water_bucket_model

Isaac Towers

19/05/2021

Here I will develop the prototype for the simple water bucket model for the **plant** model tracking the rate of change in θ

$$\frac{d\theta}{dt} = R * I(\theta) - T(\theta) - D(\theta) - E(\theta)$$

Variables

$\theta(t)$: soil volumetric water content (m^3/m^3)

Parameters

R: Rainfall $(m^3 m^{-2} yr^{-1})$ T: Transpiration (water used by plants and evaporation) $(m^3 m^{-2} yr^{-1})$ D: Drainage from plant available water from soil profile $(m^3 m^{-2} yr^{-1})$ b: Unitless parameter determining the shape of water accumulation in the soil moisture bank

$$I(\theta) = 1 - \left(\frac{\theta(t)}{\theta_{sat}}\right)^b$$

θ_{sat}: Volumetric water content at saturation (m^3/m^3)

Combining the two equations together

$$\frac{d\theta}{dt} = R * \left(1 - \left(\frac{\theta(t)}{\theta_{sat}}\right)^b\right) - T(\theta) - D(\theta)$$

Start **extremely** simple, show that soil moisture rises with yearly rainfall

$$\frac{d\theta}{dt} = R$$

```r
library(tidyverse)
```

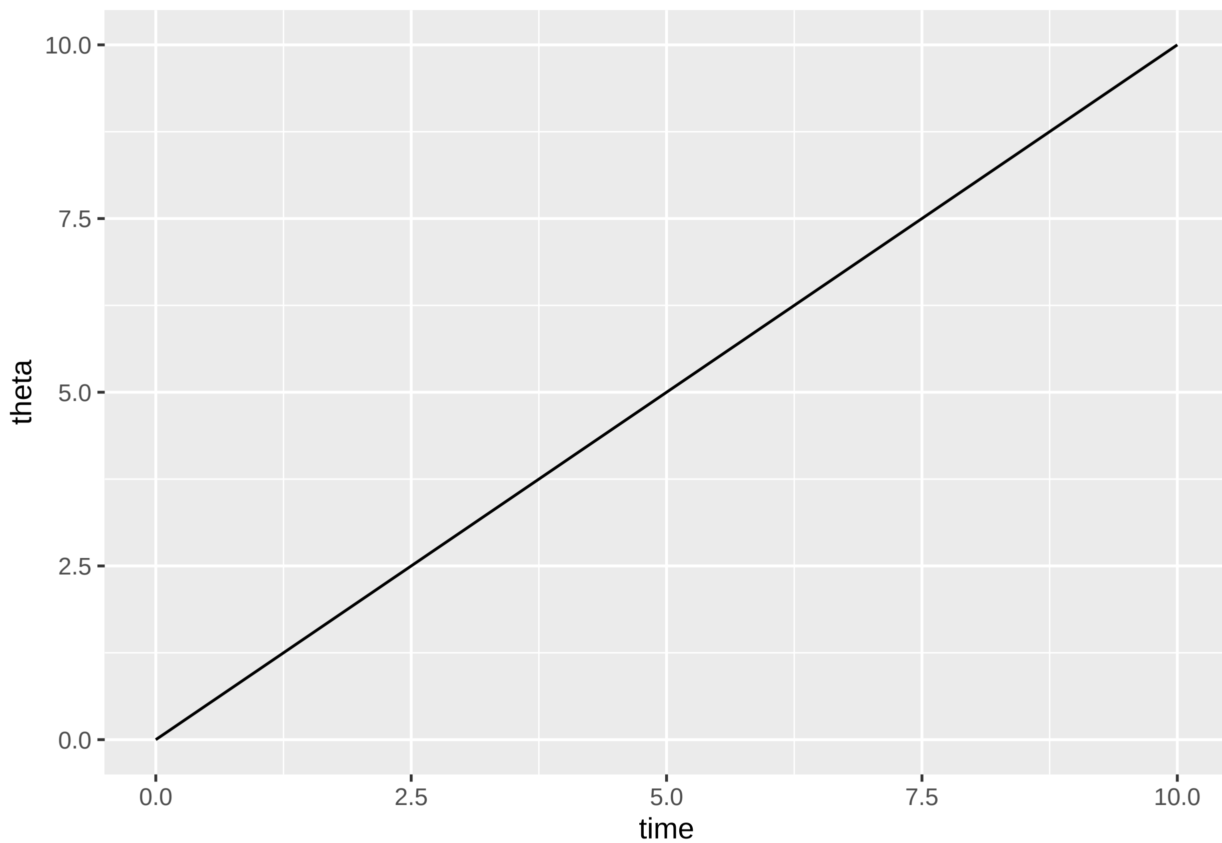
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(deSolve)
```

