Package deTestSet: testset for initial value problems of differential equations in R

Karline Soetaert
CEME
Netherlands Institute of Ecology
The Netherlands

Jeff Cash
Department of mathematics
Imperial College London
U.K.

Francesca Mazzia
Dipartimento di Matematica
Universita' di Bari
Italy

Abstract

R package **deTestSet** contains the R-version of the ODE and DAE initial value problems test set from http://www.dm.uniba.it/~testset.

If the model problem is small enough, then it is implemented in pure R . For larger models, the problem specified in FORTRAN code at the website of Jeff Cash http://www.ma.ic.ac.uk/~jcash/IVP_software were used.

These implementations were compiled as DLLs, and included in the package. The code of these models can be found in the packages inst/examples/dynload subdirectory.

For a number of small models, we show how to implement them in R .

Keywords: ordinary differential equations, differential algebraic equations, initial value problems, testset, R.

```
> out <- andrews()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "andrews", cex = 1.5)
```

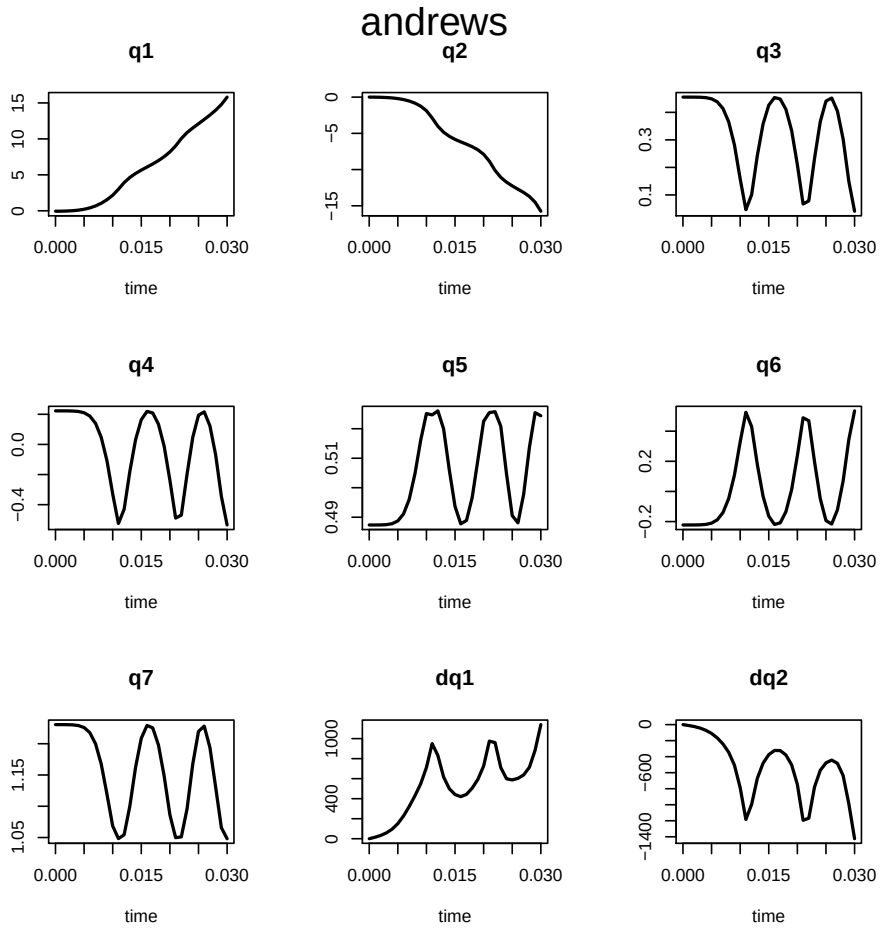

Figure 1: the andrews problem- see text for R-code

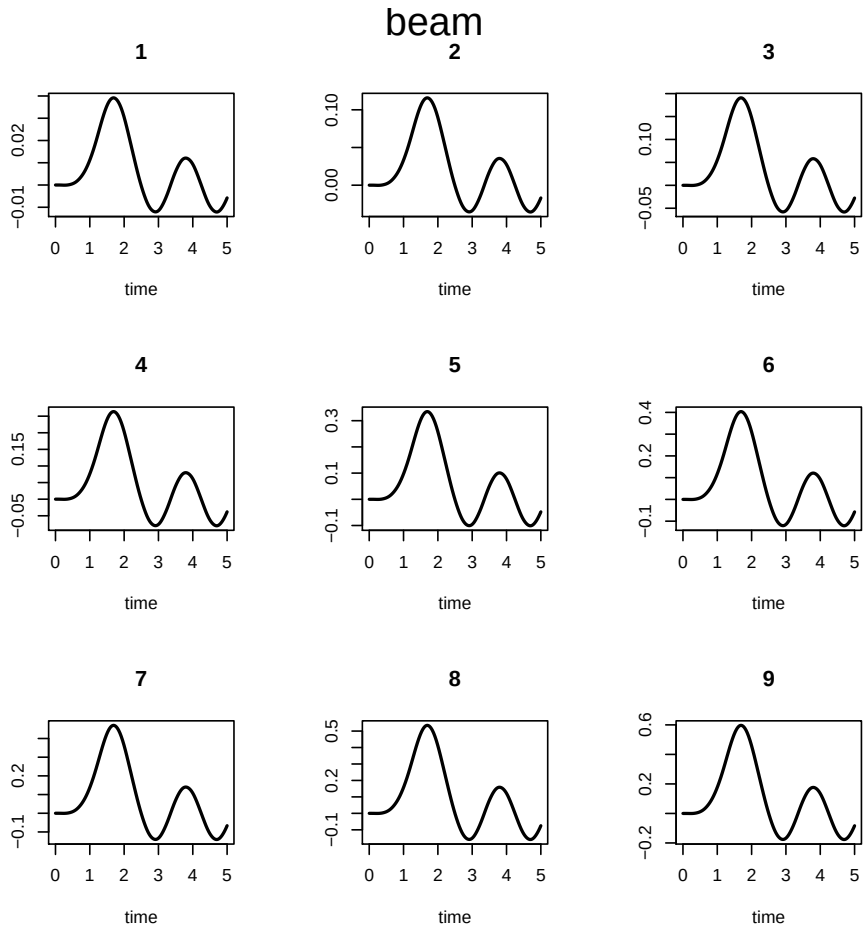

Figure 2: Solution of beam - see text for R-code

```
> out <- beam()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "beam", cex = 1.5)
```

