PC5a - Interpolation

Question 1

This following R-script can be used:

```
library(pracma)
x <- 0:3
y <- x^2+3*x+2
x_fit <- c(1/2,3/2,5/2)          # x-values for which an approximation is required
y_true <- x_fit^2+3*x_fit+2      # true y-values
for(i in 1:3){
  p <- polyfit(x[i:(i+1)],y[i:(i+1)],1)    # determine linear approximation
  y_approx <- polyval(p,x_fit[i])          # fitted value based on approximation
  cat(sprintf("x=%5.3f   y(true)=%6.3f   y(approximation)=%6.3f   error=%5.3f\n",
               x_fit[i],       y_true[i],             y_approx, y_true[i]-y_approx))
}
```

```
## x=0.500   y(true)= 3.750   y(approximation)= 4.000   error=-0.250
## x=1.500   y(true)= 8.750   y(approximation)= 9.000   error=-0.250
## x=2.500   y(true)=15.750   y(approximation)=16.000   error=-0.250
```

Question 2

The table with forward difference is given by:

x	f	Δf	$\Delta^2 f$
0	0	$1(=1-0)$	$-2(=-1-1)$
$\pi/2$	1	$-1(=0-1)$	
π	0		

This is generated by the following R-script

```
x <- c(0,pi/2,pi)
f <- c(0,1,0)
df <- diff(f)
d2f <- diff(df)
cat("    x            f          df          d2f\n")
cat("==================================================\n")
for(i in 1:length(x)){
  cat(sprintf("%11.6f",x[i]))
  for(j in 1:(3-i+1)){
     cat(sprintf("%11.6f",switch(j,f[i],df[i],d2f[i])))
```

```
  }
  cat("\n")
}
```

```
##      x             f           df           d2f
## ==================================================
##     0.000000    0.000000    1.000000   -2.000000
##     1.570796    1.000000   -1.000000
##     3.141593    0.000000
```

This leads to the following 2nd order polynomial:

$$
\begin{aligned}
f(x) &\approx 0 + \frac{1}{\pi/2}(x-0) + \frac{-2}{(\pi/2)^2 2!}(x-0)(x-\pi/2) \\
&= \frac{2}{\pi}x - \frac{4}{\pi^2}x\left(x - \frac{\pi}{2}\right) = \frac{2}{\pi}x - \frac{4}{\pi^2}x^2 + \frac{2}{\pi}x = -\frac{4}{\pi^2}x^2 + \frac{4}{\pi}x
\end{aligned}
$$

The coefficients $\left(-\frac{4}{\pi^2}, \frac{4}{\pi}, 0\right)$ correspond to the output of:

```
coef <- polyfit(x,f,2)
coef
```

```
## [1] -4.052847e-01  1.273240e+00  6.661338e-16
```

```
formatC(coef,format="f",digits=6)
```

```
## [1] "-0.405285" "1.273240"  "0.000000"
```

Note the last function `formatC()` to get the "%.6f" output for a vector!

Question 3

The points $(-\pi/2, -1), (0,0), (\pi/2, 1)$ and $(\pi, 0)$ represent:

x_1	$-\pi/2$	$f(x_1)$	-1
x_2	0	$f(x_2)$	0
x_3	$\pi/2$	$f(x_3)$	1
x_4	π	$f(x_4)$	0

