library(WriteXLS)

library(readxl)

library(minpack.lm)

library(boot)

library(stats)

library(nlme)

library(lme4)

library(MASS)

library(zoo)

library(broom)

library(tidyverse)

library(sjmisc)

#library(graphics)

#

#setwd("/devUsers/macbookair/Desktop/BerretMicroRheology/Research/")

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles")

#

Distribution\_check=function(eta,lognormal){

#cauchy only is always on deltas therefore deltaeta =eta input / log normal : deltaeta=diff(eta)

#lognormal=true when you analyze individual viscosity profiles

deltaeta=eta

meanlog=0

if(lognormal=="true"){

# lognormal\_check

meanlog=coef(fitdistr(eta,"log-normal"))[1]

sdlog=coef(fitdistr(eta,"log-normal"))[2]

median=meanlog-sdlog\*sdlog/2

mode=meanlog-3\*sdlog\*sdlog/2

etaecdf=exp(meanlog)

etaecdfhb=exp(meanlog+sdlog)

etaecdflb=exp(meanlog-sdlog)

# no ecdf plots

# xlab="eta"

xlab="Transmission"

plot(ecdf(eta),cex.lab=1.5,log="x",xlim=c(min(eta),max(eta)),xlab=xlab)

plot(ecdf(rlnorm(1000,(meanlog),(sdlog))),cex.lab=1.5,col="red",add=TRUE,do.points=FALSE)

mtext(cex=0.75,paste0("eta lognormal ",round(etaecdf,1)," hb ",round(etaecdfhb,1)," lb ",round(etaecdflb,1)),3,0.5)

deltaeta=diff(eta)

}

# cauchy\_check via fitdistr

#plot(ecdf(deltaeta),xlab="delta eta",pch=20)

meancauchy=coef(fitdistr(deltaeta,"cauchy"))[1]

scalecauchy=coef(fitdistr(deltaeta,"cauchy"))[2]

mean=round(meancauchy,2)

fwhm=2\*(round(scalecauchy,2))

#

#plot((ecdf(rcauchy(1000,meancauchy,scalecauchy))),col="red",add=TRUE,do.points=FALSE)

#mtext(cex=0.75,paste0("delta eta cauchy: mean ",mean," FWHM ",fwhm),3,0.5)

#

# gauss\_check via fitdistr

meannormal=coef(fitdistr(deltaeta,"normal"))[1]

sdnormal=coef(fitdistr(deltaeta,"normal"))[2]

if(lognormal=="false"){

#plot(ecdf(deltaeta),xlab="delta eta", pch=20)

mean=round(meannormal,2)

sd=round(sdnormal,2)

#plot((ecdf(rnorm(1000,meannormal,sdnormal))),col="red",add=TRUE,xlab="delta eta",do.points=FALSE)

#mtext(cex=0.75,paste0("delta eta gauss: mean ",mean," sd ",sd),3,0.5)

}

#

#plot routine of a vector of "experimental" values ( returns, deltas etc. )

#

#subtract mean (obtained from gaussian) from deltaeta - also for the master curve

delta\_eta=sort(deltaeta-meannormal,na.last = NA)

mykurtosis=function(x){(mean((x-mean(x,na.rm=TRUE))^4)/(sd(x,na.rm=TRUE)^4))-3}

myskewness=function(x){mean((x-mean(x,na.rm=TRUE))^3)/(sd(x,na.rm=TRUE)^3)}

kurtosis=mykurtosis(delta\_eta)

skewness=myskewness(delta\_eta)

# t Student

#fitdistr(delta\_eta,"t",start = list(m=mean(delta\_eta),s=sd(delta\_eta), df=3), lower=c(-1,0.01,1))

lower=c(-1,0.02,2)

# for series 1: if(film=="film9"){lower=c(-1,0.01,8)} - check

Param=fitdistr(delta\_eta,"t",

start = list(m=mean(delta\_eta),s=sd(delta\_eta), df=3),

lower=lower,

upper=c(1,40,50))

tm =coef(Param)[1]

ts =coef(Param)[2]

tdf=coef(Param)[3]

#

deltaeta\_neg =sort(abs(delta\_eta[delta\_eta<0]))

deltaeta\_pos =sort(delta\_eta[delta\_eta>=0])

count\_neg =length(deltaeta\_neg)

count\_pos =length(deltaeta\_pos)

count=count\_neg+count\_pos

xneg\_exp=(count\_neg:1)/count

xpos\_exp=(count\_pos:1)/count

ylim=c(1/(2\*count),1)

xlim\_max=max(deltaeta\_neg,deltaeta\_pos)

xlim\_min=min(deltaeta\_neg,deltaeta\_pos)

#xlim\_min=min(deltaeta\_neg,deltaeta\_pos)\*0.5

xlim=c(xlim\_min,xlim\_max)

if(lognormal=="false"){#for scaling law

ylim=c(1/(2\*count),1)

if(max(deltaeta\_pos)>10){#criterion for deciding on type of x axis (normalized to sigma or not)

#xlab=expression(paste("|",Delta,"",eta,"|"))

xlab="delta T"

}else{

xlab=expression(paste("|",Delta,"",eta,"| / ",sigma,"(",Delta,"",eta,")"))

xlab="delta T/sigma(T)"

}

plot(deltaeta\_neg,xneg\_exp,cex.lab=1.5,pch=20,col="red",

log="y",ylim=ylim,xlim=xlim,xlab=xlab,ylab="CDF")

}else{#for each film

if(film=="Lehigh\_data"){

plot(deltaeta\_neg,xneg\_exp,cex.lab=1.5,cex=0.75,

log="y",ylab="CDF",xlim=xlim,ylim=ylim,

pch=20,col="red",xlab=expression(paste("|",Delta,"",G,"'| (Pa.s)")))

}else{

plot(deltaeta\_neg,xneg\_exp,cex.lab=1.5,cex=0.75,

log="y",ylab="CDF",xlim=xlim,ylim=ylim,

pch=20,col="red",xlab=expression(paste("|",Delta,"",eta,"| (Pa.s)")))

}

}

points(deltaeta\_pos,xpos\_exp,pch=20)

#for every frequency/film data are written to a seperate file when "lognormal=true"

#when Scaling function is called set "false"

#scaling is now done to sd and not the mean of the viscosity as earlier (however these are proportional to eachother)

#if statement has to be changed film="filmx" first in the series of films

Format=function(number){sprintf("%3.6f",number)}#to make stardardized output

if(lognormal=="true"){

delta\_eta\_scaling=delta\_eta/sd(delta\_eta)

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

#delta\_eta\_scaling=delta\_eta/exp(meanlog)

dfscalingdeltaeta=data.frame(delta\_eta\_scaling)

WriteXLS(data.frame(delta\_eta\_scaling), paste(film,"Scalingdeltaeta.xlsx",sep=""), SheetNames = "scalinglaw")

tdistribution\_fits=data.frame(film,Format(tm),Format(ts),Format(tdf),Format(skewness),Format(kurtosis))

print(film)

if(film=="film10"){write.table(tdistribution\_fits,paste("tdistribution\_fits.txt",sep=""),col.names=TRUE, append=TRUE)

}else{

write.table(tdistribution\_fits,paste("tdistribution\_fits.txt",sep="",col.names=FALSE,append=TRUE))}

}

# plot with distributions

# cauchy

lorentz=NULL

Cauchy=function(x,meancauchy,scalecauchy){1/2-atan((x-meancauchy)/scalecauchy)/pi}

# delta\_eta with mean = 0

lines(deltaeta\_neg,Cauchy(deltaeta\_neg,0,scalecauchy),lwd=2,col="green")

#if(lognormal=="false"){

y=(1:count)/count

# gaussian

deltaeta\_gauss=qnorm(y,0,sd(delta\_eta))

deltaeta\_gauss\_neg=abs(deltaeta\_gauss[deltaeta\_gauss<0])

negcdf=y[deltaeta\_gauss<0]

deltaeta\_gauss\_pos=deltaeta\_gauss[deltaeta\_gauss>=0]

poscdf=1-y[deltaeta\_gauss>=0]

count\_gauss\_neg =length(deltaeta\_gauss\_neg)

count\_gauss\_pos =length(deltaeta\_gauss\_pos)

lines(deltaeta\_gauss\_neg,negcdf,lwd=2,col='blue')

# t student - need to develop code as qt is with scale (sigma)=1 only

# - fitdistr uses more general t student: with mean (mu), scale (sigma) and

# degrees of freedaom (nu) as output parameters)

t\_general=function(x,mu,sigma,nu){

1/sigma\*

gamma((nu+1)/2)/

sqrt(nu\*pi)/gamma(nu/2)\*

(1+((x-mu)^2)/nu/sigma/sigma)^(-(1+nu)/2)

}

min\_eta=min(delta\_eta)

max\_eta=max(delta\_eta)

if(abs(min\_eta)>max\_eta){

max\_eta=abs(min\_eta)

