Package bvpSolve, solving testproblems

Karline Soetaert
Centre for Estuarine and Marine Ecology
Netherlands Institute of Ecology
The Netherlands

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

Abstract

This document implements several testproblems that can be found on http://www.ma.ic.ac.uk/~jcash/BVP_software/PROBLEMS.PDF, using solvers from package **bvpSolve** (Soetaert and Cash 2009).

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 first-order ordinary differential equations (ODE), which for one 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_a and g_b the boundary conditions at the end points a and b.

The problem must be specified as a first-order system. Thus, higher-order ODEs need to be rewritten as a set of first-order systems. For instance:

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

can be rewritten as:

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

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

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).

Note that another package **deSolve** (Soetaert, Petzoldt, and Setzer 2009) is designed for solving initial value problems, i.e. where the boundary conditions are provided at the initial boundary point only.

Package **rootSolve** (Soetaert 2009) has functions to solve certain boundary value problems, using the method of lines approach. This is usually more efficient than the boundary value solvers from **bvpSolve**

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 ))
+ }
```

which is solved for different values of ξ

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

   user  system elapsed
   0.02    0.00    0.01
```

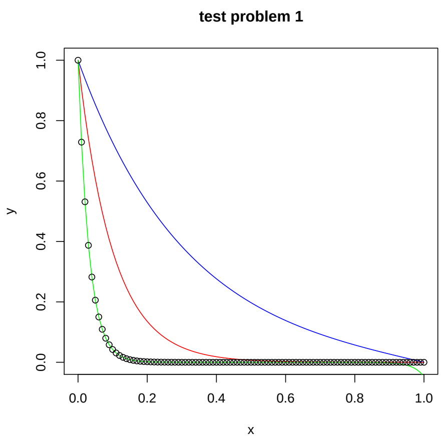

Figure 1: Solution of the BVP ODE problem 1, see text for R-code

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

```
   user  system elapsed 
   0.03    0.00    0.03 
```

for smaller ξ

```
> xi <-0.01
> shoot2  <- bvpshoot(yini=c(1,NA),yend=c(0,NA),x=seq(0,1,by=0.01),
+              func=Prob1, guess=0)
```

and for a very small value

```
> xi <-0.001
> shoot3  <- bvpshoot(yini=c(1,NA),yend=c(0,NA),x=seq(0,1,by=0.01),
+             func=Prob1, guess=0)
```

and the output plotted

```
> plot(shoot[,1],shoot[,2],type="l",main="test problem 1",col="blue",
+   xlab="x",ylab="y")
> lines(shoot2[,1],shoot2[,2],col="red")
> lines(shoot3[,1],shoot3[,2],col="green")
> # exact solution
> curve(exp(-x/sqrt(xi))-exp((x-2)/sqrt(xi))/(1-exp(-2/sqrt(xi))),
+       0,1,add=TRUE,type="p")
```

2.2. problem 2

This problem is:

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

which is rewritten as:

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

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

```
> xi <-0.1
> twp   <- bvptwp(yini=c(1,NA),yend=c(0,NA),x=seq(0,1,by=0.01),
+             func=Prob2, guess=0, atol=1e-10)
> xi <- 0.01
> twp2  <- bvptwp(yini=c(1,NA),yend=c(0,NA),x=seq(0,1,by=0.01),
+             func=Prob2, guess=0, atol=1e-10)
```

The solution can be compared with the analytical solution:

```
> plot(shoot[,1],shoot[,2],type="l",main="test problem 2",col="blue",
+   xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
> curve((1-exp((x-1)/xi))/(1-exp(-1/xi)),0,1,type="p",add=TRUE)
```

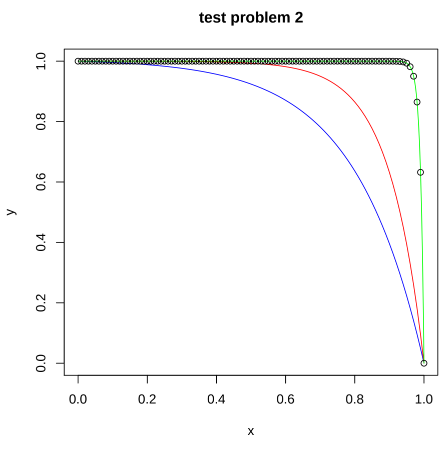

Figure 2: Solution of the BVP ODE problem 2, see text for R-code

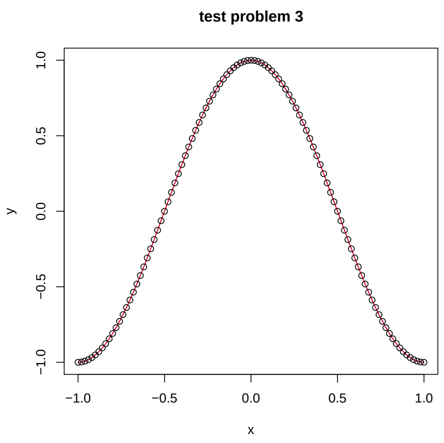

Figure 3: Solution of the BVP ODE problem 3, see text for R-code

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
> shoot  <- bvpshoot(yini=c(-1,NA),yend=c(-1,NA),x=seq(-1,1,by=0.01),
+             func=Prob3, guess=0)

> xi <-0.01
> twp    <- bvptwp(yini=c(-1,NA),yend=c(-1,NA),x=seq(-1,1,by=0.01),
+             func=Prob3, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 3",col="blue",
+   xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> curve(cos(pi*x),-1,1,type="p",add=TRUE)
```

2.4. problem 4

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

```
> Prob4 <- function(t, y, pars) {
+   list(c( y[2], (-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)
> shoot  <- bvpshoot(yini=yini,yend=yend,x=seq(-1,1,by=0.01),
+           func=Prob4, guess=0)

> xi <-0.1
> yend <- c(1+exp(-2*(1+xi)/xi),NA)
> twp    <- bvptwp(yini=yini,yend=yend,x=seq(-1,1,by=0.01),
+           func=Prob4, guess=0)

