Supplementary Materials for
Modeling Variability in Treatment Effects for Cluster Randomized Controlled Trials using By-Variable Smooth Functions in a Generalized Additive Mixed Model

Sun-Joo Cho, Vanderbilt University
Kristopher J. Preacher, Vanderbilt University
Haley E. Yaremych, Vanderbilt University
Matthew Naveiras, Vanderbilt University
Douglas Fuchs, Vanderbilt University
Lynn S. Fuchs, Vanderbilt University

April 26, 2023

Accepted for publication in *Behavior Research Methods*

Appendix S1.

Implementations for By-Variable Smooth Functions and Random Effects in the mgcv Package

When model complexity for by-variable smooth functions regarding the number of clusters J is understood, it is informative to specify the sizes of matrices for basis coefficient parameters ($\boldsymbol{\delta}$) (multiplied by known basis functions $\mathbf{b}$) and for smoothing parameters ($\boldsymbol{\lambda}$) (controlling the trade-off between goodness of fit and smoothness of a smooth function). The sizes of matrices can be specified with respect to the number of data points (N), the number of clusters (J), and the number of basis functions (K). In the by-variable smooth functions, the size of $\boldsymbol{\delta}$ with a $N \times [J(K-1)]$ $\mathbf{b}$ is $[J(K-1)] \times 1$, where $K-1$ is imposed instead of K because of a sum-to-zero constraint on the smooth function. The size of $\boldsymbol{\lambda}$ with a $[J(K-1) \times J(K-1)]$ $\mathbf{S}$ is $J \times 1$ because each cluster has a different amount of wiggliness.

Various kinds of smoothers can be selected in the mgcv package. Below, we explain specifications of smoothers for by-variable smooth functions ($f_1(x_{.j})z_j$, $f_2(x_{ij} - x_{.j})z_j$, and $f_3(x_{ij} - x_{.j})Cluster_j$) and a random intercept (u_{0j}) in the mgcv package.

For the by-variable smooth function of a continuous covariate ($x_{ij} - x_{.j}$) with multiple levels of a factor ($f_3(x_{ij} - x_{.j})Cluster_j$), a by-variable smoother can be considered. The by-variable smoother for $f_3(x_{ij} - x_{.j})Cluster_j$ allows varying functional relationships and wiggliness between ($x_{ij} - x_{.j}$) and y_{ij} across clusters. The smoother for ($x_{ij} - x_{.j}$) (denoted by `s(x1)`) can be generated for *each* level of the factor variable `cluster` indicated by the `by`-argument, and can be specified as follows:

```
cluster <- as.factor(cluster)
s(x1,by=cluster,bs="tp",k=,m=2)
```

The by-variable smoother is centered with a sum-to-zero constraint. With this constraint, cluster-specific intercepts are not shifted so that the random intercept (u_{0j}) can be estimated.

For the by-variable smooth function of a continuous covariate ($x_{.j}$ and $x_{ij} - x_{.j}$) with the two levels of a factor z_j ($f_1(x_{.j})z_j$ and $f_2(x_{ij} - x_{.j})z_j$), a by-variable smoother can also be chosen. Taking a binary (dummy-coded) TRT factor variable in R (coded as 1 for a treatment group; 0

for a control group), the by-variable smoother for $f_1(x_{.j})z_j$ is specified as follows:

```
f.TRT <- as.factor(TRT)
```

With `f.TRT`, the mean effect of `f.TRT` and the smoother for $x_{.j}$ (denoted by `s(x2)`) can be generated for *each* level of the factor variable TRT indicated by the `by`-argument:

```
f.TRT + s(x2,by=f.TRT,bs="tp",k=,m=2)
```

In a similar way, a smooth function for $f_2(x_{ij} - x_{.j})z_j$ can be specified as follows:

```
f.TRT + s(x1,by=f.TRT,bs="tp",k=,m=2)
```

Because of the relation between a smooth function in GAMM and a random effect, the random intercept (u_{0j}) can be estimated as basis coefficients using a basis `re` in the `mgcv` package:

```
s(cluster,bs="re",k=)
```

Appendix S2.