}else{

min\_eta=-max\_eta

}

DELTA=max\_eta-min\_eta

deltaeta\_tstudent=NULL

cfdt=NULL

tdist=NULL

cfdt[1]=t\_general(min\_eta,tm,ts,tdf)

tdist[1]=cfdt[1]

deltaeta\_tstudent[1]=min\_eta

for(i in 2:count){

deltaeta\_tstudent[i]=min\_eta+(i-1)\*DELTA/count

if(abs(deltaeta\_tstudent[i])<0.0001){deltaeta\_tstudent[i]=0.0001}

tdist[i]=t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

cfdt[i]=cfdt[i-1]+t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

}

cfdt=cfdt/cfdt[count]

deltaeta\_tstudent\_neg=abs(deltaeta\_tstudent[deltaeta\_tstudent<0])

negcdft=cfdt[deltaeta\_tstudent<0]

deltaeta\_tstudent\_pos=deltaeta\_tstudent[deltaeta\_tstudent>=0]

poscdft=1-cfdt[deltaeta\_tstudent>=0]

count\_tstudent\_neg =length(deltaeta\_tstudent\_neg)

count\_tstudent\_pos =length(deltaeta\_tstudent\_pos)

#points(log="xy",xneg,yneg,pch=17,cex=1.5,col="red")

#points(xpos,ypos,pch=20)

lines(deltaeta\_tstudent\_neg,negcdft,lwd=2,lty=2,col='red')

lines(deltaeta\_tstudent\_pos,poscdft,lwd=2)

mtext(cex=0.75,paste("df = ",round(tdf,2)," scale = ",round(ts,2)),3,2)

mtext(cex=0.75,paste("Skewn = ",round(skewness,2)," Kurt = ",round(kurtosis,2)),3,1)

#

#

if(film=="film9" | film=="film15" | film=="film10" | film=="film21"){#series 1

#if(film=="film12" | film=="film13" | film=="film17" | film=="film10"){#series 2

delta\_eta\_all =c(deltaeta\_neg,deltaeta\_pos,deltaeta\_gauss\_neg,deltaeta\_gauss\_pos,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

ecdf\_exp\_neg =c(xneg\_exp,rep(NA,(count\_pos+count+count)))

ecdf\_exp\_pos =c(rep(NA,(count\_neg)),xpos\_exp,rep(NA,count+count))

ecdf\_cauchy\_neg =c(Cauchy(deltaeta\_neg,0,scalecauchy),rep(NA,(count\_pos+count+count)))

ecdf\_cauchy\_pos =c(rep(NA,count\_neg),Cauchy(deltaeta\_pos,0,scalecauchy), rep(NA,count+count))

ecdf\_gauss\_neg =c(rep(NA,count),negcdf,rep(NA,count\_gauss\_pos+count))

ecdf\_gauss\_pos =c(rep(NA,count+count\_gauss\_neg),poscdf,rep(NA,count))

ecdf\_tstudent\_neg =c(rep(NA,count+count),negcdft,rep(NA,count\_tstudent\_pos))

ecdf\_tstudent\_pos =c(rep(NA,count+count+count\_tstudent\_neg),poscdft)

# plot for publication non-scaled

plot\_cdf\_delta\_eta=data.frame(Format(delta\_eta\_all),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/Dataprep/")

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plots for publication - plots for order of magnitude difference

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results")

#series 1: 9, 21; series 2: 10, 17

if(film=="film9"){

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

if(film=="film21"){

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

# plot for publication scaled data for each frequency with the respective distributions for films 9,10,15,21

delta\_eta\_scaled = delta\_eta\_all/sd(delta\_eta)

plot\_cdf\_delta\_eta\_scaling = data.frame(Format(delta\_eta\_scaled),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

write.table(plot\_cdf\_delta\_eta\_scaling,quote=FALSE,file=paste(film,"plot\_correct\_scaling.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

if(lognormal=="false"){

# plot for publication - old master curve for cdf -

# data have been scaled on individual basis - see above - therefore deltaeta\_neg/pos are scaled

# arrays of x (xneg and pos) and y (deltaeta\_pos and neg) values

delta\_eta\_vs\_eta=c(deltaeta\_neg,deltaeta\_pos,deltaeta\_gauss\_neg,deltaeta\_gauss\_pos,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

ecdf\_exp\_neg =c(xneg\_exp,rep(NA,(count\_pos+count+count)))

ecdf\_exp\_pos =c(rep(NA,(count\_neg)),xpos\_exp,rep(NA,count+count))

ecdf\_cauchy\_neg =c(Cauchy(deltaeta\_neg,0,scalecauchy),rep(NA,(count\_pos+count+count)))

ecdf\_cauchy\_pos =c(rep(NA,count\_neg),Cauchy(deltaeta\_pos,0,scalecauchy), rep(NA,count+count))

ecdf\_gauss\_neg =c(rep(NA,count),negcdf,rep(NA,count\_gauss\_pos+count))

ecdf\_gauss\_pos =c(rep(NA,count+count\_gauss\_neg),poscdf,rep(NA,count))

ecdf\_tstudent\_neg =c(rep(NA,count+count),negcdft,rep(NA,count\_tstudent\_pos))

ecdf\_tstudent\_pos =c(rep(NA,count+count+count\_tstudent\_neg),poscdft)

# dataframe for plotting sprintf("%.6s",t)

#Format=function(number){sprintf("%.6s",number)}

plot\_mastercurve\_delta\_eta=data.frame(Format(delta\_eta\_vs\_eta),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_mastercurve\_delta\_eta,quote=FALSE,file="plot\_mastercurve\_delta\_eta\_scaled.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

}# End of distribution check

Scalingdelta\_aeta=function(){

# for Series/File selection put on TRUE

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

# All FALSE for series 3

if(FALSE){

# Film Series 1 for scaling of the selected data for publication

File=c("film10Scalingdeltaeta.xlsx",

"film15Scalingdeltaeta.xlsx",

"film21Scalingdeltaeta.xlsx" )

}

if(TRUE){

# Film Series 1

File=c("film1Scalingdeltaeta.xlsx",

"film8Scalingdeltaeta.xlsx",

"film9Scalingdeltaeta.xlsx",

"film10Scalingdeltaeta.xlsx",

"film11Scalingdeltaeta.xlsx",

"film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film14Scalingdeltaeta.xlsx",

"film15Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx",

"film18Scalingdeltaeta.xlsx",

"film19Scalingdeltaeta.xlsx",

"film20Scalingdeltaeta.xlsx",

"film21Scalingdeltaeta.xlsx" )

}

if(FALSE){

# Film Series Lehigh

File=c( "SheetScalingdeltaeta.xlsx")

}

if(FALSE){

# Film Series 2 for scaling

File=c( "film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx")

}

if(FALSE){

# Film Series 2

File=c( "film10Scalingdeltaeta.xlsx",

"film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film14Scalingdeltaeta.xlsx",

"film16Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx",

"film2Scalingdeltaeta.xlsx",

"film9Scalingdeltaeta.xlsx" )

}

dim=length(File)

combined\_deltaeta=NULL

for (i\_delta\_eta in 1:dim){

delta\_eta = read\_xlsx(File[i\_delta\_eta],"scalinglaw")

#check for end of file / determination of Dimension of data - Dimdata

for (i in 1:1000){

if (is.na(delta\_eta[i,1])){

Dimdata=i-1

break}

}

deltaeta=head(unlist((delta\_eta[,1])),Dimdata)

combined\_deltaeta=c(combined\_deltaeta,deltaeta)

#Distribution\_check(deltaeta,lognormal=FALSE)

}

#### scaling law all data irrespective of frequency - one does not see this in litarature

#### this part can be used to look at limited number files for the plot distribution data

#### (FALSE) leads to analysis of limited number of data

#### provided these have been set accordingly

df=data.frame(combined\_deltaeta)

if(FALSE){WriteXLS(df,"ScalingLawDeltaEta", SheetNames ="Alldata")}else{

WriteXLS(df,"ScalingLawDeltaEta", SheetNames ="Alldata")

}

lognormal="false"

count=length(combined\_deltaeta)

Distribution\_check(combined\_deltaeta,lognormal)

#read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

ylim=c(1/(2\*count),1)

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results")

####scaling law 1 order of magnitude apart: 0.089 Hz vs 0.847 Hz (Bouchaud et al.) for delta eta

if(TRUE){scaling\_nu8=read.table(paste("film9","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

scaling\_nu80=read.table(paste("film21","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

}

if(FALSE){scaling\_nu8=read.table(paste("film10","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

scaling\_nu80=read.table(paste("film17","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

}

deltaeta\_ecdfneg8=subset(scaling\_nu8[,c(1,2)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,2])))

deltaeta\_ecdfpos8=subset(scaling\_nu8[,c(1,3)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,3])))

deltaeta\_tdfneg8=subset(scaling\_nu8[,c(1,8)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,8])))

deltaeta\_tdfpos8=subset(scaling\_nu8[,c(1,9)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,9])))

#

deltaeta\_ecdfneg80=subset(scaling\_nu80[,c(1,2)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,2])))

deltaeta\_ecdfpos80=subset(scaling\_nu80[,c(1,3)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,3])))

deltaeta\_tdfneg80=subset(scaling\_nu80[,c(1,8)], (!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,8])))

deltaeta\_tdfpos80=subset(scaling\_nu80[,c(1,9)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,9])))

#

ylim=c(1/(2\*count),1)

ylab="CDF"

xmax=max(deltaeta\_ecdfpos80,deltaeta\_ecdfneg80,deltaeta\_tdfpos80,

deltaeta\_ecdfpos8,deltaeta\_ecdfneg8,deltaeta\_tdfpos8)

logmax=trunc(log10(xmax))+1

logmax=10^logmax

linmax=(logmax/10)\*(trunc(xmax/(logmax/10))+1)

xlab=expression(paste("|",Delta,"",eta,"| (Pa.s)"))

plot(deltaeta\_ecdfneg80,cex.lab=1.5,col="red",log="xy",ylim=ylim,xlim=c(0.0001,logmax),

ylab=ylab,

xlab=xlab)

points(deltaeta\_ecdfpos80,pch=20,cex=0.5)