> xi <-0.01
> yend <- c(1+exp(-2*(1+xi)/xi),NA)
> twp2    <- bvptwp(yini=yini,yend=yend,x=seq(-1,1,by=0.01),
+           func=Prob4, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 4",ylim=c(0,1.2),
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
> curve(exp(x-1)+exp(-(1+xi)*(1+x)/xi),-1,1,type="p",add=TRUE)
```

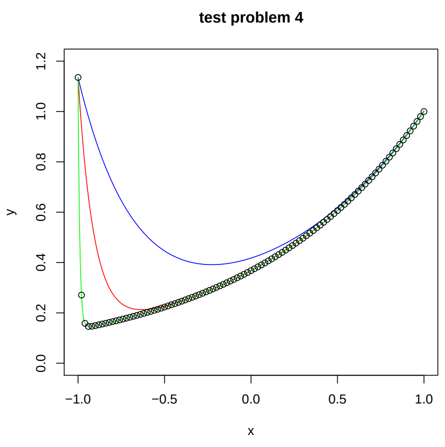

Figure 4: Solution of the BVP ODE problem 4, see text for R-code

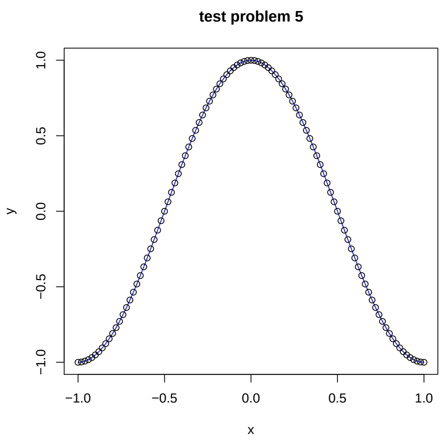

Figure 5: Solution of the BVP ODE problem 5, see text for R-code

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
> shoot  <- bvpshoot(yini=c(-1,NA),yend=c(-1,NA),x=seq(-1,1,by=0.01),
+            func=Prob5, guess=0)

> twp    <- bvptwp(yini=c(-1,NA),yend=c(-1,NA),x=seq(-1,1,by=0.01),
+            func=Prob5, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 5",
+   col="blue",xlab="x",ylab="y")
> curve(cos(pi*x),-1,1,type="p",add=TRUE)
```

2.6. problem 6

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

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

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

> xi <-0.1
> shoot  <- bvpshoot(yini=c(-2,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+             func=Prob6, guess=0)

> xi <-0.01
> twp  <- bvptwp(yini=c(-2,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+             func=Prob6, guess=0)

> xi <-0.001
> twp2  <- bvptwp(yini=c(-2,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+             func=Prob6, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 6",
+      ylim =c(-2,2),col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
> 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)
```

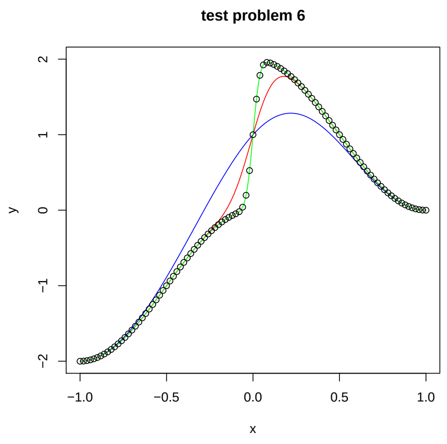

Figure 6: Solution of the BVP ODE problem 6, see text for R-code

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
> twp  <- bvptwp(yini=c(-1,NA),yend=c(1,NA),x=x,func=prob7, guess=0)

> xi <- 0.001
> twp2  <- bvptwp(yini=c(-1,NA),yend=c(1,NA),x=x,func=prob7, guess=0)

> xi <- 0.0001
> twp3  <- bvptwp(yini=c(-1,NA),yend=c(1,NA),x=x,func=prob7, guess=0)

> par(mfrow=c(1,2))
> plot(twp[,1],twp[,2],type="l",main="test problem 7",
+   col="blue",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)
> plot(twp[,1],twp[,3],type="l",main="test problem 7",
+   col="blue",xlab="x",ylab="y'")
> lines(twp2[,1],twp2[,3],col="red")
> lines(twp3[,1],twp3[,3],col="green")
> par(mfrow=c(1,1))
```

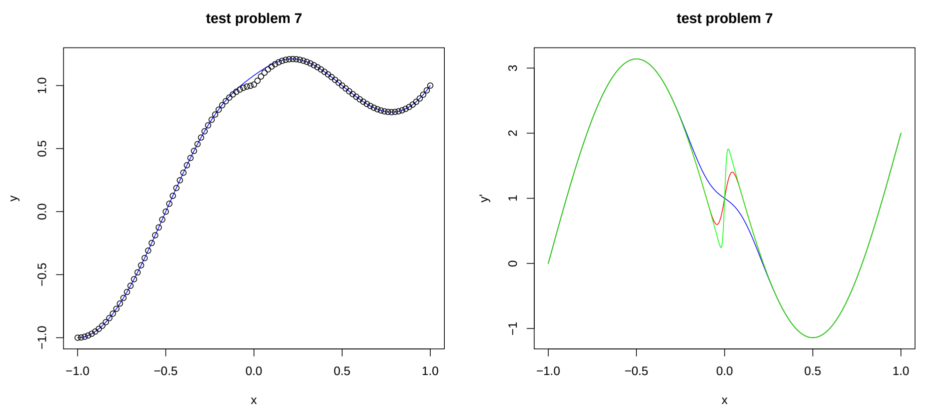

Figure 7: Solution of the BVP ODE problem 7, y and y' versus x- see text for R-code

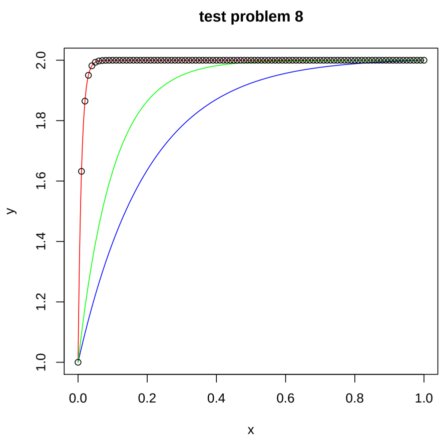

Figure 8: Solution of the BVP ODE problem 8, see text for R-code

2.8. problem 8

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

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

> xi <- 0.2
> shoot  <- bvpshoot(yini=c(1,NA),yend=c(2,NA),x=x,func=prob8,guess=0)

