Package bvpSolve, solving testproblems

Karline Soetaert
Royal Netherlands
Institute of Sea Research
Yerseke, The Netherlands

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

Francesca Mazzia
Dipartimento di Matematica
Universita' di Bari
Italy

Abstract

This document implements several testproblems that can be found on http://wwwf.imperial.ac.uk/~jcash/BVP_software/readme.html, using solvers from package **bvpSolve** (Soetaert, Cash, and Mazzia 2010a).

Keywords: ordinary differential equations, boundary value problems, shooting method, mono-implicit Runge-Kutta method, R.

1. introduction

bvpSolve numerically solves boundary value problems (BVP) of ordinary differential equations (ODE), which for one (second-order) ODE can be written as:

$$
\begin{aligned}
\frac{d^2y}{dx^2} &= f(x, y, \frac{dy}{dx}) \\
a &\leq x \leq b \\
g_1(y)|_a &= 0 \\
g_2(y)|_b &= 0
\end{aligned}
$$

where `y` is the dependent, `x` the independent variable, function `f` is the differential equation, $g_1(y)|_a$ and $g_2(y)|_b$ the boundary conditions at the end points a and b.

Although all solvers now accept higher-order systems, the problem can be specified as a first-order system. For instance:

$$\frac{d^2y}{dx^2} = f(x, y, \frac{dy}{dx})$$

can be rewritten as:

$$
\begin{aligned}
\frac{dy}{dx} &= z \\
\frac{dz}{dx} &= f(x, y, z)
\end{aligned}
$$

*†Deceased 2020

In this document, all boundary value problems that can be found on http://wwwf.imperial.ac.uk/~jcash/BVP_software/PROBLEMS.PDF, are implemented and solved using solvers from package **bvpSolve** .

Most of the time, for each solver, the default settings are used, i.e. without providing "initial guesses" of the solution.

With these settings, some methods cannot solve certain problems. This does not mean that other settings cannot be found that do solve the problem.

If available, then the analytical solution of the problem is plotted (as dots).

There are several other packages that solve differential equations in the open-source software R (R Development Core Team 2010).

Package **deSolve** (Soetaert, Petzoldt, and Setzer 2010b) is designed for solving initial value problems, i.e. where the boundary conditions are provided at the initial boundary point only.

Package **ReacTran** (Soetaert and Meysman 2010) provides numerical differences of first- and second- order derivatives and, using solvers form package **rootSolve** (Soetaert 2009), can solve certain boundary value problems. This is usually more efficient (but less precise) than the boundary value solvers from **bvpSolve** , but many problems cannot be solved this way.

We will rewrite the uneven problems as a set of first-order equations, while the even problems will be solved in higher-order form.

2. Linear problems

2.1. problem 1

This problem is:

$$\xi y'' - y = 0$$
$$y_{(x=0)} = 1, y_{(x=1)} = 0$$

which is rewritten as:

$$y_1' = y2$$
$$y_2' = y_1/\xi$$

and implemented as:

```
Prob1 <- function(t, y, pars) {
    list(c( y[2] , y[1]/xi ))
 }
```

This is solved for different values of ξ

```
xi <- 0.1
print(system.time(
   mod1 <- bvpshoot(yini = c(1, NA), yend = c(0, NA), x = seq(0, 1, by=0.01),
             func = Prob1, guess = 0)))
```

```
   user  system elapsed
  0.002   0.000   0.003
```

```
print(system.time(
   mod1 <- bvptwp(yini = c(1, NA),yend=c(0, NA),x = seq(0, 1, by = 0.01),
             func = Prob1)))
```

```
   user  system elapsed
  0.002   0.000   0.002
```

```
print(system.time(
   mod1 <- bvpcol(yini = c(1, NA),yend=c(0, NA),x = seq(0, 1, by = 0.01),
             func = Prob1)))
```

```
   user  system elapsed
  0.001   0.000   0.000
```

for smaller ξ

```
xi <-0.01
mod2  <- bvptwp(yini = c(1, NA), yend = c(0, NA), x = seq(0, 1, by = 0.01),
             func = Prob1)
```

and for a very small value

```
xi <-0.001
mod3  <- bvptwp(yini = c(1, NA), yend = c(0, NA), x = seq(0, 1, by = 0.01),
            func = Prob1)
```

and the output plotted

```
plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 1")
# exact solution
curve(exp(-x/sqrt(xi))-exp((x-2)/sqrt(xi))/(1-exp(-2/sqrt(xi))),
       0, 1, add = TRUE, type = "p")
```

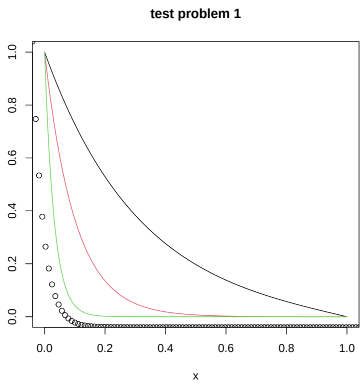

Figure 1: Solution of the BVP ODE problem 1, see text for R-code

2.2. problem 2

$$\xi y'' - y' = 0$$
$$y_{(x=0)} = 1, y_{(x=1)} = 0$$

It is solved in higher-order form.

```
Prob2 <- function(t, y, pars) {
   list( y[2]/xi )
 }
xi <-0.2
mod1 <- bvpshoot(yini = c(1, NA), yend = c(0, NA), x = seq(0, 1, by = 0.01),
                   order = 2, func = Prob2, guess = 0)
```

For lower values of ξ (<0.1) this problem cannot be solved by the shooting method, but it is solvable by mono-implicit Runge-Kutta and collocation

```
xi <-0.1
mod2 <- bvptwp(yini = c(1, NA), yend = c(0, NA), x = seq(0, 1, by = 0.01),
                 order = 2, func = Prob2, atol = 1e-10)
xi <- 0.01
mod3 <- bvptwp(yini = c(1, NA), yend = c(0, NA), x = seq(0, 1, by = 0.01),
                 order = 2, func = Prob2, atol = 1e-10)
xi <- 0.001
mod4 <- bvpcol(yini = c(1, NA), yend = c(0, NA), x = seq(0, 1, by = 0.01),
                 order = 2, func = Prob2, atol = 1e-10)
```

The solution can be compared with the analytical solution:

```
plot(mod1, mod2, mod3, mod4, which = 1, lty = 1, main = "test problem 2")
xi <- 0.01
curve((1-exp((x-1)/xi))/(1-exp(-1/xi)), 0, 1, type = "p", add = TRUE)
```