#lines(deltaeta\_tdfneg80,col="red")

lines(deltaeta\_tdfpos80,lty=3)

points(deltaeta\_ecdfneg8,pch=20,col="red")

points(deltaeta\_ecdfpos8)

#lines(deltaeta\_tdfneg8,col="red")

lines(deltaeta\_tdfpos8,lty=3)

legend("bottomleft",cex=1,

c(expression(paste("Negative, ",omega,"=5.3 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=0.55 ",rad.s^-1)),

expression(paste("Positive"))),

pch=c(1,20,20,1),

col=c("red","black","red","black"))

#

plot(deltaeta\_ecdfneg80,cex.lab=1.5,col="red",log="y",ylim=ylim,xlim=c(0,linmax),

ylab=ylab,

xlab=xlab)

points(deltaeta\_ecdfpos80,pch=20,cex=0.5)

#lines(deltaeta\_tdfneg80,col="red")

lines(deltaeta\_tdfpos80,lty=3)

points(deltaeta\_ecdfneg8,pch=20,col="red")

points(deltaeta\_ecdfpos8)

#lines(deltaeta\_tdfneg8,col="red")

lines(deltaeta\_tdfpos8,lty=3)

####end of scaling law 1 order of magnitude apart

####0###49#6######17################23#########28

####scaling law 0.089 Hz, 0.104 Hz, 0.207 Hz and 0.847 Hz Mantegna and Stanley (scaled to sd)

#in case of series 3 all FALSE

#series 1

if(TRUE){

films=c("film10","film15","film21")

nus=c(0.65,1.3,5.3)

}

#series 2

if(FALSE){

films=c("film12","film13","film17")

nus=c(0.76,0.94,5.7)

}

#correct\_scaling\_film=NULL

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

all\_scaled\_datax=NULL#input data of negative and positive values for fitdistr

all\_scaled\_datay=NULL

plot\_datax=NULL

#plotfile=TRUE for making one file else 4 files txt with a cdf version

plotfile=FALSE

if(plotfile){

plot\_index=list()

for (i in 1:(length(films)\*2)){plot\_index[i]=NULL}

#plot\_index=1>8 expcdfng\_film9 > \_film21

cdf\_=NULL

}

# to get the required data from the 4 film txt files and plotting (for publication)

# for series of frequencies/films make a plot

for (i in 1:length(films)){

file=paste(films[i],"plot\_correct\_scaling.txt",sep="")

correct\_scaling=read.table(file=file,header=TRUE)

#going from 3 columns - x y1 (>0) y2 (<0/made abs) - to 2, x y for fitdistr function

#uneven nr are negative deltas

nr=2\*i-1

cdf=subset(correct\_scaling[,c(1,2)],(!is.na(correct\_scaling[,1]) & !is.na(correct\_scaling[,2])))

cdf\_neg=-cdf[,1]#all absolute negative values > negative for fitdistr for uneven nr

all\_scaled\_datax=c(all\_scaled\_datax,cdf\_neg)#for fitdistr

plot\_datax=c(plot\_datax,cdf[,1])

#plot for Rstudio

col="red"

log="y"

if(nr==1){plot(cdf,cex.lab=1.5,log=log,type="p",pch=1,col=col,cex=2/nr,

ylim=c(0.001,1),

xlim=c(0.0001,10),

ylab="CDF",

#xlab=expression(paste("|",Delta,"",eta,"(",omega,")| / ",sigma,"(",omega,")"))

xlab=expression(paste("|",Delta,"",eta,"| / ",sigma,"(",Delta,"",eta,")"))

)

}else{points(cdf,col=col,pch=1,cex=2/nr)}

#

if(plotfile){

plot\_index[[nr]]=c(plot\_index[[nr]],cdf[,2])

for (index in (nr+1):8){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_neg)))}

if(nr>1){for (index in 1:(nr-1)){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_neg)))}}

}

#

#even nr are positive deltas

nr=2\*i

#all positive values remain positive to calculate fitdistr

cdf=subset(correct\_scaling[,c(1,3)],(!is.na(correct\_scaling[,1]) & !is.na(correct\_scaling[,3])))

cdf\_=cdf[,1]

all\_scaled\_datax=c(all\_scaled\_datax,cdf\_)

plot\_datax=c(plot\_datax,cdf[,1])

#plot for Rstudio

col="black"

points(cdf,col=col,pch=1,cex=2/(nr-1))

if(plotfile){

plot\_index[[nr]]=c(plot\_index[[nr]],cdf[,3])

for (index in 1:(nr-1)){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_)))}

for (index in (nr+1):8){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_)))}

}

}

#print(all\_data)

#lognormal="false"

#Distribution\_check(all\_scaled\_datax,lognormal)

Param=fitdistr(all\_scaled\_datax,"t",start = list(m=mean(all\_scaled\_datax),s=sd(all\_scaled\_datax), df=3), lower=c(-0.05,0.5,1),upper=c(0.05,50,50))

tm =coef(Param)[1]

ts =coef(Param)[2]

tdf=coef(Param)[3]

t\_general=function(x,mu,sigma,nu){

1/sigma\*

gamma((nu+1)/2)/

sqrt(nu\*pi)/gamma(nu/2)\*

(1+((x-mu)^2)/nu/sigma/sigma)^(-(1+nu)/2)

}

#delta\_eta is the key variable and equal to all\_scaled\_datax

delta\_eta=all\_scaled\_datax

count=length(delta\_eta)

min\_eta=min(delta\_eta)

max\_eta=max(delta\_eta)

if(abs(min\_eta)>max\_eta){

max\_eta=abs(min\_eta)

}else{

min\_eta=-max\_eta

}

DELTA=max\_eta-min\_eta

deltaeta\_tstudent=NULL

cdft=NULL

tdist=NULL

#negcfdt=NULL

#poscfdt=NULL

cdft[1]=t\_general(min\_eta,tm,ts,tdf)

tdist[1]=cdft[1]

deltaeta\_tstudent[1]=min\_eta

for(i in 2:count){

deltaeta\_tstudent[i]=min\_eta+(i-1)\*DELTA/count

if(abs(deltaeta\_tstudent[i])<0.0001){

deltaeta\_tstudent[i]=0.0001}

tdist[i]=t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

cdft[i]=cdft[i-1]+t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

}

cdft=cdft/cdft[count]

deltaeta\_tstudent\_neg=abs(deltaeta\_tstudent[deltaeta\_tstudent<0])

negcdft=cdft[deltaeta\_tstudent<0]

deltaeta\_tstudent\_pos=deltaeta\_tstudent[deltaeta\_tstudent>=0]

poscdft=1-cdft[deltaeta\_tstudent>=0]

count\_tstudent\_neg =length(deltaeta\_tstudent\_neg)

count\_tstudent\_pos =length(deltaeta\_tstudent\_pos)

#points(log="xy",xneg,yneg,pch=17,cex=1.5,col="red")

#points(xpos,ypos,pch=20)

lines(deltaeta\_tstudent\_neg,negcdft,lwd=2,col='red')

lines(deltaeta\_tstudent\_pos,poscdft,lwd=2,lty=2)

mtext(cex=0.75,paste("mean= ",round(tm,2)," df= ",round(tdf,2)," scale= ",round(ts,2)),3,1)

if(log=="xy"){position="bottomleft"}else{position="topright"}

#in case of series 3 all FALSE

if(TRUE){#Series 1

legend(position,cex=1,

#change frequencies when necessary series 1 0.104, 0.207, 0.847

#nus=c(0.65,1.3,5.3)

c(expression(paste("Negative, ",omega,"=0.65 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=1.3 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=5.3 ",rad.s^-1)),

expression(paste("Positive"))),

pt.cex=c(1.5,1.5,1,1,0.5,0.5),

pch=c(1,1,1,1,1,1),col=c("red","black","red","black","red","black"))

}

if(FALSE){#Series 2

legend(position,cex=1,

#change frequencies when necessary series 1 0.104, 0.207, 0.847

#nus=c(0.76,0.94,5.7)

c(expression(paste("Negative, ",omega,"=0.76 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=0.94 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=5.7 ",rad.s^-1)),

expression(paste("Positive"))),

pt.cex=c(1.5,1.5,1,1,0.5,0.5),

pch=c(1,1,1,1,1,1),col=c("red","black","red","black","red","black"))

}

Format=function(number){sprintf("%3.6f",number)}#to make stardardized Format output

plot\_cdfmasterstudent=data.frame(Format(c(deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)),

Format(c(negcdft,rep(NA,count\_tstudent\_pos))),

Format(c(rep(NA,count\_tstudent\_neg),poscdft)))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_cdfmasterstudent,quote=FALSE,file="plot\_mastercurve\_Student.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

#make a dataframe

#series 1 films 9, 10, 15, 21

#series 2 films 12, 13, 17

if(plotfile){

all\_scaled\_datax=c(all\_scaled\_datax,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

negcdft=c(rep(NA,length(all\_scaled\_datax)),negcdft,rep(NA,count\_tstudent\_pos))

poscfdt=c(rep(NA,length(all\_scaled\_datax)),rep(NA,count\_tstudent\_neg),poscdft)