```r
# Timestep
t <- seq(0,10, by = 0.1)

# Variables
theta_init <- c(theta=0)

# Model parameters
params <- c(theta_sat = 0.7,
            R  = 1.0,
            b = 0.4)

# Rates of change
theta_rates <- function(t, theta, params) {
  with(as.list(c(theta, params)), {

    dtheta_dt = R

    return(list(c(dtheta_dt)))
  })
}
```

```r
logistic_solution <-
  ode(
    y = theta_init,
    times = t,
    func = theta_rates,
    parms = params) %>%
  as.data.frame()
```

```r
logistic_solution %>%
  ggplot(aes(time, theta)) +
  geom_line()
```



Ok, lets first show that rain will eventually fill up the soil without any water outputs if we include a saturated water content

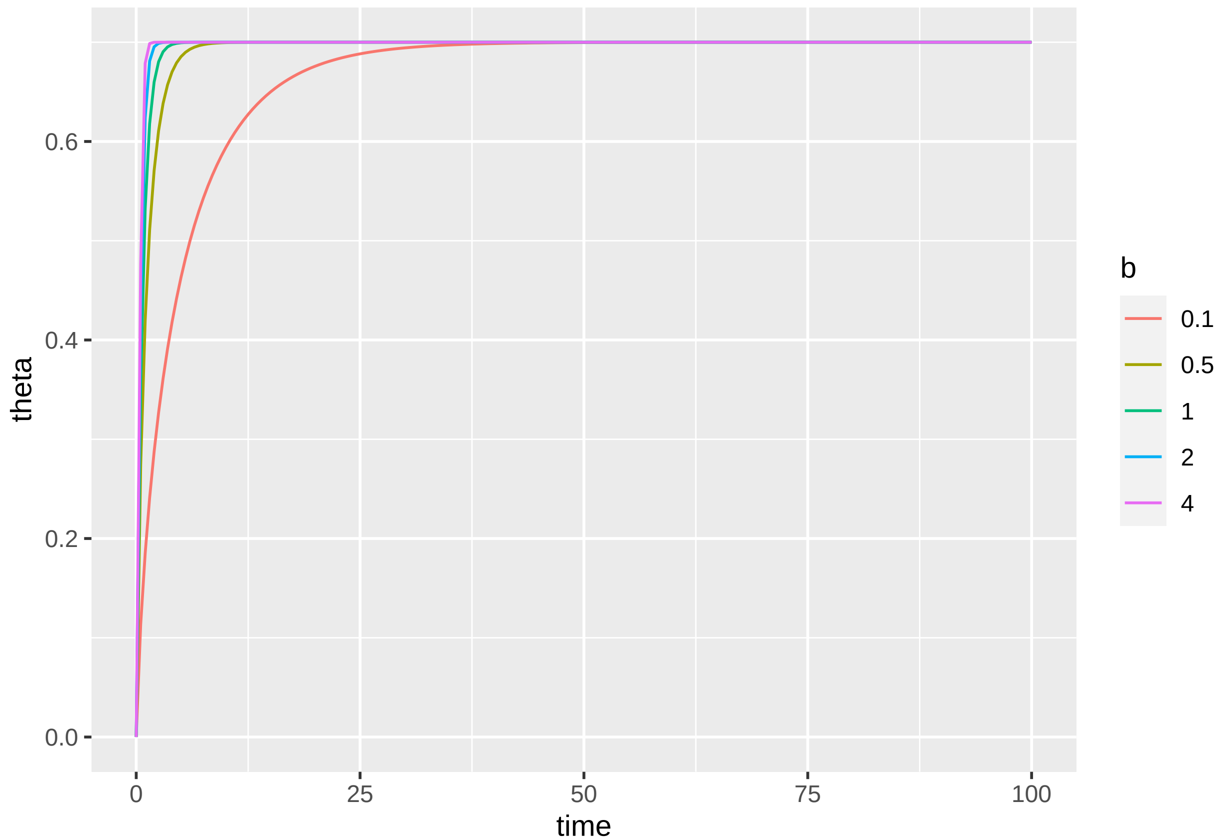
$$\frac{d\theta}{dt} = R * \left(1 - \left(\frac{\theta(t)}{\theta_{sat}}\right)^b\right)$$

