Solving Differential Equations in R (book) - DAE examples

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

Abstract

This vignette contains the R-examples of chapter 6 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 6. Solving Differential Algebraic 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: differential algebraic equations, initial value problems, examples, R.

1. A simple DAE of Index 2

```
resdae <- function (t, y, dy, p) {
  r1 <- dy[1] - y[2]
  r2 <- y[1] - cos(t)
  list(c(r1, r2))
}
library(deTestSet)
yini  <- c(y1 = cos(0), y2 = -sin(0))
dyini <- c(-sin(0), -cos(0))
times <- seq(from = 0, to = 10, by = 0.1)
index <- c(1, 1, 0)
out1   <- mebdfi(times = times, res = resdae, y = yini,
                 atol = 1e-10, rtol = 1e-10, dy = dyini,
                 parms = NULL, nind = index)
max (abs(out1[,"y1"] - cos(times)), abs(out1[,"y2"] + sin(times)))
```

```
[1] 2.349123e-09
```

```
fundae <- function (t, y, p) {
  f1 <- y[2]
  f2 <- y[1] - cos(t)
  list(c(f1, f2))
```

```
}
M <- matrix(nrow = 2, ncol = 2, data = c(1, 0, 0, 0))
out2 <- radau(times = times, fun = fundae, y = yini,
              atol = 1e-10, rtol = 1e-10, mass = M,
              parms = NULL, nind = index)
max (abs(out2[,"y1"] - cos(times)), abs(out2[,"y2"] + sin(times)))
```

```
[1] 1.098162e-06
```

2. A Nonlinear Implicit DAE of index 1

```
implicit <- function(t, y, dy, parms) {
   list(t*y^2*dy^3 - y^3*dy^2 + t*(t^2+1)*dy - t^2*y)
}
yini  <- sqrt(3/2)
times <- seq(from = 1, to = 10, by = 0.1)
library(rootSolve)
rootfun <- function (dy, y, t)
  t*y^2*dy^3 - y^3*dy^2 + t*(t^2+1)*dy - t^2*y
dyini <- multiroot(f = rootfun, start = 0, y = yini,
                  t = times[1] )$root
dyini
```

```
[1] 0.8164966
```

```
out   <- mebdfi(times = times, res = implicit, y = yini,
               dy = dyini, parms = NULL)
out2  <- daspk (times = times, res = implicit, y = yini,
               dy = dyini, parms = NULL)
max(abs(out [,2]- sqrt(times^2+0.5)))
```

```
[1] 3.017694e-06
```

```
max(abs(out2[,2]- sqrt(times^2+0.5)))
```

```
[1] 3.869883e-05
```

```
implicit2 <- function (t, y, p) {
   f1 <- y[2]
   f2 <- t*y[1]^2*y[2]^3-y[1]^3*y[2]^2+t*(t^2+1)*y[2]-t^2*y[1]
   list(c(f1, f2))
 }
M <- matrix(nrow = 2, ncol = 2, data = c(1, 0, 0, 0))
yini_li      <- c(yini,dyini)
out3 <- radau(times = times, fun = implicit2, y = yini_li,
              mass = M, parms = NULL)
out4 <- gamd (times = times, fun = implicit2, y = yini_li,
              mass = M, parms = NULL)
max(abs(out3[,2]- sqrt(times^2+0.5)))
```

```
[1] 3.410377e-08
```

```
max(abs(out4[,2]- sqrt(times^2+0.5)))
```

```
[1] 1.990159e-06
```