#assigning the right plot\_index[[i]] (a list) to the right film

if(TRUE){

plot\_expcdfneg\_film9=plot\_index[[1]]

plot\_expcdfpos\_film9=plot\_index[[2]]

plot\_expcdfneg\_film10=plot\_index[[3]]

plot\_expcdfpos\_film10=plot\_index[[4]]

plot\_expcdfneg\_film15=plot\_index[[5]]

plot\_expcdfpos\_film15=plot\_index[[6]]

plot\_expcdfneg\_film21=plot\_index[[7]]

plot\_expcdfpos\_film21=plot\_index[[8]]

plot\_all\_data=data.frame(Format(plot\_datax),

Format(plot\_expcdfneg\_film9),

Format(plot\_expcdfpos\_film9),

Format(plot\_expcdfneg\_film10),

Format(plot\_expcdfpos\_film10),

Format(plot\_expcdfneg\_film15),

Format(plot\_expcdfpos\_film15),

Format(plot\_expcdfneg\_film21),

Format(plot\_expcdfpos\_film21),

Format(negcfdt),

Format(poscfdt))

}

if(FALSE){

plot\_expcdfneg\_film12=plot\_index[[1]]

plot\_expcdfpos\_film12=plot\_index[[2]]

plot\_expcdfneg\_film13=plot\_index[[3]]

plot\_expcdfpos\_film13=plot\_index[[4]]

plot\_expcdfneg\_film17=plot\_index[[5]]

plot\_expcdfpos\_film17=plot\_index[[6]]

plot\_all\_data=data.frame(Format(plot\_datax),

Format(plot\_expcdfneg\_film12),

Format(plot\_expcdfpos\_film12),

Format(plot\_expcdfneg\_film13),

Format(plot\_expcdfpos\_film13),

Format(plot\_expcdfneg\_film17),

Format(plot\_expcdfpos\_film17),

Format(negcfdt),

Format(poscfdt))

}

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_all\_data,quote=FALSE,file="plot\_mastercurve\_delta\_eta\_scaled.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

}# End of scaling deltas

#

Rawdata\_RDT=function(analyserawdata){

# either testing (of raw data or reading txt files and writing them to excel files -initial setup)

# first testing (of raw data)

setwd("/Users/macbookair/Desktop/Series/InputData")

#Extensive Testing (time (etamin) etamin time (etamax) etamax)

#setwd("/Users/macbookair/Desktop/BerretMicroRheology/Berret/Data/Raw data txt files")

#if KO go to line 169

### AnalyseRawData must still be written

if(analyserawdata=="OK"){

dataset = read.delim("Recol\_film1 57mHz.txt", sep = "\t", nrows=500, header = TRUE, stringsAsFactors = FALSE)

dim(dataset)

#

# skip empty files / test the size of each file， and skip the file size of 0

#for (file in list.files(,"\*.txt")){

# if (file.size(file) == 0) next

# print(file)

#}

# In case header=true: first row (number 1 = first row of matrix ) is data

# check below with Pil=OK and row=1 / print individual lines Pil=OK and row=number

Pil="OK"

if (Pil=="OK")

{

#give row number

row=1800

if (is.na(dataset[row,3])){

print("stop due to NA")

print(row)}

firstline=c(dataset[row,1],dataset[row,2],dataset[row,3])

print (firstline)

}

for (i in 1000:3000){

if (is.na(dataset[i,3])){

print("stop due to NA")

print(i-1)

iend=i-1

print(dataset[i-1,3])

print(dataset[i,3])

break}

}

quit(save="ask")

}

dim=length(File)

#combined\_eta\_all=NULL

for (i\_t\_eta in 1:dim){

#print(File[i\_t\_eta])

t\_eta = read.delim(File[i\_t\_eta], sep = "\t", nrows=500, header = TRUE, stringsAsFactors = FALSE)

print(t\_eta)

#check for end of file / determination of Dimension of data - Dimdata

for (i in 1:500){

if (is.na(t\_eta[i,5])){

Dimdata=i-1

break}

}

t=head(unlist((t\_eta[,1])),Dimdata)

eta=head(unlist((t\_eta[,5])),Dimdata)

df\_t\_eta=data.frame(t,eta)

WriteXLS(df\_t\_eta, paste((File[i\_t\_eta]),".xlsx",sep=""), SheetNames = "Checkdata")

#combined\_eta\_all=c(combined\_eta\_all,eta)

}

}# End of Rawdata reading etc.

Variogram\_experimental=function(film,log\_eta,Dimdata,simulation){

#Variogram\_output=function(film,log\_aeta,Dimdata,range\_fraction,simulation,FIT,Variogram\_results){

#setting range for fitting: either a fraction or a number of lags=7 e.g.

#range=range\_fraction

#range fraction is also the choice parameter for variogram calculation < 1: yes!

#if(range\_fraction<1){

#range=range\_fraction\*Dimdata

deltaetasq=NULL

plotlag=NULL

plotdeltaetasq=NULL

for (i in 1:Dimdata){deltaetasq[i]=0}

explag=NULL

variogram=NULL

variogramerror=NULL

# All lags to be calculated

for (lag in 1:(Dimdata - 1)){

Sumn=0

for (i in 1:Dimdata){

if (i + lag > Dimdata) {

#final correction 0.5 or 1/2 in variogram formula first evaluation of data series

#We took the definition finaly from Bouchaud 4.24 (without factor 0.5 - check paper)

variogramlag=Sumn/(Dimdata-lag)

if(FALSE){

somerror=0

for (ierror in 1:(Dimdata-lag)){

# print(deltaetasq[ierror]-variogram)

somerror=somerror+(deltaetasq[ierror]/2-variogram)^2

}

variogramerror=sqrt(somerror/(Dimdata-lag))

variogramerror=c(variogramerror,variogramerror)

}

} else

{

deltai=log\_eta[i]-log\_eta[i+lag]

deltaetasq[i]=deltai\*deltai

Sumn=Sumn+deltaetasq[i]

}

}

if(lag==1){r=hist(deltaetasq/2,breaks=Dimdata-lag,plot=FALSE)

x=r$breaks[-1]

y=r$counts

Model\_powerlaw=function(x,C0\_Vcloud\_constant,C1\_Vcloud\_exponent){C0\_Vcloud\_constant\*x^C1\_Vcloud\_exponent}

powerlaw\_fit=nlsLM(y~Model\_powerlaw(x,C0\_Vcloud\_constant,C1\_Vcloud\_exponent),start=list(C0\_Vcloud\_constant=10,C1\_Vcloud\_exponent=-1))

#print(summary(powerlaw\_fit))

if(FALSE){

plot(x, y,cex.lab=1.5, xlab="variogram terms", ylab="frequency", log='xy', type='p')

x=(1:300)/1000

C0\_Vcloud\_constant=coef(powerlaw\_fit)[1]

C1\_Vcloud\_exponent=coef(powerlaw\_fit)[2]

lines(x,Model\_powerlaw(x,C0\_Vcloud\_constant,C1\_Vcloud\_exponent),col="red")

mtext(cex=0.75,paste0("cte= ",round(C0\_Vcloud\_constant,3)," power= ",round(C1\_Vcloud\_exponent,3)),3,0.5)

}

}

explag=c(explag,lag)

variogram=c(variogram,variogramlag)

}

# lag 12 analysis

# GRW

alfa=variogram[2]/variogram[1]-1

sigma\_sigma\_alfa=variogram[2]/2

# OUP

epsilon=log(1/alfa)

sigma\_sigma\_epsilon=epsilon\*variogram[1]/(1-alfa)

# output

#plot(explag,variogram,pch=20,col="grey",ylim=c(min(variogram),max(variogram)), xlim=c(0,Dimdata))

#mtext(cex=0.75,paste0(film),3,0)

# make datframes before writing to XL

dfvar=data.frame(explag,variogram)

#plot for publication

file=paste(film,"\_variogram.txt",sep="")

write.table(format(dfvar,digits=3),file=file,na="",quote=FALSE,sep=" ",col.names=TRUE,row.names=FALSE)

#

dflag12=data.frame(alfa,sigma\_sigma\_alfa,epsilon,sigma\_sigma\_epsilon)

#pragmatic way to avoid variogram of delta eta not to be saved (usually FIT with constant; FIT=5)

# writing output of the variogram data (FIT=5 usually for constant variogram of volatilities)

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

WriteXLS(dfvar, paste(film,"variogram.xlsx",sep=""))

WriteXLS(dflag12, paste(film,"variogram\_lag12.xlsx",sep=""))

}# End of experimental variogram calculation (incl. lag 1-2 etc, alfa, epsilon, sigma2)

fit\_general=function(film,fitlag,fitvariogram,averagetime,FIT,pos\_line){

lwd=2

if (FIT==0){

Modelhole=function(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_GRW,C3\_GRW)

{C0\_constant\_cosinus\*(1-cos(2\*pi\*averagetime\*fitlag/C1\_T))+

2\*C2\_constant\_GRW\*(1-C3\_GRW^fitlag)/(1-C3\_GRW^2)}

#in case of damped cosine

#/(6.28\*fitlag/C2))

maxvar=max(fitvariogram)

