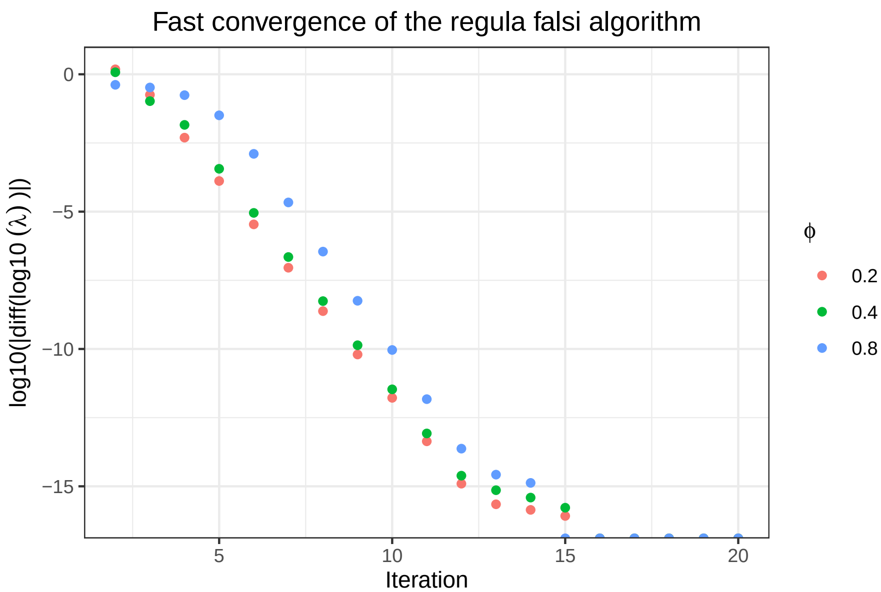
Correcting low convergence of Schall algorithm with regula falsi



Improved speed of convergence of the HFS (Harville-Fellner-Schall) algorithm when combined with the *regula falsi* algorithm for root finding. Three different values of the standard deviation (ϕ) of the noise in the simulation are presented. `R` code in `f-regula-falsi.R`

```
# Correcting low convergence of the Harville-Fellner-Schall algorithm with the regula falsi
# A graph in the book 'Practical Smoothing. The Joys of P-splines'
# Paul Eilers and Brian Marx, 2019

library(JOPS)
library(ggplot2)

# Simulate data
m = 100
set.seed(2013)
x = seq(0, 1, length = m)
r = rnorm(m)

# P-spline preparations
xmin = 0
xmax = 1
nseg = 20
B = bbase(x, xmin, xmax, nseg)
n = ncol(B)
d = 2
D = diff(diag(n), diff = d)
P = t(D) %*% D
BtB = t(B) %*% B

L = Las = NULL
nit = 20
Yhat = Y = NULL
phis = c(0.2, 0.4, 0.8)
for (phi in phis) {
  y = sin(10 * x) + r * phi
  Bty = t(B) %*% y

  lla1 = -2
  lla2 = 3
```

```
  las = NULL

  a = solve(BtB + 10 ^ lla1 * P, Bty)
  G = solve(BtB + 10 ^ lla1 * P, BtB)
  ed = sum(diag(G))
  yhat = B %*% a
  sig2 = sum((y - yhat) ^ 2) / (m - ed -2)
  tau2 = sum((D %*% a) ^ 2) / ed
  u1 = log10(tau2) + lla1 - log10(sig2)

  a = solve(BtB + 10 ^ lla2 * P, Bty)
  G = solve(BtB + 10 ^ lla2 * P, BtB)
  ed = sum(diag(G))
  yhat = B %*% a
  sig2 = sum((y - yhat) ^ 2) / (m - ed -2)
  tau2 = sum((D %*% a) ^ 2) / ed
  u2 = log10(tau2) + lla2 - log10(sig2)

  for (it in 1:nit) {
    dla = (0 - u1) * (lla2 - lla1) / (u2 - u1)
    lla = lla1 + dla
    a = solve(BtB + 10 ^ lla * P, Bty)
    G = solve(BtB + 10 ^ lla * P, BtB)
    ed = sum(diag(G))
    yhat = B %*% a
    sig2 = sum((y - yhat) ^ 2) / (m - ed - 2)
    tau2 = sum((D %*% a) ^ 2) / ed
    u = log10(tau2) + lla - log10(sig2)
    if (u * u1 > 0) {
      lla1 = lla
      u1 = u
    }  else {
      lla2 = lla
      u2 = u
    }
    las = c(las, 10 ^ lla)
  }
  # Compute convergence rate
  dla =  log10(abs(diff(log10(las))))
  Lsub = data.frame(it = 2:nit, phi = as.factor(phi), dla = dla)
  L = rbind(L, Lsub)

  # Save data and fit for plotting
  Yhat = cbind(Yhat, yhat)
  Y = cbind(Y, y)
  Las = cbind(Las, las)
}

plt2 = ggplot(L, aes(x = it, y = dla, group = phi)) +
  geom_point(aes(colour = phi)) +
  xlab("Iteration") +
  ylab(expression(paste("log10(|diff(log10"~(lambda) ~")|)"))) +
  labs(color = bquote(phi)) +
  ggtitle('Fast convergence of the regula falsi algorithm') +
  JOPS_theme()
print(plt2)

# Plot and save
print(plt2)
```