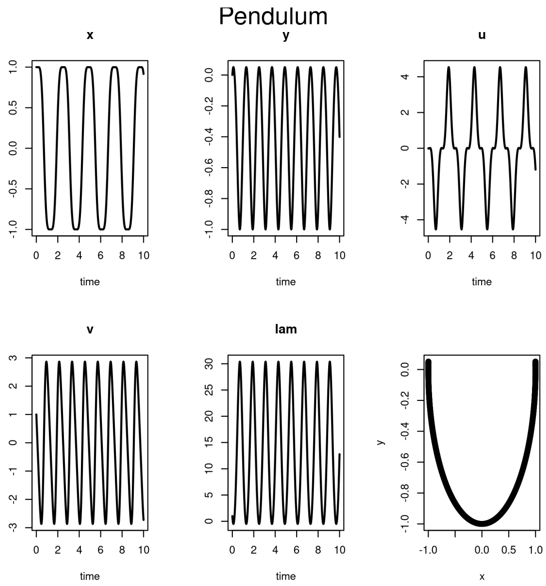

Figure 1: Solution of the pendulum problem. See book for explanation.

3. The Pendulum Problem

```
library(deTestSet)
pendulum <- function (t, y, dy, parms) {
 list(c(-dy[1] + y[3]          ,
        -dy[2] + y[4]          ,
        -dy[3] -y[5]*y[1]      ,
        -dy[4] -y[5]*y[2] - 9.8,
         y[1]^2 + y[2]^2 -1
     ))
}
yini  <- c(x = 1,  y = 0, u = 0,  v = 1   , lam = 1)
dyini <- c(dx = 0,dy = 1,du = -1,dv = -9.8,dlam = 3*9.8)
times <- seq(from = 0, to = 10, by = 0.01)
index3 <- c(2, 2, 1)
out3 <- mebdfi (y = yini, dy = dyini, res = pendulum,
              parms = NULL, times = times,
              nind = index3)

plot(out3, lwd = 2)
plot(out3[, 2:3])
mtext(side = 3, outer = TRUE, line = -1.5,
     "Pendulum", cex = 1.5)
```

4. The Car Axis problem

```
caraxis <- function(t, y, dy, parms) {
 with(as.list(y), {
   f <- rep(0, 10)
   yb <- r * sin(w * t)
   xb <- sqrt(L^2  - yb^2)
   Ll <- sqrt(xl^2 + yl^2)
   Lr <- sqrt((xr - xb)^2 + (yr - yb)^2)
   f[1:4] <- y[5:8]
   f[5] <- 1/k*((L0-Ll)*xl/Ll + lam1*xb + 2*lam2*(xl-xr))
   f[6] <- 1/k*((L0-Ll)*yl/Ll + lam1*yb + 2*lam2*(yl-yr)) -g
   f[7] <- 1/k*((L0-Lr)*(xr - xb)/Lr    - 2*lam2*(xl-xr))
   f[8] <- 1/k*((L0-Lr)*(yr - yb)/Lr    - 2*lam2*(yl-yr)) -g
   f[9] <- xb * xl + yb * yl
   f[10]<- (xl - xr)^2 + (yl - yr)^2 - L^2

   delt       <- dy - f
   delt[9:10] <- -f[9:10]

   list(delt)
 })
}
eps <- 0.01; M <- 10; k <- M * eps * eps/2
L <- 1; L0 <- 0.5; r <- 0.1; w <- 10; g <- 9.8
yini <- c(xl = 0,     yl = L0, xr = L,     yr = L0,
          ul = -L0/L, vl = 0,  ur = -L0/L, vr = 0,
          lam1 = 0, lam2 = 0)
library(rootSolve)
rootfun <- function (dyi, y, t)
  unlist(caraxis(t, y, dy = c(dyi, 0, 0),
        parms = NULL)) [1:8]
dyini <- multiroot(f = rootfun, start = rep(0,8),
                  y = yini, t = 0)$root
(dyini <- c(dyini,0,0))
```

```
 [1] -0.500000  0.000000 -0.500000  0.000000  0.000000 -9.799999  0.000000 -9.799999
 [9]  0.000000  0.000000
```

```
caraxis(t = 0, yini, dyini, NULL)
```

```
[[1]]
 [1] 2.512380e-09 0.000000e+00 2.512380e-09 0.000000e+00 0.000000e+00 8.108556e-07
 [7] 0.000000e+00 8.108556e-07 0.000000e+00 0.000000e+00
```

```
index <- c(4, 4, 2)
times <- seq(from = 0, to = 3, by = 0.01)
```