> xi <- 0.1
> twp  <- bvptwp(yini=c(1,NA),yend=c(2,NA),x=x,func=prob8, guess=0)

> xi <- 0.01
> twp2  <- bvptwp(yini=c(1,NA),yend=c(2,NA),x=x,func=prob8, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 8",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp[,1],twp[,2],col="green")
> # analytical solution
> curve(2-exp(-1/xi)-exp(-x/xi)/(1-exp(-1/xi)),0,1,add=TRUE,type="p")
```

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
> twp  <- bvptwp(yini=c(1/(1+xi),NA),yend=c(1/(1+xi),NA),x=seq(-1,1,by=0.01),
+             func=Prob9, guess=0)

> xi <-0.02
> twp2  <- bvptwp(yini=c(1/(1+xi),NA),yend=c(1/(1+xi),NA),x=seq(-1,1,by=0.01),
+             func=Prob9, guess=0)

> xi <-0.01
> twp3  <- bvptwp(yini=c(1/(1+xi),NA),yend=c(1/(1+xi),NA),x=seq(-1,1,by=0.01),
+             func=Prob9, guess=0)

> plot(twp[,1],twp[,2],type="l",main="test problem 9", ylim=c(0,100),
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
> # exact
> curve(1/(xi+x^2),-1,1,type="p",add=TRUE)
```

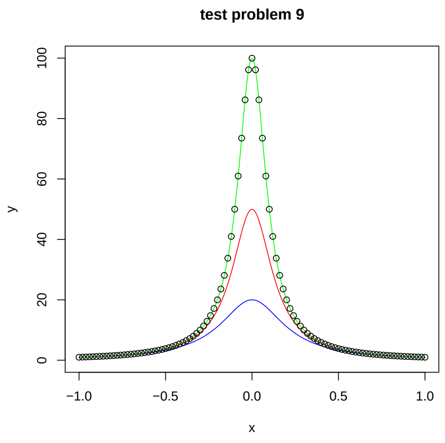

Figure 9: Solution of the BVP ODE problem 9, see text for R-code

2.10. problem 10

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

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

> xi <-0.1
> shoot  <- bvpshoot(yini=c(0,NA),yend=c(2,NA),x=seq(-1,1,by=0.01),
+         func=Prob10, guess=0,atol=1e-10)

> xi <- 0.05
> twp  <- bvptwp(yini=c(0,NA),yend=c(2,NA),x=seq(-1,1,by=0.01),
+            func=Prob10, guess=0)

> xi <- 0.01
> twp2  <- bvptwp(yini=c(0,NA),yend=c(2,NA),x=seq(-1,1,by=0.01),
+            func=Prob10, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 10",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp[,1],twp[,2],col="green")
> 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)
```

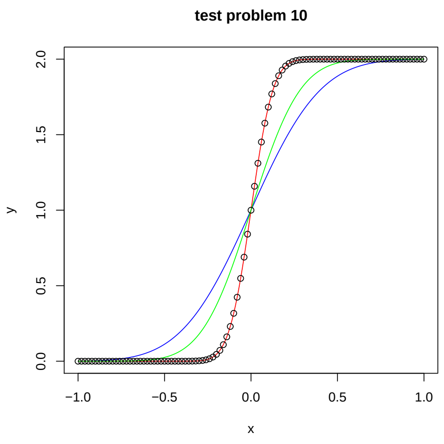

Figure 10: Solution of the BVP ODE problem 10, see text for R-code

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
> # Shooting
> print(system.time(
+ shoot  <- bvpshoot(yini=c(-1,NA),yend=c(-1,NA),x=seq(-1,1,by=0.01),
+         func=Prob11, guess=0,atol=1e-10)
+ ))
```

```
   user  system elapsed
   0.01    0.00    0.01
```

```
> print(system.time(
+ twp  <- bvptwp(yini=c(-1,NA),yend=c(-1,NA),x=seq(-1,1,by=0.01),
+         func=Prob11, guess=0,atol=1e-10)
+ ))
```

```
   user  system elapsed
   0.12    0.00    0.12
```

```
> plot(shoot[,1],shoot[,2],type="l",main="test problem 11",
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="green")
> curve(cos(pi*x),-1,1,type="p",add=TRUE)
```

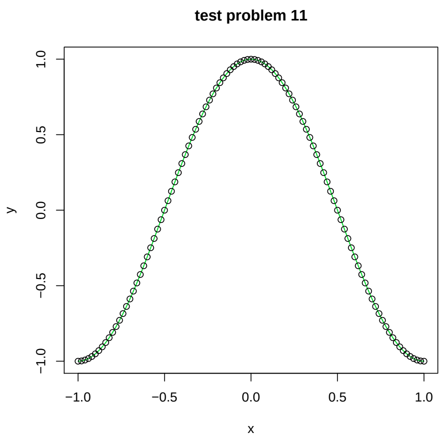

Figure 11: Solution of the BVP ODE problem 11, see text for R-code

2.12. problem 12

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

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

```
> Prob12 <- function(x, y, pars) {
+   list(c(y[2],1/xi*(y[1]-(xi*pi*pi+1)*cos(pi*x))))
+ }

> xi <-0.01
> shoot  <- bvpshoot(yini=c(-1,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+          func=Prob12, guess=0)

> xi <-0.0025
> twp  <- bvptwp(yini=c(-1,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+          func=Prob12, guess=0)

> xi <-0.0001
> twp2  <- bvptwp(yini=c(-1,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+          func=Prob12, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 12",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp[,1],twp[,2],col="green")
> curve(cos(pi*x)+exp((x-1)/sqrt(xi)),-1,1,type="p",add=TRUE)
```