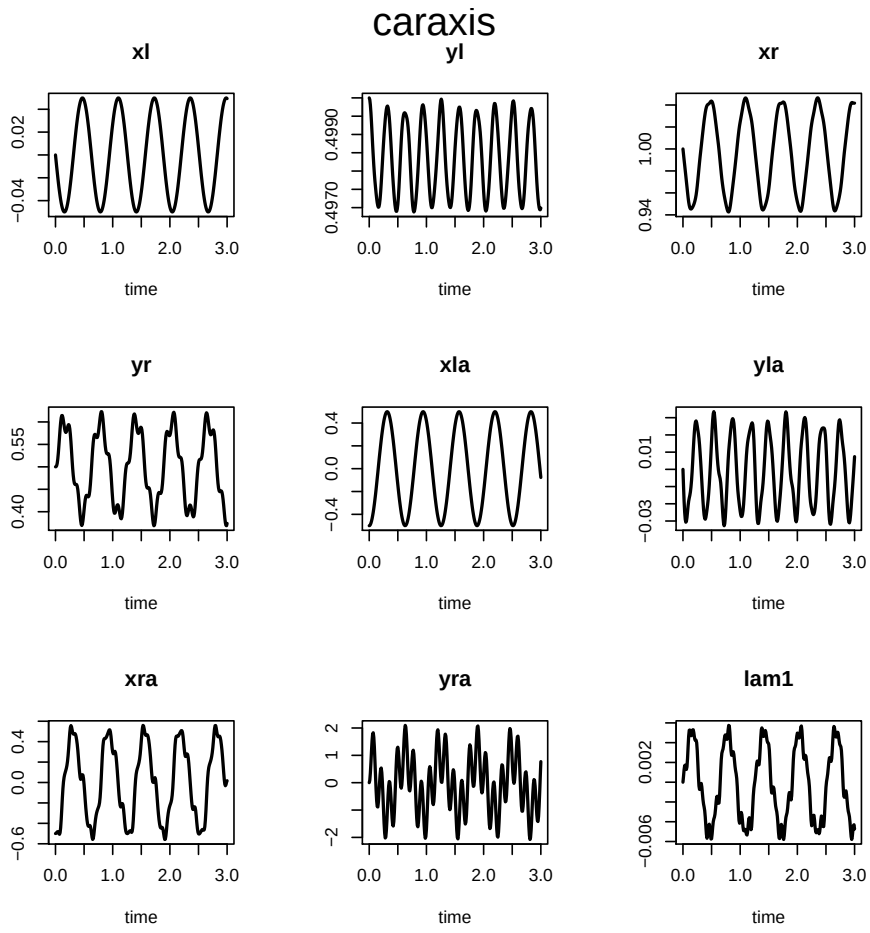

Figure 3: Solution of caraxis - see text for R-code

```
> out <- caraxis()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "caraxis", cex = 1.5)
```

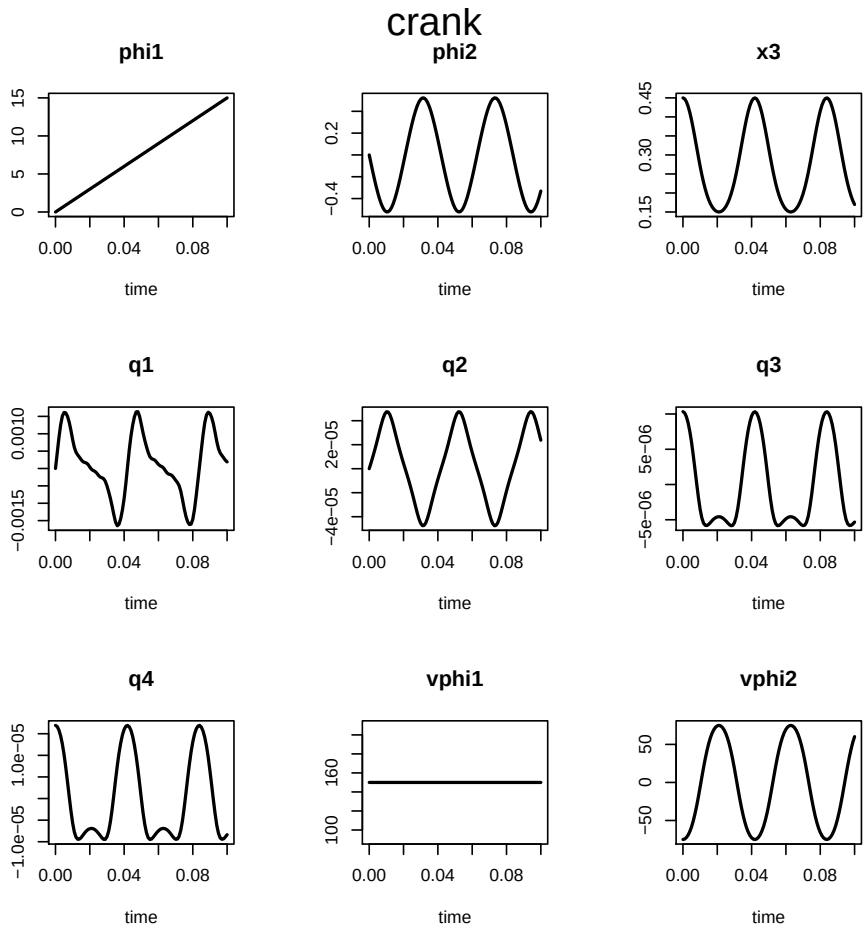

Figure 4: Solution of crank - see text for R-code

```
> out <- crank()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "crank", cex = 1.5)
```

