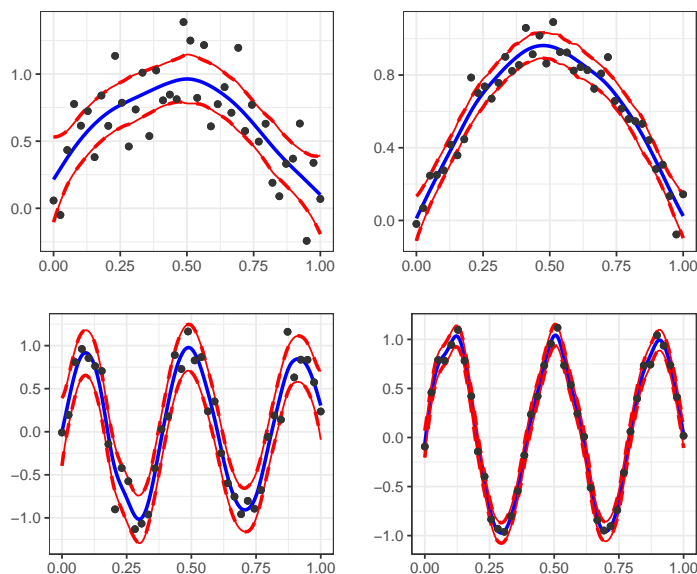


Illustration of Bayesian P-splines



Smoothing of simulated data with Bayesian P-splines (20 cubic segments, second order penalty) for different combination of smoothness and noise level. The number of Markov steps was 1000. Blue curves: exact values; red curves: plus and minus two standard deviations. R code in `f-bayes-show.R`

```
# Illustration of Bayesian P-splines
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

library(ggplot2)
library(gridExtra)
library(JOPS)

# Simulation parameters
m = 40
set.seed(23)
x = seq(0, 1, length = m)
frq = c(0.5, 0.5, 2.5, 2.5)
nse = c(0.3, 0.1, 0.3, 0.1)

# Bspline parameters
nseg = 50
B = bbase(x, 0, 1, nseg, 3)
n = ncol(B)

# Roughness penalty
E = diag(n)
d = 2
D = diff(E, diff = d)
P = t(D) %*% D

ndraw = 1000
V0 = V1 = matrix(0, ndraw, 4)

plts = list()
for (sim in 1:4) {
  t0 = Sys.time()

  # Simulate data
```

```

y = sin(2 * pi * frq[sim] * x) + rnorm(m) * nse[sim]

# Initialize
sig2 = 0.1
tau2 = 1
BB = t(B) %>% B
By = t(B) %>% y
yy = t(y) %>% y
A = matrix(0, n, ndraw)
# Run Markov chain
for (it in 1:ndraw) {

  # Update coefficients
  U = BB/sig2 + P/tau2
  Ch = chol(U)
  a0 = solve(Ch, solve(t(Ch), By))/sig2
  a = solve(Ch, rnorm(n)) + a0
  A[, it] = a

  # Update and save error variance
  r2 = yy - 2 * t(a) %>% By + t(a) %>% BB %>% a
  sig2 = c(r2/rchisq(1, m))
  V0[it, sim] = sig2

  # Update and save roughness variance
  r = D %>% a
  tau2 = c(sum(r^2)/rchisq(1, n - d))
  V1[it, sim] = tau2

}

# Compute curve on grid
am = apply(A[, -(1:100)], 1, mean)
xg = seq(0, 1, length = 200)
Bg = bbase(xg, 0, 1, nseg, 3)
mu = Bg %>% am

# Variation in curves
Mu = Bg %>% A
s = apply(Mu, 1, sd)
t1 = Sys.time() - t0
cat(t1, "\n")

# Plot data and curve
Data = data.frame(x = x, y = y)
Dfit = data.frame(x = xg, mu = mu, lo = mu - 2 * s, hi = mu + 2 * s)
plt1 = ggplot(Data, aes(x = x, y = y)) +
  geom_point(aes(x = x, y = y), size = 1.5, color = grey(0.20)) +
  geom_line(data = Dfit, aes(x = x, y = mu), size = 1, color = 'blue') +
  geom_line(data = Dfit, aes(x = x, y = lo), size = 1, color = 'red', linetype = 2) +
  geom_line(data = Dfit, aes(x = x, y = hi), size = 1, color = 'red', linetype = 2) +
  geom_line(data = Dfit, aes(x = x, y = lo), size = 0.5, color = 'red') +
  geom_line(data = Dfit, aes(x = x, y = hi), size = 0.5, color = 'red') +
  geom_point(aes(x = x, y = y), size = 1.5, color = grey(0.20)) +
  xlab('') + ylab('') +
  JOPS_theme()
plts[[sim]] = plt1
}

# Make and save plots
grid.arrange(grobs = plts, ncol = 2, nrow = 2)

```