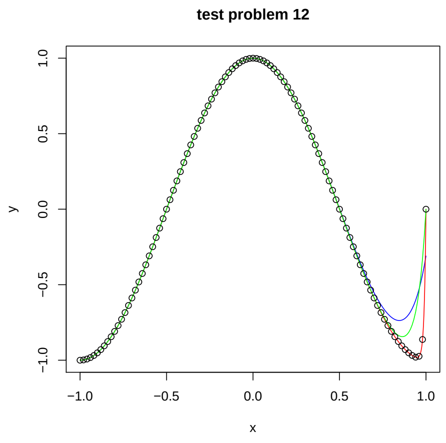

Figure 12: Solution of the BVP ODE problem 12, see text for R-code

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

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

> xi <-0.0001
> twp2  <- bvptwp(yini=c(0,NA),yend=c(-1,NA),x=seq(-1,1,by=0.01),
+         func=Prob13, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 13",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp[,1],twp[,2],col="green")
> curve(cos(pi*x)+exp(-(x+1)/sqrt(xi)),-1,1,type="p",add=TRUE)
```

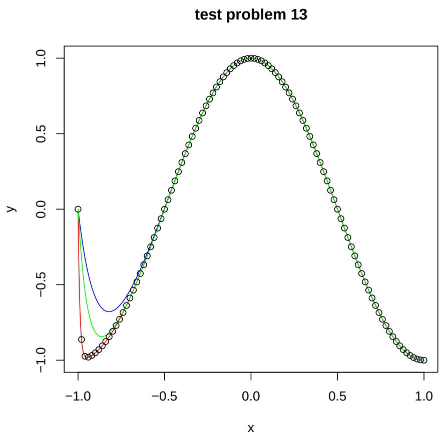

Figure 13: Solution of the BVP ODE problem 13, see text for R-code

2.14. problem 14

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

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

```
> Prob14 <- function(x, y, pars)  {
+   list(c( y[2], 1/xi*(y[1]-(xi*pi*pi+1)*cos(pi*x))))
+ }

> xi <-0.01
> shoot  <- bvpshoot(yini=c(0,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+          func=Prob14, guess=0)

> xi <-0.0025
> twp  <- bvptwp(yini=c(0,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+          func=Prob14, guess=0)

> xi <-0.0001
> twp2  <- bvptwp(yini=c(0,NA),yend=c(0,NA),x=seq(-1,1,by=0.01),
+          func=Prob14, guess=0)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 14", ylim=c(-1,1),
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp[,1],twp[,2],col="green")
> curve(cos(pi*x)+exp((x-1)/sqrt(xi))+exp(-(x+1)/sqrt(xi)),-1,1,type="p",add=TRUE)
```

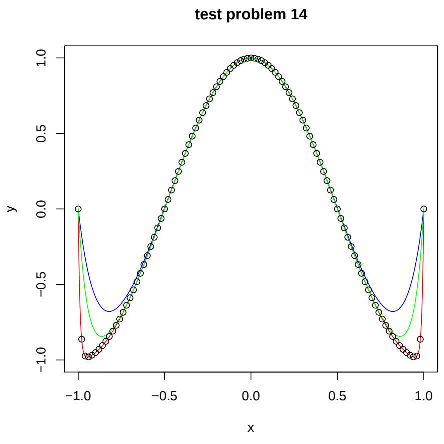

Figure 14: Solution of the BVP ODE problem 14, see text for R-code

2.15. 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(
+ shoot  <- bvpshoot(yini=c(1,NA),yend=c(1,NA),x=seq(-1,1,by=0.01),
+           func=Prob15, guess=0)
+ ))

   user  system elapsed
   0.08    0.00    0.08

> xi <- 0.005
> print(system.time(
+ twp  <- bvptwp(yini=c(1,NA),yend=c(1,NA),x=seq(-1,1,by=0.01),
+           func=Prob15, guess=0)
+ ))

   user  system elapsed
   0.08    0.00    0.08

> xi <- 0.01
> print(system.time(
+ twp2  <- bvptwp(yini=c(1,NA),yend=c(1,NA),x=seq(-1,1,by=0.01),
+           func=Prob15, guess=0)
+ ))

   user  system elapsed
   0.08    0.00    0.08

> plot(shoot[,1],shoot[,2],type="l",main="test problem 15",
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
```

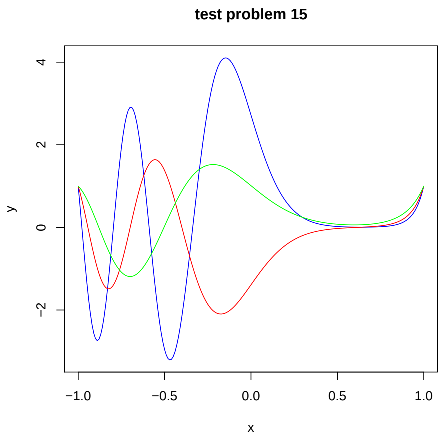

Figure 15: Solution of the BVP ODE problem 15, see text for R-code

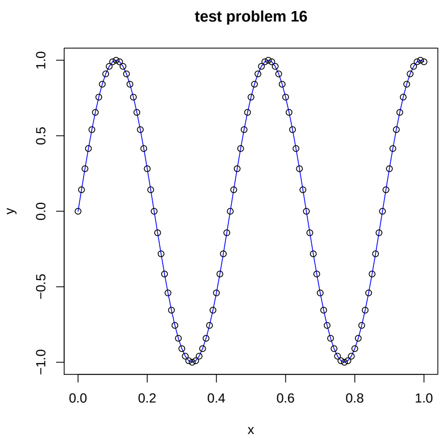

Figure 16: Solution of the BVP ODE problem 16, see text for R-code

2.16. problem 16

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

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

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

   user  system elapsed
   0.09    0.00    0.09

> plot(shoot[,1],shoot[,2],type="l",main="test problem 16",
+   col="blue",xlab="x",ylab="y")
> curve(sin(pi*x/2/xi),0,1,type="p",add=TRUE)
```

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
> twp1  <- bvptwp(yini=c(-0.1/sqrt(xi+0.01),NA),
+                     yend=c(0.1/sqrt(xi+0.01),NA),x=xseq,
+                     func=Prob17, guess=0,atol=1e-10)

> xi <- 0.001
> twp2  <- bvptwp(yini=c(-0.1/sqrt(xi+0.01),NA),
+                     yend=c(0.1/sqrt(xi+0.01),NA),x=xseq,
+                     func=Prob17, guess=0,atol=1e-8)

> xi <- 0.0001
> twp3  <- bvptwp(yini=c(-0.1/sqrt(xi+0.01),NA),
+                     yend=c(0.1/sqrt(xi+0.01),NA),x=xseq,
+                     func=Prob17, guess=0,atol=1e-8)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 17",ylim=c(-1,1),
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="red")
> curve(x/sqrt(xi+x^2),-0.1,0.1,type="p",add=TRUE)
```