#0.5 range covered by periodic function on Variogram axis

fraction\_period=0.5

C0l=fraction\_period\*maxvar/5

C0m=C0l\*5

#period=2\*averagetime\*fitlag[which.max(fitvariogram)]

period=150

int=100

# series 1

if(FALSE){

if(film=="film20"|film=="film18"|film=="film21"){

period=150

int=30

}

if(film=="film1"|film=="film19"){

period=80

int=40

}

if(film=="film14"){

period=40

int=20

}

}

# series 2

if(FALSE){

if(film=="film17"){

period=50

int=25

}

}

# series 3

if(FALSE){

if(film=="film20"|film=="film18"){

period=80

int=20

}

}

# Lehigh\_data

if(film=="Lehigh\_data"|film=="Lehigh\_data\_VolatilityClustering"){

period=20

int=10

}

# series 4

if(T){

if(film=="run1-Curo"){

#C0l=0.00005

#C0m=0.00005

period=250

int=25

C3l=0.97

C3m=0.97

#C2l=0.00002

#C2m=0.00002

}

if(film=="run2-Curo+silica"){

#C0l=0.00001

#C0m=0.00001

period=400

int=100

C3l=0.95

C3m=0.95

#C2l=0.00002

#C2m=0.00002

}

if(film=="run3-Curo+alumina"){

period=150

int=50

C3l=0.479

C3m=0.479

}

C1l=period-int

C1m=period+int

C2l=0.1\*C0l

C2m=C0m

# constant C2/C3= fixed

if(F){

C2l=0.036

C2m=C2l

C3l=0.658

C3m=C3l

}

#

C0s=(C0l+C0m)/2

C1s=(C1l+C1m)/2

C2s=(C2l+C2m)/2

C3s=(C3l+C3m)/2

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_GRW,C3\_GRW),

start=list(C0\_constant\_cosinus=C0s,C1\_T=C1s,C2\_constant\_GRW=C2s,C3\_GRW=C3s),

lower=c(C0l,C1l,C2l,C3l),

#upper=c(C0l,C1l,C2l,C3l))

upper=c(C0m,C1m,C2m,C3m))

# C0 amplitude C1 period C2 alfa C3 sigma^2

C0\_constant\_cosinus =coef(variogramfit)[1]

C1\_T =coef(variogramfit)[2]

C2\_constant\_GRW =coef(variogramfit)[3]

C3\_GRW =coef(variogramfit)[4]

#lines(lwd=lwd,predict(variogramfit),col='red')

x=(10:(10\*length(fitlag)))/10

lines(lwd=lwd,predict(variogramfit,list(x)),col='red')

lines(lwd=lwd,x,2\*C2\_constant\_GRW\*(1-C3\_GRW^x)/(1-C3\_GRW^2),col="blue")

mtext(cex=0.75,paste0("T(s)=",round(C1\_T,1)," alfa=",round(C3\_GRW,3)," sigma^2=",round(C2\_constant\_GRW,3)),3,pos\_line)

# plot for publication

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

GRW=2\*C2\_constant\_GRW\*(1-C3\_GRW^fitlag)/(1-C3\_GRW^2)

Periodic\_plus\_GRW=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,Periodic\_plus\_GRW,GRW)

file=paste(film,"plot\_fit\_variogram\_GRW.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,na="",quote=FALSE,sep=" ",col.names=TRUE,row.names=FALSE)

}

}

}

#(damped) cosinus with Ornstein-Uhlenbeck contribution (Mean Reverting Mechanism)

if (FIT==1){

Modelhole=function(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_OU,C3\_correlationtime)

{C0\_constant\_cosinus\*(1-cos(2\*pi\*averagetime\*fitlag/C1\_T))+

C2\_constant\_OU\*(C3\_correlationtime/averagetime)\*(1-exp(-fitlag\*averagetime/C3\_correlationtime))}

#in case of damped cosine

#/(6.28\*fitlag/C2))

maxvar=max(fitvariogram)

#0.5 range covered by periodic function on Variogram axis

fraction\_period=0.5

C0l=fraction\_period\*maxvar/5

C0m=C0l\*5

#C1=6.28/estimated period = 100 lags > 0.0628

#period=2\*fitlag[which.max(fitvariogram)]

#period=2\*averagetime\*fitlag[which.max(fitvariogram)]

period=150

int=100

# all FALSE for series 3

# series 1

if(TRUE){

if(film=="film20"|film=="film18"|film=="film21"){

period=120

int=30

}

if(film=="film1"|film=="film19"){

period=80

int=40

}

if(film=="film14"){

period=40

int=20

}

}

# series 2

if(FALSE){

if(film=="film17"){

period=50

int=25

}

}

# series 3

if(FALSE){

if(film=="film20"|film=="film18"){

period=80

int=20

}

}

if(FALSE){

# Lehigh\_data

if(film=="Lehigh\_data"|film=="Lehigh\_data\_VolatilityClustering"){

period=20

int=10

}

}

# series 4

if(FALSE){

if(film=="run1-Curo"){

#C0l=0.00005

#C0m=0.00005

period=250

int=25

C3l=22

C3m=22

#C2l=0.00002

#C2m=0.00002

}

if(film=="run2-Curo+silica"){

#C0l=0.00001

#C0m=0.00001

period=400

int=100

C3l=22

C3m=22

#C2l=0.00002

#C2m=0.00002

}

if(film=="run3-Curo+alumina"){

period=150

int=50

C3l=0.68

C3m=0.68

}

}

C1l=period-int

C1m=period+int

C2l=0.1\*C0l

C2m=C0m

#C3l=0.01

##C3m=0.5\*fitlag[which.max(fitvariogram)]

#C3m=5\*averagetime

# constant C2/C3= fixed for the master curve

if(FALSE){

C2l=0.058

C2m=C2l

C3l=2.386

C3m=C3l

}

#

C0s=(C0l+C0m)/2

C1s=(C1l+C1m)/2

C2s=(C2l+C2m)/2

C3s=(C3l+C3m)/2

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_OU,C3\_correlationtime),

start=list(C0\_constant\_cosinus=C0s,C1\_T=C1s,C2\_constant\_OU=C2s,C3\_correlationtime=C3s),

lower=c(C0l,C1l,C2l,C3l),

#upper=c(C0l,C1l,C2l,C3l))

upper=c(C0m,C1m,C2m,C3m))

# C0 amplitude C1 period C2 UO prefactor sigma^2 C3 1/relaxation lag

C0\_constant\_cosinus =coef(variogramfit)[1]

C1\_T =coef(variogramfit)[2]

C2\_constant\_OU =coef(variogramfit)[3]

C3\_correlationtime =coef(variogramfit)[4]

#lines(lwd=lwd,predict(variogramfit),col='red')

x=(10:(10\*length(fitlag)))/10

lines(lwd=lwd,predict(variogramfit,list(x)),col='red')

lines(lwd=lwd,x,C2\_constant\_OU\*(C3\_correlationtime/averagetime)\*(1-exp(-x\*averagetime/C3\_correlationtime)),col="blue")

mtext(cex=0.75,paste0("T(s)=",round(C1\_T,1)," tau(s)=",round(C3\_correlationtime,3)," sigma^2=",round(C2\_constant\_OU,3)),3,pos\_line)

# plot for publication - dending on the series

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

OUP=C2\_constant\_OU\*(C3\_correlationtime/averagetime)\*(1-exp(-fitlag\*averagetime/C3\_correlationtime))

Periodic\_plus\_OUP=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,Periodic\_plus\_OUP,OUP)

file=paste(film,"plot\_fit\_variogram\_OUP.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE)

}

}

#GRW mean reverting mechanism

if (FIT==2){

Modelhole=function(fitlag,C0\_constant\_GRW,C1\_GRW){2\*C0\_constant\_GRW\*(1-C1\_GRW^fitlag)/(1-C1\_GRW^2)}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_GRW,C1\_GRW),start=list(C0\_constant\_GRW=0.05,C1\_GRW=0.5))

C0\_constant\_GRW =coef(variogramfit)[1]

C1\_GRW =coef(variogramfit)[2]

lines(lwd=lwd,predict(variogramfit),col='blue')

mtext(cex=0.75,paste0("alfa=",round(C1\_GRW,3)," sigma^2=",round(C0\_constant\_GRW,3)),3,pos\_line)

# plot for publication

if(length(fitlag)>7){

GRW=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,GRW)

file=paste(film,"plot\_fit\_variogram\_GRW\_only.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,na="",quote=FALSE,sep=" ",col.names=TRUE,row.names=FALSE)

}

}

#Ornstein Uhlenbeck mean reverting mechanism

if (FIT==3){

Modelhole=function(fitlag,C0\_constant\_OUP,C1\_correlationtime){C0\_constant\_OUP\*(C1\_correlationtime/averagetime)\*(1-exp(-fitlag\*averagetime/C1\_correlationtime))}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_OUP,C1\_correlationtime),start=list(C0\_constant\_OUP=0.1,C1\_correlationtime=5))

C0\_constant\_OUP =coef(variogramfit)[1]

C1\_correlationtime =coef(variogramfit)[2]

lines(lwd=lwd,predict(variogramfit),col='blue')

mtext(cex=0.75,paste0("tau=",round(C1\_correlationtime,3)," sigma^2=",round(C0\_constant\_OUP,3)),3,pos\_line)

# plot for publication

if(length(fitlag)>7){

OUP=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,OUP)

file=paste(film,"plot\_fit\_variogram\_OUP\_only.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE)

}

}

#(damped) cosinus - hast to be rewritten (constant names constant in terms of seconds)

if (FIT==4){

Modelhole=function(fitlag,C0,C1,C2){C0+C1\*(1-cos(C2\*fitlag))}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0,C1,C2),start=list(C0=0.32,C1=0.1,C2=0.05),

lower=c(0.04,0.01,0.01),

upper=c(0.4,0.15,0.5))