```
# Timestep
t <- seq(0,100, by = 0.5)

# Variables
theta_init <- c(theta=0)

# Model parameters
params <-
  expand_grid(theta_sat = 0.7,
            R  = 1.0,
            b = c(1, 0.5, 2, 4, 0.1)) %>%
        split(., .$b)

# Rates of change
theta_rates <- function(t, theta, params) {
  with(as.list(c(theta, params)), {

    dtheta_dt = R*(1-(theta/theta_sat)^b)

    return(list(c(dtheta_dt)))
```

```
  })
}
```

```
logistic_solution <-
  map_dfr(
    params,
    ~ode(
    y = theta_init,
    times = t,
    func = theta_rates,
    parms = .x) %>%
  as.data.frame(), .id = "b")
```

```
logistic_solution %>%
  ggplot(aes(time, theta, col=b)) +
  geom_line()
```



Cool, let's now include a drainage factor

$$\frac{d\theta}{dt} = R * \left(1 - \left(\frac{\theta(t)}{\theta_{sat}}\right)^b\right) - D(\theta)$$

Drainage is equal to the hydraulic conductivity of the soil K_s when $\theta = \theta_{sat}$ but declines with soil soil moisture (Zavala et al. 2005)

$$D(\theta) = K_s \left(\frac{\theta(t)}{\theta_{sat}}\right)^c$$

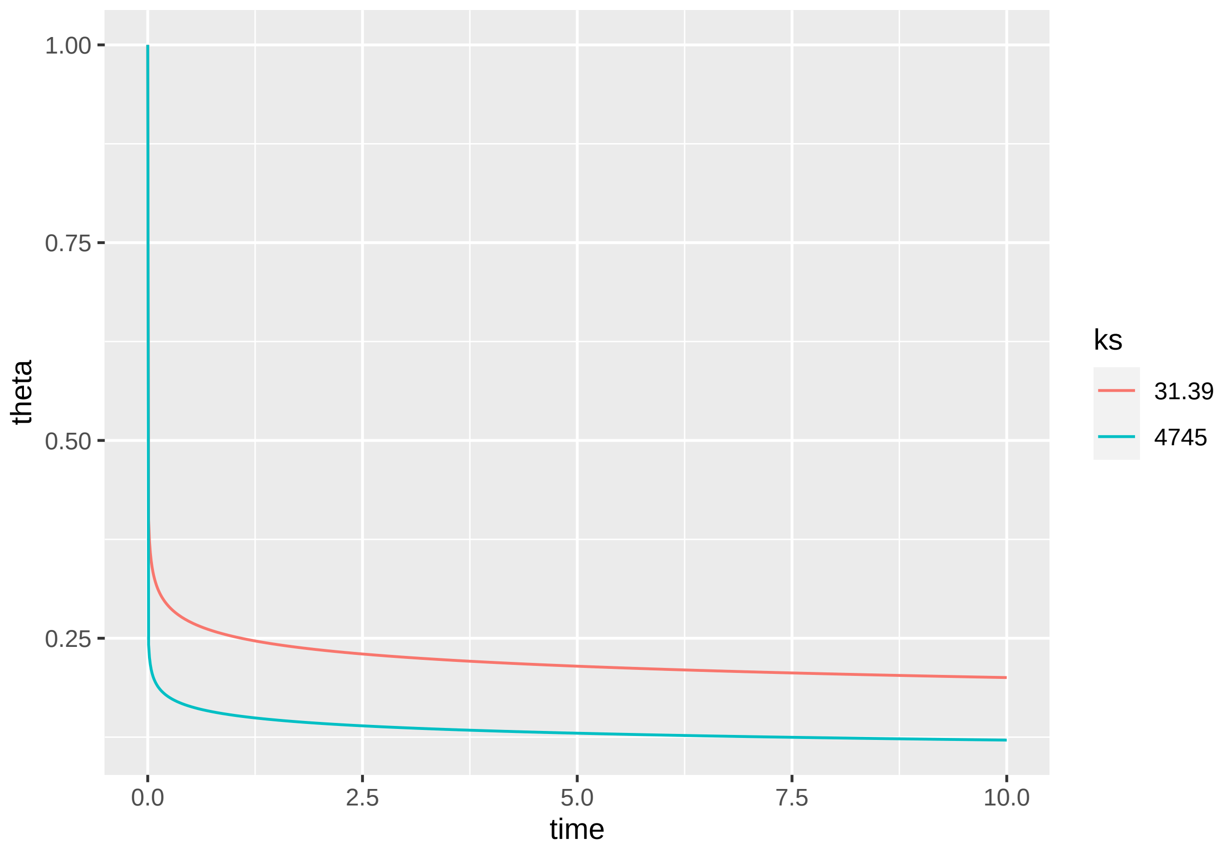
Parameters

K_s: Saturated soil hydraulic conductivity $(m^3 m^{-2} yr^{-1})$ c: Unitless empirical soil parameter