Substitution into the formula

$$\begin{aligned} f_3(x) =& \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} f(x_1) + \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)} f(x_2) + \\ & \frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)} f(x_3) + \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)} f(x_4) \end{aligned}$$

gives

$$\begin{aligned} f_3(x) &= \frac{(x-0)(x-\frac{\pi}{2})(x-\pi)}{(\frac{-\pi}{2}-0)(\frac{-\pi}{2}-\frac{\pi}{2})(\frac{-\pi}{2}-\pi)} \cdot -1 + \frac{(x-\frac{-\pi}{2})(x-0)(x-\pi)}{(\frac{\pi}{2}-\frac{-\pi}{2})(\frac{\pi}{2}-0)(\frac{\pi}{2}-\pi)} \cdot 1 \\ &= \frac{1}{\frac{\pi}{2}\pi\frac{3\pi}{2}} x(x-\tfrac{\pi}{2})(x-\pi) + \frac{1}{-\pi\frac{\pi}{2}\frac{\pi}{2}} x(x+\tfrac{\pi}{2})(x-\pi) \\ &= \frac{4}{3\pi^3} x(x-\tfrac{\pi}{2})(x-\pi) - \frac{4}{\pi^3} x(x+\tfrac{\pi}{2})(x-\pi) = -\frac{8}{3\pi^3}x^3 + \frac{8}{3\pi}x. \end{aligned}$$

Using the symbolic Yacas system, we define a symbolic variable `x` and get (note that $f(x_2) = 0$ and $f(x_4) = 0$):

```
#install.packages("Ryacas")
# Yacas = Yet Another Computer Algebra System
# Ryacas is an R interface to the 'Yacas' Computer Algebra System
library(Ryacas)
x <- ysym("x")     # treat x as a symbolic instead of numeric variable
x1 <- -pi/2
x2 <- 0
x3 <- pi/2
x4 <- pi
f1 <- -1
f3 <- 1
# below f is computed using x1-x4 and f1&f3 as numerical values
# f is an yacas object
f=(x-x2)*(x-x3)*(x-x4)/((x1-x2)*(x1-x3)*(x1-x4))*f1+
  (x-x1)*(x-x2)*(x-x4)/((x3-x1)*(x3-x2)*(x3-x4))*f3
# execute the command Expand(f) in yacas
fexpand<-y_fn(f,"Expand")
# execute the command Coef(fexpand,x,0 .. 3) in yacas
# to obtain the coefficients of x^0,...,x^3 (in this order)
coef<-y_fn(fexpand,"Coef","x","0 .. 3")
# N(coef,6) give the coefficient in 6 digits
# as_r() convert yacas object into R object
coefNum<-as_r(y_fn(coef,"N","6"))
# show coefficients using 6 digits in reverse order
# (to make compatible with output polyfit!)
formatC(coefNum[4:1],digits=6,format="f")
```

```
## [1] "-0.086004" "-0.000000" "0.848826"  "0.000000"
```

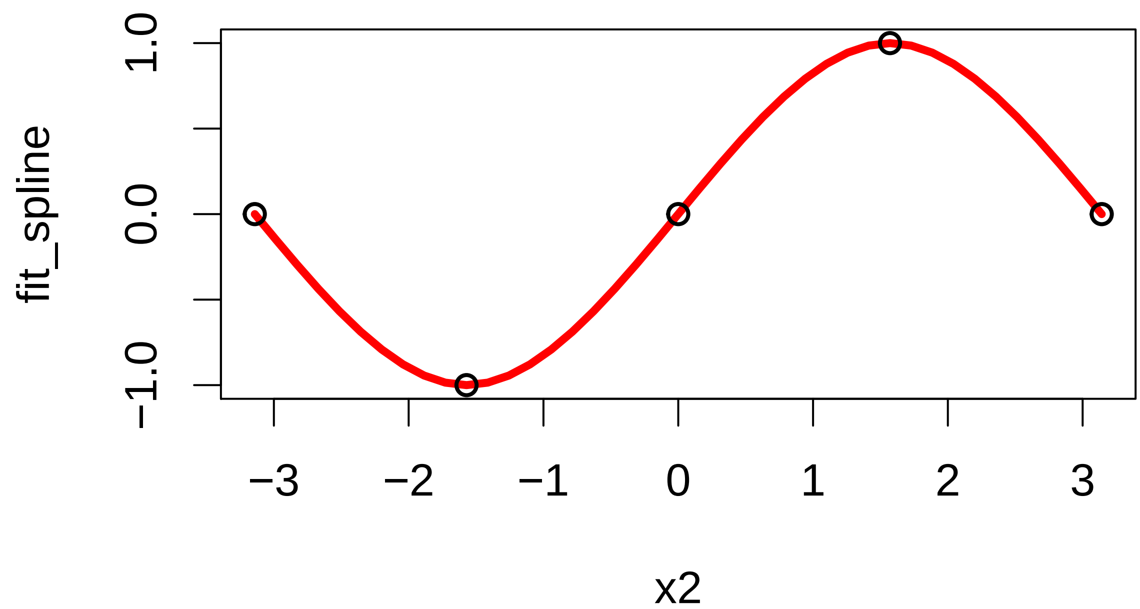
This corresponds to the output of the function `polyfit`:

```
x <- seq(-pi/2,pi,pi/2)
f <- c(-1,0,1,0)
coef <- polyfit(x,f,3)
formatC(coef,format="f",digits=6)
```

```
## [1] "-0.086004" "0.000000"  "0.848826"  "-0.000000"
```

