Below is the R script Forrest wrote on Nov., 2012. We were able to replicate the PLSR and generate the same regressor scores as those in SAS. However, we could not replicate the PROCTPSPLINE package in SAS. I think Forrest seemed to think this had something to do with the ?ordering of the basis functions? or some such. I am sure I am misquoting him here.

####Read in geology\_basic data set####

#geo = read.table("C:/Users/Forrest\_Williamson/Dropbox/SoilGeochemProj/R stuff/analysis/geology\_basic.csv",header=T,sep=",")

geo = read.table("C:/Users/gstinchcomb/Dropbox/MurrayStateUniversity/Publications/StinchEtAl\_PPM2.0/geology\_basic.csv",header=T,sep=",")

geo = read.table("C:/Users/Gary/Dropbox/MurrayStateUniversity/Publications/StinchEtAl\_PPM2.0/geology\_basic.csv",header=T,sep=",")

subgeo = geo[,c(1,2,14,15,16,17)]

####Set up data frames####

oxide = as.matrix(geo[,3:13])

response = as.matrix(geo[,1:2])

MAP = as.matrix(geo[,1])

MAT = as.matrix(geo[,2])

factors = as.matrix(geo[,14:17])

Geo = model.frame(response ~ oxide)

####PLS using NIPALS, Combined Response####

require(pls)

fit.pls = plsr(response ~ oxide, ncomp = 11, data = geo, method = "oscorespls", validation = 'CV')

plot(RMSEP(fit.pls), legendpos = "topright")

fit.pls = plsr(response ~ oxide, ncomp = 4, data = geo, method = "oscorespls", validation = 'CV')

comps = oxide%\*%as.matrix(fit.pls$projection)

Geo = model.frame(response ~ oxide + comps + factors)

###Correlations loading plot###

windows(); par(mfrow=c(1,2))

corrplot(fit.pls, comps = 1:2, labels="names", radii = c(sqrt(1/2), 1),

identify = FALSE, type = "p")

corrplot(fit.pls, comps = 3:4, labels="names", radii = c(sqrt(1/2), 1),

identify = FALSE, type = "p")

####PLS using NIPALS, MAP reponse####

require(pls)

fit.pls = plsr(MAP ~ oxide, ncomp = 11, data = geo, method = "oscorespls", validation = 'CV')

plot(RMSEP(fit.pls), legendpos = "topright")

fit.pls = plsr(MAP ~ oxide, ncomp = 4, data = geo, method = "oscorespls", validation = 'CV')

comps = oxide%\*%as.matrix(fit.pls$projection)

Geo = model.frame(MAP ~ oxide + comps + factors)

#Spline

require(mgcv)

b <- gam((MAP~s(factors,k=15, bs="tp")),family=mvn(d=2),data=Geo)

fit<-s(method="REML")

require(fields)

Tps(factors, response)

require(mgcv); n <- 100; set.seed(2)

x <- runif(n); y <- x + x^2\*.2 + rnorm(n) \*.1

## is smooth significantly different from straight line?

summary(gam(y~s(x,m=c(2,0))+x,method="REML")) ## not quite

## is smooth significatly different from zero?

summary(gam(y~s(x),method="REML")) ## yes!

## Fool bam(...,discrete=TRUE) into (strange) nested

## model fit...

set.seed(2) ## simulate some data...

dat <- gamSim(1,n=400,dist="normal",scale=2)

dat$x1a <- dat$x1 ## copy x1 so bam allows 2 copies of x1

## Following removes identifiability problem, by removing

## linear terms from second smooth, and then re-inserting

## the one that was not a duplicate (x2)...

b <- bam(y~s(x0,x1)+s(x1a,x2,m=c(2,0))+x2,data=dat,discrete=TRUE)

#Spline - earth

require(fields)

Y = scale(response)

fit.spline = Tps(factors, Y)

rmse = sqrt(apply(fit.spline$residuals^2,2,mean))

windows(); par(mfrow=c(1,2))

boxplot(fit.spline$residuals[,1],main="MAP residuals \n earth",xlab=bquote("RMSE = "~.(rmse[1])))

boxplot(fit.spline$residuals[,2],main="MAT residuals \n earth",xlab=bquote("RMSE = "~.(rmse[2])))

#Spline = mda

require(mda)

fit.mda = mars(comps,response)

plotmo(fit.mda)

rmse.mda = sqrt(apply(fit.mda$residuals\*\*2,2,mean))

windows(); par(mfrow=c(1,2))

boxplot(fit.mda$residuals[,1],main="MAP residuals \n MARS",xlab=bquote("RMSE = "~.(rmse.mda[1])))

boxplot(fit.mda$residuals[,2],main="MAT residuals \n MARS",xlab=bquote("RMSE = "~.(rmse.mda[2])))

#Spline = mda non-dataframe

require(mda)

fit.mda2 = mars(subgeo[,c(3,4,5,6)],subgeo[,c(1,2)])

plotmo(fit.mda2, nresponse=1)

plotmo(fit.mda2, nresponse=2)

rmse.mda2 = sqrt(apply(fit.mda2$residuals\*\*2,2,mean))

windows(); par(mfrow=c(1,2))

boxplot(fit.mda2$residuals[,1],main="MAP residuals \n MARS",xlab=bquote("RMSE = "~.(rmse.mda2[1])))

boxplot(fit.mda2$residuals[,2],main="MAT residuals \n MARS",xlab=bquote("RMSE = "~.(rmse.mda2[2])))

library(mgcv)

## simulate some data...

V <- matrix(c(2,1,1,2),2,2)

f0 <- function(x) 2 \* sin(pi \* x)

f1 <- function(x) exp(2 \* x)

f2 <- function(x) 0.2 \* x^11 \* (10 \* (1 - x))^6 + 10 \*

(10 \* x)^3 \* (1 - x)^10

n <- 300

x0 <- runif(n);x1 <- runif(n);

x2 <- runif(n);x3 <- runif(n)

y <- matrix(0,n,2)

for (i in 1:n) {

mu <- c(f0(x0[i])+f1(x1[i]),f2(x2[i]))

y[i,] <- rmvn(1,mu,V)

}

dat <- data.frame(y0=y[,1],y1=y[,2],x0=x0,x1=x1,x2=x2,x3=x3)

## fit model...

b <- gam(list(y0~s(x0)+s(x1),y1~s(x2)+s(x3)),family=mvn(d=2),data=dat)

b

summary(b)

plot(b,pages=1)

solve(crossprod(b$family$data$R)) ## estimated cov matrix