Solving Differential Equations in R (book) - DDE examples

Karline Soetaert
Royal Netherlands Institute of Sea Research (NIOZ)
Yerseke, The Netherlands

Abstract

This vignette contains the R-examples of chapter 8 from the book:

Soetaert, K., Cash, J.R. and Mazzia, F. (2012). Solving Differential Equations in R. UseR series, Springer, 248 pp.

www.springer.com/statistics/computational+statistics/book/978-3-642-28069-6.

Chapter 8. Solving Delay Differential Equations in R.

Here the code is given without documentation. Of course, much more information about each problem can be found in the book.

Keywords: delay differential equations, initial value problems, examples, R.

1. Two simple examples

```
DDE1 <- function(t, y, parms) {
  tlag <- t - 1
  if (tlag <= 0)
    ylag <- 1
  else
    ylag <- lagvalue(tlag)

  list(dy = - ylag, ylag = ylag)
}
yinit <- 1
times <- seq(from = 0, to = 10, by = 0.1)
yout  <- dede(y = yinit, times = times, func = DDE1,
             parms = NULL, atol = 1e-10, rtol = 1e-10 )
tt <- which(times >= 1 & times <= 2)
analytic <- c(1-times[times <1] , 0.5*times[tt]^2 - 2*times[tt]+3/2)
max(abs(yout[times <= 2,2] - analytic))
```

```
[1] 1.388897e-10
```

```
DDE2 <- function(t, y, parms) {
  tlag <- t - 1
```

```
  if (tlag <= 0)
    ylag <- 1
  else
    ylag <- lagderiv(tlag)

  list(dy = - ylag, ylag = ylag)
}
yout2 <- dede(y = yinit, times = times, func = DDE2,
              parms = NULL )
```

2. Chaotic Production of White Blood Cells

```
mackey <- function(t, y, parms, tau) {
  tlag <- t - tau
  if (tlag <= 0)
    ylag <- 0.5
  else
    ylag <- lagvalue(tlag)
  dy <- 0.2 * ylag * 1/(1+ylag^10) - 0.1 * y
  list(dy = dy, ylag = ylag)
}
yinit <- 0.5
times <- seq(from = 0, to = 300, by = 0.1)
yout1 <- dede(y = yinit, times = times, func = mackey,
             parms = NULL, tau = 10)
yout2 <- dede(y = yinit, times = times, func = mackey,
             parms = NULL, tau = 20)

plot(yout1, lwd = 2, main = "tau=10",
    ylab = "y", mfrow = c(2, 2), which = 1)
plot(yout1[,-1], type = "l", lwd = 2, xlab = "y")
plot(yout2, lwd = 2, main = "tau=20",
    ylab = "y", mfrow = NULL, which = 1)
plot(yout2[,-1], type = "l", lwd = 2, xlab = "y")
```

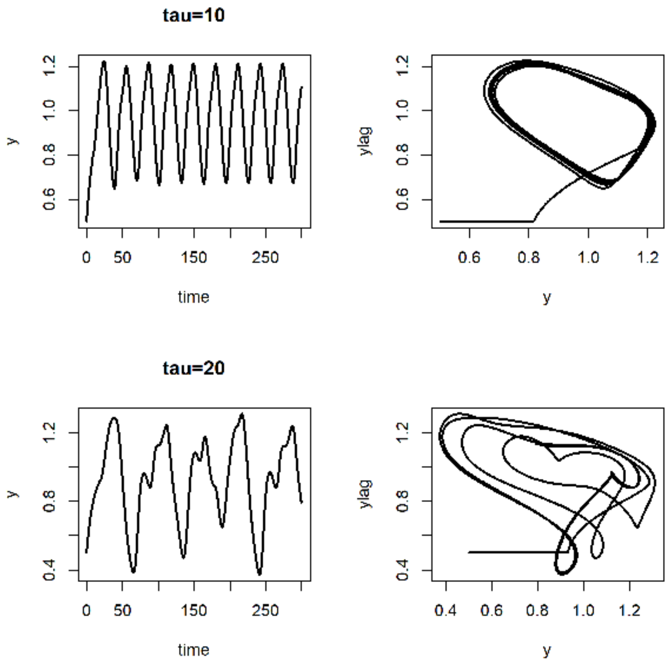

Figure 1: The Mackey-Glass DDE. See book for more information.