`R` Code Used for the Empirical Study

The `mgcv` session information is as follows:

```
> library(mgcv)
> sessionInfo(package = "mgcv")
R version 3.6.0 (2019-04-26)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19043)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
character(0)

other attached packages:
[1] mgcv_1.8-35

loaded via a namespace (and not attached):
 [1] viridis_0.6.1          jsonlite_1.7.2         viridisLite_0.4.0
 [4] splines_3.6.0          foreach_1.5.1          carData_3.0-4
 [7] shiny_1.6.0            statmod_1.4.36         grDevices_3.6.0
[10] cellranger_1.1.0       pillar_1.6.1           lattice_0.20-38
[13] glue_1.4.2             base_3.6.0             digest_0.6.27
[16] manipulateWidget_0.11.0 RColorBrewer_1.1-2     qgam_1.3.3
[19] promises_1.2.0.1       minqa_1.2.4            colorspace_2.0-1
[22] htmltools_0.5.1.1      httpuv_1.6.1           Matrix_1.2-17
[25] plyr_1.8.6             pkgconfig_2.0.3        haven_2.4.1
[28] purrr_0.3.4            xtable_1.8-4           scales_1.1.1
[31] openxlsx_4.2.3         later_1.2.0            rio_0.5.27
[34] lme4_1.1-26            tibble_3.1.1           generics_0.1.0
[37] farver_2.1.0           datasets_3.6.0         car_3.0-11
[40] ggplot2_3.3.4          ellipsis_0.3.2         cachem_1.0.4
[43] withr_2.4.2            readxl_1.3.1           magrittr_2.0.1
[46] crayon_1.4.1           mime_0.10              memoise_2.0.0
[49] methods_3.6.0          GGally_2.1.2           fs_1.5.0
[52] fansi_0.4.2            doParallel_1.0.16      nlme_3.1-139
[55] MASS_7.3-51.4          forcats_0.5.1          foreign_0.8-71
[58] utils_3.6.0            data.table_1.14.0      tools_3.6.0
[61] hms_1.1.0              lifecycle_1.0.0        matrixStats_0.58.0
[64] gamm4_0.2-6            munsell_0.5.0          zip_2.1.1
[67] stats_3.6.0            compiler_3.6.0         pkgdown_1.6.1
[70] rlang_0.4.11           grid_3.6.0             nloptr_1.2.2.2
[73] iterators_1.0.13       graphics_3.6.0         htmlwidgets_1.5.3
[76] crosstalk_1.1.1        miniUI_0.1.1.1         labeling_0.4.2
[79] boot_1.3-22            gtable_0.3.0           codetools_0.2-16
[82] abind_1.4-5            reshape_0.8.8          curl_4.3.1
[85] R6_2.5.0               gridExtra_2.3          mgcViz_0.1.6
[88] knitr_1.33             dplyr_1.0.6            fastmap_1.1.0
[91] utf8_1.2.1             KernSmooth_2.23-15     stringi_1.6.1
[94] parallel_3.6.0         Rcpp_1.0.6             vctrs_0.3.8
[97] rgl_0.106.8            tidyselect_1.1.1       xfun_0.22
```

```
library(nlme) #for the "lme" function
library(mgcv) #for the "gam" and "gamm" functions
library(mgcViz) #for figures of smooth functions
library(ggplot2) #for figures
library(itsadug)  #for the ‘‘find_difference’’ function
library(psych) #for calculating quantiles
library(car) #for QQplot of residuals


#################################################################
##Step 1: Unconditional GAMM and Exploratory Graphical Analysis##
#################################################################

#data importing
data <- read.table("C:/Banarov_hamd.txt", header=TRUE)

data$uc <- as.factor(data$uc) #cluster
data$group <- as.factor(data$group) #treatment vs. control
data$group2 <- as.factor(data$group2) #group_cluster id for figures

#unconditional GAMM
gamm.uncon <- gam(hamd_6m ~ 1 + s(uc,bs="re",k=5), data=data,method="REML")
gam.vcomp(gamm.uncon)

#Figure 2
fit.cluster <- ggplot(data, aes(x = x1, y = hamd_6m)) +
      geom_point() +
      geom_smooth(method="loess",se=FALSE, col="red") +
      geom_smooth(method=lm, se=FALSE, col="blue", linetype="dotted") +
      facet_wrap(facets = vars(group2)) +
              xlab("Cluster-Mean Centered Baseline HDR") +
              ylab("6-Month Follow-Up HDR") +
              theme_bw() +
              theme(legend.direction = "horizontal", legend.position = "bottom", legend.key = element_blank(),
      legend.background = element_rect(fill = "white", colour = "gray30")) +
              guides(fill = guide_legend(keywidth = 1, keyheight = 1), linetype=guide_legend(keywidth = 3, keyheight = 1),
      colour=guide_legend(keywidth = 3, keyheight = 1))

fit.cluster + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))

#Figure 3
fit.L11 <- ggplot(data=data,
aes(x = x1, y = hamd_6m, shape=group, color=group)) +
geom_point(aes(colour = factor(group))) +
geom_smooth(aes(linetype="Smooth"), method="loess", se=T, fullrange=F) +
geom_smooth(aes(linetype="Linear"),method="lm", se=F, fullrange=F) +
xlab("Cluster-Mean Centered Baseline HDR") +
ylab("6-Month Follow-Up HDR") +
theme_bw() +
theme(legend.direction = "horizontal", legend.position = "bottom", legend.key = element_blank(),
        legend.background = element_rect(fill = "white", colour = "gray30")) +
guides(fill = guide_legend(keywidth = 1, keyheight = 1), linetype=guide_legend(keywidth = 3, keyheight = 1),
         colour=guide_legend(keywidth = 3, keyheight = 1))

fit.L11 + scale_color_brewer(palette="Dark2")


fit.L22 <- ggplot(data=data,
aes(x = x2, y = hamd_6m, shape=group, color=group)) +
geom_point(aes(colour = factor(group))) +
geom_smooth(aes(linetype="Smooth"), method="loess", se=T, fullrange=F) +
geom_smooth(aes(linetype="Linear"),method="lm", se=F, fullrange=F) +
xlab("Cluster-Means of Baseline HDR") +
ylab("6-Month Follow-Up HDR") +
theme_bw() +
theme(legend.direction = "horizontal", legend.position = "bottom", legend.key = element_blank(),
        legend.background = element_rect(fill = "white", colour = "gray30")) +
```

```
guides(fill = guide_legend(keywidth = 1, keyheight = 1), linetype=guide_legend(keywidth = 3, keyheight = 1),
         colour=guide_legend(keywidth = 3, keyheight = 1))

fit.L22 + scale_color_brewer(palette="Dark2")



#####################################################################
##Step 2: Adding Covariates (TRT and COV) to the Unconditional GAMM##
#####################################################################

gamm <- gam(hamd_6m ~ 1 + group + s(x2,by=group,bs="tp",k=5,m=2) + s(x1,by=group,bs="tp",k=5,m=2) +
s(x1,by=uc,bs="tp",k=5,m=2) + s(uc,bs="re",k=5), data=data,method="REML")
gam.check(gamm) #k-index
summary(gamm) #results of fixed effects and smooth functions
gam.vcomp(gamm) #results of variance components

#MLM was fit in Step 2 for the comparison with GAMM in Figure 4.
mlm <- gamm(hamd_6m ~ 1 + x1*group + x2*group, random=list(uc=~1+x1),data=data,method="REML")
summary(mlm$lme)

#Figure 4
fit <- ggplot(data, aes(x = hamd_baseline, y = hamd_6m) ) +
      geom_point() +
      geom_line(aes(y = fitted(mlm$lme), linetype="MLM"), size = 1, color="blue") +
      geom_line(aes(y = fitted(gamm), linetype="GAMM"), size = 1, color="red") +
      scale_linetype_manual(name="Model:", values=c("solid", "dotted"), breaks=c("GAMM", "MLM")) +
      facet_wrap(facets = vars(group2)) +
xlab("Baseline HDR") +
ylab("6-Month Follow-Up HDR") +
theme_bw() +
theme(legend.direction = "horizontal", legend.position = "bottom", legend.key = element_blank(),
      legend.background = element_rect(fill = "white", colour = "gray30")) +
guides(fill = guide_legend(keywidth = 1, keyheight = 1), linetype=guide_legend(keywidth = 3, keyheight = 1),
      colour=guide_legend(keywidth = 3, keyheight = 1))

fit + scale_color_brewer(palette="Dark2")
# Use grey scale
fit + scale_color_grey()
fit + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
panel.background = element_blank(), axis.line = element_line(colour = "black"))


################################
##Step 3: Interpreting Results##
################################

#Figure 5(a)
quantile(data$x2,probs = c(0.1,0.25,0.5,0.75,0.9))

x2.1 <- 12.68182
x2.2 <- 13.71429
x2.3 <- 14.40000
x2.4 <- 15.47619
x2.5 <- 16.78947

gamm.gamma00 <- 8.5934
gamm.gamma02 <-  -4.1727

newdata.z0 <- data.frame(z=c(0,0,0,0,0),group=c(0,0,0,0,0),uc=c(1,1,1,1,1),x1=c(0,0,0,0,0),x2=c(x2.1,x2.2,x2.3,x2.4,x2.5))

newdata.z1 <- data.frame(z=c(1,1,1,1,1),group=c(1,1,1,1,1),uc=c(1,1,1,1,1),x1=c(0,0,0,0,0),x2=c(x2.1,x2.2,x2.3,x2.4,x2.5))

predict.z0 <- predict(gamm,newdata.z0,type="iterms",se=TRUE)
predict.z1 <- predict(gamm,newdata.z1,type="iterms",se=TRUE)
predict.z0$fit[,2]
predict.z1$fit[,3]
```

```
gamm.1.y.z0 <- gamm.gamma00 + predict.z0$fit[1,2]
gamm.1.y.z1 <- gamm.gamma00 + gamm.gamma02 + predict.z1$fit[1,3]

gamm.2.y.z0 <- gamm.gamma00 + predict.z0$fit[2,2]
gamm.2.y.z1 <- gamm.gamma00 + gamm.gamma02 + predict.z1$fit[2,3]

gamm.3.y.z0 <- gamm.gamma00 + predict.z0$fit[3,2]
gamm.3.y.z1 <- gamm.gamma00 + gamm.gamma02 + predict.z1$fit[3,3]

gamm.4.y.z0 <- gamm.gamma00 + predict.z0$fit[4,2]
gamm.4.y.z1 <- gamm.gamma00 + gamm.gamma02 + predict.z1$fit[4,3]

gamm.5.y.z0 <- gamm.gamma00 + predict.z0$fit[5,2]
gamm.5.y.z1 <- gamm.gamma00 + gamm.gamma02 + predict.z1$fit[5,3]


z <- rbind("Control","Treatment","Control","Treatment","Control","Treatment","Control","Treatment","Control","Treatment")
x2 <- rbind("0.1","0.1","0.25","0.25","0.5","0.5","0.75","0.75","0.9","0.9")
gamm.y <- rbind(gamm.1.y.z0,gamm.1.y.z1,gamm.2.y.z0,gamm.2.y.z1,gamm.3.y.z0,
gamm.3.y.z1,gamm.4.y.z0,gamm.4.y.z1,gamm.5.y.z0,gamm.5.y.z1)
gamm.y <- round(gamm.y, digits = 3)
gamm.y.z <- data.frame(cbind(z,x2,gamm.y))
names(gamm.y.z) <- c("z","x.j","gamm.y")
gamm.y.z$gamm.y <- as.numeric(gamm.y)
gamm.y.z


g.y.z <- ggplot(gamm.y.z, aes(x=z, y=gamm.y, group=x.j, color=x.j)) +
         geom_line(aes(linetype=x.j), size = 1.5) +
         geom_point(aes(shape=x.j), size = 4) +
   ylab("Class-Means of 6-Month Follow-Up HDR") +
         scale_y_continuous(breaks=seq(3.0, 10.0, 1),limits=c(3.0, 10.0))

g.y.z + scale_color_brewer(palette="Dark2")
# Use grey scale
g.y.z + scale_color_grey()
g.y.z + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),panel.background = element_blank(),
axis.line = element_line(colour = "black"))


#Figure 5(b)
vcov <- as.matrix(vcov(gamm))
vcov

coef <- coef(gamm)
coef

b <- getViz(gamm)

plot(sm(b,1))
plot(sm(b,2))

x2.diff <- plotDiff(s1 = sm(b, 2), s2 = sm(b, 1)) + l_ciPoly() +
  l_fitLine() + geom_hline(yintercept = 0, linetype = 2)

x2.diff$data$fit

#95% CI
x2.dat <- data.frame(x2.diff$data$fit)
x2.mean <- x2.dat$y
x2.se <- x2.dat$se
x2.x <- x2.dat$x

find_difference(x2.mean,x2.se,x2.x)

#level2 simple effect=gamma01 + x2.diff vs. x2
x2.trt.mean <- x2.dat$y + coef[2]
x2.trt.se <- x2.se + sqrt(vcov[2,2])
```

```
lower <- x2.trt.mean - 1.96*x2.trt.se
upper <- x2.trt.mean + 1.96*x2.trt.se

x2.trt.data <- data.frame(x2.trt.mean,x2.trt.se,x2.x,lower,upper)

gamm.fit.x2 <- ggplot(x2.trt.data, aes(x2.x, y=x2.trt.mean)) + geom_line() +
      geom_ribbon(aes(ymin = x2.trt.mean - 1.96*x2.trt.se, ymax = x2.trt.mean + 1.96*x2.trt.se),
      linetype="dashed",alpha=0.2,fill="white",colour="black") +
      geom_hline(yintercept = 0,linetype="dashed") +
      geom_vline(xintercept = c(11.04762,16.80348), color="red", linetype = 3, size=1.5) +
      ylim(-10,2) +
xlab("Cluster-Means of Baseline HDR") +
ylab("6-Month Follow-Up HDR") +
theme_bw() +
theme(legend.direction = "horizontal", legend.position = "bottom", legend.key = element_blank(),
      legend.background = element_rect(fill = "white", colour = "gray30")) +
guides(fill = guide_legend(keywidth = 1, keyheight = 1), linetype=guide_legend(keywidth = 3, keyheight = 1),
      colour=guide_legend(keywidth = 3, keyheight = 1))

gamm.fit.x2 + scale_color_brewer(palette="Dark2")
# Use grey scale
gamm.fit.x2 + scale_color_grey()
gamm.fit.x2 + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),panel.background = element_blank(),
axis.line = element_line(colour = "black"))



####################################
##Comparisons between GAMM and MLM##
####################################

#RMSEI for GAMM
gamm.std.con <- data.frame(std.con=resid(gamm,type="p",level=1))
qqPlot(gamm.std.con[,1])
gamm.mean <- mean(gamm.std.con[,1]^2)
gamm.rmsei <- sqrt(gamm.mean)
gamm.rmsei

#RMSEI for MLM
mlm.std.con <- data.frame(std.con=resid(mlm$gam,type="p",level=1))
qqPlot(mlm.std.con[,1])
mlm.mean <- mean(mlm.std.con[,1]^2)
mlm.rmsei <- sqrt(mlm.mean)
mlm.rmsei
```