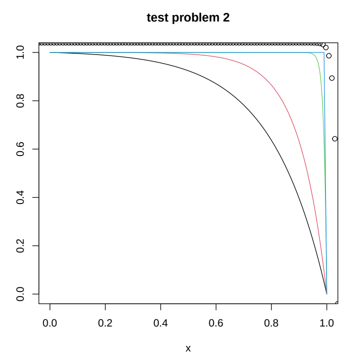

Figure 2: Solution of the BVP ODE problem 2, see text for R-code

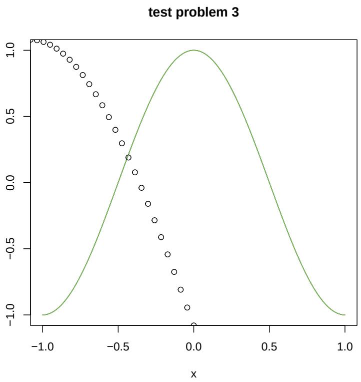

Figure 3: Solution of the BVP ODE problem 3, see text for R-code

2.3. problem 3

$$\xi y'' + (2+\cos(\pi x))y' - y = -(1+\xi\pi^2)\cos(\pi x) - (2+\cos(\pi x))\pi\sin(\pi x)$$
$$y_{(x=-1)} = y_{(x=1)} = -1$$

```
Prob3 <- function(x, y, pars) {
   list(c( y[2],
          1/xi * (-(2+cos(pi*x)) * y[2] + y[1]-
           (1 + xi*pi*pi) * cos(pi*x)-
           (2 + cos(pi*x))* pi * sin(pi*x))
       ))
 }

xi <-0.1
mod1 <- bvpshoot(yini = c(-1, NA), yend = c(-1, NA),
                    x = seq(-1, 1, by=0.01), func = Prob3, guess = 0)

xi <-0.01
mod2 <- bvptwp(yini = c(-1, NA), yend = c(-1, NA),
                  x = seq(-1, 1, by=0.01), func = Prob3)
xi <-0.001
mod3 <- bvpcol(yini = c(-1, NA), yend = c(-1, NA),
                  x = seq(-1, 1, by=0.01), func = Prob3)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 3")
curve(cos(pi*x), -1, 1, type = "p", add = TRUE)
```

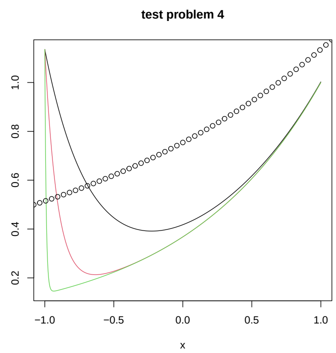

Figure 4: Solution of the BVP ODE problem 4, see text for R-code

2.4. problem 4

$$\xi y'' + y' - (1+\xi)y = 0$$
$$y_{(x=-1)} = 1 + \exp(-2)$$
$$y_{(x=1)} = 1 + \exp(-2(1+\xi)/\xi)$$

It is solved in higher-order form.

```
Prob4 <- function(t, y, pars) {
   list((-y[2] + (1+xi)*y[1])/xi )
 }
yini <- c(1 + exp(-2), NA)

xi   <- 0.5
yend <- c(1 + exp(-2*(1+xi)/xi), NA)
mod1  <- bvpshoot(yini = yini, yend = yend, x = seq(-1, 1, by = 0.01),
                    order = 2, func = Prob4, guess = 0)
xi   <- 0.1
yend <- c(1 + exp(-2*(1+xi)/xi), NA)
mod2  <- bvptwp(yini = yini, yend = yend, x = seq(-1, 1, by = 0.01),
                  order = 2, func = Prob4)
xi <- 0.01
yend <- c(1 + exp(-2*(1+xi)/xi), NA)
mod3  <- bvptwp(yini = yini,yend = yend,x = seq(-1, 1, by = 0.01),
                   order = 2, func = Prob4)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 4")
curve(exp(x-1) + exp(-(1+xi)*(1+x)/xi), -1, 1, type = "p", add = TRUE)
```

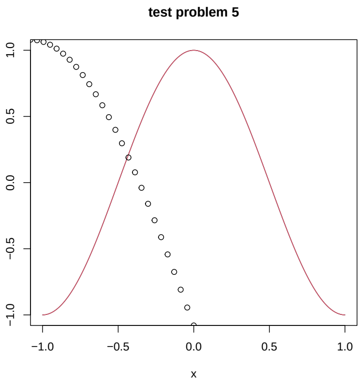

Figure 5: Solution of the BVP ODE problem 5, see text for R-code

2.5. problem 5

$$\xi y'' - xy' - y = -(1 + \xi\pi^2)\cos(\pi x) + (\pi x))\sin(\pi x)$$
$$y_{(x=-1)} = y_{(x=1)} = -1$$

```
Prob5 <- function(x, y, pars) {
   list(c( y[2],
           x * y[2] + y[1] - (1+pi*pi) * cos(pi*x) + pi*x*sin(pi*x) ))
 }

xi <- 0.1
mod1 <- bvpshoot(yini = c(-1, NA), yend = c(-1, NA),
                   x=seq(-1, 1, by = 0.01), func = Prob5, guess = 0)

mod2 <- bvptwp(yini = c(-1, NA), yend = c(-1, NA),
                 x=seq(-1, 1, by = 0.01), func = Prob5)

plot(mod1, mod2, which = 1, lty = 1, main = "test problem 5")
curve(cos(pi*x), -1, 1, type = "p", add = TRUE)
```

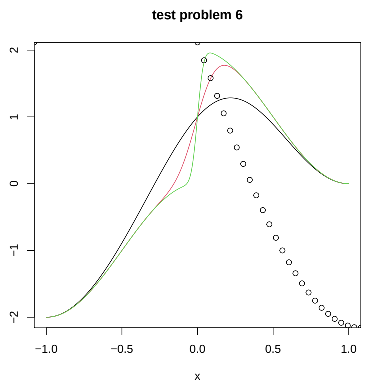

Figure 6: Solution of the BVP ODE problem 6, see text for R-code

2.6. problem 6

$$\xi y'' + xy' = -\xi\pi^2 \cos(\pi x) - \pi x \sin(\pi x)$$
$$y_{(x=-1)} = -2$$
$$y_{(x=1)} = 0$$

It is solved in higher-order form.

This problem cannot be solved by the shooting method, except for the largest value of xi

```
Prob6 <- function(t, y, pars) {
   list(1/xi * (-t*y[2] - xi*pi*pi*cos(pi*t) - pi*t*sin(pi*t)) )
 }

xi    <- 0.1
mod1 <- bvpshoot(yini = c(-2, NA), yend = c(0, NA),
             order = 2, x = seq(-1, 1, by = 0.01), func = Prob6, guess = 0)
xi    <- 0.01
mod2 <- bvptwp(yini = c(-2, NA), yend = c(0, NA),
             order = 2, x = seq(-1, 1, by = 0.01), func = Prob6)
xi    <- 0.001
mod3 <- bvptwp(yini = c(-2, NA), yend = c(0, NA),
             order = 2, x = seq(-1, 1, by = 0.01), func = Prob6)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 6")
erf <- function(x) 2 * pnorm(x * sqrt(2)) - 1
curve(cos(pi*x) + erf(x/sqrt(2*xi))/erf(1/sqrt(2*xi)), -1, 1,
       type = "p", add = TRUE)
```

2.7. problem 7

$$
\begin{aligned}
\xi y'' + xy' - y &= -(1+\xi\pi^2)\cos(\pi x) - \pi x \sin(\pi x) \\
y(-1) &= -1 \\
y(1) &= 1
\end{aligned}
$$

This problem cannot be solved with the shooting method for small xi.

```
prob7 <- function(x, y, pars) {
   list(c(y[2],
      1/xi * (-x*y[2]+y[1] - (1+xi*pi*pi)*cos(pi*x)-pi*x*sin(pi*x)))   )
 }
x  <- seq(-1, 1, by = 0.01)

xi   <- 0.01
mod1  <- bvptwp(yini = c(-1, NA), yend = c(1, NA), x = x, func = prob7)
xi   <- 0.001
mod2 <- bvptwp(yini = c(-1, NA), yend = c(1, NA), x = x, func = prob7)
```