Question 4

```
x <- seq(-pi,pi,pi/2)
f <- c(0,-1,0,1,0)
pp <- cubicspline(x, f)
x2 <- seq(-pi,pi,pi/20)
fit_spline <- ppval(pp,x2)
par(cex=1.4)
plot(x2,fit_spline,type="l",lty=1,lwd=4,col="red")
points(x,f,lwd=2)
```



Question 5

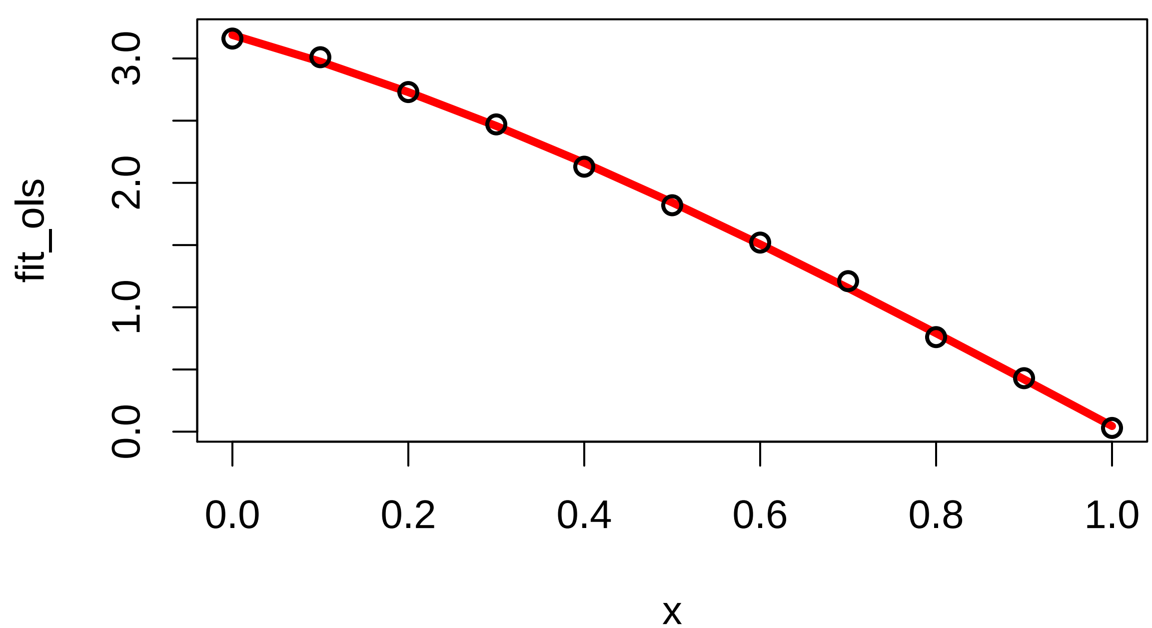
```
rm(list = ls())
data <- read.table("C:/Surfdrive/PNA/Week6/PC6a/YX.txt",header=TRUE)
attach(data)

SSR <- function(coef,y,x){
  fit <- coef[1]*sin(x)+coef[2]*cos(x)
  residuals <- y-fit
  return(sum(residuals^2))
}

results <- optim(c(0,0),SSR,gr=NULL,y,x,method = c("BFGS"))
coef <- results$par
cat(sprintf("a=%10.4f  b=%10.4f\n",coef[1],coef[2]))
```

```
## a=    -1.9941  b=     3.1892
```

```
fit_ols <- coef[1]*sin(x)+coef[2]*cos(x)
par(cex=1.25)
plot(x,fit_ols,type="l",lty=1,lwd=4,col="red")
points(x,y,lwd=2)
```




```
detach(data)
```