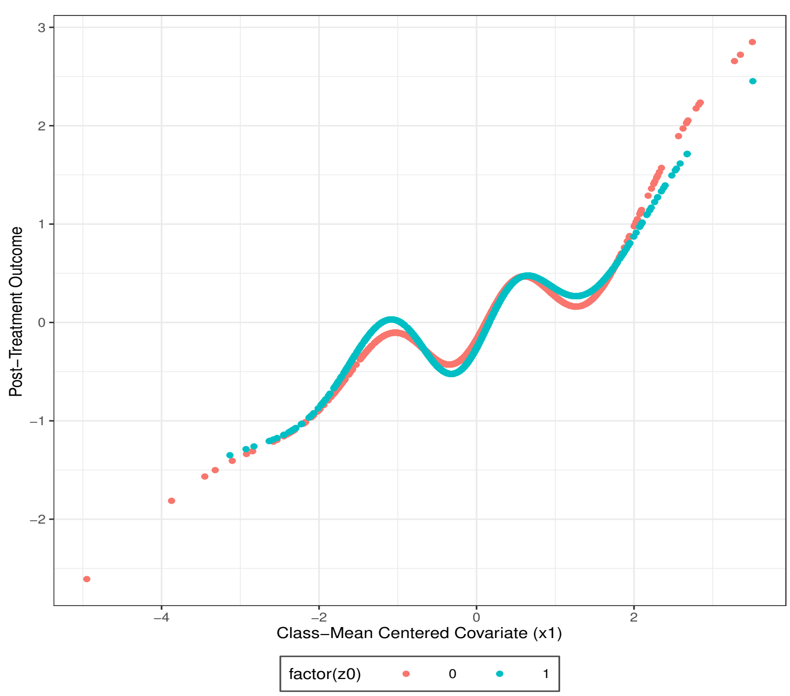
Appendix S3.