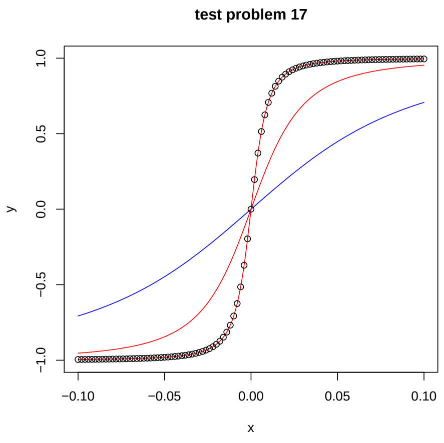

Figure 17: Solution of the BVP ODE problem 17, see text for R-code

2.18. problem 18

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

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

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

> xi <- 0.1
> twp  <- bvptwp(yini=c(1,NA),yend=c(exp(-1/xi),NA),x=xseq,
+             func=Prob18, guess=0,atol=1e-10)

> xi <- 0.01
> twp2  <- bvptwp(yini=c(1,NA),yend=c(exp(-1/xi),NA),x=xseq,
+             func=Prob18, guess=0,atol=1e-10)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 18",ylim=c(0,1),
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
> curve(exp(-x/xi),0,1,type="p",add=TRUE)
```

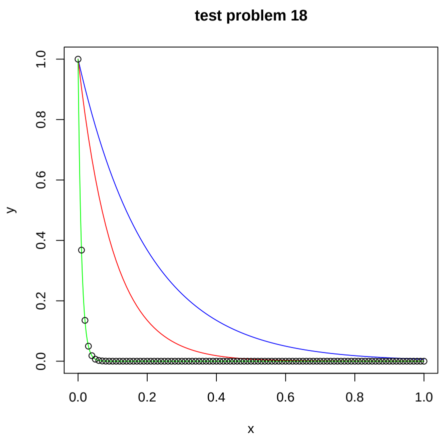

Figure 18: Solution of the BVP ODE problem 18, see text for R-code

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
> shoot  <- 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
> twp  <- bvptwp(yini=c(0,NA),yend=c(0,NA),x=seq(0,1,by=0.01),
+            func=Prob19, guess=0,ksi=xi,atol=1e-15)

> xi <- 0.005
> print(system.time(
+
+ twp2  <- bvptwp(yini=c(0,NA),yend=c(0,NA),x=seq(0,1,by=0.01),
+            func=Prob19, guess=0,ksi=xi, atol=1e-10)
+ ))

   user  system elapsed
   0.42    0.00    0.43

> plot(shoot[,1],shoot[,2],type="l",main="test problem 19", ylim=c(-0.7,0),
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
```

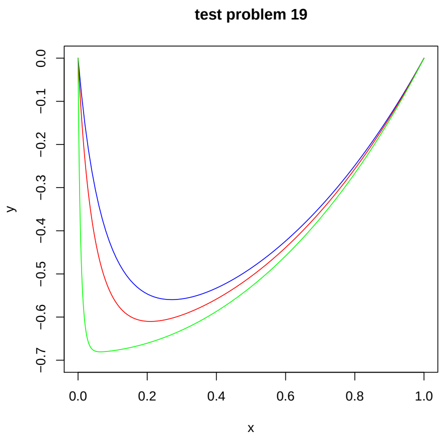

Figure 19: Solution of the BVP ODE problem 19, see text for R-code

3.2. problem 20

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

```
> Prob20 <- function(x, y, pars) {
+   list(c( y[2] , 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)
> twp1  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob20, guess=0)

> xi <-0.3
> ini <- c(1+xi * log(cosh(0.745/xi)),NA)
> end <- c(1+xi * log(cosh(0.255/xi)),NA)
> twp2  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob20, guess=0)

> xi <-0.01
> ini <- c(1+xi * log(cosh(0.745/xi)),NA)
> end <- c(1+xi * log(cosh(0.255/xi)),NA)
> twp3  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob20, guess=0)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 20", ylim=c(1,1.8),
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
> curve(1+xi * log(cosh((x-0.745)/xi)),0,1,add=TRUE,type="p")
```

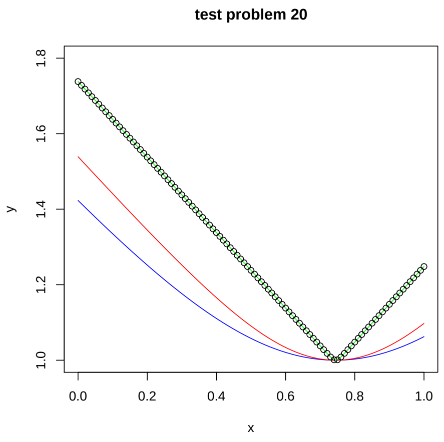

Figure 20: Solution of the BVP ODE problem 20, see text for R-code

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)
> shoot  <- 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)
> twp  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob21, guess=0, xi=xi)