```
# Timestep
t <- seq(0,10, by = 0.01)

# Variables
theta_init <- c(theta=1)

# Model parameters
params <-
  expand_grid(theta_sat = 0.482,
              R  = 1.0,
              ks = c(31.39, 4745),
              c = 11,
              b = 4) %>%
        split(., .$ks)

# Rates of change
theta_rates <- function(t, theta, params) {
  with(as.list(c(theta, params)), {

    dtheta_dt = -ks*(theta/theta_sat)^c

    return(list(c(dtheta_dt)))
  })
}
```

```
logistic_solution <-
  map_dfr(
    params,
    ~ode(
    y = theta_init,
    times = t,
    func = theta_rates,
    parms = .x) %>%
  as.data.frame(), .id = "ks")
```

```
logistic_solution %>%
  ggplot(aes(time, theta, col=ks)) +
  geom_line()
```



Parameters

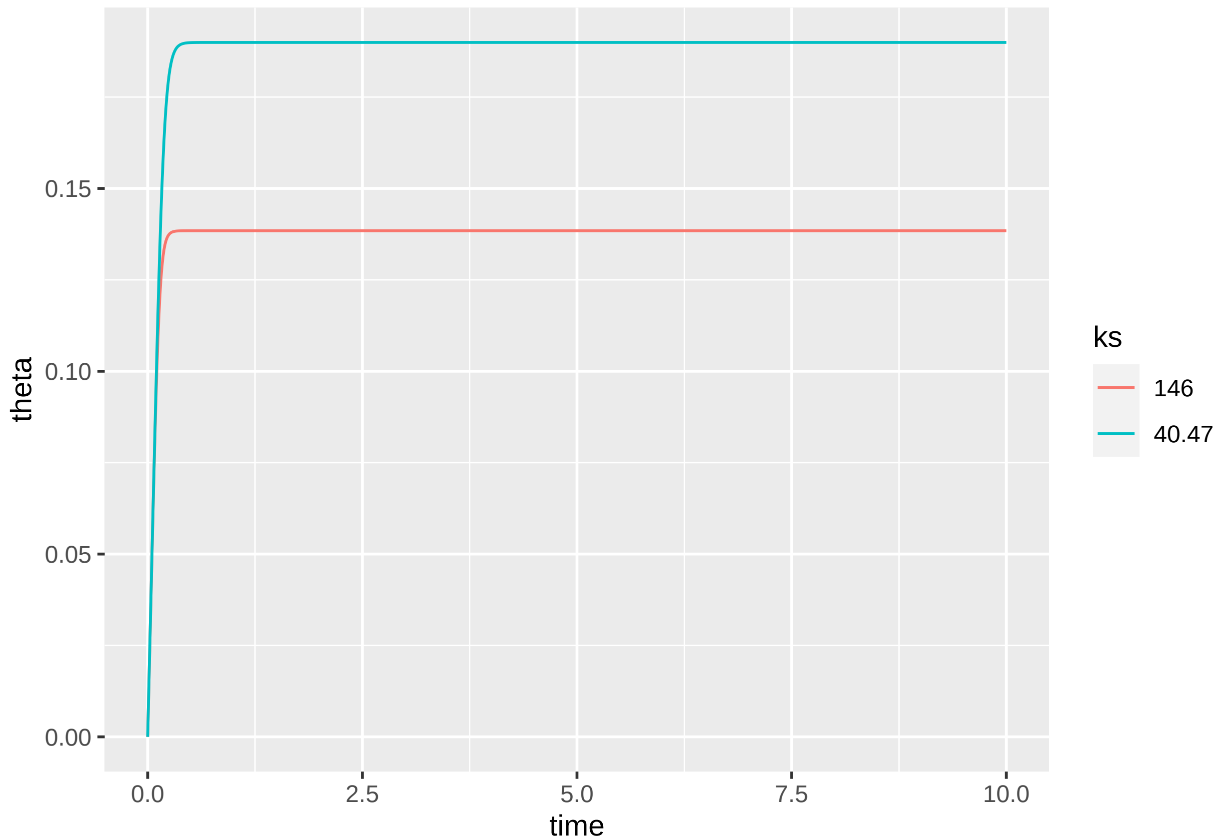
K_s: Saturated soil hydraulic conductivity $(m^3 m^{-2} yr^{-1})$ c: Unitless empirical soil parameter