Data Generation in the Simulation Study

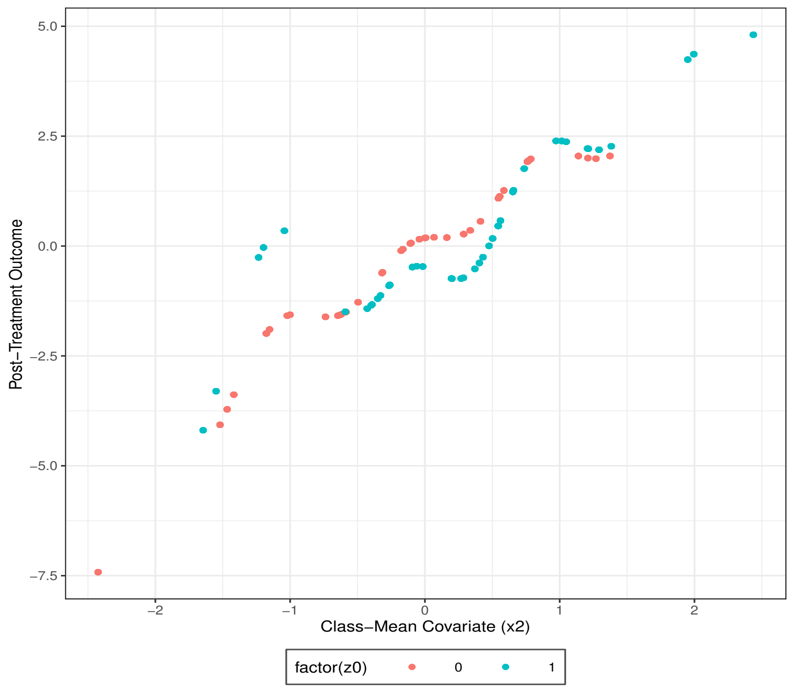
The following is an example of generating by-variable smooth functions for the condition of $J = 80$, $n_j = 30$, and $ICC = 0.30$. Nine basis functions ($\mathbf{b}_h$ for TPRS) were obtained using the `interpret.gam` and `smoothCon` functions in the `mgcv` packages. Example code for $f_2(x_{ij} - x_{.j})(z_j = 0)$ is as follows:

```
smooth.spec.object.x1.z1 <- interpret.gam(y~s(x1,by=z,bs="tp",k=10,m=2))$smooth.spec[[1]]
SM.x1.z1 <- smoothCon(smooth.spec.object.x1.z1,data=data.2,knots=NULL,absorb.cons=TRUE)[[1]]
X.x1.z1 <- as.matrix(SM.x1.z1$X)
dim(X.x1.z1)
S.x1.z1 <- SM.x1.z1$S[[1]]
```

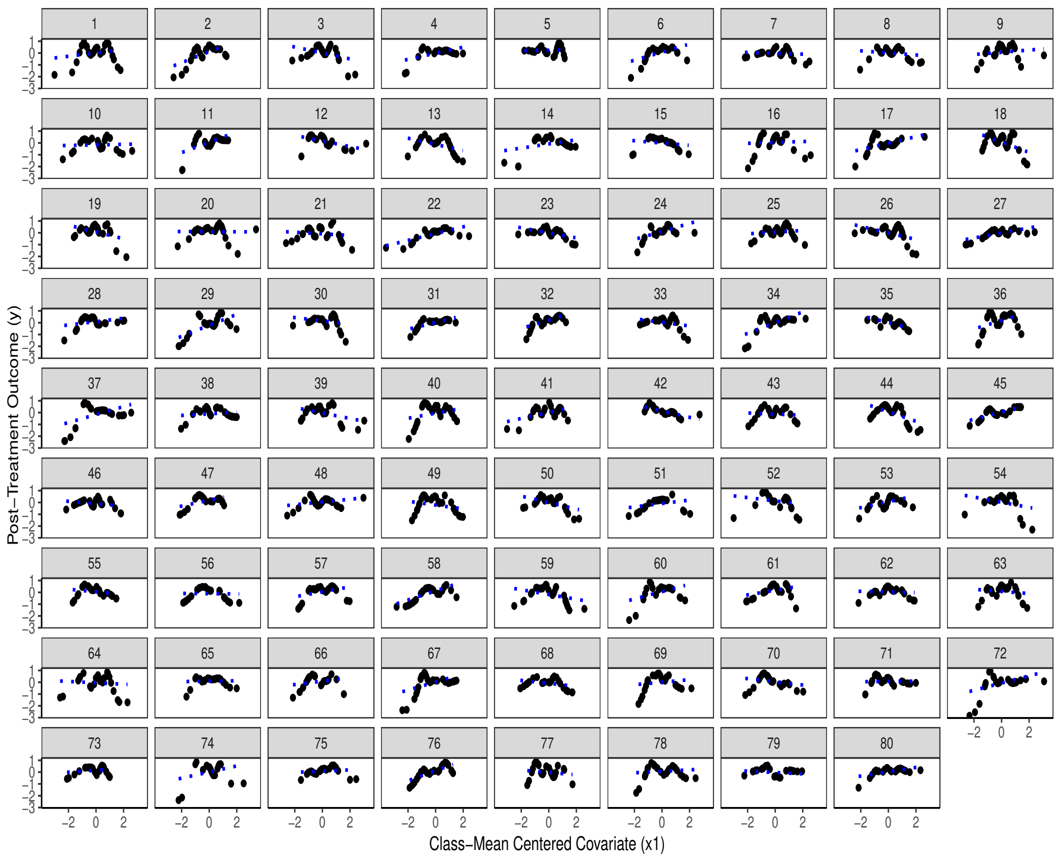
The true basis coefficients were set as $[0.5, 0.1, 0.4, 0.3, 0.1, 0.4, 0.1, 0.9, 0.8]'$ for $f_2(x_{ij} - x_{.j})(z_j = 0)$ and $[0.5, 0.2, 0.3, 0.4, 0.1, 0.4, 0.2, 0.9, 0.8]'$ for $f_2(x_{ij} - x_{.j})(z_j = 1)$. Based on TPRS basis functions with $K = 10 - 1$ and the true basis coefficients, the generated by-variable smooth functions $f_2(x_{ij} - x_{.j})(z_j = 0)$ and $f_2(x_{ij} - x_{.j})(z_j = 1)$ are as follows:



The true basis coefficients were set as $[0.1, 0.1, 0.2, 0.3, 0.1, 0.4, 1.1, 0.3, 1.3]'$ for $f_1(x_{.j})(z_j = 0)$ and $[0.9, 1.9, 0.5, 0.5, 0.4, 0.9, 1.8, 1.5, 1.1]'$ for $f_1(x_{.j})(z_j = 1)$. Based on TPRS basis functions with $K = 10 - 1$ and the true basis coefficients, the generated by-variable smooth functions $f_1(x_{.j})(z_j = 0)$ and $f_1(x_{.j})(z_j = 1)$ are as follows:



The generated by-variable smooth functions $f_3(x_{ij} - x_{.j})(Cluster_j = j)$s are as follows:



Note. Dotted lines indicate linear functions to show their deviation from the 'true' by-variable smooth functions (dots).

Appendix S4.