3. A DDE involving a Root Function

```
xb  <- -0.427; a <- 0.16; xi <- 0.02; u <- 0.5; tau <- 1
yinit <- c(y = 0.6)
mariott <- function(t, y, parms) {
  tlag <- t - 12
  if (tlag <= 0)
    ylag <- 0.6
  else
    ylag <- lagvalue(tlag)

  Delt  <- ylag - xb
  sDelt <- sign(Delt)

  dy <- (-y + pi*(a + xi*sDelt - u*(sin(Delt))^2))/tau
  list(dy)
}
times <- seq(from = 0, to = 120, by = 0.5)
yout <- dede(y = yinit, times = times, func = mariott,
            parms = NULL)
root <- function(t, y, parms) {
  tlag <- t - 12
  if (tlag <= 0)
    return (1) # not a root
  else
    return(lagvalue(tlag)- xb)
}
event <- function(t, y, parms) return(y)
yout <- dede(y = yinit, times = times, func = mariott,
            parms = NULL, rootfun = root,
            events = list(func = event, root = TRUE))
attributes(yout)$troot
```

```
[1]  14.01588  24.49263  67.54678  75.18141 118.43615
```

```
plot(yout, lwd = 2,
    main = "Controller problem")
abline(v = attributes(yout)$troot, col = "grey")
```

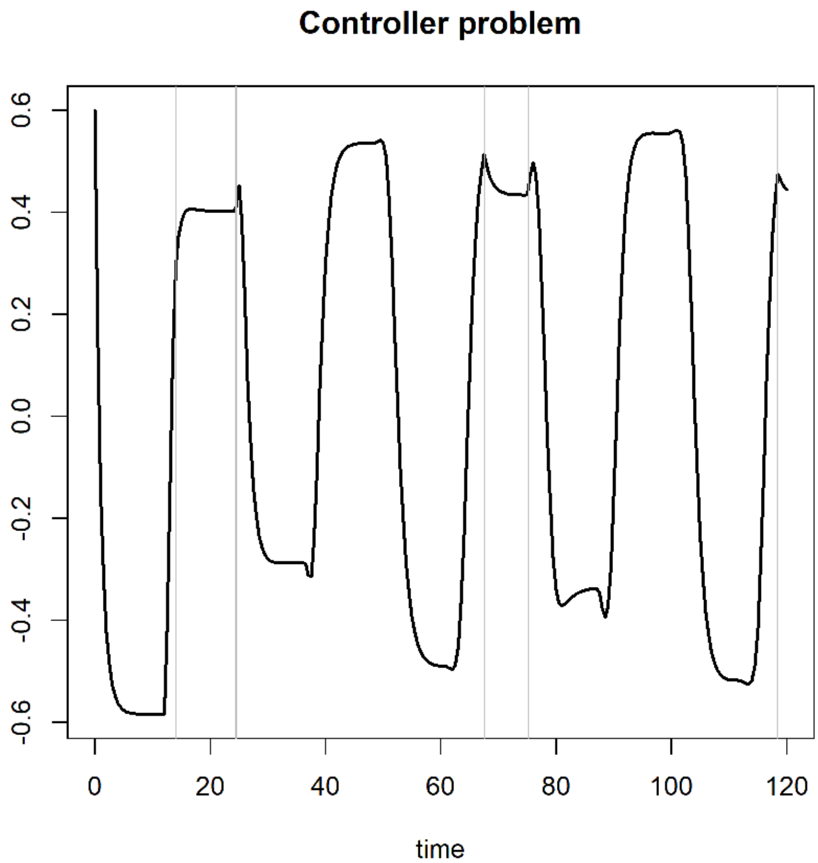

Figure 2: Solution of the Controller problem. See book for explanation.