```
# Timestep
t <- seq(0,10, by = 0.01)

# Variables
theta_init <- c(theta=0)

# Model parameters
params <-
  expand_grid(theta_sat = 0.482,
              R  = 1.0,
              ks = c(40.47, 146),
              c = 4,
              b = 4) %>%
        split(., .$ks)

# Rates of change
theta_rates <- function(t, theta, params) {
  with(as.list(c(theta, params)), {

    dtheta_dt = R*(1-(theta/theta_sat)^b)-ks*(theta/theta_sat)^c

    return(list(c(dtheta_dt)))
  })
```

```
}
```

```
logistic_solution <-
  map_dfr(
    params,
    ~ode(
    y = theta_init,
    times = t,
    func = theta_rates,
    parms = .x) %>%
  as.data.frame(), .id = "ks")
```

```
logistic_solution %>%
  ggplot(aes(time, theta, col=ks)) +
  geom_line()
```



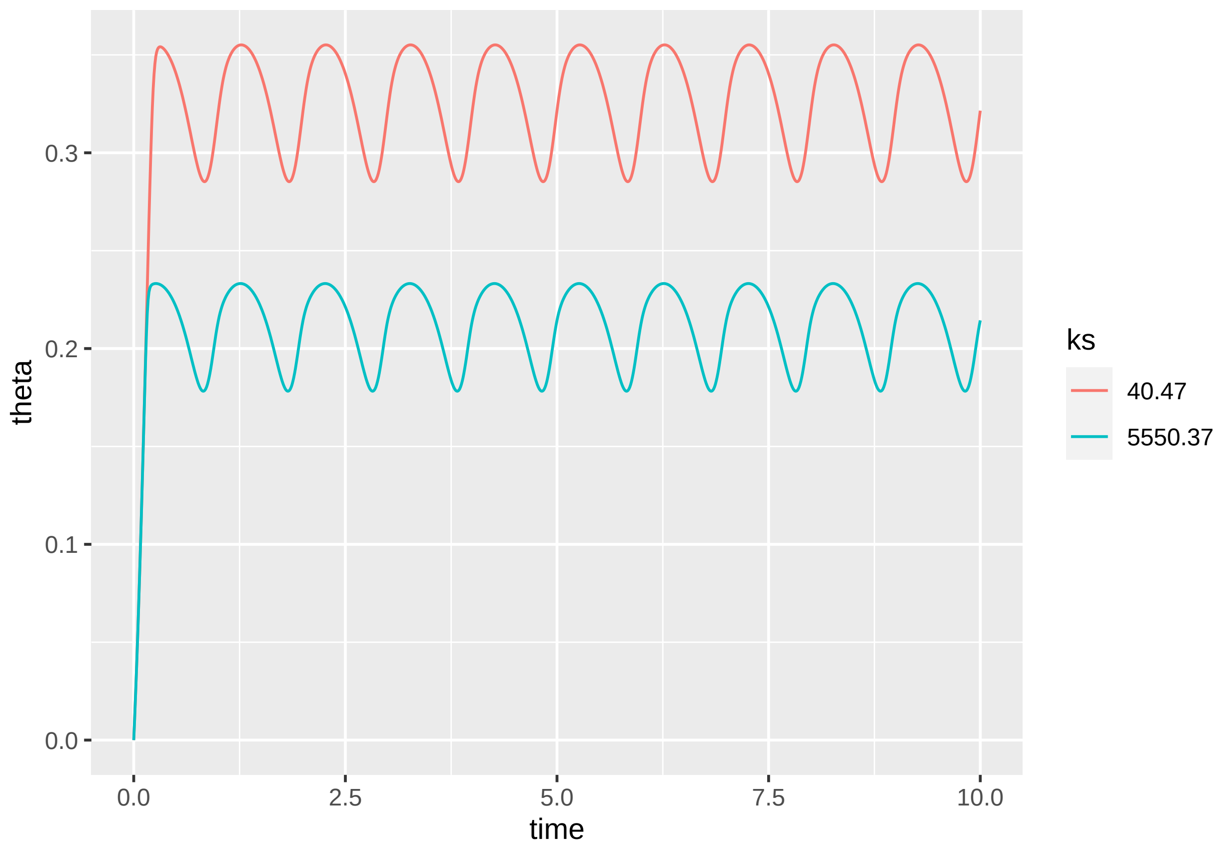
Now with climatic variability of amplitude = the annual mean rainfall

```
# Timestep
t <- seq(0,10, by = 0.01)

# Variables
theta_init <- c(theta=0)

# Model parameters
params <-
```

```r
  expand_grid(theta_sat = 0.482,
              R  = 1.0,
              ks = c(40.47, 5550.37),
              c = 11,
              b = 4) %>%
        split(., .$ks)

# Rates of change
theta_rates <- function(t, theta, params) {
  with(as.list(c(theta, params)), {

    dtheta_dt = (R+R*sin(t/(2*pi)^-1))*(1-(theta/theta_sat)^b)-ks*(theta/theta_sat)^c

    return(list(c(dtheta_dt)))
  })
}
```

```r
logistic_solution <-
  map_dfr(
    params,
    ~ode(
    y = theta_init,
    times = t,
    func = theta_rates,
    parms = .x) %>%
  as.data.frame(), .id = "ks")
```