C0 =coef(variogramfit)[1]

C1 =coef(variogramfit)[2]

C2 =coef(variogramfit)[3]

lines(predict(variogramfit),col='red')

mtext(cex=0.75,paste0("T= ",round(2\*pi\*averagetime/C2,1)),3,pos\_line)

}

#constant

if (FIT==5){

Modelhole=function(fitlag,C0){C0+0\*fitlag}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0),start=list(C0=1))

C0 =coef(variogramfit)[1]

lines(predict(variogramfit),col='red')

mtext(cex=0.75,paste0("cte= ",round(C0,3)),3,pos\_line)

}

print(summary(variogramfit))

#csv file> , to ; >. to , then text to columns

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

write.table(tidy(variogramfit),paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),append=FALSE)

#print(coef(variogramfit))

#print(confint(variogramfit))

#plot(fitlag,residuals(variogramfit),pch=20,col="grey")

if(length(fitlag)>10){hist(residuals(variogramfit))}

Variogram\_fit=list(summary(variogramfit))

}# End of general fitting (x,y,fit)

Variogram\_theoretical=function(film,Dimdata,averagetime,range\_fraction,simulation,Fit,Variogram\_fit){

# setting range for fitting: either a fraction or a number of lags=7 e.g.

range=range\_fraction

#range fraction is also the choice parameter for variogram calculation < 1: yes!

if(range\_fraction<1){range=range\_fraction\*(Dimdata-1)}

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

variogram\_data = read\_xlsx(paste(film,"variogram.xlsx",sep=""),n\_max = (Dimdata-1),col\_names = TRUE)

explag=unlist((variogram\_data[,1]))

variogram=unlist(variogram\_data[,2])

#put number of lags if you want to have it for a constant number in a series

#random=NULL

#if (rand=="OK"){

# fitlag=explag

# random=variogram

# randomvariogram=head(random,range)

#}

fitlag=head(explag,range)

fitvariogram=head(variogram,range)

if(max(fitvariogram)<mean(fitvariogram)+sd(fitvariogram)){FIT=FIT+1}

ymin=0.9\*min(fitvariogram)

ymax=1.1\*max(fitvariogram)

xlab=expression(italic(Lag~l))

ylab=expression(italic(Variogram))

plot(fitlag,fitvariogram,xlab=xlab,ylab="",type="l",ylim=c(0,ymax), xlim=c(0,range),cex.lab=1.5)

title(ylab=ylab,line=2.5,cex.lab=1.5)

pos\_line=1

#simulation yes or no

if(simulation=="yes"){

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

WriteXLS(df, paste(amplitude,nu,trend,"simulationvariogram.xlsx",sep=""), SheetNames = "variogram")

mtext(cex=0.75,paste0("amplitude= ",amplitude,", period= ",round(1/nu,1)," s, trend= ",round(trend,1)),3,0)

pos\_line=0

}

# FIT=0 hole + mean reverting GRW; 1 hole + mean reverting (Ornstein Uhlenbeck)

# 2 mean reverting GRW; 3 mean reverting OU mechanism; 4 cosinus; 5 constant

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

}# End of theoretical variogram fit (a film file must be read)

#

####

# START of program

####

#

# overall data reading / testing raw data - reading viscosity files & analysis combining files

#

# first\_reading\_testing\_rawdata ="no", default because data have been looked at already (just when there are new data)

# data reading all files saving to separate excel files

first\_reading\_testing\_rawdata="no"

if(first\_reading\_testing\_rawdata=="yes"){

analyserawdata="KO"

Rawdata\_RDT(analyserawdata)

}

#

# default equidistant = "yes" (which make equidistant or "no" to treat raw data or already equidistant data)

# simulation = "yes" only variogram is done on a lognormally generated sample or an another model

# > variogram theory can be done

# > assumed equidistant = "yes" for the x points (time axis) - by default

# simulation ="no" > reading of experimental data + set equidistant = "yes" or "no"

# rand = OK for random sampling; rand !="OK" so no randomization via sample

# (demonstration of the uniquen features of the time series results)

#

# Data files to be analyzed (TRUE assigns the data to be analyzed)

# if(TRUE){datablock: Film\_series,Film\_frequencies}

#

ID\_path="/Users/macbookair/Desktop/Series/InputData"

print(ID\_path)

Film\_series=NULL

alldatafiles=list.files(path=ID\_path,".txt",full.names=TRUE)

#print(length(alldatafiles))

characteristic\_string="film"#characteristic string in file definition

#Lehigh in case of Lehigh data, run in case of series 4

characteristic\_string="run"

for (i in 1:length(alldatafiles)){

if(length(alldatafiles)==1){characteristic\_string="Lehigh"}

start\_filmstring=(unlist(gregexpr(pattern=characteristic\_string,alldatafiles[i]))[1])

if (data=="run"){

end\_filmstring=(unlist(gregexpr(pattern="-",alldatafiles[i]))[1])-1

}else{

end\_filmstring=(unlist(gregexpr(pattern=".txt",alldatafiles[i]))[1])-1

}

films=substr(alldatafiles[i],as.numeric(start\_filmstring),as.numeric(end\_filmstring))

Film\_series=c(Film\_series,films)

}

# Analysis settings

simulation="no"#no simulations - needs to be checked

variogram\_theory="no"#no GRW alfa variations - needs to be checked

make\_equidistant="yes"

rand="K"#"OK"=randomize (all)

#

# Initialisation of outputarrays

#

Film\_frequencies=NULL

Film\_acf\_tau=NULL

Film\_acf\_T=NULL

Film\_maxV\_T=NULL

Film\_ito\_tau=NULL

Film\_eta\_rawdata=NULL

Film\_eta\_no\_outliers=NULL

Film\_eta\_detrended=NULL

Film\_eta\_ito=NULL

Film\_eta=NULL

Film\_sdetamin\_rawdata=NULL

Film\_sdetamin\_no\_outliers=NULL

Film\_sdetamin\_detrended=NULL

Film\_sdetamin=NULL

Film\_sdetaplus\_rawdata=NULL

Film\_sdetaplus\_no\_outliers=NULL

Film\_sdetaplus\_detrended=NULL

Film\_sdetaplus=NULL

Film\_skewness\_rawdata=NULL

Film\_skewness\_no\_outliers=NULL

Film\_skewness\_detrended=NULL

Film\_skewness=NULL

Film\_kurtosis\_rawdata=NULL

Film\_kurtosis\_no\_outliers=NULL

Film\_kurtosis\_detrended=NULL

Film\_kurtosis=NULL

Film\_averagetime=NULL

Film\_outliers=NULL

Film\_numberofdata=NULL

Film\_alfa=NULL

Film\_ito\_alfa=NULL

Film\_sigma\_sigma\_alfa=NULL

Film\_sigma\_sigma\_ito=NULL

Film\_epsilon=NULL

Film\_tau=NULL

Film\_sigma\_sigma\_epsilon=NULL

Film\_sigma\_alfa\_eta=NULL

Film\_sigma\_epsilon\_eta=NULL

Film\_GRW\_constant\_cosinus\_all=NULL

Film\_GRW\_T\_all=NULL

Film\_GRW\_constant\_all=NULL

Film\_GRW\_alfa\_all=NULL

Film\_OU\_constant\_cosinus\_all=NULL

Film\_OU\_T\_all=NULL

Film\_OU\_constant\_all=NULL

Film\_OU\_epsilon\_all=NULL

Film\_OU\_tau\_all=NULL

Film\_GRW\_constant\_short\_time=NULL

Film\_GRW\_alfa\_short\_time=NULL

Film\_OU\_constant\_short\_time=NULL

Film\_OU\_epsilon\_short\_time=NULL

Film\_OU\_tau\_short\_time=NULL

Film\_GRW\_amplitude\_sigma=NULL

Film\_OU\_amplitude\_sigma=NULL

Film\_GRW\_sigma\_eta\_all=NULL

Film\_GRW\_sigma\_eta\_short\_time=NULL

Film\_OU\_sigma\_eta\_all=NULL

Film\_OU\_sigma\_eta\_short\_time=NULL

y\_GRW\_master\_plot=NULL

x\_GRW\_master\_plot=NULL

y\_OUP\_master\_plot=NULL

x\_OUP\_master\_plot=NULL

#

# Start of analysis loop for series of films

#

#output of fit data from screen

sink("Fit Output.txt",append=FALSE)

#

cat("#########################################",sep="\n")

cat(Film\_series,sep="\n")

cat("#########################################",sep="\n")

for(i in 1:length(Film\_series)){

# yes: simulation of data and t. eta using lognormal distribution

# in the new version this part should be eliminated

if(simulation=="yes"){

# in the case of random log generated numbers

# generation of nr random numbers logn distributed (nr=300, with a mean value of 20 and sdev 2.5)

# rlnorm(300, log(20), log(2.5))

# adding a trend and a cosine function

nr=300

t=c(1:nr)

if(variogram\_theory=="yes"){

ialfa=0

alfa=NULL

# change alfa

alfa[1]=0.5

alfa[2]=0.7

alfa[3]=0.9

alfa[4]=0.99

theor\_variogram\_norm=NULL

for (ialfa in 1:4){

theor\_variogram=(1-alfa[ialfa]^t)/(1-(alfa[ialfa])^2)

for (i in 1:(nr-1)){

theor\_variogram\_norm[i]=theor\_variogram[i]/theor\_variogram[1]