Simulation Study Results

GAMM as a Data-Generating Model (Research Questions (a) and (b))

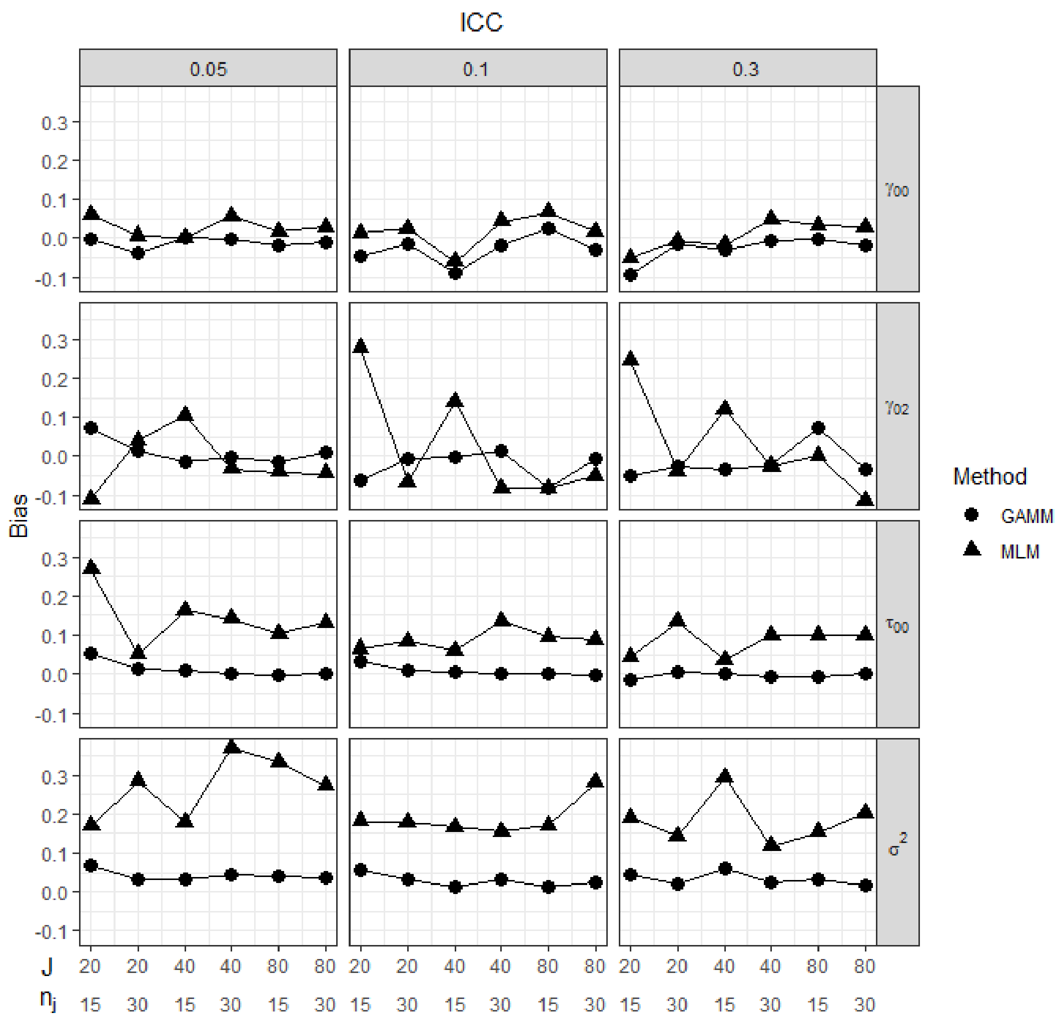


Figure S4.1 Bias of parameter estimates in GAMM and MLM.

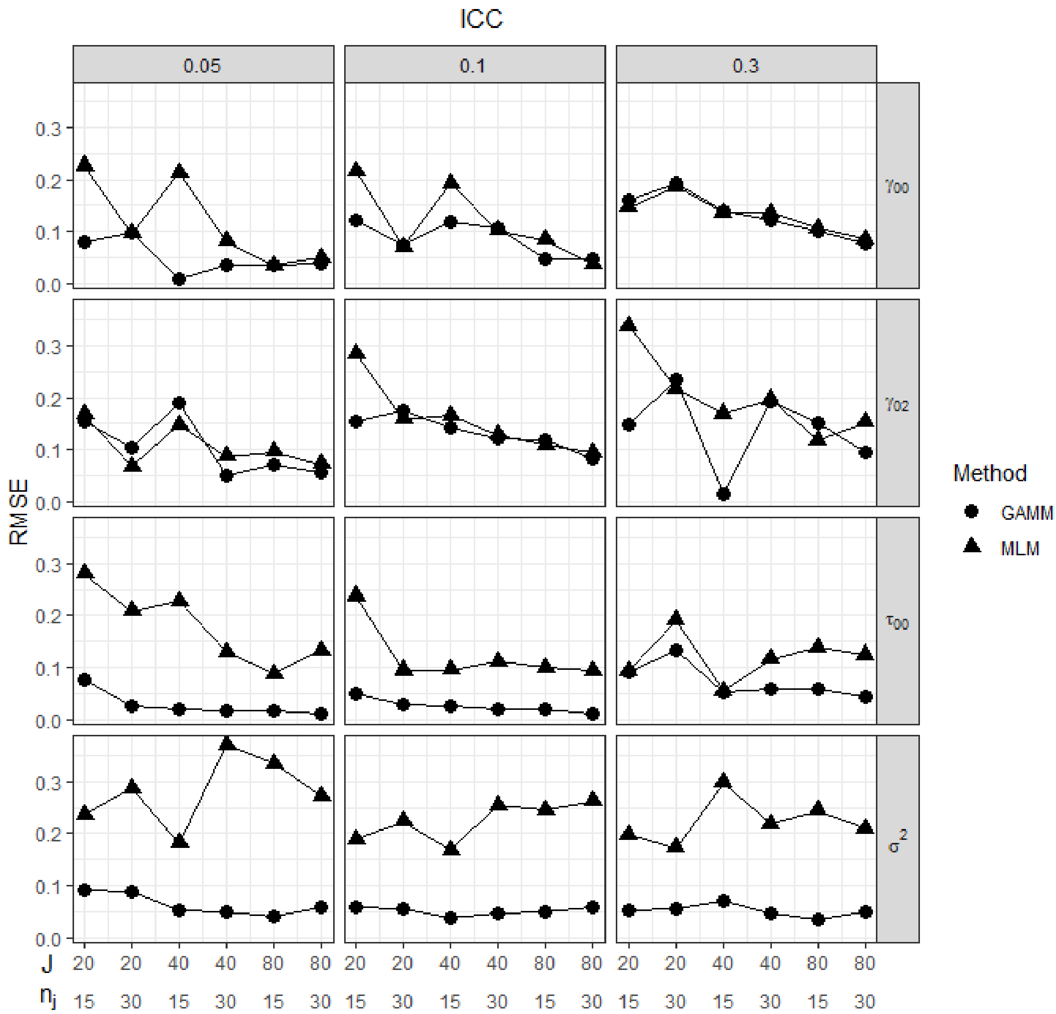


Figure S4.2 RMSE of parameter estimates in GAMM and MLM.