```r
logistic_solution %>%
  ggplot(aes(time, theta, col=ks)) +
  geom_line()
```

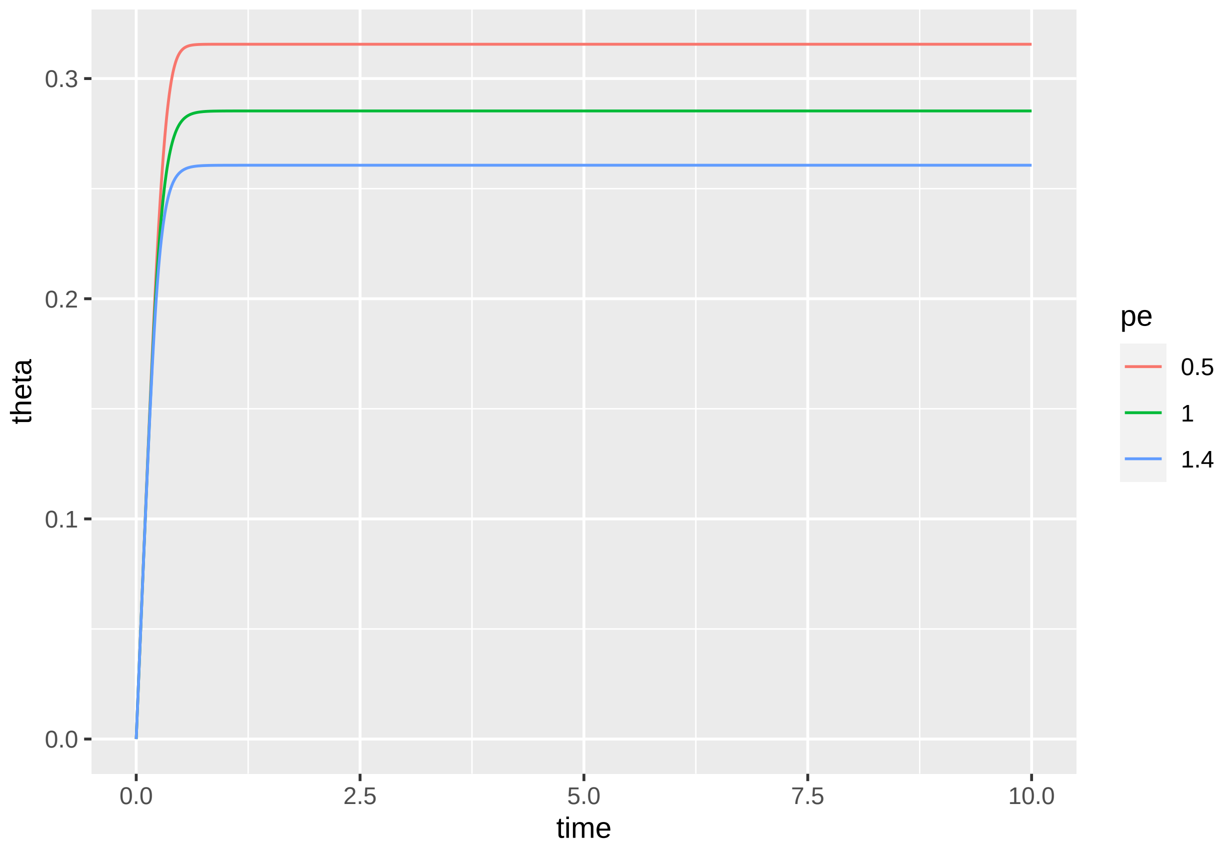



```r
# Timestep
t <- seq(0,10, by = 0.01)

# Variables
theta_init <- c(theta=0)

# Model parameters
params <-
  expand_grid(theta_sat = 0.482,
            R  = 1.0,
            ks = c(40.47),
            c = 11,
            b = 4,
            pe = c(0.5,1,1.4),
            k = 0.46,
            LAI = 0) %>%
        split(., .$pe)


# Rates of change
theta_rates <- function(t, theta, params) {
  with(as.list(c(theta, params)), {

    dtheta_dt = R*(1-(theta/theta_sat)^b)-ks*(theta/theta_sat)^c -
    pe*(exp(-k * LAI)/(1 + exp (-12 * (theta /theta_sat - 0.5))))
```

```r
    return(list(c(dtheta_dt)))
  })
}
```

```r
logistic_solution <-
  map_dfr(
    params,
    ~ode(
    y = theta_init,
    times = t,
    func = theta_rates,
    parms = .x) %>%
  as.data.frame(), .id = "pe")
```