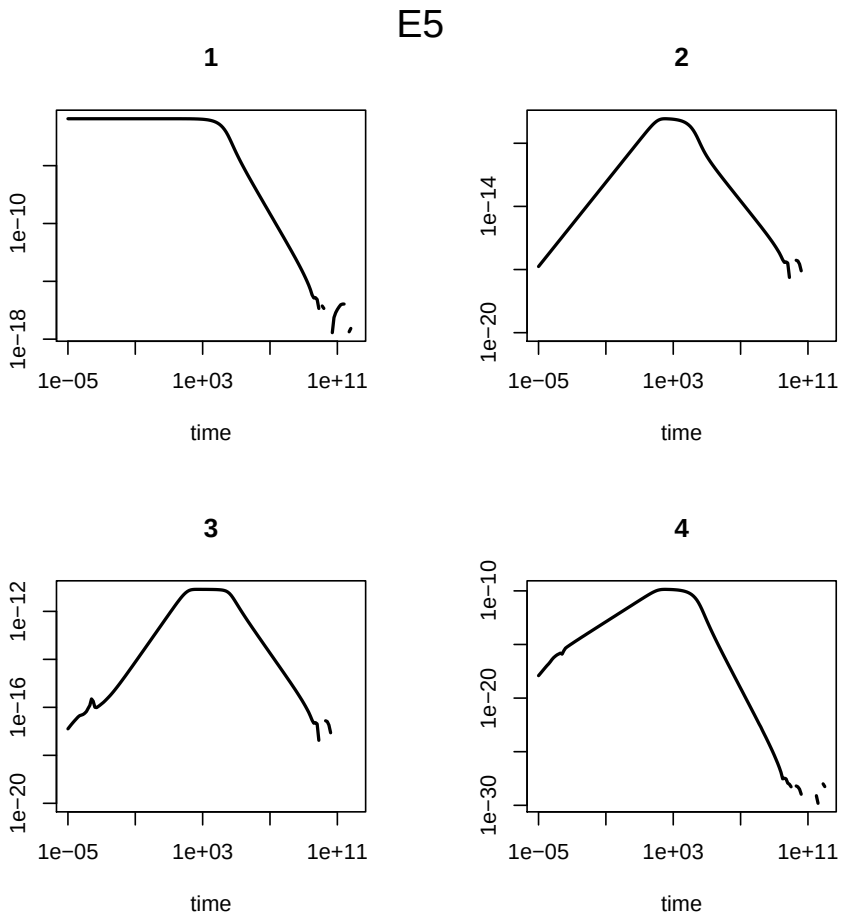

Figure 5: Solution of E5 - see text for R-code

```
> out <- E5()

> plot(out, lwd = 2, ask = FALSE, log = "xy")
> mtext(outer = TRUE, side = 3, line = -1.5, "E5", cex = 1.5)
```

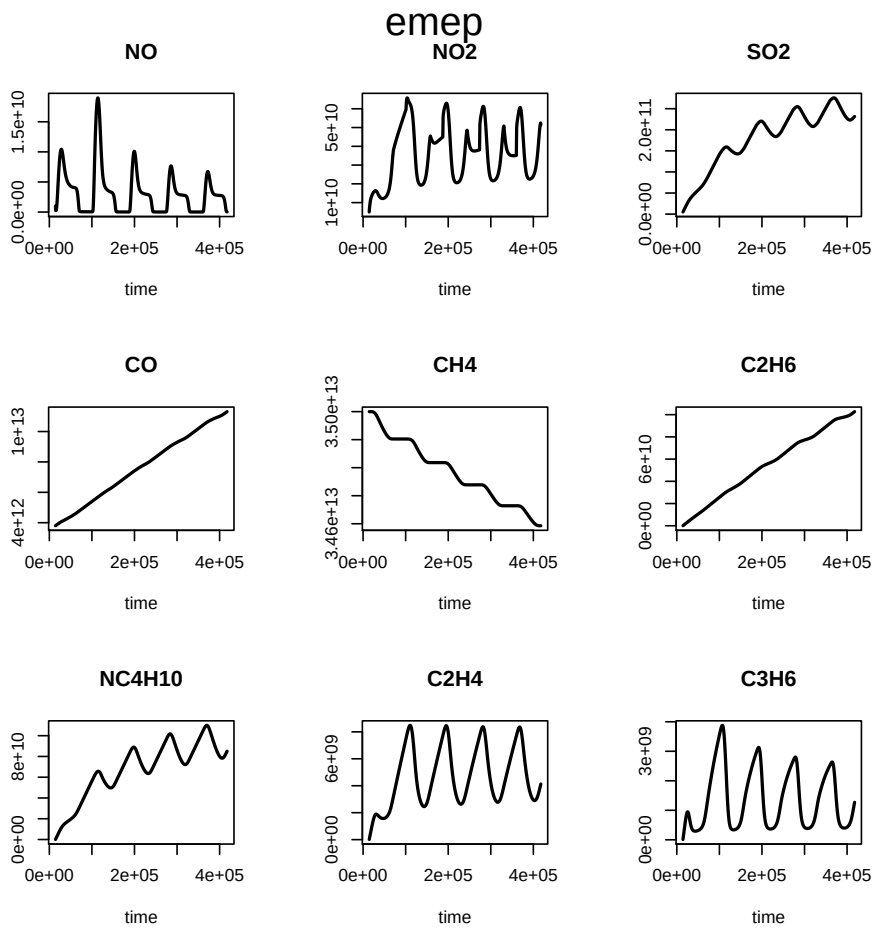

Figure 6: Solution of emep - see text for R-code

```
> out <- emep()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "emep", cex = 1.5)
```

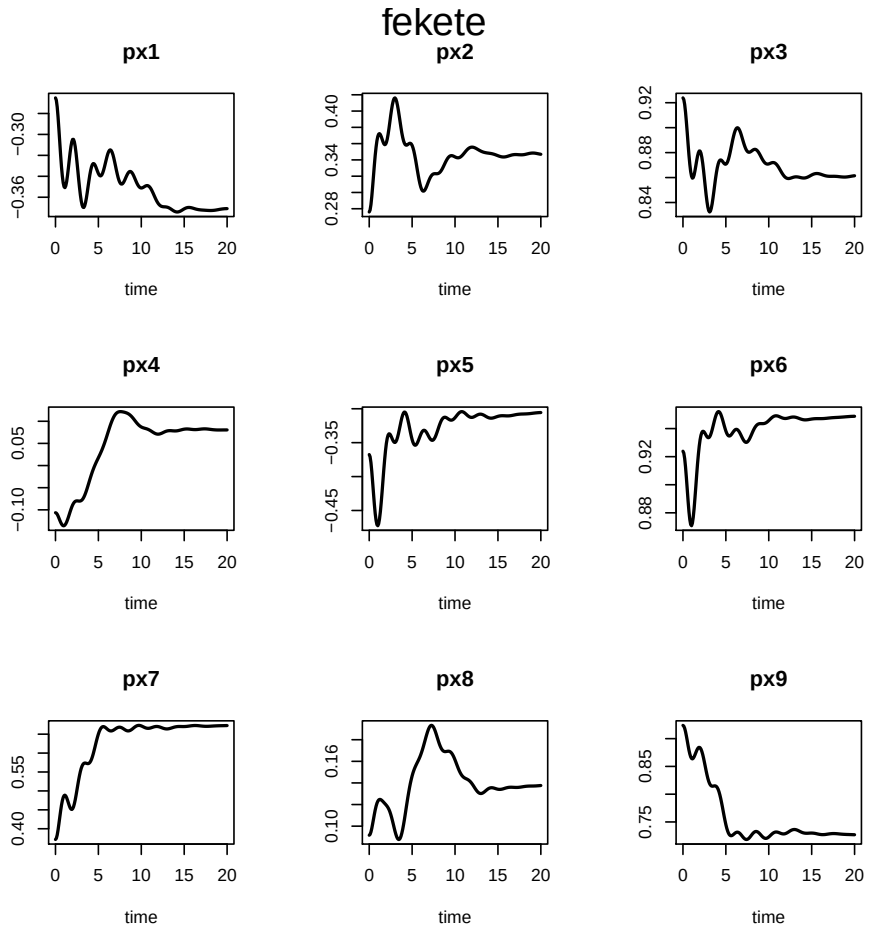

Figure 7: Solution of fekete - see text for R-code

```
> out <- fekete()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "fekete", cex = 1.5)
```

