Woods\_Rate\_Fittings\_multiple.R

Phil

2022-11-10

## Rate Fittings   
## Masters Thesis: Stretch activation and fatigue   
## Philip C. Woods  
## Created: 5/23/22  
## Last updated: 7/22/22

# packages to load  
library(tcltk)  
library(tidyverse)  
library(readxl)  
library(dygraphs)  
library(RcppRoll)  
library(minpack.lm)  
library(writexl)  
library(ggpubr)  
library(broom)  
theme\_set(theme\_classic())  
  
rm(list = ls())  
if(!is.null(dev.list())) dev.off()  
cat("\014")   
  
# formula for fits  
my\_forumula <- Force\_One ~ (a\*exp(-b\*time0))+   
 (c\*(1.0-exp(-d\*time0))) +   
 (e\*exp(-g\*time0))  
  
# function to graph each parameter seperately  
get\_seperate\_phases <- function(model\_tidy, time0){  
 opt\_a <- filter(model\_tidy, term == 'a')  
 opt\_b <- filter(model\_tidy, term == 'b')  
 opt\_c <- filter(model\_tidy, term == 'c')  
 opt\_d <- filter(model\_tidy, term == 'd')  
 opt\_e <- filter(model\_tidy, term == 'e')  
 opt\_g <- filter(model\_tidy, term == 'g')  
   
 p2 <- opt\_a$estimate \* exp(-opt\_b$estimate \* time0)  
 p3 <- opt\_c$estimate \* (1 - exp(-opt\_d$estimate \* time0))  
 p4 <- opt\_e$estimate \* exp(-opt\_g$estimate \* time0)  
   
 phase2 <- data.frame(time0 = time0,  
 Force\_One = p2,  
 phase = '2')  
   
 phase3<- data.frame(time0 = time0,  
 Force\_One = p3,  
 phase = '3')  
 phase4 <- data.frame(time0 = time0,  
 Force\_One = p4,  
 phase = '4')  
   
 rbind(phase2,phase3,phase4)  
}  
  
  
## read data in-----------------------------------------------------------------  
  
# Brent Messing around  
# read\_fiber <- function(file){  
# read\_excel(file, skip = 29) %>%  
# dplyr::select(Time, Force\_One)  
# }  
#   
# my\_data11 <- map(my\_files, read\_fiber)  
  
  
setwd(tk\_choose.dir("Choose X"))  
my\_files <- list.files(pattern = "Run")  
my\_data <- map(my\_files, ~ read\_excel(.x, skip = 29) %>%  
 dplyr::select(Time, Force\_One))  
  
names(my\_data) <- my\_files  
  
## Run 2: Fatigue pCa [[1]] ------------------------------------------------------  
  
dygraph(my\_data$Run2.xlsx)  
  
r2 <- my\_data$Run2.xlsx %>%   
 filter(Time >= 0.067875, Time <= 0.25) %>%   
 mutate(time0 = Time - Time[[1]], .before = Force\_One) %>%   
 select(-Time)  
  
dygraph(r2)  
#   
# r2\_phase2 <- r2 %>%   
# filter(time0 <= 0.05)  
#   
# r2\_lm <- lm(log10(r2\_phase2$Force\_One) ~ r2\_phase2$time0)  
#   
# r2\_phase2$lm <- predict(r2\_lm)  
#   
# r2\_phase2\_model <- nlsLM(Force\_One ~ (a\*exp(-b\*time0)),  
# data = r2\_phase2,  
# start = list(a = (10^r2\_lm$coefficients[[1]]),  
# b = (-r2\_lm$coefficients[[2]])/(log10(exp(1)))),  
# control = nls.control(maxiter = 100))  
#   
# r2\_phase2\_model\_summary <- broom::tidy(r2\_phase2\_model)  
#   
# grd2 <- list(a = r2\_phase2\_model\_summary$estimate[[1]],  
# b = r2\_phase2\_model\_summary$estimate[[2]],  
# c = tail(my\_data$Run2.xlsx$Force\_One, n=1),  
# d = r2\_phase2\_model\_summary$estimate[[2]]/2,  
# e = r2\_phase2\_model\_summary$estimate[[1]],  
# g = r2\_phase2\_model\_summary$estimate[[2]]/4)  
  