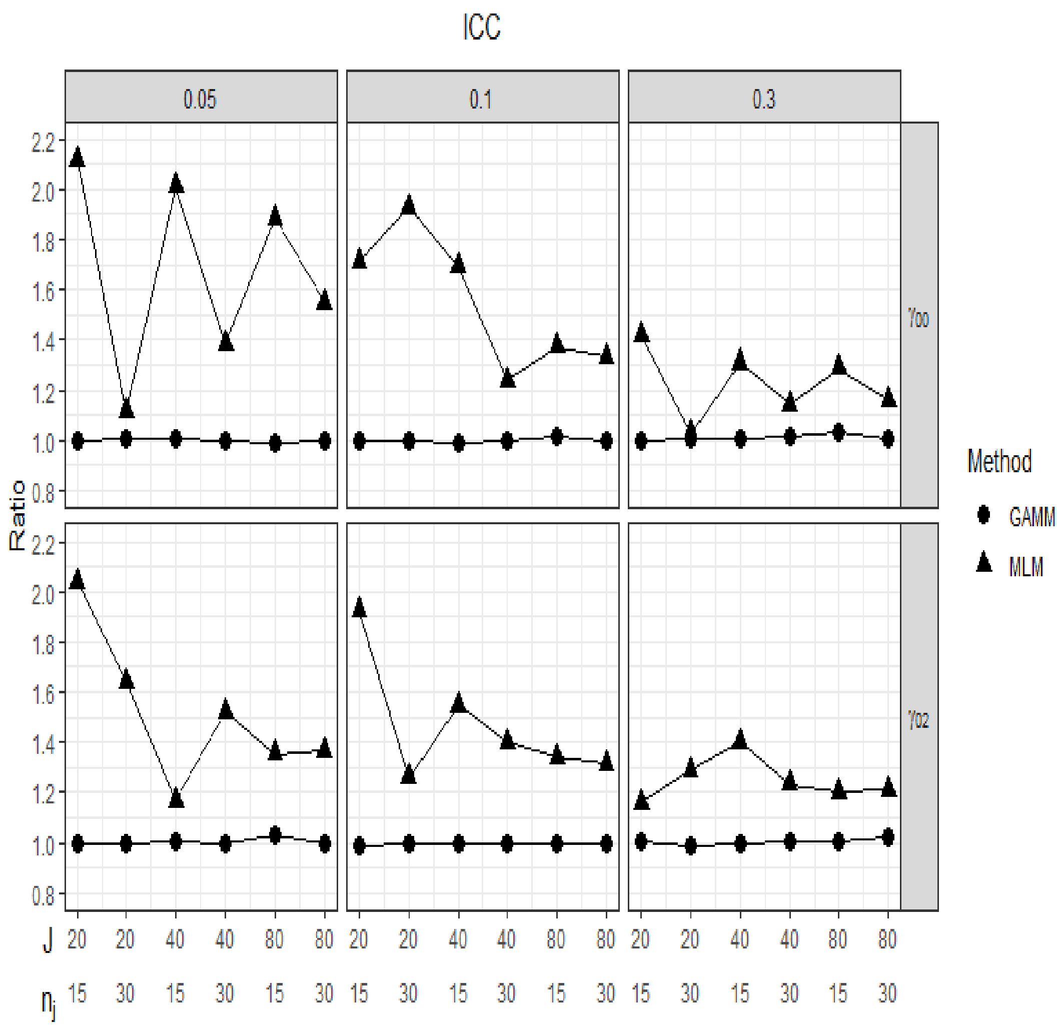


Figure S4.3 Ratio of M(SE) to SD for estimates of fixed effects in GAMM and MLM.

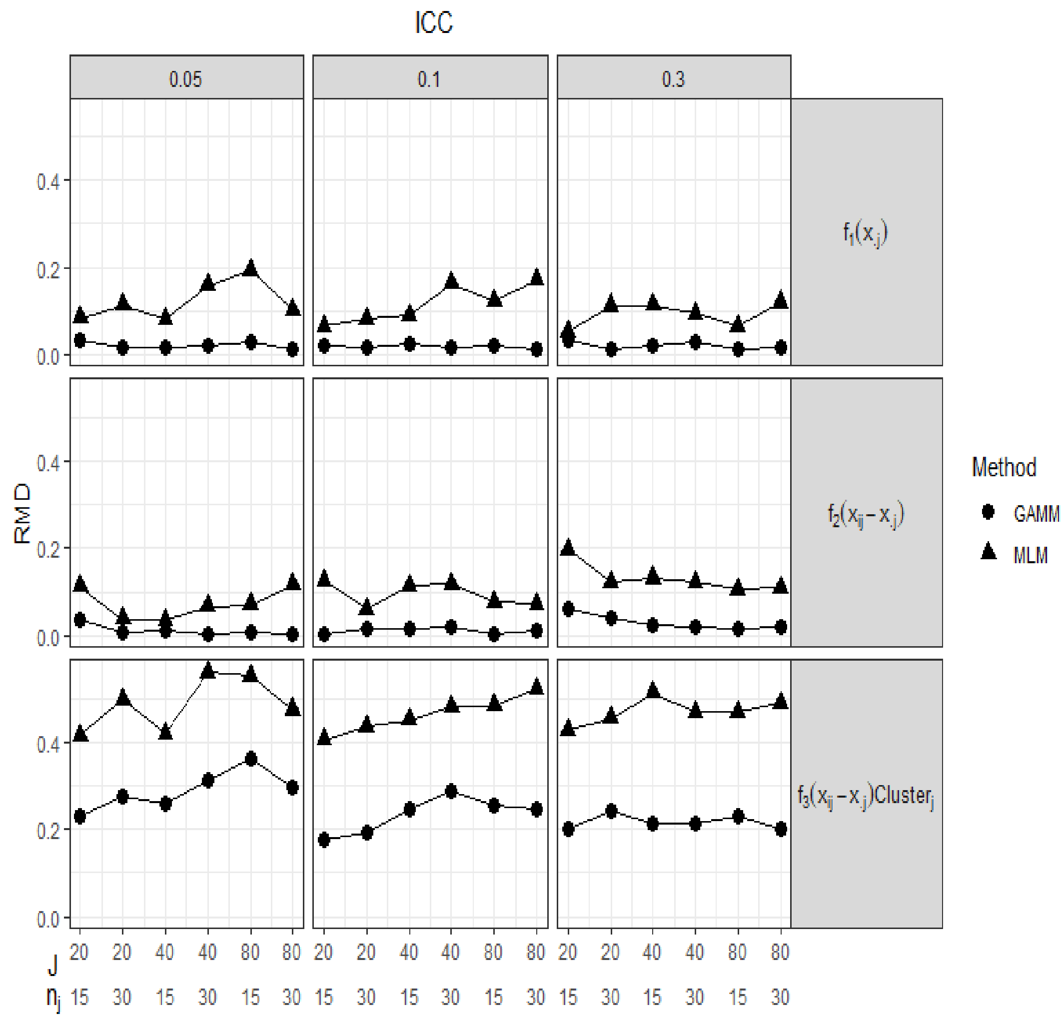


Figure S4.4 RMD of smooth functions in GAMM and MLM.