```r
logistic_solution %>%
  ggplot(aes(time, theta, col=pe)) +
  geom_line()
```

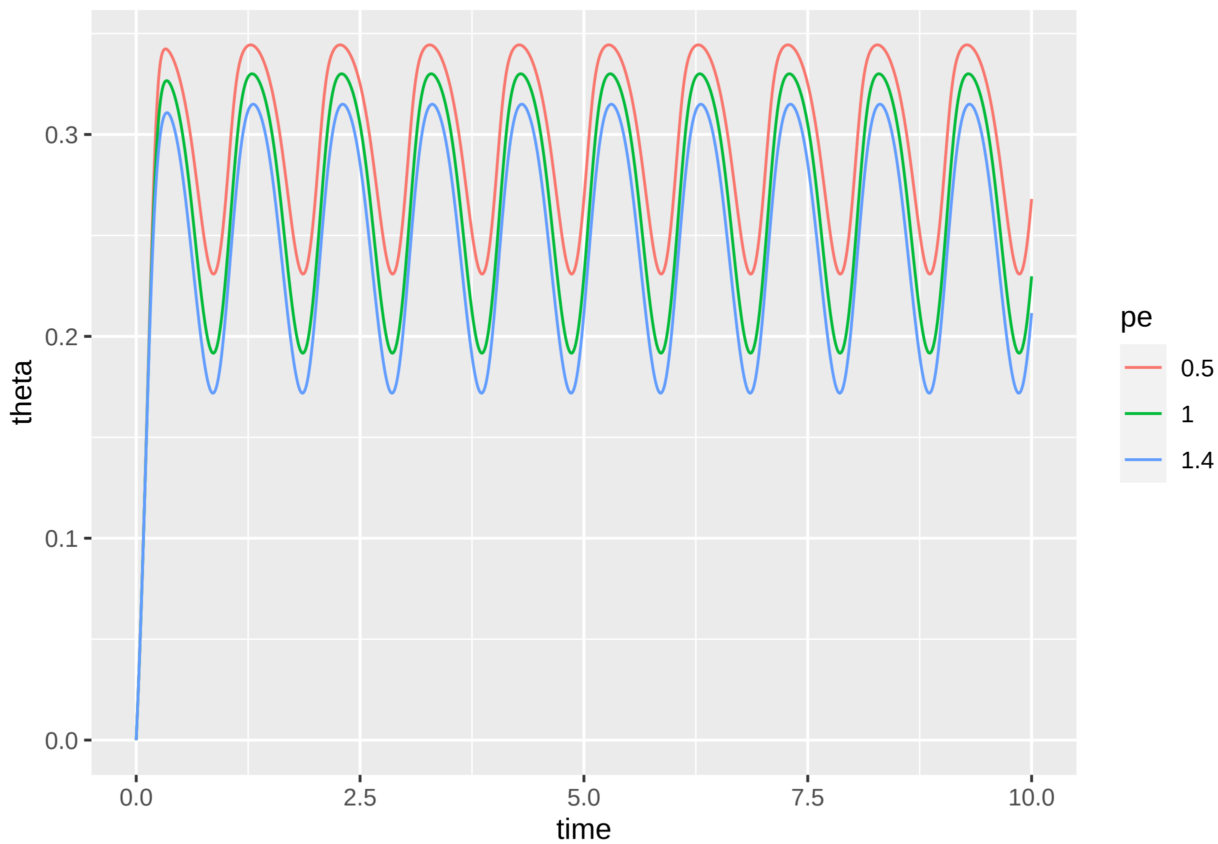



```r
# Timestep
t <- seq(0,10, by = 0.01)

# Variables
theta_init <- c(theta=0)

# Model parameters
```

```r
params <-
  expand_grid(theta_sat = 0.482,
              R  = 1.0,
              ks = c(40.47),
              c = 11,
              b = 4,
              pe = c(0.5, 1,1.4),
              k = 0.46,
              LAI = 0) %>%
        split(., .$pe)


# Rates of change
theta_rates <- function(t, theta, params) {
  with(as.list(c(theta, params)), {

    dtheta_dt = (R+R*sin(t/(2*pi)^-1))*(1-(theta/theta_sat)^b)-ks*(theta/theta_sat)^c -
    pe*(exp(-k * LAI)/(1 + exp (-12 * (theta /theta_sat - 0.5))))

    return(list(c(dtheta_dt)))
  })
}
```

```r
logistic_solution <-
  map_dfr(
    params,
    ~ode(
    y = theta_init,
    times = t,
    func = theta_rates,
    parms = .x) %>%
  as.data.frame(), .id = "pe")
```

```r
logistic_solution %>%
  ggplot(aes(time, theta, col=pe)) +
  geom_line()
```



0.3
0.2
0.1
0.0
theta
0.0
2.5
5.0
7.5
10.0
time
pe
0.5
1
1.4