# grd2 <- list(a = run3\_model\_tidy$estimate[[1]],  
# b = run3\_model\_tidy$estimate[[2]],  
# c = run3\_model\_tidy$estimate[[3]],  
# d = run3\_model\_tidy$estimate[[4]],  
# e = run3\_model\_tidy$estimate[[5]],  
# g = run3\_model\_tidy$estimate[[6]])  
  
#soleus  
grd2 <- list(a = 0.02,  
 b = 200,  
 c = 0.02,  
 d = 5,  
 e = 0.02,  
 g = 5)  
# #   
# grd2 <- list(a = 0.005,  
# b = 300,  
# c = 0.02,  
# d = 50,  
# e = 0.02,  
# g = 10)  
  
#EDL  
# grd2 <- list(a = 0.02,  
# b = 500,  
# c = 0.02,  
# d = 300,  
# e = 0.02,  
# g = 50)  
  
# grd2 <- grd4  
  
run2\_model <- nlsLM(my\_forumula,  
 data = r2,  
 start = grd2,  
 control = nls.control(maxiter = 100))  
  
r2$fit <- predict(run2\_model)  
  
(run2.graph <- ggplot(data = r2, aes(x = time0, y = Force\_One)) +  
 geom\_point()+  
 geom\_line(aes(y = fit), size = 0.8, col = "red") +  
 ggtitle("Run2")  
)  
  
run2\_model\_tidy <- tidy(run2\_model)  
  
run2\_seperate <- get\_seperate\_phases(run2\_model\_tidy, r2$time0)  
  
(run2\_all <- ggplot() +  
 geom\_line(data = run2\_seperate,   
 aes(x = time0, y = Force\_One, color = phase)) +  
 geom\_line(data = r2,   
 aes(x = time0, y = fit), size = 0.8, col = "red") +  
 ggtitle("Run 2 Seperated")  
)  
  
run2\_info <- list(data.frame(grd2),  
 data.frame(r2),  
 data.frame(run2\_model\_tidy),  
 run2\_seperate)  
  
names(run2\_info) <- list("Starting Parameters",  
 "Truncated Data",  
 "Model",  
 "Rates Seperated")  
  
## Run 3: Fatigue pCa [[2]] ------------------------------------------------------  
  
dygraph(my\_data$Run3.xlsx)  
  
r3 <- my\_data$Run3.xlsx %>%   
 filter(Time >=0.0675, Time <= 0.25) %>%   
 mutate(time0 = Time - Time[[1]], .before = Force\_One) %>%   
 select(-Time)  
  
dygraph(r3)  
#   
# r3\_phase2 <- r3 %>%   
# filter(time0 <= 0.01025)  
#   
# r3\_lm <- lm(log10(r3\_phase2$Force\_One) ~ r3\_phase2$time0)  
#   
# r3\_phase2$lm <- predict(r3\_lm)  
#   
# r3\_phase2\_model <- nlsLM(Force\_One ~ (a\*exp(-b\*time0)),  
# data = r3\_phase2,  
# start = list(a = (10^r3\_lm$coefficients[[1]]),  
# b = (-r3\_lm$coefficients[[2]])/(log10(exp(1)))),  
# control = nls.control(maxiter = 100))  
#   
# r3\_phase2\_model\_summary <- broom::tidy(r3\_phase2\_model)  
  
# grd3 <- list(a = r3\_phase2\_model\_summary$estimate[[1]],  
# b = r3\_phase2\_model\_summary$estimate[[2]],  
# c = tail(r3$Force\_One, n=1),  
# d = r3\_phase2\_model\_summary$estimate[[2]]/2,  
# e = r3\_phase2\_model\_summary$estimate[[1]],  
# g = r3\_phase2\_model\_summary$estimate[[2]]/4)  
#   
# grd3 <- list(a = run2\_model\_tidy$estimate[[1]],  
# b = run2\_model\_tidy$estimate[[2]],  
# c = run2\_model\_tidy$estimate[[3]],  
# d = run2\_model\_tidy$estimate[[4]],  
# e = run2\_model\_tidy$estimate[[5]],  
# g = run2\_model\_tidy$estimate[[6]])  
  
grd3 <- list(a = 0.02,  
 b = 300,  
 c = 0.02,  
 d = 10,  
 e = 0.02,  
 g = 5)  
  
