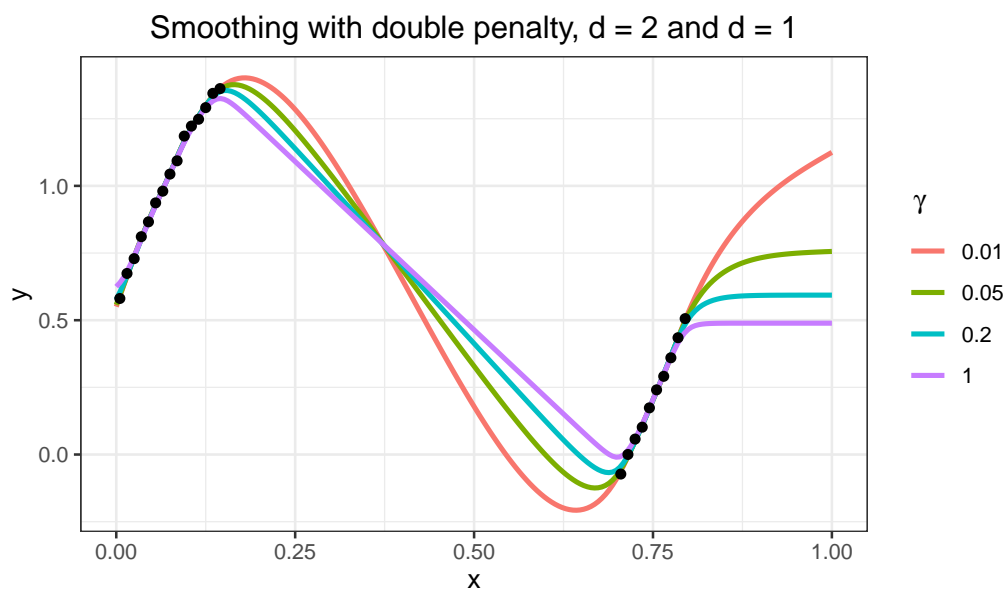


Interpolation with double penalization using first and second orders



Combining first and second order penalties to get tight interpolation and extrapolation (100 cubic B-spline segments, $\lambda = 1$). R code in `f-double-pen.R`

```
# Interpolation with double penalization (first and second order)
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019
```

```
library(ggplot2)
library(JOPS)
```

```
# Simulate data
m0 = 100
x = ((1:m0) - 0.5) / m0
set.seed(911)
y = 0.9 * sin(2.5 * pi * x) - 0.45 + rnorm(m0) * 0.01 + 1
sel = x < 0.15 | (x > 0.7 & x < 0.8)
x = x[sel]
y = y[sel]
m = length(x)
```

```
# Prepare basis and penalty matrices
nseg = 100
B = bbase(x, xl = 0, xr = 1, nseg = nseg)
n = ncol(B)
D1 = diff(diag(n))
D2 = diff(D1)
ng = 200
xg = seq(0, 1, length = ng)
Bg = bbase(xg, xl = 0, xr = 1, nseg = nseg)
```

```
# Do the smoothing
lambda = 1
gams = c(0.01, 0.05, 0.2, 1)
nc = length(gams)
Z = NULL
for (gam in gams) {
  P = lambda * t(D2) %>% D2 + gam * sqrt(lambda) * t(D1) %>% D1
  a = solve(t(B) %>% B + P, t(B) %>% y)
```

```
z = Bg %**% a
Z = cbind(Z, z)
}

# Make data frames for ggplot
Dat = data.frame(x = x, y = y)
Fz = data.frame(x = rep(xg, nc), y = as.vector(Z),
               gamma = as.factor(rep(gams, each = ng)))

# Make the plot
plt = ggplot() +
  geom_line(aes(x = x, y = y, color = gamma), data = Fz, size = 1) +
  geom_point(aes(x = x, y = y), data = Dat) +
  ggtitle('Smoothing with double penalty, d = 2 and d = 1') +
  scale_color_discrete(name = bquote(~ gamma)) +
  JOPS_theme()

# Save graph
print(plt)
```
