_4} \tag{2}$$

This allows us to reduce the system to a pooled B and C group that converts to and from A. Let $\tilde{d}(t)$ be the total concentration in the pool $B + C$ (*i.e.* $\tilde{d}(t) = \tilde{b}(t) + \tilde{c}(t)$). The fractions of B and C are fixed by the equilibrium ratio. Thus, combining this with Eq 2, we obtain:

$$\begin{aligned}\tilde{d}(t) &= \tilde{b}(t) + \tilde{c}(t)\\ &= \tilde{b}(t) + \tilde{b}(t)\frac{k_3}{k_4}\\ &= \tilde{b}(t)\frac{k_4 + k_3}{k_4}\end{aligned}$$

And solving for $\tilde{b}(t)$ gives:

$$\tilde{b}(t) = \tilde{d}(t)\frac{k_4}{k_4 + k_3} \tag{3}$$

And

$$\tilde{c}(t) = \tilde{d}(t) - \tilde{b}(t) = \tilde{d}(t)\frac{k_3}{k_4 + k_3} \tag{4}$$

The pool D converts to and from A at rate $k_1\tilde{a}(t) - k_2\tilde{b}(t)$:

$$\begin{aligned}\frac{d}{dt}\tilde{d}(t) &= k_1\tilde{a}(t) - k_2\tilde{b}(t) \\ &= k_1\tilde{a}(t) - k_2\left(\frac{k_4}{k_4 + k_3}\tilde{d}(t)\right) \\ &= k_1\tilde{a}(t) - \frac{k_2 k_4}{k_4 + k_3}\tilde{d}(t)\end{aligned} \tag{5}$$

We can use Eq 5 along with $\frac{d}{dt}\tilde{a}(t) = (-k_1 + k_2)\tilde{d}(t)$ to calculate the long-term equilibrium of A and D, and then use the algebraic expressions 3 and 4 to calculate B and C at each time step.

Numerical solution of reduced model

```
# Define a function for the reduced system
reduced_system <- function(t, values, params){
  # Extract parameters
  k1 <- params[1]
  k2 <- params[2]
  k3 <- params[3]
  k4 <- params[4]

  # Extract state variables
  a <- values[1]
  d <- values[2]

  # Calculate rates
  dadt <- (-k1 * a) + (k2 * d)
  dddt <- (k1 * a) - (k2*((k4)/(k4+k3))*d)

  return(list(c(dadt, dddt)))
}
```

```
# Define new initial conditions
d <- b + c
values <- c(a, d)

# Numerical solution
reduced <- lsoda(y = values, times = t, func = reduced_system, parms = params) %>%
  as.data.frame() %>%
  magrittr::set_colnames(value = c("time", "A", "D")) %>%
  mutate(B = D*(k4/(k4+k3)),
         C = D*(k3/(k4+k3))) %>%
  gather(Letter, Value, -time) %>%
  mutate(Model = "Reduced")
```

```r
rbind(full, reduced) %>%
  ggplot(aes(x = time, y = Value, color = Letter, linetype = Model)) +
  geom_line(size = 1) +
  cowplot::theme_cowplot() +
  scale_color_brewer(palette = "Set1")
```

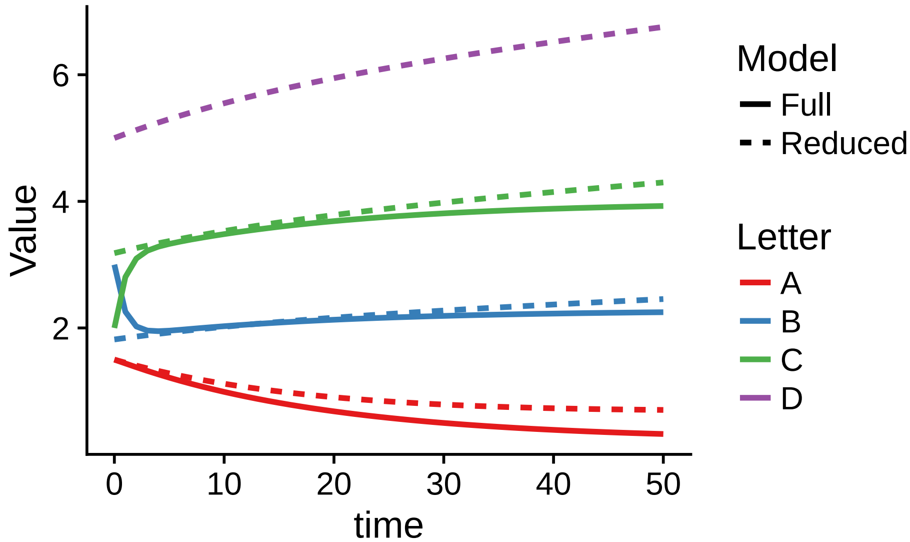

Figure 2: Comparision of transient and steady-state behavior of state variables A, B, and C between the full model and the reduced model.