# grd3 <- grd2  
  
run3\_model <- nlsLM(my\_forumula,  
 data = r3,  
 start = grd3,  
 control = nls.control(maxiter = 100))   
  
r3$fit <- predict(run3\_model)  
  
(run3.graph <- ggplot(data = r3, aes(x = time0, y = Force\_One)) +  
 geom\_point()+  
 geom\_line(aes(y = fit), size = 0.8, col = "red") +  
 ggtitle("Run3")  
)  
  
run3\_model\_tidy <- tidy(run3\_model)  
  
run3\_seperate <- get\_seperate\_phases(run3\_model\_tidy, r3$time0)  
  
(run3\_all <- ggplot() +  
 geom\_line(data = run3\_seperate,   
 aes(x = time0, y = Force\_One, color = phase)) +  
 geom\_line(data = r3,   
 aes(x = time0, y = fit), size = 0.8, col = "red") +  
 ggtitle("Run 3 Seperated")  
)  
  
run3\_info <- list(data.frame(grd3),  
 data.frame(r3),  
 data.frame(run3\_model\_tidy),  
 run3\_seperate)  
  
names(run3\_info) <- list("Starting Parameters",  
 "Truncated Data",  
 "Model",  
 "Rates Seperated")  
  
ggs <- ggarrange(run3.graph,run3\_all,ncol=1)  
  
  
ggexport(ggs, filename = "Woods\_M4F15\_Fat5.1\_3Exp.pdf")  
write\_xlsx(run3\_info, path = "Woods\_M4F15\_Fat5.1\_3Exp.xlsx")  
  
## Run 4: Fatigue pCa 4.5 ------------------------------------------------------  
  
dygraph(my\_data$Run4.xlsx)  
  
r4 <- my\_data$Run4.xlsx %>%   
 filter(Time >=0.067375, Time <= 0.25) %>%   
 mutate(time0 = Time - Time[[1]], .before = Force\_One) %>%   
 select(-Time)  
  
dygraph(r4)  
#   
# r4\_phase2 <- r4 %>%   
# filter(time0 <= 0.0064)  
#   
# r4\_lm <- lm(log10(r4\_phase2$Force\_One) ~ r4\_phase2$time0)  
#   
# r4\_phase2$lm <- predict(r4\_lm)  
#   
# r4\_phase2\_model <- nlsLM(Force\_One ~ (a\*exp(-b\*time0)),  
# data = r4\_phase2,  
# start = list(a = (10^r4\_lm$coefficients[[1]]),  
# b = (-r4\_lm$coefficients[[2]])/(log10(exp(1)))),  
# control = nls.control(maxiter = 100))  
#   
# r4\_phase2\_model\_summary <- broom::tidy(r4\_phase2\_model)  
  
# grd4 <- list(a = r4\_phase2\_model\_summary$estimate[[1]],  
# b = r4\_phase2\_model\_summary$estimate[[2]],  
# c = tail(my\_data$Run4.xlsx$Force\_One, n=1),  
# d = r4\_phase2\_model\_summary$estimate[[2]]/2,  
# e = r4\_phase2\_model\_summary$estimate[[1]],  
# g = r4\_phase2\_model\_summary$estimate[[2]]/4)  
  
grd4 <- list(a = run5\_model\_tidy$estimate[[1]],  
 b = run5\_model\_tidy$estimate[[2]],  
 c = run5\_model\_tidy$estimate[[3]],  
 d = run5\_model\_tidy$estimate[[4]],  
 e = run5\_model\_tidy$estimate[[5]],  
 g = run5\_model\_tidy$estimate[[6]])  
  
# grd4 <- list(a = 0.02,  
# b = 200,  
# c = 0.02,  
# d = 20,  
# e = 0.02,  
# g = 10)  
  
# grd4 <- list(a = 0.02,  
# b = 100,  
# c = 0.02,  
# d = 10,  
# e = 0.02,  
# g = 10)  
  
run4\_model <- nlsLM(my\_forumula,  
 data = r4,  
 start = grd4,  
 control = nls.control(maxiter = 100))  
  
r4$fit <- predict(run4\_model)  
  
(run4.graph <- ggplot(data = r4, aes(x = time0, y = Force\_One)) +  
 geom\_point()+  
 geom\_line(aes(y = fit), size = 0.8, col = "red") +  
 ggtitle("Run4")  
)  
  
run4\_model\_tidy <- tidy(run4\_model)  
  
run4\_seperate <- get\_seperate\_phases(run4\_model\_tidy, r4$time0)  
  
