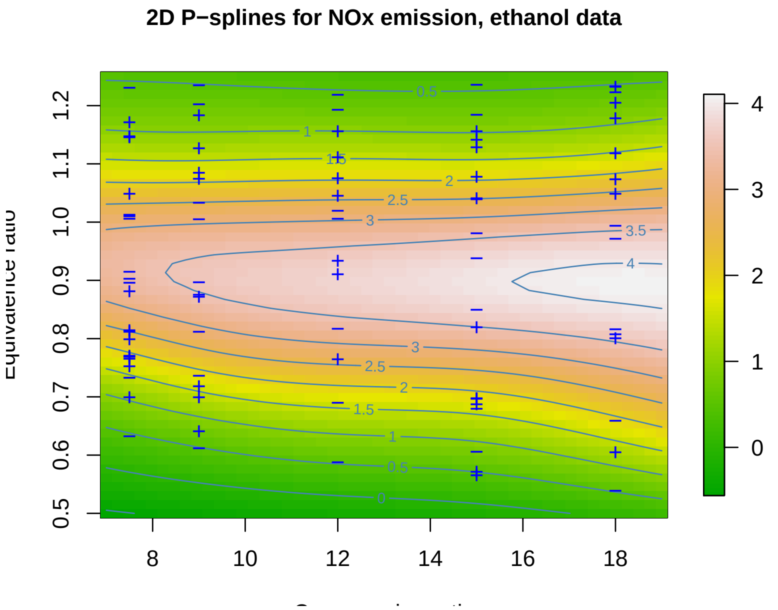
Tensor product P-spline fit (Ethanol data)



The blue + and - symbols indicate the positions of the data points and the signs of the residuals. Each basis has 10 segments and a second order penalty, with $\lambda = 1$ (for the C axis) and $\breve{\lambda} = 0.1$ (for the E axis). R code in `f-ethanol-psp2.R`

```
# Tensor product P-spline fit (Ethanol data)
# A graph in the book ’Practical Smoothing. The Joys of P-splines’
# Paul Eilers and Brian Marx, 2019

library(SemiPar)
library(fields)
library(spam)
library(JOPS)

# Get the data
data(ethanol)
m = nrow(ethanol)
x = ethanol$C
y = ethanol$E
z = ethanol$NOx

# Set parameters for domain
xlo = 7
xhi = 19
ylo = 0.5
yhi = 1.25

# Set P-spline parameters, fit and compute surface
xseg = 10
xdeg = 3
xpars = c(xlo, xhi, xseg, xdeg)
yseg = 10
ydeg = 3
ypars = c(ylo, yhi, yseg, ydeg)

# Compute one-dimensional bases
Bx = bbase(x, xpars[1], xpars[2], xpars[3], xpars[4])
By = bbase(y, ypars[1], ypars[2], ypars[3], ypars[4])
nx = ncol(Bx)
```

```
ny = ncol(By)

# Compute tensor products
B1 = kronecker(t(rep(1, ny)), Bx)
B2 = kronecker(By, t(rep(1, nx)))
B = B1 * B2
n = ncol(B)

# Compute penalty matrices
Dx = diff(diag(nx), diff = 2)
Dy = diff(diag(ny), diff = 2)
delta = 1e-10
Px = kronecker(diag(ny), t(Dx) %*% Dx + delta * diag(nx))
Py = kronecker(t(Dy) %*% Dy + delta * diag(ny), diag(nx))
E = diag(n)

# Fit the model
lambdax = 1
lambday = 0.1
a = solve(t(B) %*% B + lambdax * Px + lambday * Py, t(B) %*% z)
zhat = B %*% a
r = z - zhat
cat("SD of residuals:", sd(r), "\n")

# Compute grid for predicted surface
nu = 50
nv = 50
u = seq(xlo, xhi, length = nu)
v = seq(ylo, yhi, length = nv)
Bgx = bbase(u, xpars[1], xpars[2], xpars[3], xpars[4])
Bgy = bbase(v, ypars[1], ypars[2], ypars[3], ypars[4])
A = matrix(a, nx, ny)
Fit = Bgx %*% A %*% t(Bgy)

# Plot result and data
cols = c("blue", "red")[(z > zhat) + 1]
pchs = c("+", "-")[(z > zhat) + 1]
image.plot(u, v, Fit, col = terrain.colors(100), xlab = "Compression ratio",
    ylab = "Equivalence ratio")
contour(u, v, Fit, add = T, col = "steelblue", labcex = 0.7)
points(x, y, pch = pchs, col = "blue", cex = 1.1, )
title("2D P-splines for NOx emission, ethanol data", cex.main = 1)
```