For even smaller ξ, we need to provide good initial guesses:

```
xi <- 0.0005
mod3  <- bvptwp(yini = c(-1, NA), yend = c(1, NA), x = x, func = prob7,
   xguess = mod2[,1], yguess = t(mod2[,-1]))

plot(mod1, mod2, mod3, which = c(2,1), type = "l", lty = 1,
      main = c("dy", "y"), xlab = "x", ylab = "y")
erf <- function(x) 2 * pnorm(x * sqrt(2)) - 1
curve(cos(pi*x) + x + (x*erf(x/sqrt(2*xi))+sqrt(2*xi/pi)*exp(-x^2/2/xi))/
          (erf(1/(2*xi))+sqrt(2*xi/pi)*exp(-1/2/xi)),
          -1, 1, type = "p", add = TRUE)
```

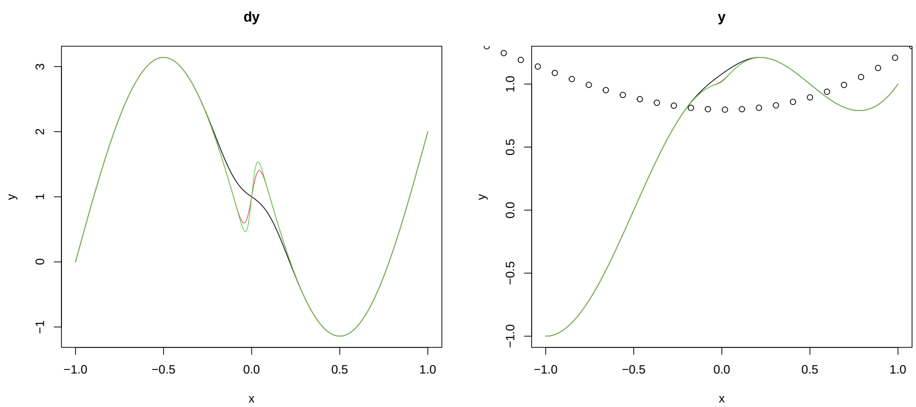

Figure 7: Solution of the BVP ODE problem 7, y and y' versus x- see text for R-code

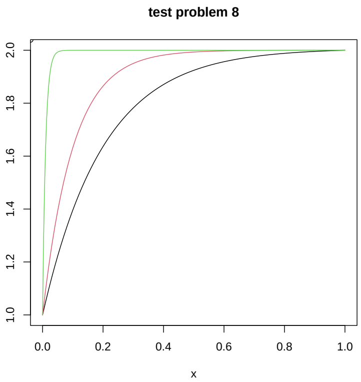

Figure 8: Solution of the BVP ODE problem 8, see text for R-code

2.8. problem 8

$$
\begin{aligned}
\xi y'' + y' &= 0 \\
y(0) &= 1 \\
y(1) &= 2
\end{aligned}
$$

It is solved in higher-order form.

```
prob8 <- function(x, y, pars) {
   list(-1/xi*y[2])
 }
x  <- seq(0,1,by=0.01)

xi    <- 0.2
mod1 <- bvpshoot(yini = c(1, NA), yend = c(2, NA), x = x,
                   order = 2, func = prob8, guess = 0)
xi   <- 0.1
mod2 <- bvptwp(yini = c(1, NA), yend = c(2, NA), x = x,
                   order = 2, func = prob8)
xi   <- 0.01
mod3 <- bvptwp(yini = c(1, NA), yend = c(2, NA), x = x,
                   order = 2, func = prob8)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 8")
curve(2-exp(-1/xi)-exp(-x/xi)/(1-exp(-1/xi)),
       0, 1, add = TRUE, type = "p")
```

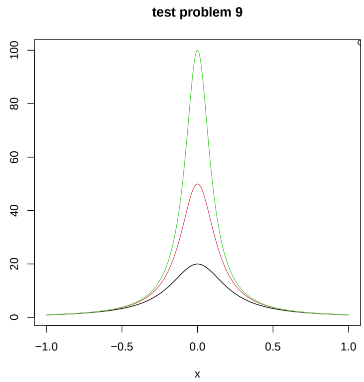

Figure 9: Solution of the BVP ODE problem 9, see text for R-code

2.9. problem 9

$$(\xi + x^2)y'' + 4xy' + 2y = 0$$
$$y_{(x=-1)} = y_{(x=1)} = 1/(1+\xi)$$

This problem cannot be solved by the shooting method

```
Prob9 <- function(x, y, pars) {
   list(c( y[2], -1/(xi+x^2)*(4*x*y[2]+2*y[1]) ))
 }

xi   <-0.05
mod1 <- bvptwp(yini = c(1/(1+xi), NA), yend = c(1/(1+xi), NA),
                x = seq(-1, 1, by = 0.01), func = Prob9)

xi   <-0.02
mod2 <- bvptwp(yini = c(1/(1+xi), NA), yend = c(1/(1+xi), NA),
                x = seq(-1, 1, by = 0.01), func = Prob9)

xi   <-0.01
mod3 <- bvptwp(yini = c(1/(1+xi), NA), yend = c(1/(1+xi), NA),
                x = seq(-1, 1, by = 0.01), func = Prob9)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 9")
# exact
curve(1/(xi+x^2), -1, 1, type = "p", add = TRUE)
```

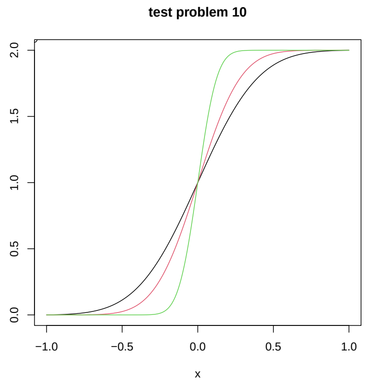

Figure 10: Solution of the BVP ODE problem 10, see text for R-code

2.10. problem 10

$$\xi y'' + xy' = 0$$
$$y_{(x=-1)} = 0$$
$$y_{(x=1)} = 2$$

It is solved in higher-order form.

```
Prob10 <- function(x, y, pars) {
   list( -1/xi*x*y[2] )
 }

xi    <-0.1
mod1 <- bvpshoot(yini = c(0, NA), yend = c(2, NA),
            order = 2, x = seq(-1, 1, by = 0.01), func=Prob10, guess = 0)
xi   <- 0.05
mod2 <- bvpcol(yini = c(0, NA), yend = c(2, NA),
            order = 2, x = seq(-1, 1, by = 0.01), func=Prob10)
xi   <- 0.01
mod3 <- bvptwp(yini = c(0, NA), yend = c(2, NA),
            order = 2, x = seq(-1, 1, by = 0.01), func=Prob10)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 10")
erf <- function(x) 2 * pnorm(x * sqrt(2)) - 1
curve(1+erf(x/sqrt(2*xi))/erf(1/sqrt(2*xi)),
       -1, 1, type = "p", add = TRUE)
```

2.11. problem 11

$$
\begin{aligned}
\xi y'' - y &= -(\xi\pi^2 + 1)\cos(\pi x) \\
y_{(x=-1)} &= -1 \\
y_{(x=1)} &= -1
\end{aligned}
$$