(run4\_all <- ggplot() +  
 geom\_line(data = run4\_seperate,   
 aes(x = time0, y = Force\_One, color = phase)) +  
 geom\_line(data = r4,   
 aes(x = time0, y = fit), size = 0.8, col = "red") +  
 ggtitle("Run 4 Seperated")  
)  
  
run4\_info <- list(data.frame(grd4),  
 data.frame(r4),  
 data.frame(run4\_model\_tidy),  
 run4\_seperate)  
  
names(run4\_info) <- list("Starting Parameters",  
 "Truncated Data",  
 "Model",  
 "Rates Seperated")  
  
## Run 5: Active ---------------------------------------------------------------  
  
dygraph(my\_data$Run5.xlsx)  
  
r5 <- my\_data$Run5.xlsx %>%   
 filter(Time >=0.067375, Time <= 0.12) %>%   
 mutate(time0 = Time - Time[[1]], .before = Force\_One) %>%   
 select(-Time)  
  
dygraph(r5)  
  
# r5\_phase2 <- r5 %>%  
# filter(time0 <= 0.03)  
# #  
#   
# r5\_lm <- lm(log10(r5\_phase2$Force\_One) ~ r5\_phase2$time0)  
# #  
# r5\_phase2$lm <- predict(r5\_lm)  
# #  
# r5\_phase2\_model <- nlsLM(Force\_One ~ (a\*exp(-b\*time0)),  
# data = r5\_phase2,  
# start = list(a = (10^r5\_lm$coefficients[[1]]),  
# b = (-r5\_lm$coefficients[[2]])/(log10(exp(1)))),  
# control = nls.control(maxiter = 100))  
# #  
# r5\_phase2\_model\_summary <- broom::tidy(r5\_phase2\_model)  
#   
# grd5 <- list(a = r5\_phase2\_model\_summary$estimate[[1]],  
# b = r5\_phase2\_model\_summary$estimate[[2]],  
# c = tail(my\_data$Run5.xlsx$Force\_One, n=1),  
# d = r5\_phase2\_model\_summary$estimate[[2]]/2,  
# e = r5\_phase2\_model\_summary$estimate[[1]],  
# g = r5\_phase2\_model\_summary$estimate[[2]]/4)  
  
# grd5 <- list(a = run4\_model\_tidy$estimate[[1]],  
# b = run4\_model\_tidy$estimate[[2]],  
# c = run4\_model\_tidy$estimate[[3]],  
# d = run4\_model\_tidy$estimate[[4]],  
# e = run4\_model\_tidy$estimate[[5]],  
# g = run4\_model\_tidy$estimate[[6]])  
  
# # Starting parameters for Type I trace  
# grd5 <- list(a = 0.02,  
# b = 300,  
# c = 0.02,  
# d = 50,  
# e = 0.02,  
# g = 10)  
  
grd5 <- list(a = 0.02,  
 b = 800,  
 c = 0.02,  
 d = 300,  
 e = 0.02,  
 g = 50)  
  
# grd5 <- grd4  
  
  
run5\_model <- nlsLM(my\_forumula,  
 data = r5,  
 start = grd5,  
 control = nls.control(maxiter = 100))  
  
r5$fit <- predict(run5\_model)  
  
(run5.graph <- ggplot(data = r5, aes(x = time0, y = Force\_One)) +  
 geom\_point()+  
 geom\_line(aes(y = fit), size = 0.8, col = "red") +  
 ggtitle("Run5")  
)  
  
run5\_model\_tidy <- tidy(run5\_model)  
  
run5\_seperate <- get\_seperate\_phases(run5\_model\_tidy, r5$time0)  
  
(run5\_all <- ggplot() +  
 geom\_line(data = run5\_seperate,   
 aes(x = time0, y = Force\_One, color = phase)) +  
 geom\_line(data = r5,   
 aes(x = time0, y = fit), size = 0.8, col = "red") +  
 ggtitle("Run 5 Seperated")  
)  
  
run5\_info <- list(data.frame(grd5),  
 data.frame(r5),  
 data.frame(run5\_model\_tidy),  
 run5\_seperate)  
  
names(run5\_info) <- list("Starting Parameters",  
 "Truncated Data",  
 "Model",  
 "Rates Seperated")  
  