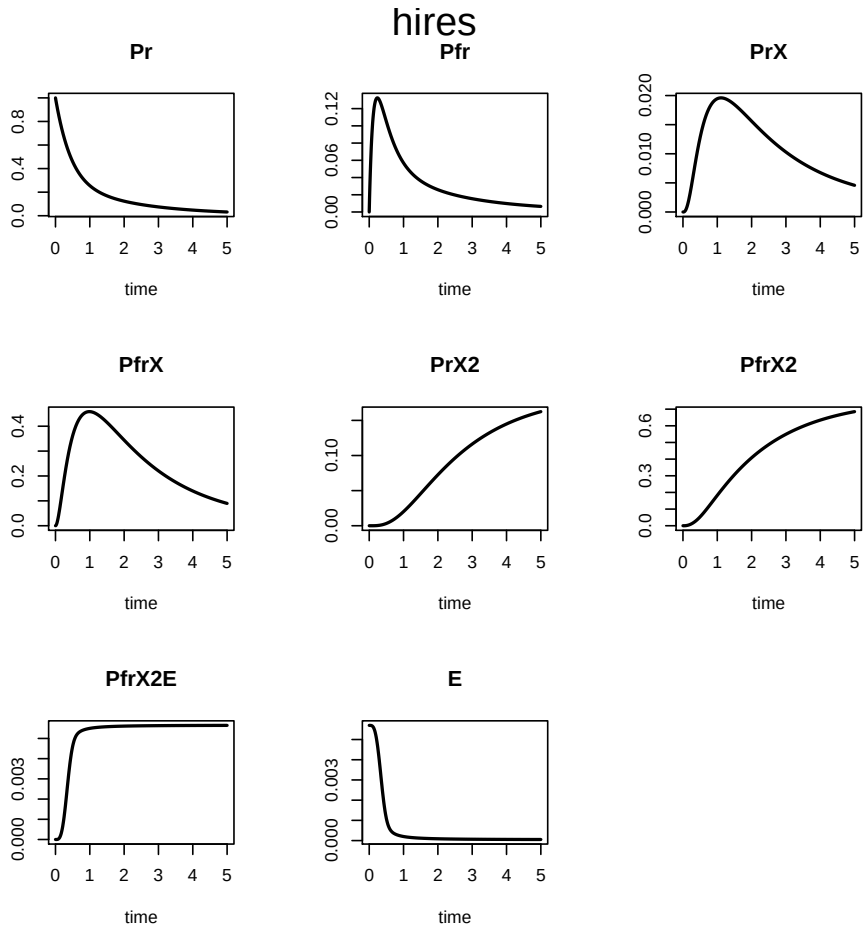

Figure 8: Solution of hires - see text for R-code

```
> out <- hires()

> plot(out, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "hires", cex = 1.5)
```

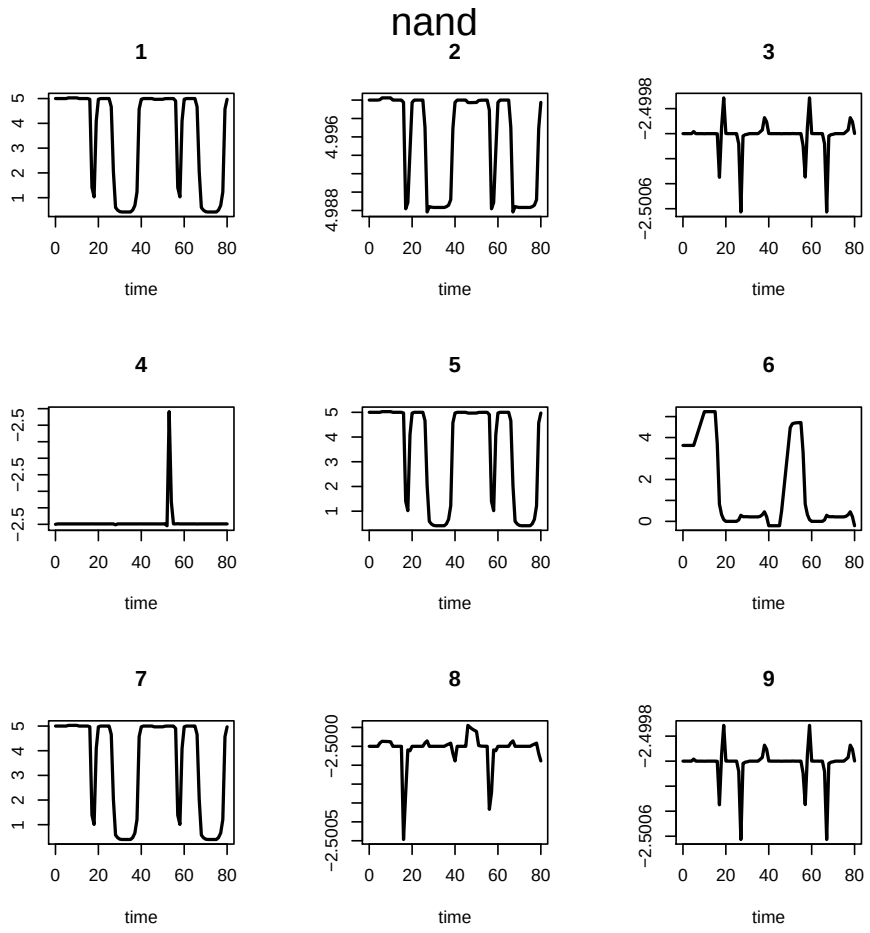

Figure 9: Solution of nand - see text for R-code

The nand problem is most efficiently solved with `daspk`

```
> out <- nand(method = daspk)

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "nand", cex = 1.5)
```