All xi give the same result

```
Prob11 <- function(x, y, pars) {
   list(c(y[2], 1/xi * (y[1]-(xi*pi*pi+1)*cos(pi*x)) ))
 }

xi <-0.1
print(system.time(
 mod1 <- bvpshoot(yini = c(-1, NA), yend = c(-1, NA), guess = 0,
                   x = seq(-1, 1, by=0.01), func = Prob11, atol = 1e-10)
 ))

  user  system elapsed
 0.004   0.000   0.004

print(system.time(
 mod2 <- bvptwp(yini = c(-1, NA), yend = c(-1, NA),
               x = seq(-1, 1, by=0.01), func = Prob11, atol = 1e-10)
 ))

  user  system elapsed
 0.006   0.000   0.005

plot(mod1, mod2, which = 1, lty = 1, main = "test problem 11")
curve(cos(pi*x), -1, 1, type = "p", add= TRUE)
```

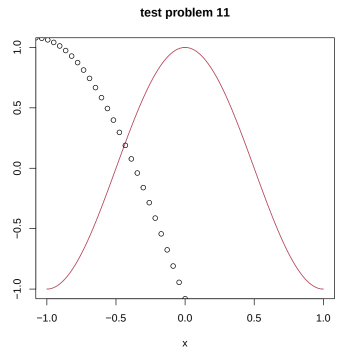

Figure 11: Solution of the BVP ODE problem 11, see text for R-code

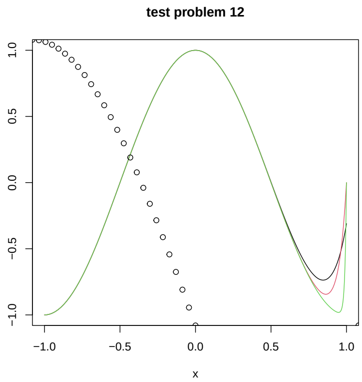

Figure 12: Solution of the BVP ODE problem 12, see text for R-code

2.12. problem 12

The same as problem 11, but with different boundary values:

$$\xi y'' = y = -(\xi \pi^2 + 1)\cos(\pi x)$$
$$y_{(x=-1)} = -1$$
$$y_{(x=1)} = 0$$

It is solved in higher-order form.

```
Prob12 <- function(x, y, pars) {
   list(1/xi * (y[1]-(xi*pi*pi+1)*cos(pi*x)))
 }

xi    <- 0.01
mod1 <- bvpshoot(yini = c(-1, NA), yend = c(0, NA),
            order = 2, x = seq(-1, 1, by = 0.01), func = Prob12, guess = 0)

xi   <- 0.0025
mod2 <- bvptwp(yini = c(-1, NA), yend = c(0, NA),
            order = 2, x = seq(-1, 1, by = 0.01), func = Prob12)

xi   <- 0.0001
mod3 <- bvptwp(yini = c(-1, NA), yend = c(0, NA),
            order = 2, x = seq(-1, 1, by = 0.01), func = Prob12)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 12")
curve(cos(pi*x)+exp((x-1)/sqrt(xi)),
       -1, 1, type = "p", add = TRUE)
```

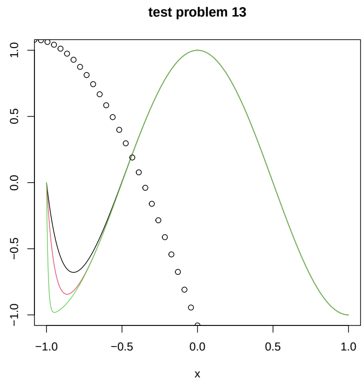

Figure 13: Solution of the BVP ODE problem 13, see text for R-code

2.13. problem 13

The same as problem 11, but with different boundary values:

$$\xi y'' = y = -(\xi\pi^2 + 1)\cos(\pi x)$$
$$y_{(x=-1)} = 0$$
$$y_{(x=1)} = -1$$

```
Prob13 <- function(x, y, pars)  {
   list(c( y[2], 1/xi*(y[1]-(xi*pi*pi+1)*cos(pi*x)) ))
 }

xi     <- 0.01
mod1 <- bvpshoot(yini = c(0, NA), yend = c(-1, NA),
                   x = seq(-1, 1, by=0.01), func = Prob13, guess = 0)

xi    <- 0.0025
mod2 <- bvptwp(yini = c(0, NA), yend = c(-1, NA),
                x = seq(-1, 1, by=0.01), func = Prob13)

xi     <- 0.0001
mod3 <- bvptwp(yini = c(0, NA), yend = c(-1, NA),
                 x = seq(-1, 1, by=0.01), func = Prob13)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 13")
curve(cos(pi*x)+exp(-(x+1)/sqrt(xi)),
       -1, 1, type = "p", add = TRUE)
```

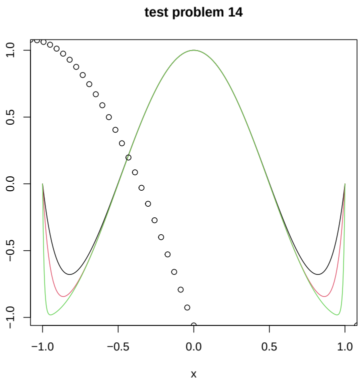

Figure 14: Solution of the BVP ODE problem 14, see text for R-code

2.14. problem 14

The same as problem 11, but with different boundary values:

$$\xi y'' = y = -(\xi\pi^2 + 1)\cos(\pi x)$$
$$y_{(x=-1)} = 0$$
$$y_{(x=1)} = 0$$

It is solved in higher-order form.

```
Prob14 <- function(x, y, pars)  {
   list(1/xi*(y[1]-(xi*pi*pi+1)*cos(pi*x)))
 }

xi    <- 0.01
mod1 <- bvpshoot(yini = c(0, NA), yend = c(0, NA),
          order = 2, x = seq(-1, 1, by = 0.01), func = Prob14, guess = 0)

xi   <- 0.0025
mod2 <- bvptwp(yini = c(0, NA), yend = c(0, NA),
             order = 2, x = seq(-1, 1, by = 0.01), func = Prob14)

xi    <- 0.0001
mod3 <- bvptwp(yini = c(0, NA), yend = c(0, NA),
             order = 2, x = seq(-1, 1, by = 0.01), func = Prob14)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 14")
curve(cos(pi*x)+exp((x-1)/sqrt(xi))+exp(-(x+1)/sqrt(xi)),
       -1, 1, type = "p", add = TRUE)
```

2.15. problem 15

$$\xi y'' - xy = 0$$
$$y_{(x=-1)} = y_{(x=1)} = 1$$

```
Prob15 <- function(x, y, pars)  {
   list(c( y[2], 1/xi*x*y[1] ))
 }

xi <-0.003
print(system.time(
 mod1 <- bvpshoot(yini = c(1, NA), yend = c(1, NA),
                   x = seq(-1, 1, by = 0.01), func = Prob15, guess = 0)
 ))

  user  system elapsed
 0.011   0.000   0.011

xi <- 0.005
print(system.time(
 mod2 <- bvptwp(yini = c(1, NA), yend = c(1, NA),
               x = seq(-1, 1, by = 0.01), func = Prob15)
 ))

  user  system elapsed
 0.005   0.001   0.005

xi <- 0.005
print(system.time(
 mod3 <- bvpcol(yini = c(1, NA), yend = c(1, NA),
               x = seq(-1, 1, by = 0.01), func = Prob15)
 ))

  user  system elapsed
 0.009   0.000   0.009

xi <- 0.01
print(system.time(
 mod4 <- bvptwp(yini = c(1, NA), yend = c(1, NA),
            x = seq(-1, 1, by = 0.01), func = Prob15)
 ))

  user  system elapsed
 0.005   0.000   0.005

plot(mod1, mod2, mod3, mod4, which = 1, lty = 1, main = "test problem 15")
```