## Run 6: Active Remeasure -----------------------------------------------------  
  
dygraph(my\_data$Run6.xlsx)  
  
r6 <- my\_data$Run6.xlsx %>%   
 filter(Time >=0.069, Time <= 0.12) %>%   
 mutate(time0 = Time - Time[[1]], .before = Force\_One) %>%   
 select(-Time)  
  
dygraph(r6)  
  
# r6\_phase2 <- r6 %>%  
# filter(time0 <= 0.047875)  
#   
# r6\_lm <- lm(log10(r6\_phase2$Force\_One) ~ r6\_phase2$time0)  
#   
# r6\_phase2$lm <- predict(r6\_lm)  
#   
# r6\_phase2\_model <- nlsLM(Force\_One ~ (a\*exp(-b\*time0)),  
# data = r6\_phase2,  
# start = list(a = (10^r6\_lm$coefficients[[1]]),  
# b = (-r6\_lm$coefficients[[2]])/(log10(exp(1)))),  
# control = nls.control(maxiter = 100))  
#   
# r6\_phase2\_model\_summary <- broom::tidy(r6\_phase2\_model)  
  
# grd6 <- list(a = r6\_phase2\_model\_summary$estimate[[1]],  
# b = r6\_phase2\_model\_summary$estimate[[2]],  
# c = tail(my\_data$Run6.xlsx$Force\_One, n=1),  
# d = r6\_phase2\_model\_summary$estimate[[2]]/2,  
# e = r6\_phase2\_model\_summary$estimate[[1]],  
# g = r6\_phase2\_model\_summary$estimate[[2]]/4)  
  
grd6 <- list(a = run5\_model\_tidy$estimate[[1]],  
 b = run5\_model\_tidy$estimate[[2]],  
 c = run5\_model\_tidy$estimate[[3]],  
 d = run5\_model\_tidy$estimate[[4]],  
 e = run5\_model\_tidy$estimate[[5]],  
 g = run5\_model\_tidy$estimate[[6]])  
  
# grd6 <- list(a = 0.04,  
# b = 120,  
# c = 0.02,  
# d = 15,  
# e = 0.01,  
# g = 5)  
  
# grd6 <- list(a = 0.02,  
# b = 500,  
# c = 0.02,  
# d = 200,  
# e = 0.02,  
# g = 50)  
  
  
run6\_model <- nlsLM(my\_forumula,  
 data = r6,  
 start = grd6,  
 control = nls.control(maxiter = 100))  
  
r6$fit <- predict(run6\_model)  
  
(run6.graph <- ggplot(data = r6, aes(x = time0, y = Force\_One)) +  
 geom\_point()+  
 geom\_line(aes(y = fit), size = 0.8, col = "red") +  
 ggtitle("Run6")  
)  
  
run6\_model\_tidy <- tidy(run6\_model)  
  
run6\_seperate <- get\_seperate\_phases(run6\_model\_tidy, r6$time0)  
  
(run6\_all <- ggplot() +  
 geom\_line(data = run6\_seperate,   
 aes(x = time0, y = Force\_One, color = phase)) +  
 geom\_line(data = r6,   
 aes(x = time0, y = fit), size = 0.8, col = "red") +  
 ggtitle("Run 6 Seperated")  
)  
  
run6\_info <- list(data.frame(grd6),  
 data.frame(r6),  
 data.frame(run6\_model\_tidy),  
 run6\_seperate)  
  
names(run6\_info) <- list("Starting Parameters",  
 "Truncated Data",  
 "Model",  
 "Runs Seperated")  
  
  
## Single & Double Exp Decay Fits -------------------------------------------------  
  
# Fatigue pCa 1  
df <- r2 %>%   
 select(time0, Force\_One) %>%   
 mutate(Ten = Force\_One - tail(Force\_One, n=1), .before = Force\_One) %>%   
 select(-Force\_One)  
  
# df <- df %>%  
# filter(time0 <= 0.015)  
  
## Single  
single.mdl <- nlsLM((Ten ~ (a\*exp(-b\*time0))),  
 data = df,  
 start = list(a = 0.1,  
 b = 300),  
 control = nls.control(maxiter = 100))  
  
df$single.fit <- predict(single.mdl)  
  
single.tidy <- tidy(single.mdl)  
  
