

RTM_0D template

```
# Initial conditions of the state variables
yini <- c(O2 = 0.25, BOD = 0.5) # both in [mol/m3]

# Model parameters
pars <- c(
  rDecay = 0.05, # [/d]      first-order rate constant
  kO2    = 0.001, # [mol/m3] half-saturation O2 concentration
  satO2  = 0.3,  # [mol/m3]  O2 solubility
  k       = 0.1, # [/d]      reaeration rate constant
  inputBOD = 0.001 # [mol/m3/d] constant BOD input rate
)

# Model function: calculates time-derivatives and other output
BODmodel <-function(t, state, pars) {
  # t: time, state: state variables, pars: model parameters
  with (as.list(c(state, pars)),{

    # rate expressions [mol/m3/d]
    Decay      <- rDecay * BOD * O2/(O2+kO2) # BOD decay
    Reaeration <- k * (satO2-O2)              # air-water exchange

    # Time-derivatives: dC/dt = production - consumption [mol/m3/d]
    dO2dt      <- Reaeration - Decay
    dBODdt     <- inputBOD - Decay

    # return time-derivatives and ordinary variables as a list
    list(c(dO2dt, dBODdt), # vector with derivatives
         # (the same order as state variables!)
         Decay = Decay, # other output
         Reaeration = Reaeration)
  })
}
```

```
require(deSolve) # package with integration methods
# vector of output times
outtimes <- seq(from = 1, to = 100, length.out = 100)
# ode integrates the model
out <- ode(y=yini, parms=pars, func=BODmodel, times=outtimes)
# plot the model output
plot(out)
```

Simplest COVID model (Exercise 14)

```
# Initial conditions of the state variables (D=deceased)
yini <- c(S = 17.5e6, I = 1e3, R = 0, D = 0) # number of people

# Model parameters
pars <- c(
  b = 0.00000002, # [/ind/d]  infection rate constant
  g = 0.07,       # [/d]      recovery rate constant
  m = 0.007       # [/d]      mortality rate constant
)

# Model function: calculates time-derivatives and other output
SIRmodel <-function(t, state, pars) {
  # t: time, state: state variables, pars: model parameters
  with (as.list(c(state, pars)),{

    # rate expressions [ind/d]
    Infection <- b * I * S # infection rate
    Recovery  <- g * I     # recovery rate
    Mortality <- m * I     # mortality rate

    # Time-derivatives: dC/dt = production - consumption [ind/d]
    dSdt      <- -Infection
    dIdt      <- Infection - Recovery - Mortality
    dRdt      <- Recovery
    dDdt      <- Mortality

    # return time-derivatives and ordinary variables as a list
    list(c(dSdt, dIdt, dRdt, dDdt), # vector with derivatives
         # (the same order as state variables!)
         Infection = Infection, # other output
         Mortality = Mortality)
  })
}
```

```
require(deSolve) # package with integration methods
# vector of output times
outtimes <- seq(from = 1, to = 100, length.out = 100)
# ode integrates the model
out <- ode(y=yini, parms=pars, func=SIRmodel, times=outtimes)
# plot the model output
plot(out)
```

(f)