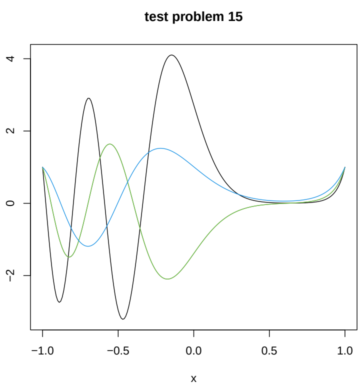

Figure 15: Solution of the BVP ODE problem 15, see text for R-code

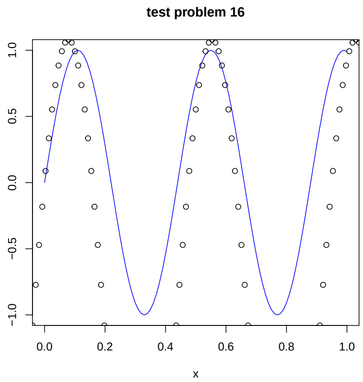

Figure 16: Solution of the BVP ODE problem 16, see text for R-code

2.16. problem 16

$$\xi^2 y'' + \pi^2 y/4 = 0$$
$$y_{(x=0)} = 0$$
$$y_{(x=1)} = \sin(\pi/(2\xi))$$

It is solved in higher-order form.

```
Prob16 <- function(x, y, pars) {
   list(-1/xi^2*pi^2*y[1]/4 )
 }

xi <-0.11
print(system.time(
 mod1 <- bvpshoot(yini = c(0,NA),yend = c(sin(pi/2/xi), NA),
                   x = seq(0, 1, by=0.01), func = Prob16, guess = 0,
                   order = 2, atol = 1e-10)
 ))
```

```
  user  system elapsed
 0.016   0.000   0.017
```

```
plot(mod1, which = 1, main = "test problem 16", col = "blue")
curve(sin(pi*x/2/xi), 0, 1, type = "p", add = TRUE)
```

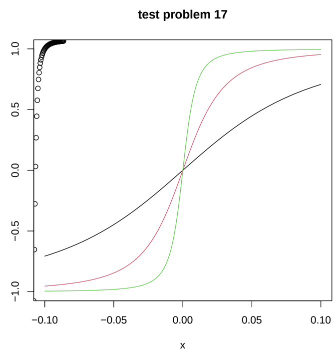

Figure 17: Solution of the BVP ODE problem 17, see text for R-code

2.17. problem 17

$$y'' = -3\xi y/(\xi + x^2)^2$$
$$y_{(x=0.1)} = -y(-0.1) = \frac{0.1}{\sqrt{(}\xi + 0.01)}$$

only bvptwp works.

```
Prob17 <- function(x, y, pars)  {
   list(c( y[2], -3*xi*y[1]/(xi+x^2)^2 ))
 }
xseq <- seq(-0.1, 0.1, by = 0.001)

xi   <- 0.01
mod1 <- bvptwp(yini = c(-0.1/sqrt(xi+0.01), NA),
                 yend = c(0.1/sqrt(xi+0.01), NA), x = xseq,
                 func = Prob17, atol = 1e-10)
xi   <- 0.001
mod2 <- bvptwp(yini = c(-0.1/sqrt(xi+0.01), NA),
                 yend = c(0.1/sqrt(xi+0.01), NA), x = xseq,
                 func = Prob17, atol = 1e-8)
xi   <- 0.0001
mod3 <- bvptwp(yini = c(-0.1/sqrt(xi+0.01), NA),
                 yend = c(0.1/sqrt(xi+0.01), NA), x = xseq,
                 func = Prob17, atol = 1e-8)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 17")
curve(x/sqrt(xi+x^2), -0.1, 0.1, type = "p", add = TRUE)
```

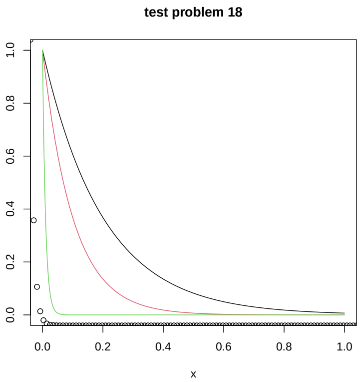

Figure 18: Solution of the BVP ODE problem 18, see text for R-code

2.18. problem 18

$$\xi y'' = -y'$$
$$y_{(x=0)} = 1$$
$$y_{(x=1)} = \exp(-1/\xi)$$

It is solved in higher-order form.

```
Prob18 <- function(x, y, pars) {
   list( -1/xi*y[2])
 }
xseq<-seq(0,1,by=0.01)

xi <-0.2
mod1 <- bvpshoot(yini = c(1, NA), yend = c(exp(-1/xi), NA), x = xseq,
               order = 2, func = Prob18, guess = 0, atol = 1e-10)

xi <- 0.1
mod2 <- bvptwp(yini = c(1, NA), yend = c(exp(-1/xi), NA), x = xseq,
               order = 2, func = Prob18, atol = 1e-10)

xi <- 0.01
mod3 <- bvptwp(yini = c(1, NA), yend = c(exp(-1/xi), NA), x = xseq,
               order = 2, func = Prob18, atol = 1e-10)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 18")
curve(exp(-x/xi), 0, 1, type = "p", add = TRUE)
```

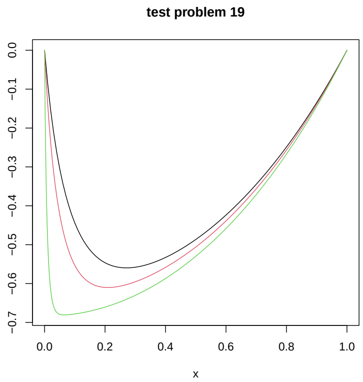

Figure 19: Solution of the BVP ODE problem 19, see text for R-code

3. nonlinear problems

For the nonlinear problems, the analytical solution is often not known.

3.1. problem 19

$$\xi y'' + \exp(y)y' - \frac{\pi}{2}\sin(\pi x/2)\exp(2y) = 0$$
$$y_{(x=0)} = y_{(x=1)} = 0$$

```
Prob19 <- function(t, y, pars, ksi) {
   pit = pi*t
   list(c(y[2],(pi/2*sin(pit/2)*exp(2*y[1])-exp(y[1])*y[2])/ksi))
 }

xi    <- 0.05
mod1 <- bvpshoot(yini = c(0, NA), yend = c(0, NA),
          x = seq(0, 1, by = 0.01), func = Prob19, guess = 0, ksi = xi)
xi <- 0.03
mod2 <- bvptwp(yini = c(0, NA), yend = c(0, NA),
                x = seq(0, 1, by = 0.01), func = Prob19, ksi = xi,
                atol = 1e-15)
xi <- 0.005
mod3 <- bvptwp(yini = c(0, NA), yend = c(0, NA),
                 x = seq(0, 1, by = 0.01), func = Prob19, ksi = xi,
                 atol = 1e-10)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 19")
```

3.2. problem 20

$$\begin{aligned} \xi y'' + y'^2 &= 1 \\ y_{x=0} &= 1 + \xi \ln(\cosh(0.745/\xi)) \\ y_{x=1} &= 1 + \xi \ln(\cosh(0.255/\xi)) \end{aligned}$$