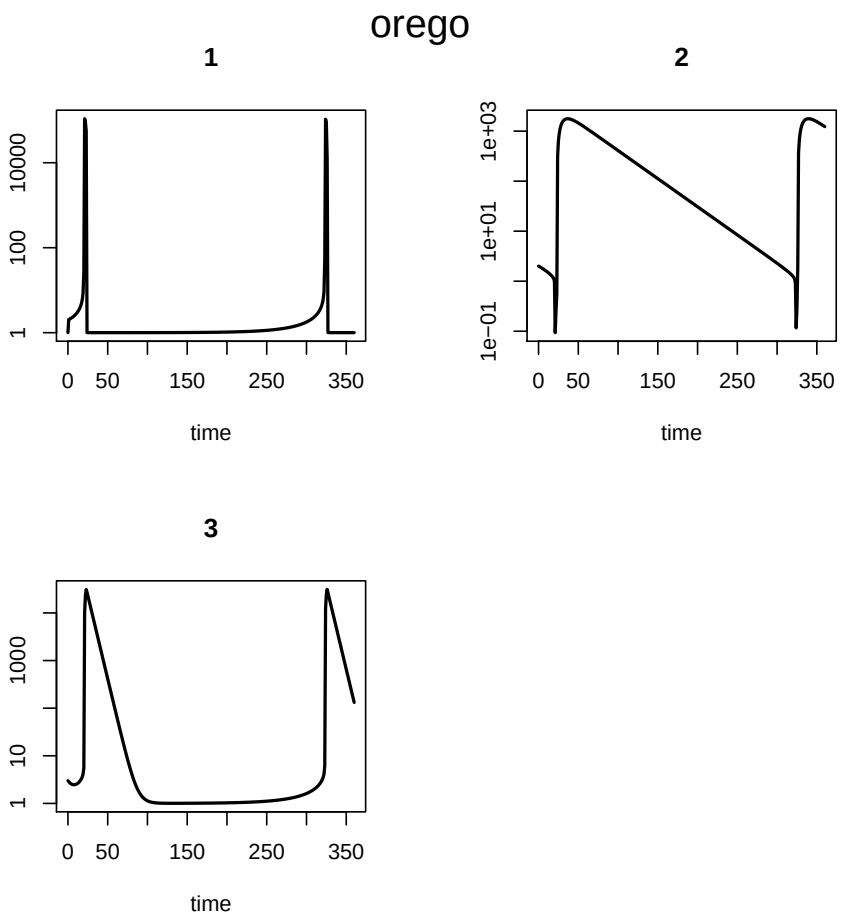

Figure 10: Solution of orego - see text for R-code

```
> out <- orego()

> plot(out, lwd = 2, ask = FALSE, log = "y")
> mtext(outer = TRUE, side = 3, line = -1.5, "orego", cex = 1.5)
```

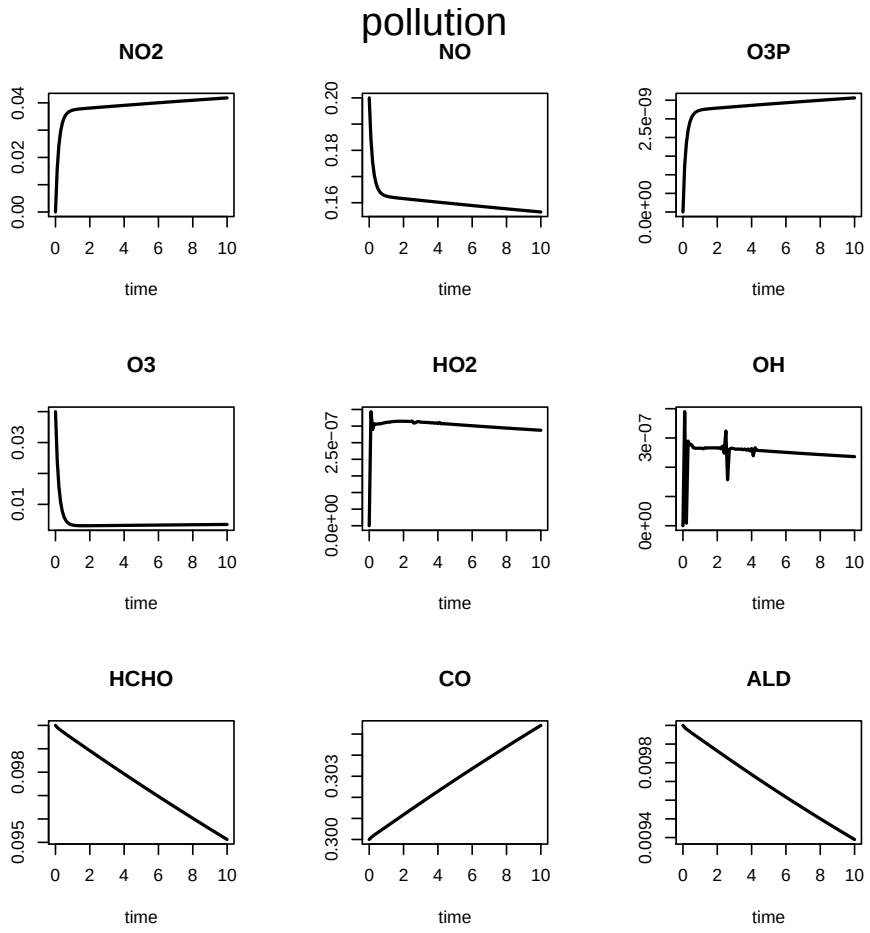

Figure 11: Solution of pollution - see text for R-code

```
> out <- pollution()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "pollution", cex = 1.5)
```