> xi <-0.01
> end <- c(exp(-1/sqrt(xi)),NA)
> twp2  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob21, guess=0, xi=xi)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 21", ylim=c(0,1),
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
> curve(exp(-x/sqrt(xi)),0,1,add=TRUE,type="p")
```

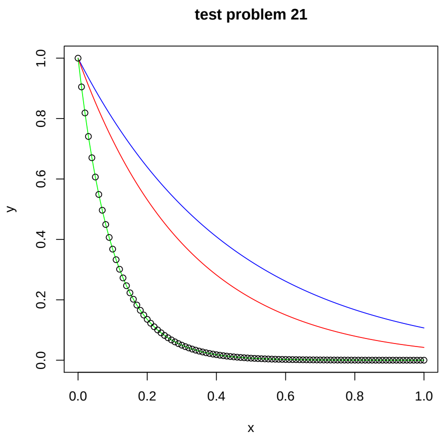

Figure 21: Solution of the BVP ODE problem 21, see text for R-code

3.4. problem 22

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

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

> xi <-0.1
> shoot  <- bvpshoot(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob22, guess=0, xi=xi)

> xi <-0.05
> twp  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob22, guess=0, xi=xi)

> xi <- 0.01
> twp2  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob22, guess=0,xi=xi)

> plot(shoot[,1],shoot[,2],type="l",main="test problem 22", ylim=c(0,1),
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
```

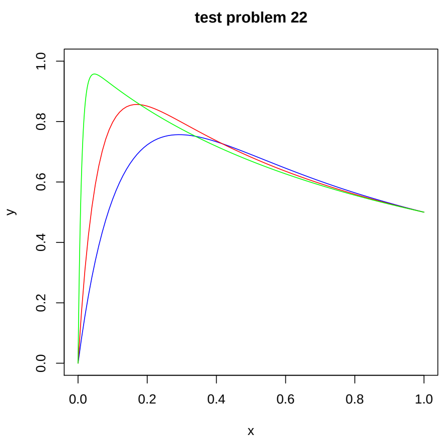

Figure 22: Solution of the BVP ODE problem 22, see text for R-code

3.5. problem 23

This is a difficult problem that cannot be solved with bvpshoot

$$y'' = \mu sinh(\mu y)$$
$$y_{(x=0)} = y_{(x=1)} = 1$$

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

> xi <- 1/5
> twp1  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob23, guess=c(0),xi=xi)

> xi <- 1/7
> twp  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob23, guess=0,xi=xi)

> xi <- 1/9
> twp2  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob23, guess=0,xi=xi)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 23",
+   col="blue",xlab="x",ylab="y")
> lines(twp[,1],twp[,2],col="red")
> lines(twp2[,1],twp2[,2],col="green")
```

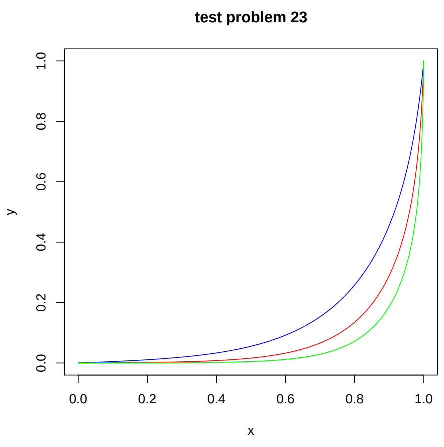

Figure 23: Solution of the BVP ODE problem 23, see text for R-code

3.6. problem 24

This is a particularly difficult problem to solve

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

$$A(x) = 1 + x^2$$

$$y_{(x=0)} = 0.9129$$

$$y_{(x=1)} = 0.375$$

```
> Prob24 <- function(t, y, pars, xi) {
+   A <- 1+t*t
+   AA <- 2*t
+   ga <- 1.4
+   list(c(y[2],
+        (((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),
+         func=Prob24, guess=0.9 ,xi=xi)

> xi <-0.025
> mod2  <- bvpshoot(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob24, guess=0.9 ,xi=xi)

> xi <-0.02
> mod3  <- bvpshoot(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob24, guess=0.9 ,xi=xi)
> attributes(mod3)$roots         # has FAILED: f.root too large!

       root     f.root iter
1 0.8359306 -0.4355728   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 <- 0.01
> mod3  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob24, xi=xi, xguess = mod2[,1], yguess = t(mod2[,2:3]))

> xi <- 0.005
> mod4  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob24, xi=xi, xguess = mod3[,1], yguess = t(mod3[,2:3]))
```

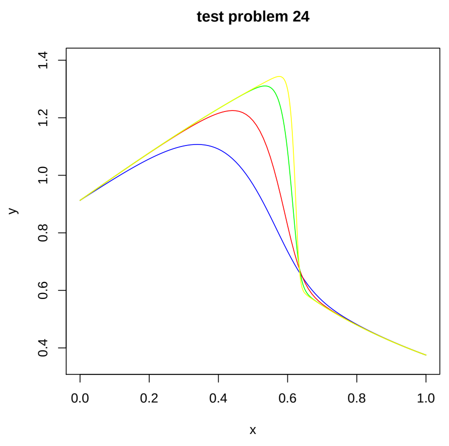

Figure 24: Solution of the BVP ODE problem 24, see text for R-code

```
> plot(mod1[,1],mod1[,2],type="l",main="test problem 24",
+   col="blue",xlab="x",ylab="y",ylim=c(0.35,1.4))
> lines(mod2[,1],mod2[,2],col="red")
> lines(mod3[,1],mod3[,2],col="green")
> lines(mod4[,1],mod4[,2],col="yellow")
```

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
> twp1  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+          func=Prob25, guess=0, xi=xi)

> xi <-0.01
> twp2  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+          func=Prob25, guess=0,xi=xi)

> xi <- 0.001
> twp3  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+          func=Prob25, guess=0,xi=xi)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 25",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

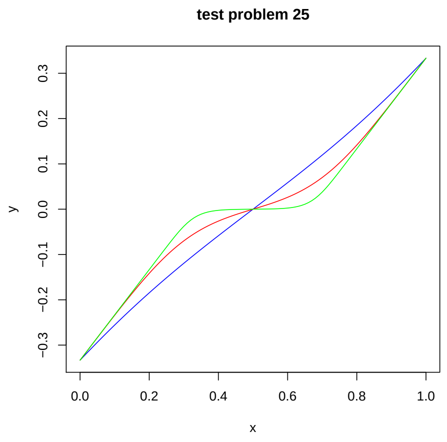

Figure 25: Solution of the BVP ODE problem 25, see text for R-code

3.8. problem 26

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

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