It is solved in higher-order form.

```
Prob20 <- function(x, y, pars) {
   list( 1/xi *(1-y[2]^2) )
 }

xi  <- 0.5
ini <- c(1+xi * log(cosh(0.745/xi)), NA)
end <- c(1+xi * log(cosh(0.255/xi)), NA)
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by=0.01),
             order = 2, func = Prob20)
xi  <- 0.3
ini <- c(1+xi * log(cosh(0.745/xi)), NA)
end <- c(1+xi * log(cosh(0.255/xi)), NA)
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by=0.01),
             order = 2, func = Prob20)
xi  <- 0.01
ini <- c(1+xi * log(cosh(0.745/xi)), NA)
end <- c(1+xi * log(cosh(0.255/xi)), NA)
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by=0.01),
                 order = 2, func = Prob20)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 20")
curve(1+xi * log(cosh((x-0.745)/xi)), 0, 1, add = TRUE, type = "p")
```

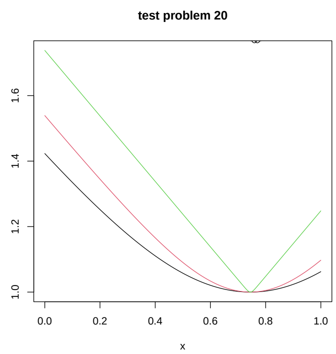

Figure 20: Solution of the BVP ODE problem 20, see text for R-code

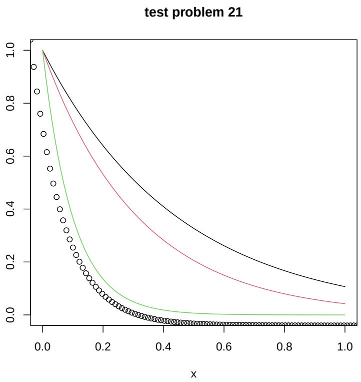

Figure 21: Solution of the BVP ODE problem 21, see text for R-code

3.3. problem 21

$$\xi y'' = y + y^2 - \exp(-2x/\sqrt(\xi))$$
$$y_{x=0} = 1$$
$$y_{x=1} = \exp(-1/\sqrt(\xi))$$

```
Prob21 <- function(x, y, pars, xi) {
   list(c( y[2], 1/xi *(y[1]+y[1]^2-exp(-2*x/sqrt(xi))) ))
 }
ini <- c(1, NA)

xi  <- 0.2
end <- c(exp(-1/sqrt(xi)), NA)
mod1 <- bvpshoot(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                    func = Prob21, guess = 0, xi = xi)
xi  <- 0.1
end <- c(exp(-1/sqrt(xi)), NA)
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob21, xi = xi)
xi  <- 0.01
end <- c(exp(-1/sqrt(xi)), NA)
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                 func = Prob21, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 21")
curve(exp(-x/sqrt(xi)), 0, 1, add = TRUE, type = "p")
```

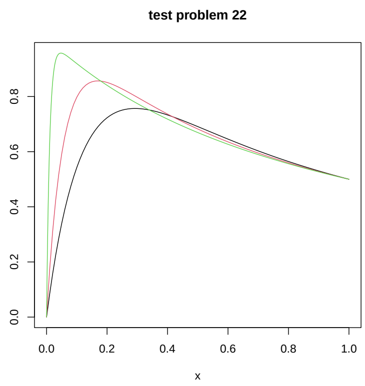

Figure 22: Solution of the BVP ODE problem 22, see text for R-code

3.4. problem 22

$$\xi y'' + y' + y^2 = 0$$
$$y_{x=0} = 0$$
$$y_{x=1} = 1/2$$

It is solved in higher-order form.

```
Prob22 <- function(t, y, pars, xi) {
   list( -1/xi *(y[2]+y[1]^2) )
 }
ini <- c(0, NA)
end <- c(1/2, NA)

xi <-0.1
mod1 <- bvpshoot(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob22, guess = 0, xi = xi)

xi <-0.05
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob22, xi = xi)

xi <- 0.01
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob22, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 22")
```

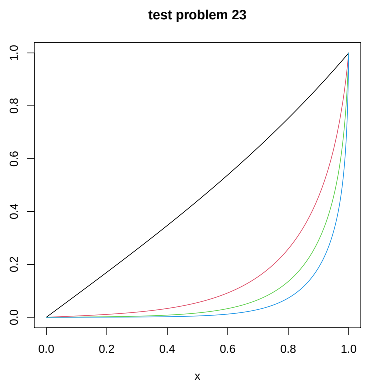

Figure 23: Solution of the BVP ODE problem 23, see text for R-code

3.5. problem 23

This is a difficult problem that cannot be solved with bvpshoot

$$y'' = \xi \sinh(\xi y)$$
$$y(x = 0) = 0, \; y(x = 1) = 1$$

```
Prob23 <- function(t, y, pars, xi) {
   list(c( y[2], xi*sinh(xi*y[1])) )
 }
ini <- c(0, NA)
end <- c(1, NA)

xi <- 1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                 func = Prob23, xi = xi)

xi <- 5
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                 func = Prob23, xi = xi)

xi <- 7
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob23, xi = xi)

xi <- 9
mod4 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                 func = Prob23, xi = xi)

plot(mod1, mod2, mod3, mod4, which = 1, lty = 1, main = "test problem 23")
```

3.6. problem 24

This is a particularly difficult problem to solve

$$\xi A(x)yy'' - (\frac{1+1.4}{2} - \xi A'(x))yy' + \frac{y'}{y} + \frac{A'(x)}{A(x)}(1 - \frac{1.4-1}{2}y^2) = 0$$
$$A(x) = 1 + x^2$$
$$y_{(x=0)} = 0.9129$$
$$y_{(x=1)} = 0.375$$

It is solved in higher-order form.

```
Prob24 <- function(t, y, pars, xi) {
   A <- 1+t*t
   AA <- 2*t
   ga <- 1.4
   list((((1+ga)/2 -xi*AA)*y[1]*y[2]-y[2]/y[1]-
        (AA/A)*(1-(ga-1)*y[1]^2/2))/(xi*A*y[1])  )
 }
ini <- c(0.9129, NA)
end <- c(0.375, NA)

xi   <- 0.05
mod1 <- bvpshoot(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
             order = 2, func = Prob24, guess = 0.9, xi = xi)

xi <- 0.02
mod2 <- bvpshoot(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
             order = 2,  func = Prob24, guess = 0.9, xi = xi)
attributes(mod2)$roots         # has FAILED: f.root too large!
```

```
       root     f.root iter
1 0.8359306 -0.4355725   23
```

Function `bvpshoot` cannot solve this problem for small ξ

Function `bvptwp` can solve it for small ξ if initiated with good initial guesses:

```
xi <- 0.02
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                 order = 2, func = Prob24, xi = xi,
                 xguess = mod2[,1], yguess = t(mod2[,2:3]))
xi <- 0.01
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                 order = 2, func = Prob24, xi = xi,
                 xguess = mod2[,1], yguess = t(mod2[,2:3]))

plot(mod1, mod2, mod3,  which = 1, lty = 1, main = "test problem 24")
```