}

power=log(theor\_variogram\_norm[2])/log(2)

print(power)

plot(head(t,(nr-1)),theor\_variogram\_norm,cex.lab=1.5,type='l',log="xy")

lines(t,t,col="red")

mtext(cex=0.75,paste0("alfa = ",alfa[ialfa]," power = ",round(power,2)),3,0)

}

}else{

amplitude=20

phase=1

period=150

nu=1/period

trend=0

alfa=0.5

x=NULL

sd=1.5

theor\_variogram\_norm=NULL

theor\_variogram=NULL

average\_viscosity=20

ksi\_previous=rlnorm(1, log(average\_viscosity), log(sd))

x[1]=ksi\_previous-average\_viscosity

# BIG Q: do I need to include periodic function in the random walk loop???

for (i in 2:nr){

x[i]=(alfa\*x[i-1] +

# from Harvey

# https://books.google.be/books?hl=nl&lr=&id=Kc6tnRHBwLcC&oi=fnd&pg=PR9&dq=harvey,+a.+c.+(1989)

amplitude \* (cos(6.28\*nu\*i) - cos(6.28\*nu\*(i-1))) +

ksi\_previous-average\_viscosity +

trend)

ksi\_previous=rlnorm(1, log(average\_viscosity), log(sd))

for (k in 1:10){

if ((abs(x[i]+average\_viscosity)<average\_viscosity/5)){

ksi\_previous=rlnorm(1, log(average\_viscosity), log(sd))

x[i]=(alfa\*x[i-1]+

amplitude \* (cos(6.28\*nu\*i) - cos(6.28\*nu\*(i-1))) +

ksi\_previous-average\_viscosity +

trend)

}else{

break

}

}

}

theor\_variogram[1]=1/(1+alfa)

for (i in 1:(nr-1)){

theor\_variogram\_norm[i]=(1-alfa^i)/(1-alfa^2)/theor\_variogram[1]

}

power=log(theor\_variogram\_norm[2])/log(2)

# plot of normalized variogram

plot(head(t,(nr-1)),theor\_variogram\_norm,cex.lab=1.5,type='l',log="xy")

lines(t,t,col="red")

mtext(cex=0.75,paste0("alfa = ",alfa," power = ",round(power,2)),3,0)

# plot of generated viscosities

plot(t,x,cex.lab=1.5,type="l",ylim=c(min(x),max(x)+average\_viscosity),xlim=c(0,max(at)))

#plot(at,x,cex.lab=1.5,type='l',lwd='1', xlim=c(0,max(at)))

mtext(cex=0.75,paste0("amplitude= ",amplitude,", period= ",round(period,1)," s, trend= ",round(trend,1)),3,0)

# transform x log to linear by exponential add ksi linear and take log

eta=abs(x+average\_viscosity)

points(t,eta,pch=20)

dfsimulation=data.frame(t,eta)

WriteXLS(dfsimulation, paste("RW Periodic Simulation.xlsx",sep=""), SheetNames = "Simulation")

#

Distribution\_check(eta,lognormal)

#

eta=log(eta)

#plot(cex.lab=1.5,t,eta,pch=20)

original\_numberofdata=nr

film="Simulation"

#

}

# NO: reading of experimental data from excel file

}else{

############# file reading and preparing t, eta data

film=Film\_series[i]

#print(film)

#film="Film21"

#film="film2"

#test=paste("3 - ",film,".txt",sep="")

test=alldatafiles[i]

t\_eta = read.delim(test, sep = "\t", nrows=1000, header = TRUE,stringsAsFactors = FALSE)

if(characteristic\_string=="Lehigh"){

column=2

#give angular frequency (1Hz in Lehigh thesis)

frequency=6.28

}else{#series 1 =5 series 3=4

if(characteristic\_string=="film"){

column=4

frequency=(unlist((t\_eta[1,2])))

}else{

column=2

frequency=0}

}

for (i\_count in 1:1000){

#check column number first in txt files "viscosity (Pa s)" second series (4), first series (5)

if (is.na(t\_eta[i\_count,column])){

Dimdata=i\_count-1

break}

}

Film\_frequencies=c(Film\_frequencies,frequency)

t=head(unlist((t\_eta[,1])),Dimdata)

eta=head(unlist((t\_eta[,column])),Dimdata)

#Lehigh:

if(characteristic\_string=="Lehigh"){eta=0.1\*head(unlist((t\_eta[,column])),Dimdata)}

##conv of dyn/cm2 to Pa \*0.1

# sink() if you want to do tests with printing (need to de-# and re-# sink (output) command)

cat("------------------------------------------------------------",sep="\n")

cat((paste(film," frequency (1/s)= ",round(frequency,3),sep="")),sep="\n")

cat("------------------------------------------------------------",sep="\n")

original\_numberofdata=length(t)

deltat=max(t)/length(t)

#

# 4 plots per page

par(mfrow=c(2,4))

par(xaxs="i",yaxs="i",mar=c(5,5,5,5))

#

#sampling to discard physical process, in case rand="OK" - making it really random

if (rand=="OK"){

for(i\_rand in 1:3){eta=sample(eta)}

} # End of randomization test (check uniqueness of time series)

# plot of experimental data

#first plot title

if (sd(eta)/mean(eta)>0.01){

ymin=exp(trunc(min(log(eta))))

ymax=exp(trunc(max(log(eta)))+1)

if(characteristic\_string=="run"){ymax=exp(max(log(eta)))}

}else{

ymin=min(eta)

ymax=max(eta)

}

ylim=c(ymin,ymax)

ylab=expression(paste(eta," ",(Pa.s)))

xlab="t"

if(characteristic\_string=="Lehigh"){

ylab=expression(paste(G,"' ",(Pa))) # freq. NOT angular freq. (Berret date have been corrected)

}

plot(t,eta,cex.lab=1.5,log="y",xlab=xlab,ylab=ylab,type="l",ylim=ylim)

title(film,line=2.5)

#eta\_uncleaned=eta

mtext(cex=0.75,paste0("ang. freq= ",round(frequency,3)," rad/s"),3,1)

# mean+sd calculation (1) + tracing of values

trace\_meaneta=NULL

trace\_sdeta=NULL

trace\_kurtosiseta=NULL

trace\_skewnesseta=NULL

meaneta=mean(log(eta),na.rm=TRUE)

sdeta=sd(log(eta),na.rm=TRUE)

mykurtosis=function(x){(mean((x-mean(x,na.rm=TRUE))^4)/(sd(x,na.rm=TRUE)^4))-3}

myskewness=function(x){mean((x-mean(x,na.rm=TRUE))^3)/(sd(x,na.rm=TRUE)^3)}

trace\_meaneta=c(trace\_meaneta,meaneta)

trace\_sdeta=c(trace\_sdeta,sdeta)

skewnesseta=myskewness(log(eta))

kurtosiseta=mykurtosis(log(eta))

trace\_skewnesseta=c(trace\_skewnesseta,skewnesseta)

trace\_kurtosiseta=c(trace\_kurtosiseta,kurtosiseta)

#

mtext(cex=0.75,paste0("<eta> ",round(meaneta,2)," sd ",round(sdeta,1)," <eta>lin ",round(exp(meaneta),2)," dt ", round(deltat,2)),3,0)

if(FALSE){

pacf(log(eta),lag.max=original\_numberofdata,main=film)

acf(log(eta),lag.max=original\_numberofdata,main=film)

}

# plot of original and cleaned data

plot(t,eta,cex.lab=1.5,log="y",pch=20,ylim=ylim,ylab=ylab)

# filter for outliers > new dataset with Dimdata

# https://www.r-bloggers.com/whisker-of-boxplot/

#

# extremes on logarithmic data LOG="true" less outliers (more robust)

# more outliers when data is treated on a linear scale LOG=FALSE

# low value outliers

LOG=FALSE

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/Dataprep/")

if(LOG){

lower\_vis=as.numeric(boxplot(log(eta),plot=FALSE)$stats[1])

upper\_vis=as.numeric(boxplot(log(eta),plot=FALSE)$stats[5])

# exp(-1) is the lowest viscosity - to be checked - else exp (-2) (so there are now outliers)

if(film!="Lehigh\_data"){lower\_vis=-1}#so all the data are taken on the low side for microrheology

mtext(cex=0.75,paste0("data filter ",round(exp(lower\_vis),2)," ",round(exp(upper\_vis),2)),3,1)

# plot for publication with elimination of outliers

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

file=paste(film,"eta\_uncleaned\_eta.txt",sep="")

eta\_uncleaned=eta

eta\_NA=eta

eta\_NA[log(eta)<=lower\_vis | log(eta)>=upper\_vis]=NA

eta\_cleaned=eta\_NA

pub\_eta\_uncleaned\_eta\_t=data.frame(sprintf("%.6s",t),sprintf("%.6s",eta\_uncleaned),sprintf("%.6s",eta\_cleaned))

write.table(pub\_eta\_uncleaned\_eta\_t,file=file,quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

t =t [log(eta)>lower\_vis & log(eta)<upper\_vis]

eta=eta[log(eta)>lower\_vis & log(eta)<upper\_vis]

}else{

lower\_vis=as.numeric(boxplot(eta,plot=FALSE)$stats[1])

upper\_vis=as.numeric(boxplot(eta,plot=FALSE)$stats[5])