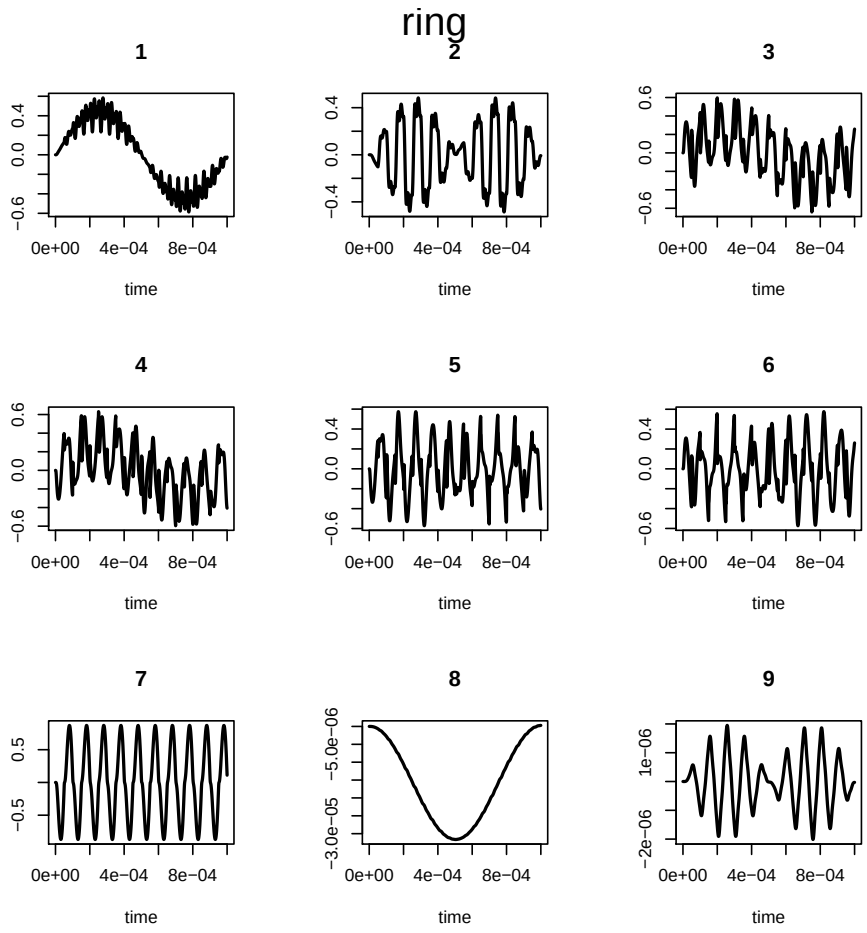

Figure 12: Solution of ring - see text for R-code

```
> out <- ring()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "ring", cex = 1.5)
```

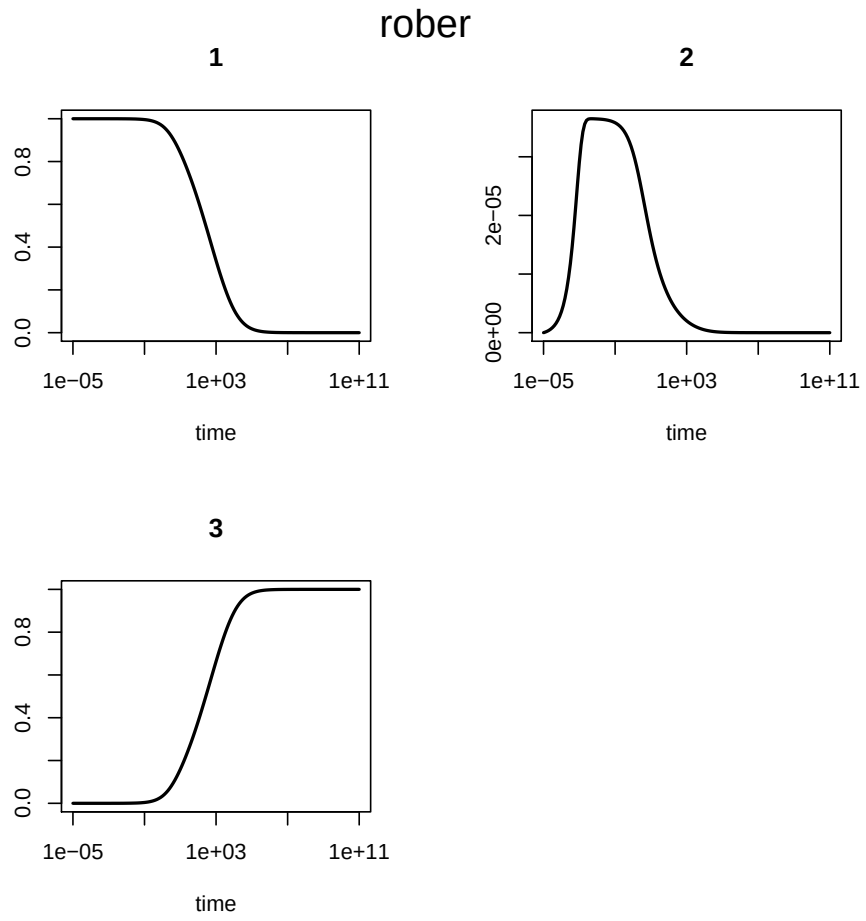

Figure 13: Solution of rober - see text for R-code

```
> out <- rober()

> plot(out, lwd = 2, ask = FALSE, log = "x")
> mtext(outer = TRUE, side = 3, line = -1.5, "rober", cex = 1.5)
```

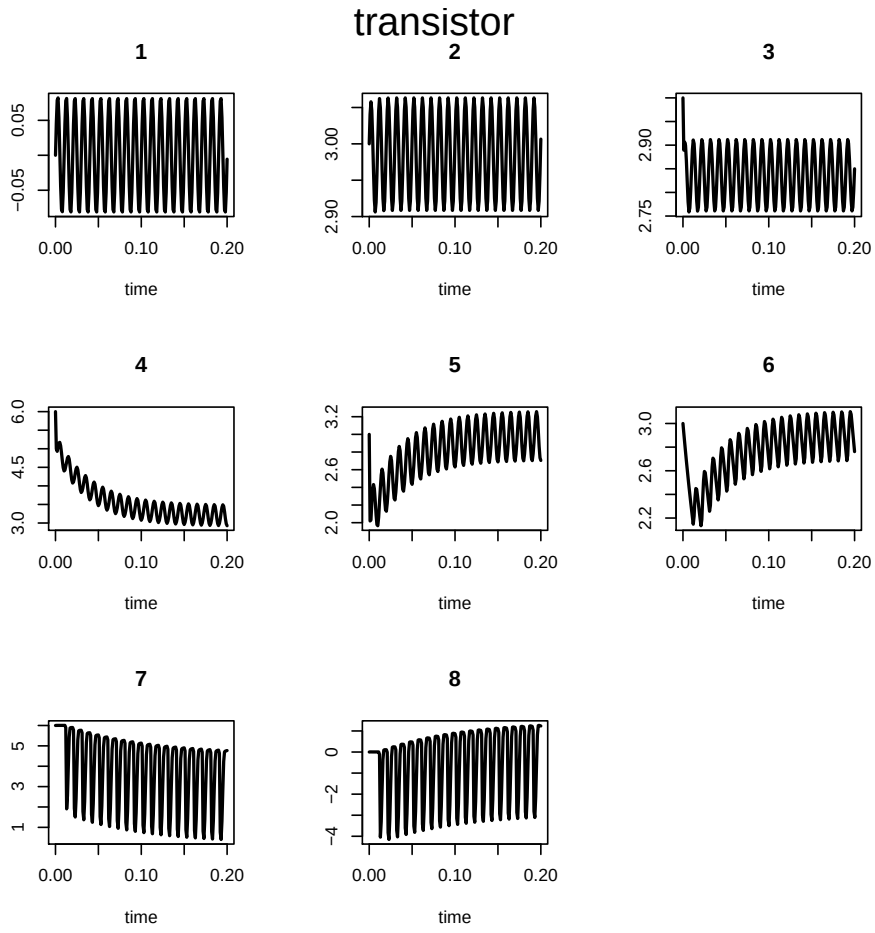

Figure 14: Solution of transistor - see text for R-code

```
> out <- transistor()

> plot(out, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "transistor", cex = 1.5)
```