(single.graph <- ggplot(data = df, aes(x = time0, y = Ten)) +  
 geom\_point()+  
 geom\_line(aes(y = single.fit), size = 0.8, col = "red") +  
 ggtitle("M4F15 Fatigue 5.2 Single Exponential Fit")  
)   
  
## Double  
dbl.mdl <- nlsLM((Ten ~ (a\*exp(-b\*time0)) +   
 (e\*exp(-g\*time0))),  
 data = df,  
 start = list(a = 0.1,  
 b = 300,   
 e = 0.02,  
 g = 10),  
 control = nls.control(maxiter = 100))  
  
df$dbl.fit <- predict(dbl.mdl)  
  
dbl.tdy <- tidy(dbl.mdl)  
  
(dbl.graph <- ggplot(data = df, aes(x = time0, y = Ten)) +  
 geom\_point()+  
 geom\_line(aes(y = dbl.fit), size = 0.8, col = "red") +  
 ggtitle("M4F15 Fatigue 5.2 Double Exponential Fit")  
)   
  
ggs <- ggarrange(single.graph,dbl.graph,ncol=1)  
wp <- list(single.tidy,dbl.tdy)  
  
ggexport(ggs, filename = "Woods\_M4F15\_Fat5.2\_Single+DblExp.pdf")  
write\_xlsx(wp, path = "Woods\_M4F15\_Fat5.2\_Single+DblExp.xlsx")  
  
## Double Fits (Decay + Growth)-------------------------------------------------  
  
  
df2 <- r5 %>%   
 select(time0, Force\_One) #%>%   
# mutate(Ten = Force\_One - tail(Force\_One, n=1), .before = Force\_One) %>%   
# select(-Force\_One)  
  
dbl.mdl2 <- nlsLM((Force\_One ~ (a\*exp(-b\*time0)) + (c\*(1.0-exp(-d\*time0)))),  
 data = df2,  
 start = list(a = 0.02,  
 b = 1000,   
 c = 0.02,  
 d = 150),  
 control = nls.control(maxiter = 100))  
  
df2$dbl.fit <- predict(dbl.mdl2)  
  
dbl.tdy2 <- tidy(dbl.mdl2)  
  
(dbl2.graph <- ggplot(data = df2, aes(x = time0, y = Force\_One)) +  
 geom\_point()+  
 geom\_line(aes(y = dbl.fit), size = 0.8, col = "red") +  
 ggtitle("Double Exponential Fit")  
)   
  
  
  
## Saving ----------------------------------------------------------------------  
  
p <- list(plot1 = ggarrange(run2.graph, run2\_all, ncol = 1),  
 plot2 = ggarrange(run3.graph, run3\_all, ncol = 1),  
 plot3 = ggarrange(run4.graph, run4\_all, ncol = 1),  
 plot4 = ggarrange(run5.graph, run5\_all, ncol = 1),  
 plot5 = ggarrange(run6.graph, run6\_all, ncol = 1)  
)  
  
modelz <- list(run2\_info,  
 run3\_info,  
 run4\_info,  
 run5\_info,  
 run6\_info)  
  
names(modelz) <- c('Rates - Fatigue[[1]]',  
 'Rates - Fatigue[[2]]',  
 'Rates - Fatigue 4.5',  
 'Rates - Active',  
 'Rates - Active 2.0')  
  
ggexport(p, filename = "Woods\_Fiberx\_Phase3.pdf")  
pmap(list(modelz,   
 names(modelz)), ~ write\_xlsx(.x,   
 path = str\_c(.y, ".xlsx")))   
  
  
  
# Other attempts to save data  
# map2(modelz,   
# names(modelz), ~ write\_xlsx(.x,   
# names(modelz),  
# path = str\_c(.y),  
# ".xlsx"))  
#   
# map(modelz, ~ write\_xlsx(.x, path = str\_c(names(modelz), ".xlsx")))  
#   
# walk2(modelz, names(modelz), ~ write\_xlsx(.x, path = str\_c(.y, ".xlsx")))  
#   
# imap(modelz, names(modelz), ~write\_xlsx(.x, path = paste0(.y, ".xlsx")))  
  
# writexl::write\_xlsx(modelz, path = 'Woods\_Fiberx\_Fits.xlsx')   
# capture.output(sp, file = 'Woods\_Fiberx\_StartingParameters.txt')