Question 6a

```
x <- seq(-pi,pi,pi/2)
f <- c(0,-1,0,1,0)
pp <- cubicspline(x, f)
x2 <- seq(-pi,pi,pi/10)
f_spline <- ppval(pp,x2)
f_true <- sin(x2)
SSR <- sum((f_true-f_spline)^2)
cat(sprintf("SSR = %10.4f\n",SSR))
```

```
## SSR =     0.0036
```

Question 6b

```r
# the coefficients of the splines
pp$coefs
```

```
##            [,1]       [,2]          [,3] [,4]
## [1,]  0.1290061  0.0000000 -9.549297e-01    0
## [2,] -0.1290061  0.6079271  1.110223e-16   -1
## [3,] -0.1290061  0.0000000  9.549297e-01    0
## [4,]  0.1290061 -0.6079271 -1.110223e-16    1
```

```r
# now in 4 decimals
formatC(pp$coefs,format="f",digits=4)
```

```
##      [,1]      [,2]      [,3]      [,4]
## [1,] "0.1290"  "0.0000"  "-0.9549" "0.0000"
## [2,] "-0.1290" "0.6079"  "0.0000"  "-1.0000"
## [3,] "-0.1290" "0.0000"  "0.9549"  "0.0000"
## [4,] "0.1290"  "-0.6079" "-0.0000" "1.0000"
```

Question 6c

The spline left from 0 is given by:

$$s_2(x) = -0.1290(x - \frac{-\pi}{2})^3 + 0.6079(x - \frac{-\pi}{2})^2 + -1$$

The spline right from 0 is given by:

$$s_3(x) = -0.1290(x - 0)^3 + 0.9549(x - 0)$$

Derivative of the spline left from 0:

$s_2'(x) = -0.1290 \cdot 3 \cdot (x - \frac{-\pi}{2})^2 + 0.6079 \cdot 2 \cdot (x - \frac{-\pi}{2}) = -0.387(x - \frac{-\pi}{2})^2 + 1.2158(x - \frac{-\pi}{2})$
$s_2'(0) = -0.387(0 - \frac{-\pi}{2})^2 + 1.2158(0 - \frac{-\pi}{2}) = 0.9549$

Derivative of the spline right from 0:

$$s_3'(x) = -0.1290 \cdot 3 \cdot (x - 0)^2 + 0.9549 = -0.387x^2 + 0.9549$$
$$s_3'(0) = 0.9549$$

Second-order derivative of the spline left from 0:

$$s_2''(x) = -0.387 \cdot 2 \cdot (x - \tfrac{-\pi}{2}) + 1.2158$$
$$s_2''(0) = -0.387 \cdot 2 \cdot (0 - \tfrac{-\pi}{2}) + 1.2158 = 3.6431 \cdot 10^{-6}$$

Second-order derivative of the spline right from 0:

$$s_3''(x) = -0.387 \cdot 2 \cdot x$$
$$s_3''(0) = 0$$