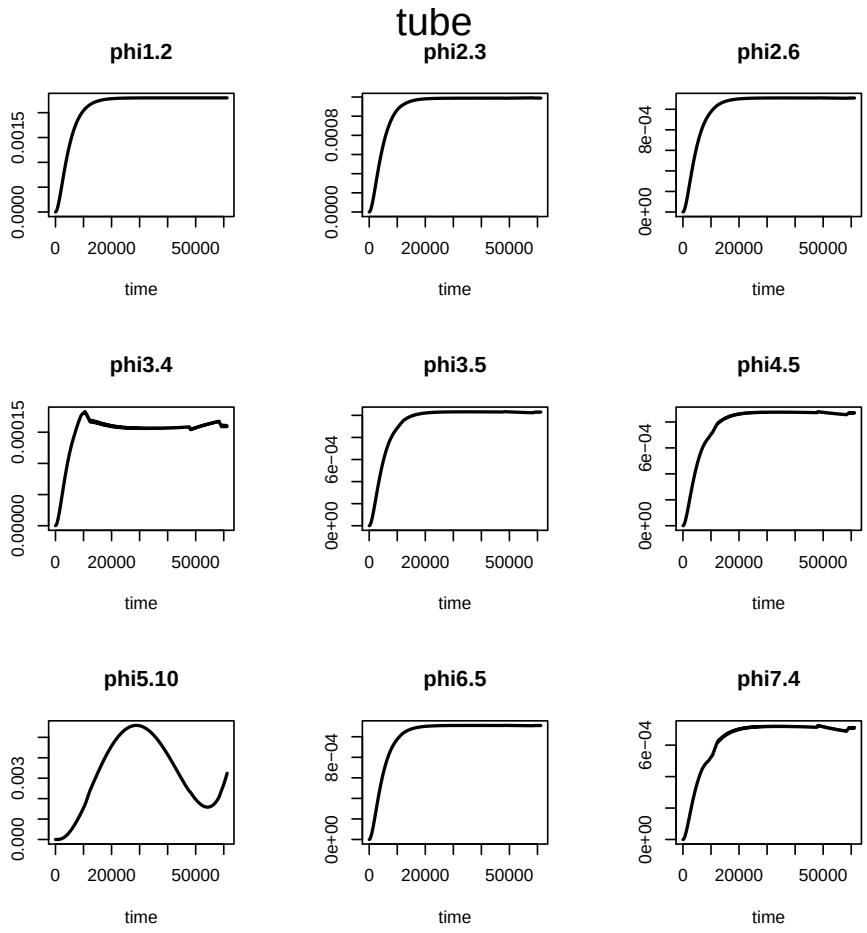

Figure 15: Solution of tube - see text for R-code

```
> out <- tube()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "tube", cex = 1.5)
```

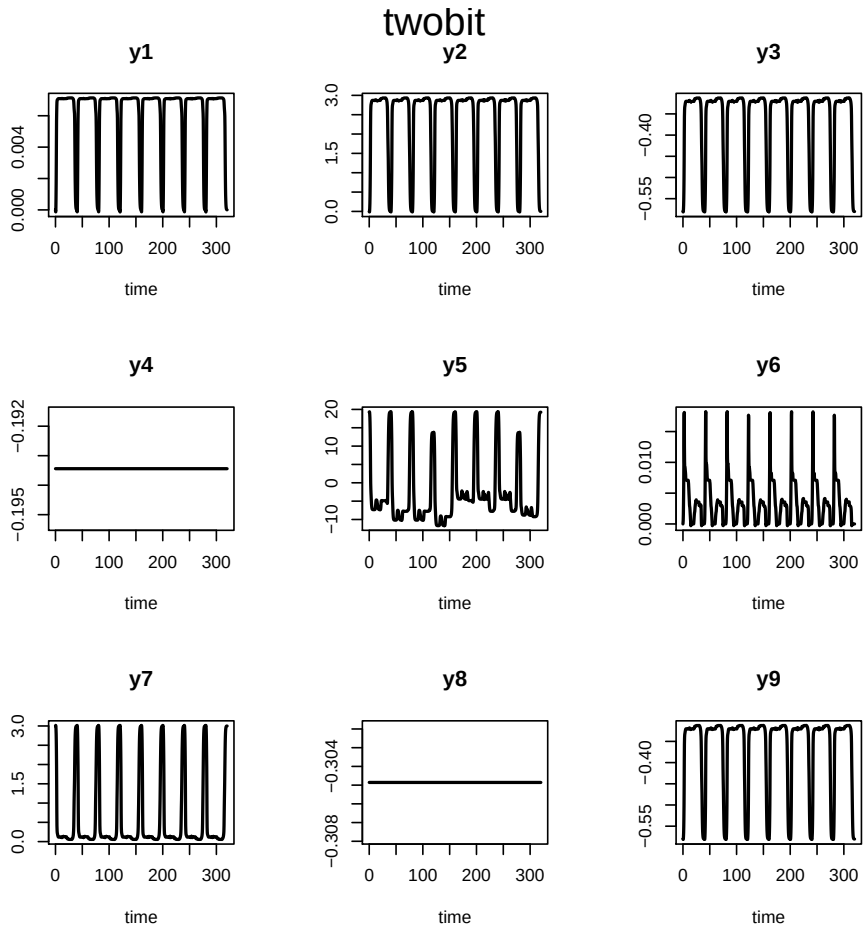

Figure 16: Solution of twobit - see text for R-code

```
> out <- twobit()

> plot(out, which = 1:9, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "twobit", cex = 1.5)
```