MLM as a Data-Generating Model (Research Question (c))

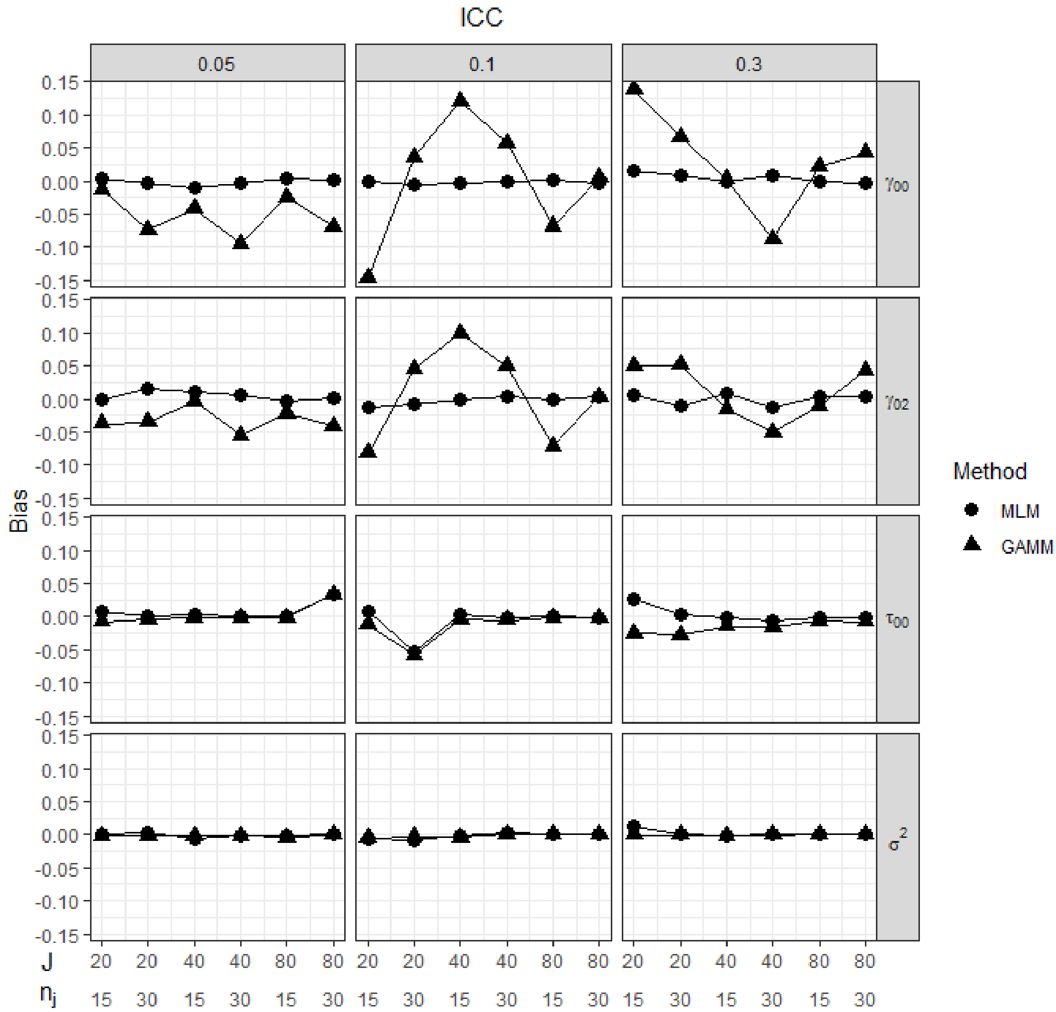


Figure S4.5 Bias of parameter estimates in GAMM and MLM.

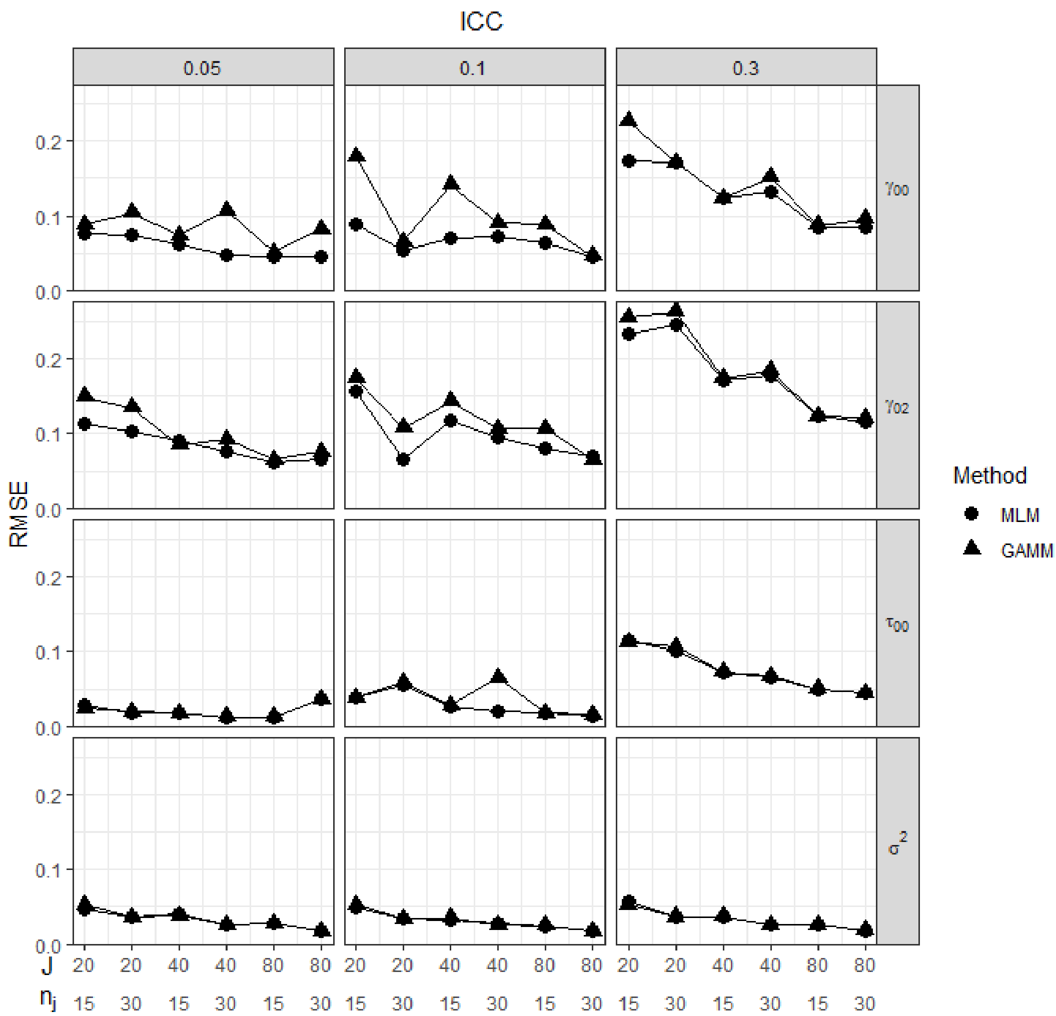


Figure S4.6 RMSE of parameter estimates in GAMM and MLM.