```
> Prob26 <- 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
> twp1  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob26, guess=0,xi=xi)

> xi <-0.02
> twp2  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob26, guess=0,xi=xi)

> xi <-0.005
> twp3  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob26, guess=0,xi=xi)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 26",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

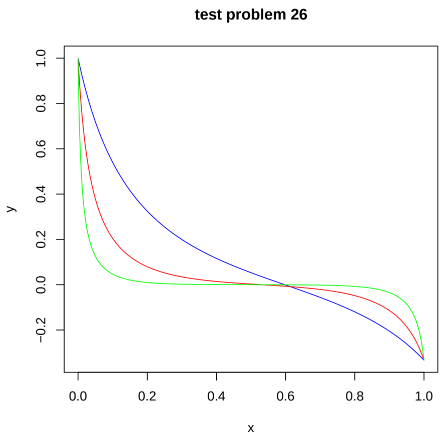

Figure 26: Solution of the BVP ODE problem 26, see text for R-code

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
> twp1 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob27, guess=0,xi=xi)

> xi   <-0.02
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob27, guess=0,xi=xi)

> xi <-0.005
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob27, guess=0,xi=xi)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 27", ylim=c(0,1),
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

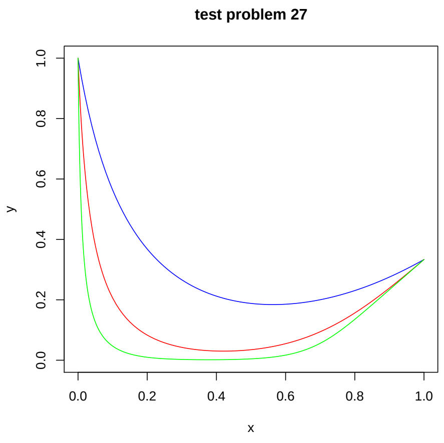

Figure 27: Solution of the BVP ODE problem 27, see text for R-code

3.10. problem 28

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

$$
\begin{aligned}
y_{x=0} &= 1 \\
y_{x=1} &= 3/2
\end{aligned}
$$

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

> xi   <-0.1
> twp1 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob28, guess=0,xi=xi)

> xi   <-0.02
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob28, guess=0,xi=xi)

> xi <-0.005
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob28, guess=0,xi=xi)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 28", ylim=c(0.4,1.5),
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

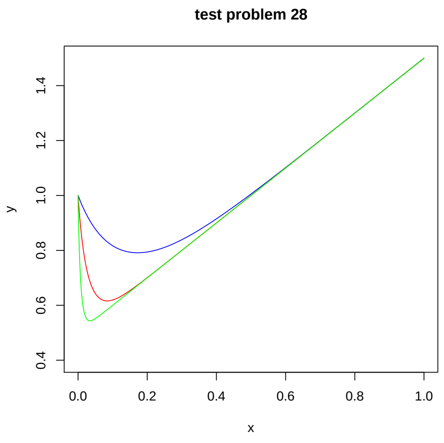

Figure 28: Solution of the BVP ODE problem 28, see text for R-code

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
> twp1 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob29, guess=0,xi=xi)

> xi   <-0.02
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob29, guess=0,xi=xi)

> xi <-0.005
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob29, guess=0,xi=xi)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 29",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

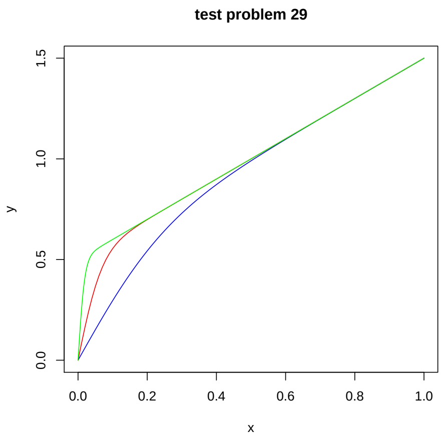

Figure 29: Solution of the BVP ODE problem 29, see text for R-code

3.12. problem 30

Similar to previous problems, with different boundary conditions:

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

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

> xi   <-0.1
> twp1 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob30, guess=0,xi=xi)

> xi   <-0.02
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob30, guess=0,xi=xi)

> xi <-0.01
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+         func=Prob30, guess=0,xi=xi)

> plot(twp1[,1],twp1[,2],type="l",main="test problem 30",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

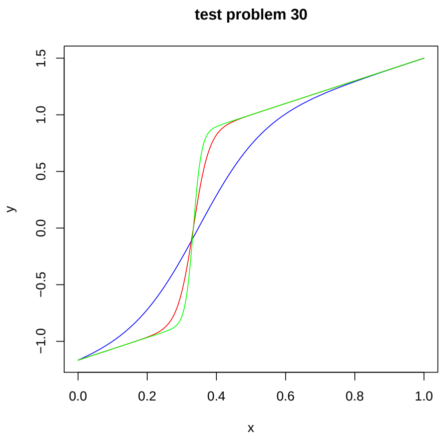

Figure 30: Solution of the BVP ODE problem 30, see text for R-code

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
> twp  <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob31, guess=c(0,0),atol=1e-10)

> xi <- 0.05
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob31, guess=c(0,0),atol=1e-10)

> xi <- 0.01
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob31, guess=c(0,0),atol=1e-10)

> plot(twp[,1],twp[,4],type="l",main="test problem 31",
+   col="blue",xlab="x",ylab="y3")
> lines(twp2[,1],twp2[,4],col="red")
> lines(twp3[,1],twp3[,4],col="green")
```

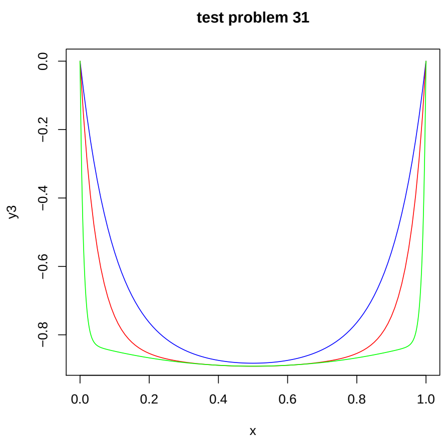

Figure 31: Solution of the BVP ODE problem 31, see text for R-code

3.14. problem 32

$$y'''' \quad = \quad 1/\xi(y'y'' - yy''')$$

where

$$y_{x=0} = y'_{x=0} = 0 y_{x=1} = 1 y'_{x=1} = 0$$