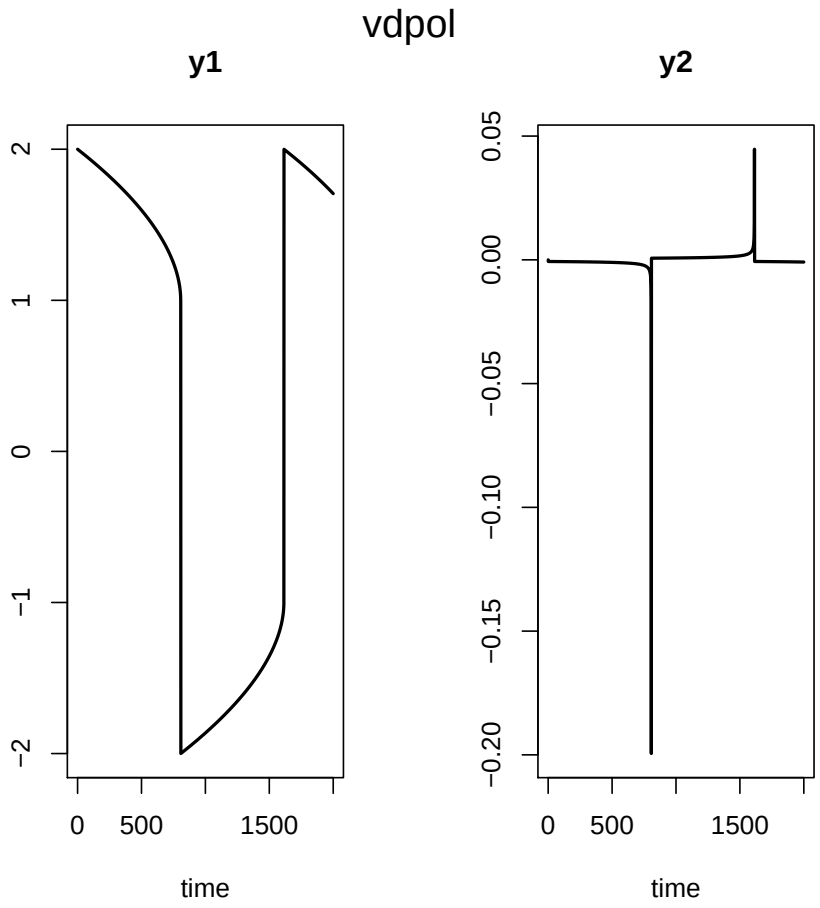

Figure 17: Solution of vdpol - see text for R-code

```
> out <- vdpol()

> plot(out, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "vdpol", cex = 1.5)
```

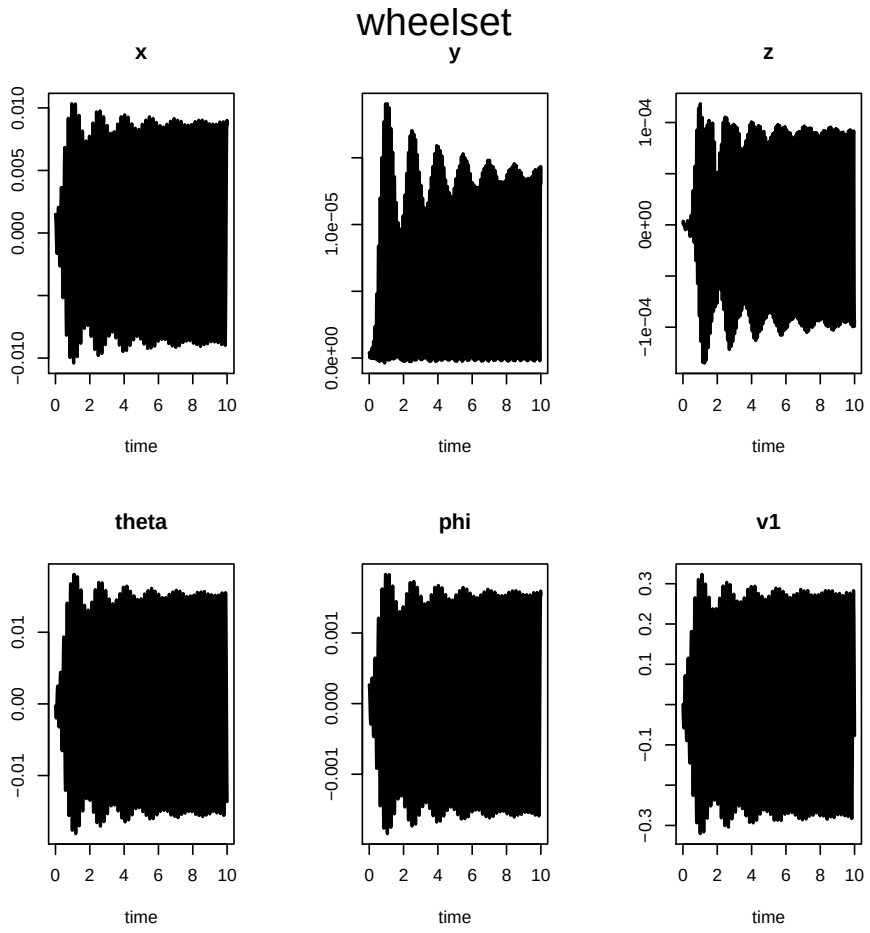

Figure 18: Solution of wheelset an implicit differential equation model - index 2, dimension 17 - see text for R-code

```
> out <- wheelset()

> plot(out, which = 1:6, lwd = 2, ask = FALSE)
> mtext(outer = TRUE, side = 3, line = -1.5, "wheelset", cex = 1.5)
```

1. References

F. Mazzia and C. Magherini. Test Set for Initial Value Problem Solvers, release 2.4. Department of Mathematics, University of Bari and INdAM, Research Unit of Bari, February 2008. Available at http://www.dm.uniba.it/~testset.

Karline Soetaert, Thomas Petzoldt and R. Woodrow Setzer (2009). deSolve: General solvers for initial value problems of ordinary differential equations (ODE), partial differential equations (PDE), differential algebraic equations (DAE), and delay differential equations (DDE). R package version 1.7.

Karline Soetaert (2009). rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations. R-package version 1.6

Affiliation:

Karline Soetaert
Centre for Estuarine and Marine Ecology (CEME)
Netherlands Institute of Ecology (NIOO)
4401 NT Yerseke, Netherlands
E-mail: k.soetaert@nioo.knaw.nl
URL: http://www.nioo.knaw.nl/users/ksoetaert

Jeff Cash
Imperial College London
South Kensington Campus
London SW7 2AZ, U.K.
E-mail: j.cash@imperial.ac.uk
URL: http://www.ma.ic.ac.uk/~jcash

Francesca Mazzia
Dipartimento di Matematica
Universita' di Bari
Via Orabona 4,
70125 BARI
Italy E-mail: mazzia@dm.uniba.it
URL: http://pitagora.dm.uniba.it/~mazzia