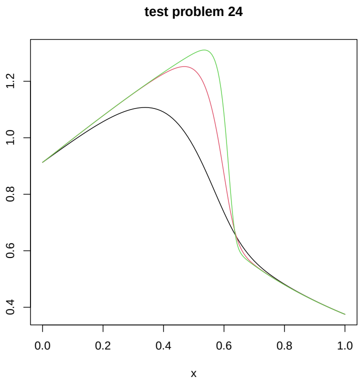

Figure 24: Solution of the BVP ODE problem 24, see text for R-code

3.7. problem 25

Now come a series of similar problems (problem 25-30), that differ only by their boundary conditions:
The differential equation is:

$$\xi y'' + yy' - y = 0$$

For problem 25, the boundary conditions are:

$$y_{x=0} = -1/3$$
$$y_{x=1} = 1/3$$

These problems are most easily solved with `bvptwp`

```
Prob25 <- function(t, y, pars, xi) {
   list(c( y[2], -1/xi *(y[1]*y[2]-y[1]) ))
 }
ini <- c(-1/3 ,NA)
end <- c(1/3, NA)

xi   <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob25, xi = xi)
xi   <- 0.01
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob25, xi = xi)
xi   <- 0.001
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob25, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 25")
```

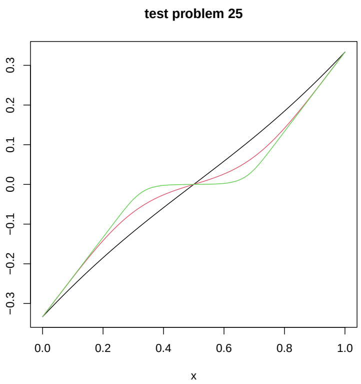

Figure 25: Solution of the BVP ODE problem 25, see text for R-code

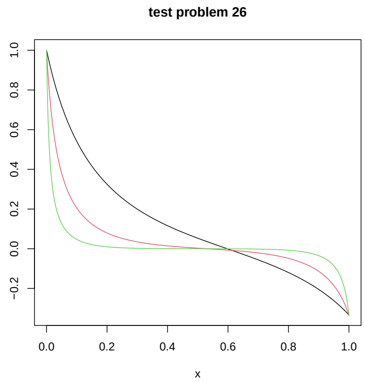

Figure 26: Solution of the BVP ODE problem 26, see text for R-code

3.8. problem 26

This problem equals previous problem, but with different boundary conditions:

$$y_{x=0} = 1$$
$$y_{x=1} = -1/3$$

It is solved in higher-order form.

```
Prob26 <- function(t, y, pars, xi)  {
   list(  -1/xi *(y[1]*y[2]-y[1]) )
 }
ini <- c(1, NA)
end <- c(-1/3, NA)

xi   <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                order = 2, func = Prob26, xi = xi)

xi   <- 0.02
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                order = 2, func = Prob26, xi = xi)

xi   <- 0.005
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                order = 2, func = Prob26, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 26")
```

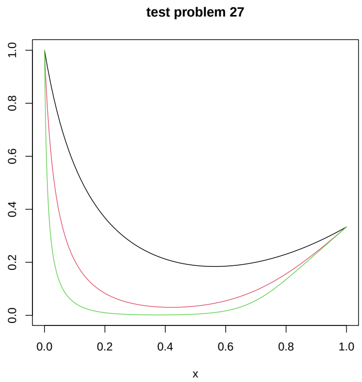

Figure 27: Solution of the BVP ODE problem 27, see text for R-code

3.9. problem 27

This problem equals previous problem, but with different boundary conditions:

$$y_{x=0} = 1$$
$$y_{x=1} = 1/3$$

```
Prob27 <- function(t, y, pars, xi) {
   list(c( y[2], -1/xi *(y[1]*y[2]-y[1]) ))
 }
ini <- c(1, NA)
end <- c(1/3, NA)

xi   <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob27, xi = xi)

xi   <- 0.02
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob27, xi = xi)

xi   <- 0.005
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob27, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 27")
```

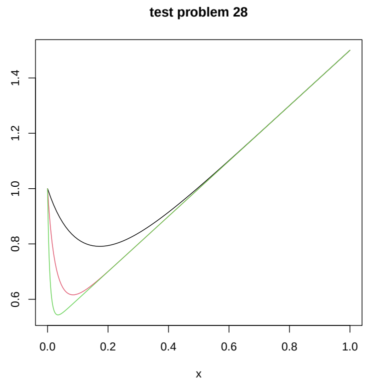

Figure 28: Solution of the BVP ODE problem 28, see text for R-code

3.10. problem 28

This problem equals previous problem, but with different boudnary conditions:

$$y_{x=0} = 1$$
$$y_{x=1} = 3/2$$

It is solved in higher-order form.

```
Prob28 <- function(t, y, pars, xi)  {
   list(  -1/xi *(y[1]*y[2]-y[1]))
 }
ini <- c(1, NA)
end <- c(3/2, NA)

xi   <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                order = 2, func = Prob28, xi = xi)

xi   <-0.02
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                order = 2, func = Prob28, xi = xi)

xi <-0.005
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                order = 2, func = Prob28, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 28")
```

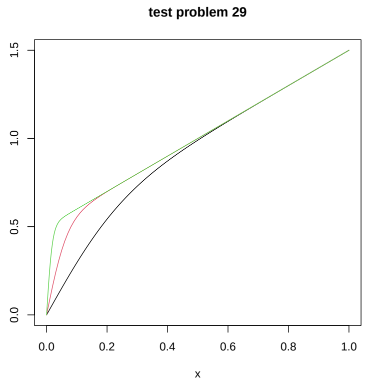

Figure 29: Solution of the BVP ODE problem 29, see text for R-code

3.11. problem 29

This problem equals previous problem, but with different boundary conditions:

$$y_{x=0} = 0$$
$$y_{x=1} = 3/2$$

```
Prob29 <- function(t, y, pars, xi)  {
   list(c( y[2], -1/xi *(y[1]*y[2]-y[1]) ))
 }
ini <- c(0,NA)
end <- c(3/2,NA)

xi   <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob29, xi = xi)

xi   <- 0.02
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob29, xi = xi)

xi   <- 0.005
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob29, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 29")
```

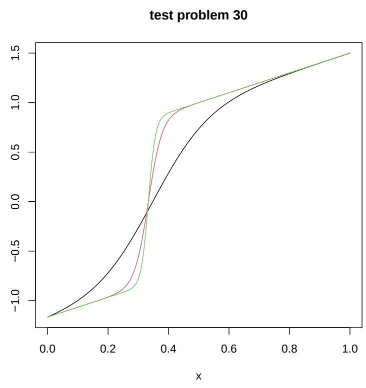

Figure 30: Solution of the BVP ODE problem 30, see text for R-code

3.12. problem 30

Similar to previous problems, with different boundary conditions:

$$y_{x=0} = -7/6$$
$$y_{x=1} = 3/2$$

It is solved in higher-order form.

```
Prob30 <- function(t, y, pars, xi)  {
   list( -1/xi *(y[1]*y[2]-y[1]) )
 }
ini <- c(-7/6, NA)
end <- c(3/2, NA)

xi   <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob30, xi = xi)
xi   <- 0.02
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob30, xi = xi)
xi   <- 0.01
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob30, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 30")
```