```
> Prob32 <- function(t, y, pars, xi) {
+   list(c( y[2], y[3], y[4], 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
> twp <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+             func=Prob32, guess=c(0,0), xi=xi)

> xi <-0.002
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+             func=Prob32, guess=c(0,0), xi=xi)

> xi <-0.0001
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+             func=Prob32, guess=c(0,0), xi=xi)

> plot(twp[,1],twp[,3],type="l",main="test problem 32",
+   col="blue",xlab="x",ylab="y'")
> lines(twp2[,1],twp2[,3],col="red")
> lines(twp3[,1],twp3[,3],col="green")
```

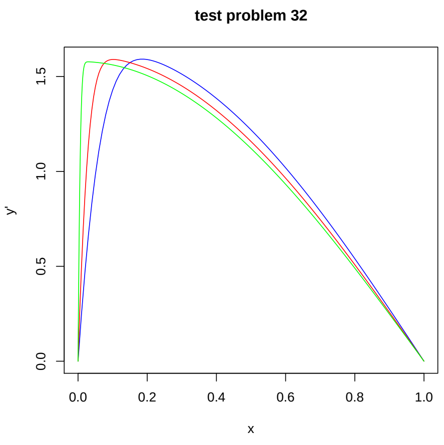

Figure 32: Solution of the BVP ODE problem 32, see text for R-code

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
> twp <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob33, guess=c(0,0,0), xi=xi)

> xi <-0.01
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob33, guess=c(0,0,0), xi=xi)

> xi <-0.001
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob33, guess=c(0,0,0), xi=xi)

> plot(twp[,1],twp[,2],type="l",main="test problem 33",
+   col="blue",xlab="x",ylab="y",ylim=c(-0.05,0.05))
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

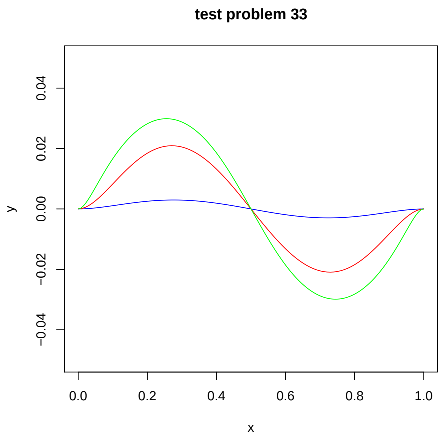

Figure 33: Solution of the BVP ODE problem 33, see text for R-code

3.16. problem 34

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

where

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

```
> Prob34 <- function(t, y, pars, xi) {
+   list(c( y[2], -xi*exp(y[1])))
+
+ }
> ini <- c(0,NA)
> end <- c(0,NA)

> xi <-0.1
> twp <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob34, guess=c(0), xi=xi)

> xi <-0.01
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob34, guess=c(0), xi=xi)

> xi <-0.001
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(0,1,by=0.01),
+            func=Prob34, guess=c(0), xi=xi)

> plot(twp[,1],twp[,2],type="l",main="test problem 34",
+   col="blue",xlab="x",ylab="y")
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

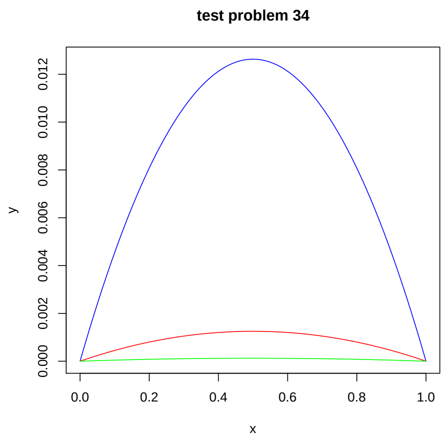

Figure 34: Solution of the BVP ODE problem 34, see text for R-code

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
> twp <- bvptwp(yini=ini,yend=end,x=seq(-1,1,by=0.05),
+            func=Prob35, guess=c(0), xi=xi)

> xi <-0.1
> twp2 <- bvptwp(yini=ini,yend=end,x=seq(-1,1,by=0.05),
+            func=Prob35, guess=c(0), xi=xi)

> xi <-0.01
> twp3 <- bvptwp(yini=ini,yend=end,x=seq(-1,1,by=0.05),
+            func=Prob35, guess=c(0), xi=xi)

> plot(twp[,1],twp[,2],type="l",main="test problem 35",
+   col="blue",xlab="x",ylab="y",ylim=c(-1,4))
> lines(twp2[,1],twp2[,2],col="red")
> lines(twp3[,1],twp3[,2],col="green")
```

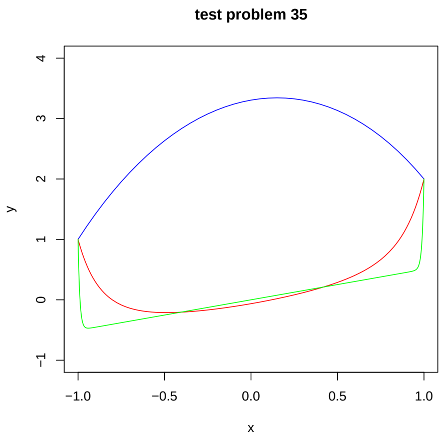

Figure 35: Solution of the BVP ODE problem 35, see text for R-code

References

Soetaert K (2009). *rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations.* R package version 1.4.

Soetaert K, Cash J (2009). *bvpSolve: solvers for boundary value problems of ordinary differential equations.* R package version 1.1.

Soetaert K, Petzoldt T, Setzer RW (2009). *deSolve: General solvers for initial value problems of ordinary differential equations (ODE), partial differential equations (PDE) and differential algebraic equations (DAE).* R package version 1.3.

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