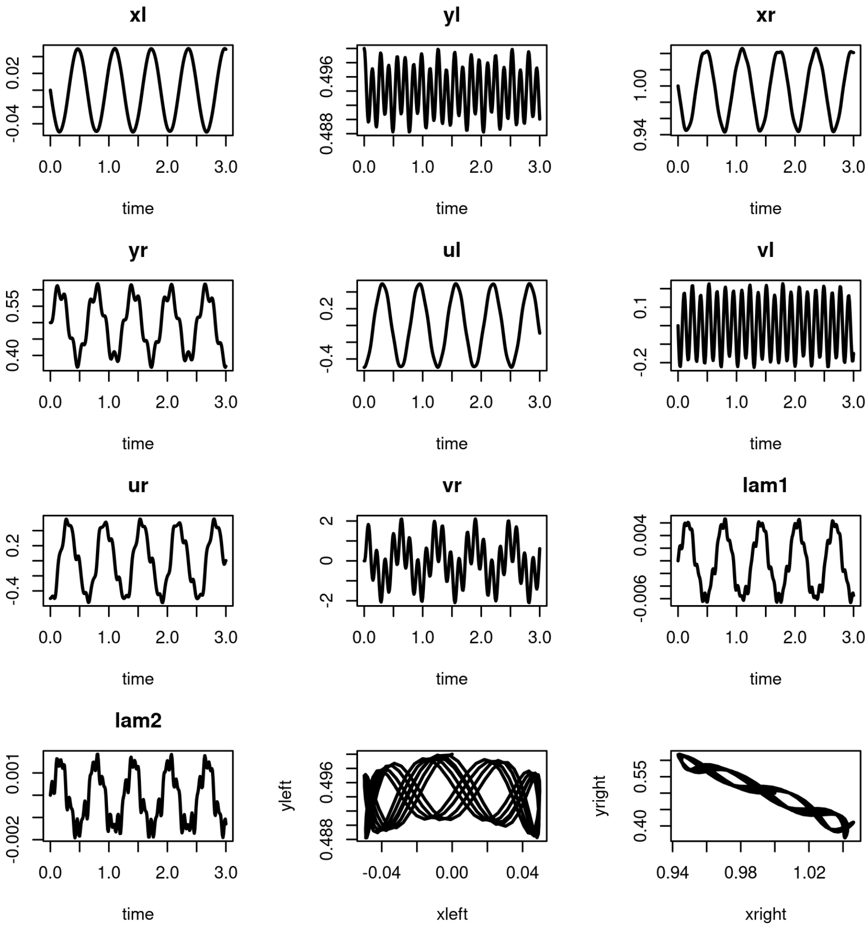

Figure 2: The car axis problem. See book for explanation.

```
out <- mebdfi(y = yini, dy = dyini, times = times,
              res = caraxis, parms = parameter, nind = index)

par(mar = c(4, 4, 3, 2))
plot(out, lwd = 2, mfrow = c(4,3))
plot(out[,c("xl", "yl")], xlab = "xleft", ylab = "yleft",
    type = "l", lwd = 2)
plot(out[,c("xr", "yr")], xlab = "xright", ylab = "yright",
    type = "l", lwd = 2)
```