3.13. problem 31

$$
\begin{aligned}
y' &= \sin(\theta)\\
\theta' &= M\\
\xi M' &= -Q\\
\xi Q' &= (y-1)\cos(\theta) - MT\\
T &= \sec(\theta) + \xi Q \tan(\theta)
\end{aligned}
$$

where

$$y_{x=0} = y_{x=1} = M_{x=0} = M_{x=1} = 0$$

```
Prob31 <- function(t, Y, pars)  {
   with (as.list(Y), {
     dy    <- sin(Tet)
     dTet  <- M
     dM    <- -Q/xi
     T     <- 1/cos (Tet) +xi*Q*tan(Tet)
     dQ    <- 1/xi*((y-1)*cos(Tet)-M*T)
     list(c( dy, dTet, dM, dQ))
   })
 }
ini <- c(y = 0, Tet = NA, M = 0, Q = NA)
end <- c(y = 0, Tet = NA, M = 0, Q = NA)
```

Shooting does not work... But the mono-implicit Runge-Kutta method does...

```
xi <-0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob31, atol = 1e-10)
xi <- 0.05
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob31, atol = 1e-10)
xi <- 0.01
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
                func = Prob31, atol = 1e-10)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 31")
```

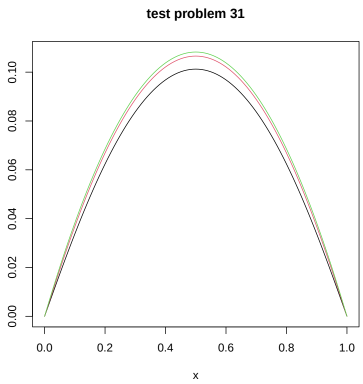

Figure 31: Solution of the BVP ODE problem 31, see text for R-code

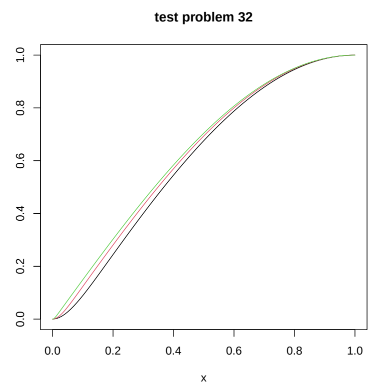

Figure 32: Solution of the BVP ODE problem 32, see text for R-code

3.14. problem 32

$$
\begin{aligned}
y'''' &= 1/\xi(y'y'' - yy''') \\
y_{x=0} = y'_{x=0} &= 0 \\
y_{x=1} &= 1 \\
y'_{x=1} &= 0
\end{aligned}
$$

It is solved in higher-order form.

```
Prob32 <- function(t, y, pars, xi) {
   list(1/xi*(y[2]*y[3]-y[1]*y[4]))
 }
ini <- c(0, 0, NA, NA)
end <- c(1, 0, NA, NA)

xi  <- 0.01
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 4, func = Prob32, xi = xi)
xi   <- 0.002
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 4, func = Prob32, xi = xi)
xi   <- 0.0001
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 4, func = Prob32, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 32")
```

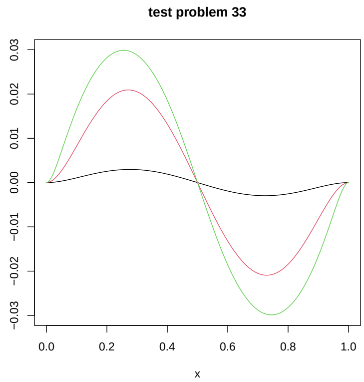

Figure 33: Solution of the BVP ODE problem 33, see text for R-code

3.15. problem 33

$$\begin{aligned} \xi z'''' &= -z \cdot z''' - y \cdot y' \\ \xi y'' &= y \cdot z' - z \cdot y' \end{aligned}$$

where

$$y_{x=0} = -1 y_{x=1} = 1 z_{x=0} = z'_{x=0} = z_{x=1} = z'_{x=1} = 0$$

```
Prob33 <- function(t, z, pars, xi) {
   list(c( z[2], z[3], z[4], 1/xi*(z[1]*z[4]-z[5]*z[6]),
           z[6], 1/xi*(z[5]*z[2]-z[1]*z[6])))
 }
ini <- c(0, 0, NA, NA, -1, NA)
end <- c(0, 0, NA, NA,  1, NA)

xi  <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               func = Prob33, xi = xi)
xi  <- 0.01
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               func = Prob33, xi = xi)
xi  <- 0.001
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               func = Prob33, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 33")
```

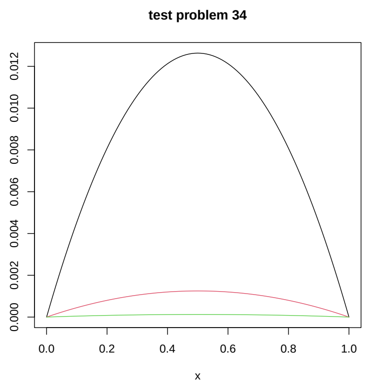

Figure 34: Solution of the BVP ODE problem 34, see text for R-code

3.16. problem 34

$$y'' = -\xi \cdot \exp(y)$$

where

$$y_{x=0} = y_{x=1} = 0$$

It is solved in higher-order form.

```
Prob34 <- function(t, y, pars, xi) {
   list(-xi*exp(y[1]))
 }
ini <- c(0, NA)
end <- c(0, NA)

xi  <- 0.1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob34, xi = xi)

xi   <- 0.01
mod2 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob34, xi = xi)

xi   <- 0.001
mod3 <- bvptwp(yini = ini, yend = end, x = seq(0, 1, by = 0.01),
               order = 2, func = Prob34, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 34")
```

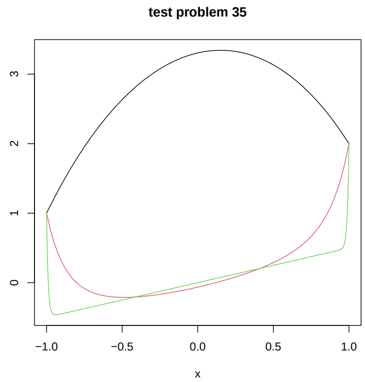

Figure 35: Solution of the BVP ODE problem 35, see text for R-code

3.17. problem 35

$$\xi y'' = xy' - y$$

where

$$y_{x=-1} = 1; y_{x=1} = 2$$

```
Prob35 <- function(x, y, pars, xi) {
   list(c( y[2], 1/xi*(x * y[2]-y[1])))
 }
ini <- c(1, NA)
end <- c(2, NA)

xi  <- 1
mod1 <- bvptwp(yini = ini, yend = end, x = seq(-1, 1, by = 0.05),
               func = Prob35, xi = xi)

xi   <- 0.1
mod2 <- bvptwp(yini = ini, yend = end, x = seq(-1, 1, by = 0.05),
               func = Prob35, xi = xi)

xi   <- 0.01
mod3 <- bvptwp(yini = ini, yend = end, x = seq(-1, 1, by = 0.05),
               func = Prob35, xi = xi)

plot(mod1, mod2, mod3, which = 1, lty = 1, main = "test problem 35")
```

References

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL https://www.R-project.org.

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

Soetaert K, Cash J, Mazzia F (2010a). *bvpSolve: solvers for boundary value problems of ordinary differential equations*. R package version 1.2.

Soetaert K, Meysman F (2010). *ReacTran: Reactive transport modelling in 1D, 2D and 3D*. R package version 1.2.

Soetaert K, Petzoldt T, Setzer RW (2010b). "Solving Differential Equations in R: Package deSolve." *Journal of Statistical Software*, **33**(9), 1–25. ISSN 1548-7660. URL https://www.jstatsoft.org/v33/i09.

Affiliation:

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

Jeff Cash
Imperial College London
South Kensington Campus
London SW7 2AZ, U.K.

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