4. Vanishing Time Delay

```
vanishing <- function(t, y, parms, cc) {

  tlag <- t*y^2
  if (tlag <= 0) {
    ylag  <- 0
    dylag <- 0
  } else {
    ylag  <- lagvalue(tlag)
    dylag <- lagderiv(tlag)
  }
  dy <- cos(t)*(1+ylag) + cc*y*dylag +
       (1-cc)*sin(t)*cos(t*sin(t)^2) - sin(t+t*sin(t)^2)

  list(dy)
}
yinit <- c(y = 0)
times <- seq(from = 0, to = 2*pi, by = 0.1)
yout <- dede(y = 0, times = times, func = vanishing,
             parms = NULL, cc = -0.5,
             atol = 1e-10, rtol = 1e-10)
print(max(abs(yout[,2] - sin(yout[,1]))))
```

```
[1] 1.397827e-06
```

5. Predator-Prey Dynamics with Harvesting

```
LVdede <- function(t, y, p) {
 if (t > tau1) Lag1 <- lagvalue(t - tau1) else Lag1 <- yini
 if (t > tau2) Lag2 <- lagvalue(t - tau2) else Lag2 <- yini

 dy1 <- r * y[1] *(1 - Lag1[1]/K) - a*y[1]*y[2]
 dy2 <- a * b * Lag2[1]*Lag2[2] - d*y[2]

 list(c(dy1, dy2))
}
rootfun <- function(t, y, p)
  return(y[1] - Ycrit)
eventfun <- function(t, y, p)
  return (c(y[1] * 0.7, y[2]))
r <- 1; K <- 1; a <- 2; b <- 1; d <- 1; Ycrit <- 1.2*d/(a*b)
tau1 <- 0.2; tau2 <- 0.2
yini <- c(y1 = 0.2, y2 = 0.1)
times <- seq(from = 0, to = 200, by = 0.01)
yout <- dede(func = LVdede, y = yini, times = times,
           parms = 0, rootfun = rootfun,
           events = list(func = eventfun, root = TRUE))
attributes(yout)$troot [1:10]
```

```
[1]  2.125283  3.057600  3.991063  4.926748  5.863709  6.803034  7.743653  8.685925
[9]  9.630663 10.577380
```

```
plot(yout[,-1], type = "l")
```

Affiliation:

Karline Soetaert
Royal Netherlands Institute of Sea Research (NIOZ)
4401 NT Yerseke, Netherlands E-mail: karline.soetaert@nioz.nl
URL: http://www.nioz.nl

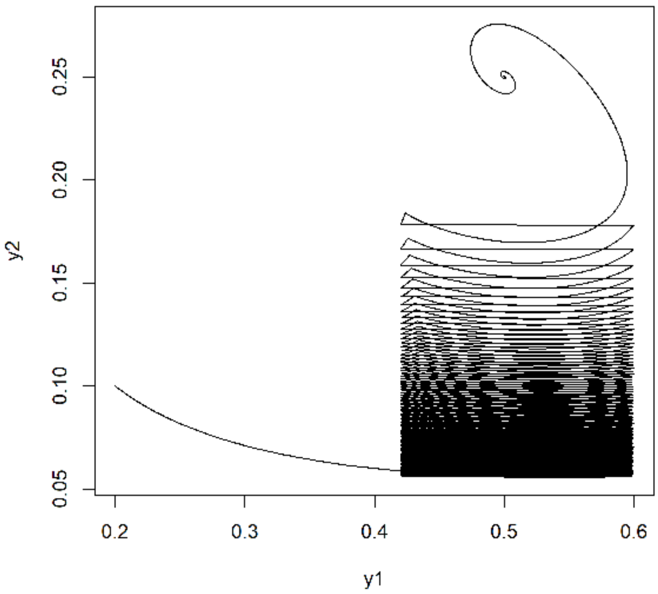

Figure 3: Solution of the predator-prey DDE model. See book for explanation.