5. The Transistor Amplifier

```
library(deSolve)
Transistor <- function(t, u, du, pars) {
  delt <- vector(length = 8)
   uin  <- 0.1 * sin(200 * pi * t)
   g23  <- beta * (exp( (u[2] - u[3]) / uf) - 1)
   g56  <- beta * (exp( (u[5] - u[6]) / uf) - 1)

   delt[1] <- (u[1] - uin)/R0
   delt[2] <- u[2]/R1 + (u[2]-ub)/R2 + (1-alpha) * g23
   delt[3] <- u[3]/R3 - g23
   delt[4] <- (u[4] - ub) / R4 + alpha * g23
   delt[5] <- u[5]/R5 + (u[5]-ub)/R6 + (1-alpha) * g56
   delt[6] <- u[6]/R7 - g56
   delt[7] <- (u[7] - ub) / R8 + alpha * g56
   delt[8] <- u[8]/R9
   list(delt)
}
ub <- 6; uf <- 0.026; alpha <- 0.99; beta <- 1e-6; R0 <- 1000
R1 <- R2 <- R3 <- R4 <- R5 <- R6 <- R7 <- R8 <- R9 <- 9000
C1 <- 1e-6; C2 <- 2e-6; C3 <- 3e-6; C4 <- 4e-6; C5 <- 5e-6
mass <- matrix(nrow = 8, ncol = 8, byrow = TRUE, data = c(
      -C1,C1, 0,  0,  0,  0,  0,  0,
      C1,-C1, 0,  0,  0,  0,  0,  0,
      0,  0,-C2,  0,  0,  0,  0,  0,
      0,  0,  0,-C3, C3,  0,  0,  0,
      0,  0,  0, C3,-C3,  0,  0,  0,
      0,  0,  0,  0,  0,-C4,  0,  0,
      0,  0,  0,  0,  0,  0,-C5, C5,
      0,  0,  0,  0,  0,  0, C5,-C5
))
yini <- c(0, ub/(R2/R1+1), ub/(R2/R1+1),
         ub, ub/(R6/R5+1), ub/(R6/R5+1), ub, 0)
names(yini) <- paste("u", 1:8, sep = "")
ind   <- c(8, 0, 0)
times <- seq(from = 0, to = 0.2, by = 0.001)
out <- radau(func = Transistor, y = yini, parms = NULL,
            times = times, mass = mass, nind = ind)

plot(out, lwd = 2, which = c("u1", "u5", "u8"),
    mfrow = c(1, 3))
```

Affiliation:

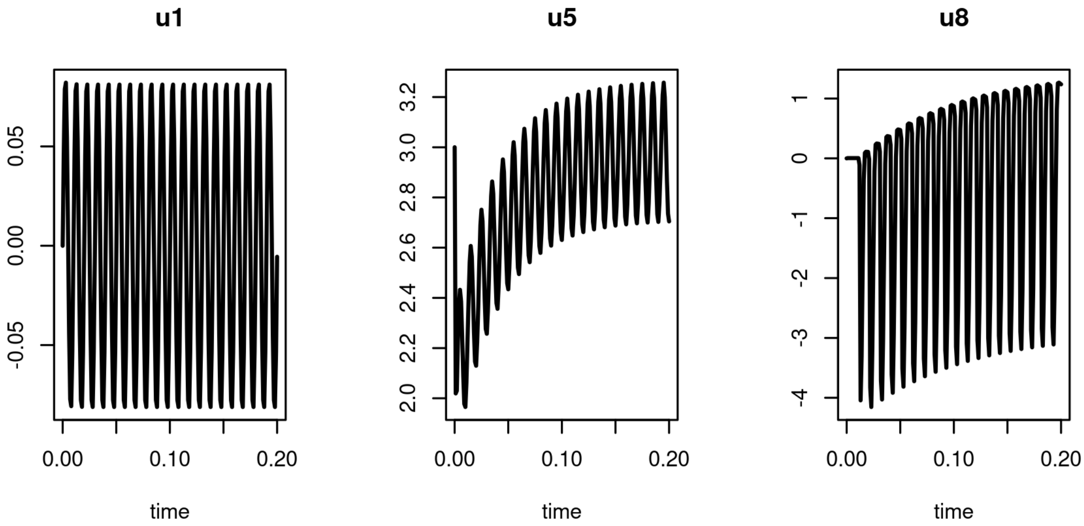

Figure 3: The transistor amplifier. See book for more information.

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