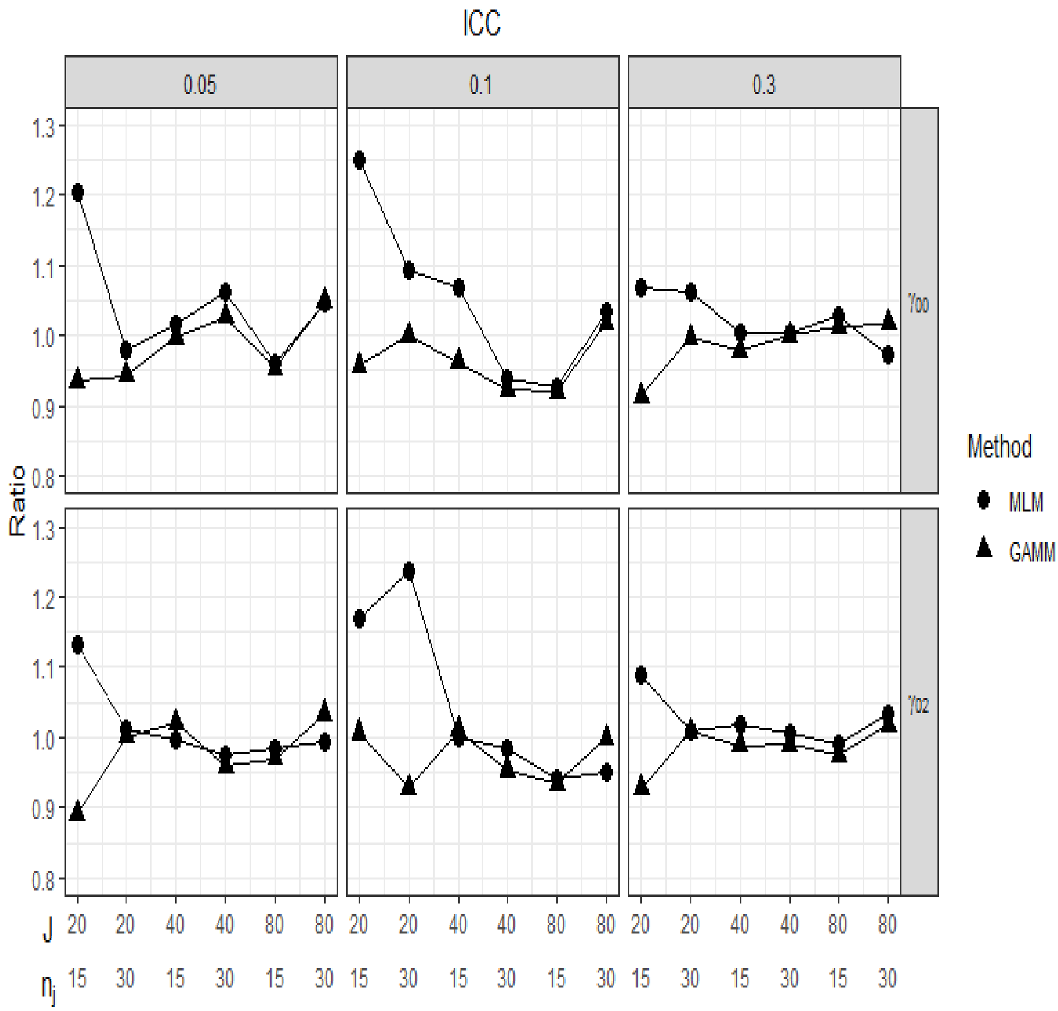


Figure S4.7 Ratio of M(SE) to SD for estimates of fixed effects in GAMM and MLM.

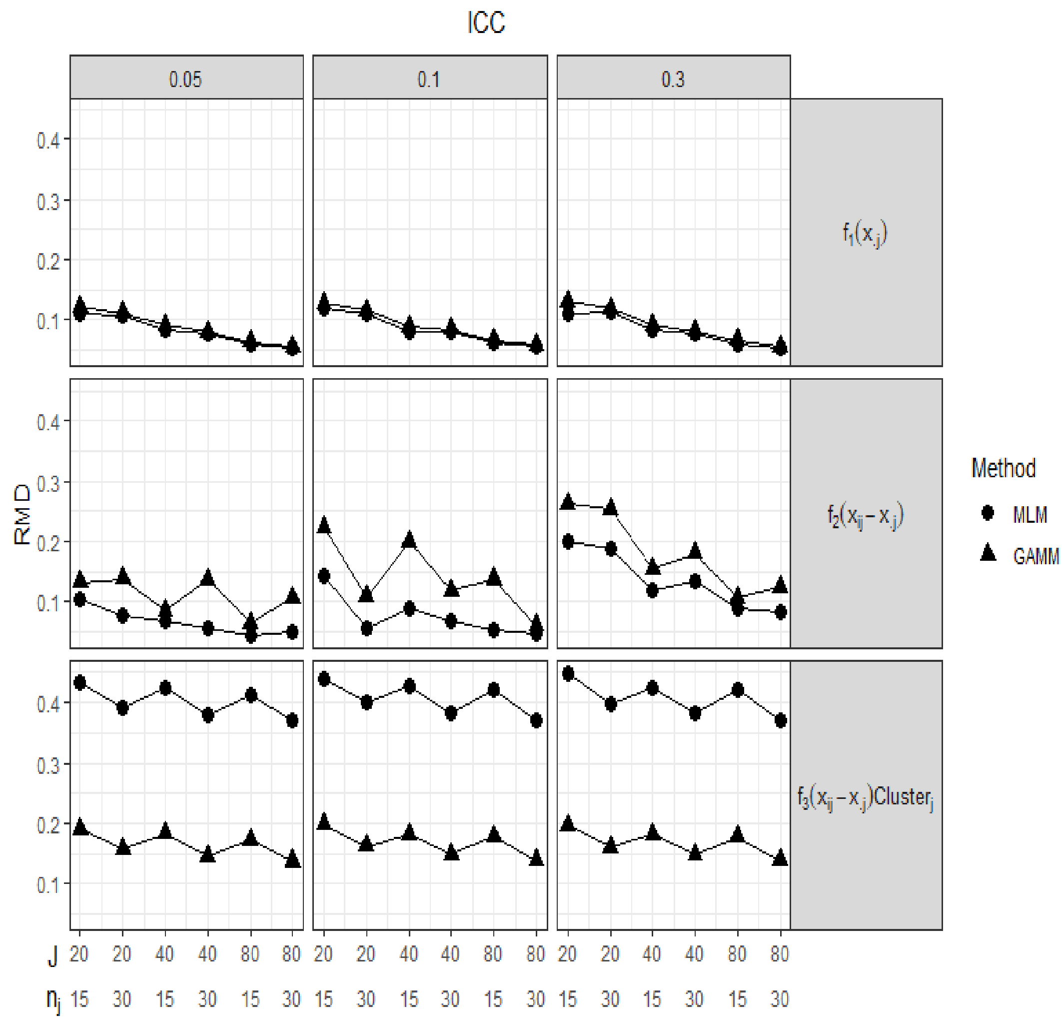


Figure S4.8 RMD of smooth functions in GAMM and MLM.