#no lower outliers: all viscosities are ok in case of microrheology

if(film!="Lehigh\_data"){lower\_vis=0}

mtext(cex=0.75,paste0("data filter ",round(lower\_vis,2)," ",round(upper\_vis,2)),3,1)

# plot for publication with elimination of outliers

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

file=paste(film,"eta\_uncleaned\_eta.txt",sep="")

eta\_uncleaned=eta

eta\_NA=eta

eta\_NA[eta<=lower\_vis | eta>=upper\_vis]=NA

eta\_cleaned=eta\_NA

pub\_eta\_uncleaned\_eta\_t=data.frame(sprintf("%.6s",t),sprintf("%.6s",eta\_uncleaned),sprintf("%.6s",eta\_cleaned))

write.table(pub\_eta\_uncleaned\_eta\_t,file=file,quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

t =t [eta>lower\_vis & eta<upper\_vis]

eta=eta[eta>lower\_vis & eta<upper\_vis]

}

lines(t,eta,col="red")

meaneta=mean(log(eta),na.rm=TRUE)

sdeta=sd(log(eta),na.rm=TRUE)

trace\_meaneta=c(trace\_meaneta,meaneta)

trace\_sdeta=c(trace\_sdeta,sdeta)

skewnesseta=myskewness(log(eta))

kurtosiseta=mykurtosis(log(eta))

trace\_skewnesseta=c(trace\_skewnesseta,skewnesseta)

trace\_kurtosiseta=c(trace\_kurtosiseta,kurtosiseta)

#

Dimdata=length(t)

Film\_outliers=c(Film\_outliers,(original\_numberofdata-Dimdata))

Film\_numberofdata=c(Film\_numberofdata,original\_numberofdata)

mtext(cex=0.75,paste0("original data # ",original\_numberofdata," new data # ",Dimdata),3,2)

mtext(cex=0.75,paste0("<eta> ",round((meaneta),2)," sd ",round((sdeta),2)," <eta> ",round(exp(meaneta),2)),3,0)

#distribution check of cleaned data

lognormal="true"

Distribution\_check(eta,lognormal)

mtext(cex=0.75,paste0("Data without outliers"))

#PACF and ACF of cleaned data

plot\_pacf=pacf(log(eta),lag.max=Dimdata,main=film)

plot\_acf=acf(log(eta),lag.max=Dimdata,main=film)

if(rand=="OK"){break}

# plot for publication

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

df\_pacf=data.frame(plot\_pacf$lag,sprintf("%.6s",plot\_pacf$acf))

write.table(df\_pacf,file=paste(film,"PACF\_cleaned.txt",sep=""),quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

file=paste(film,"ACF\_cleaned.txt",sep="")

df\_acf=data.frame(plot\_acf$lag,sprintf("%.6s",plot\_acf$acf))

write.table(df\_acf,file=paste(film,"ACF\_cleaned.txt",sep=""),quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

#

# Detrending: transformation to log data for eta

# check input data make fit and plots (incl. residuals) + check for trend/trend

log\_eta=log(eta)

# max y and x values

xmax=max(t)+5

#ymax=1.1\*max(log\_eta)

#ymin=0.9\*min(log\_eta)

if (sd(eta)/mean(eta)>0.01){

ymin=exp(trunc(min(log\_eta)))

ymax=exp(trunc(max(log\_eta))+1)

if(characteristic\_string=="run"){ymax=exp(max(log\_eta))}

}else{

ymin=min(eta)

ymax=max(eta)

}

ylim=c(ymin,ymax)

# nonlinear least sq's fit

Modeleta=function(t,trend,cte){t\*trend+cte}

etafit=nlsLM(log\_eta~Modeleta(t,trend,cte),start=list(trend=0.1,cte=0.5\*(max(log\_eta)-min(log\_eta))))

trend =as.numeric(coef(etafit)[1])

cte =as.numeric(coef(etafit)[2])

# plot of fitted curve

plot(t,eta,cex.lab=1.5,log="y",pch=20,xlab="t",ylab=ylab,ylim=ylim,xlim=c(0,xmax))

mtext(cex=0.75,paste0("trend= ",round(trend,5)),3,1)

# detrending mean at average in the center

# calculation of new log\_eta

# new y = [res (y) = y - (cte + trend x t)] + cte + trend x 0.5 x max(t)

# eta\_t = alfa x t + beta + epsilon\_t

# <eta\_t> = alfa x tMAX/2 + beta + 0 (assuming epsilon\_t has zero average)

# assuming a new eta detrended based on <eta>

# eta\_t (detrended) = <eta\_t> + epsilon\_t

# eta\_t (detrended) = alfa x tMAX/2 + beta - (epsilon\_t = eta\_t - alfa x t - beta)

# eta\_t (detrended) = eta\_t - alfa x ( t - tMAX/2 )

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# alternative via FT see: http://www.di.fc.ul.pt/~jpn/r/fourier/fourier.html

# X.k <- fft(trajectory) trajectory=log\_eta

# plot.frequency.spectrum(X.k,xlimits=c(0,acq.freq/2)) acq.freq=Dimdata

log\_eta\_detr=log\_eta+trend\*(0.5\*max(t)-t) # see above derivation

lines(t,exp(log\_eta\_detr),cex=0.5,col="red")

lines(t,exp(Modeleta(t,trend,cte)),type="l",col="black",lwd=2)

etafit\_detr=nlsLM(log\_eta\_detr~Modeleta(t,trend,cte),start=list(trend=0.1,cte=0.5\*(max(log\_eta\_detr)-min(log\_eta\_detr))))

trend0 =as.numeric(coef(etafit\_detr)[1])

cte0 =as.numeric(coef(etafit\_detr)[2])

# mean+sd calculation (3) assuming log-normal distr (mean,sd) + tracing

meaneta=mean(log\_eta\_detr)

sdeta=sd(log\_eta\_detr)

trace\_meaneta=c(trace\_meaneta,meaneta)

trace\_sdeta=c(trace\_sdeta,sdeta)

skewnesseta=myskewness(log\_eta\_detr)

kurtosiseta=mykurtosis(log\_eta\_detr)

trace\_skewnesseta=c(trace\_skewnesseta,skewnesseta)

trace\_kurtosiseta=c(trace\_kurtosiseta,kurtosiseta)

col="red"

if(Dimdata>200){col="white"}

abline(lwd=2,h=exp(meaneta),col=col)

mtext(cex=0.75,paste0("trend= ",round(trend0,5)," eta -trend= ",round(exp(meaneta),2)),3,col="red",0)

# plot for publication

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

paste(film,"detrended\_equidist.txt",sep="")

#plot\_detrending=data.frame(t,sprintf("%.6s",log\_eta),sprintf("%.6s",Modeleta(t,trend,cte)),sprintf("%.6s",log\_eta\_detr),sprintf("%.6s",Modeleta(t,trend0,cte0)))

plot\_detrending=data.frame(t,sprintf("%.6s",exp(log\_eta)),sprintf("%.6s",exp(Modeleta(t,trend,cte))),sprintf("%.6s",exp(log\_eta\_detr)),sprintf("%.6s",exp(Modeleta(t,trend0,cte0))))

write.table(plot\_detrending,file=paste(film,"detrended\_equidist.txt",sep=""),quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

# make linear for lognormal/cauchy/gaussian distribution checks

if(FALSE){

eta\_detr=exp(log\_eta\_detr)

lognormal="true"

Distribution\_check(eta\_detr,lognormal)

mtext(cex=0.75,paste0("Detrended points "))

log\_eta\_detr=log(eta\_detr)

}

#take orginal number of points as number of points to make equidistant

Dimdata=original\_numberofdata

#calculate average time between points

averagetime=max(t)/Dimdata

#choice for lognormalfit yes or no - equidistant yes or no

#parameter for number of equidistant points (=Dimdata)

if (make\_equidistant=="yes"){

aeta=0

#making equidistant with approx

at=unlist((approx (t, log\_eta\_detr, method = "linear", n = Dimdata, rule = 2, ties = mean)[1]))

log\_aeta=unlist((approx (t, log\_eta\_detr, method = "linear", n = Dimdata, rule = 2, ties = mean)[2]))

#plot(approx (t, eta, method = "linear", n = Dimdata, rule = 1, ties = mean),type="p",lwd="1",ylim=c(0,ymax),xlim=c(0,xmax))

# make linear for lognormal/cauchy/gaussian distribution checks

aeta=exp(log\_aeta)

#

ito\_delta\_eta=diff(log\_aeta)

ito\_eta=head(log\_aeta,(Dimdata-1))

plot(ito\_eta,ito\_delta\_eta,main="Ito Calculus ",col="red")

fitline=lm(ito\_delta\_eta~ito\_eta)

abline(fitline,lwd=2)

summary(fitline)

intercept=as.numeric(coefficients(fitline)[1])

slope=as.numeric(coefficients(fitline)[2])

resid\_fit=residuals(fitline)

sigma=sd(resid\_fit)

beta\_0=intercept

beta\_1=slope

alfa\_Mod=1+beta\_1

alfa=-beta\_1/delta\_t

mu\_star=beta\_0/delta\_t/alfa

t12=log(2)/alfa

#print(240 - total time span - \*log(2)/5/ 5 -number of days- in a week /4 -number of days in a month) calculation of Mitsui

sigma\_ito=round(sigma^2,2)

alfa\_ito=round(alfa\_Mod,2)

tau\_ito=round(1/alfa,2)

eta\_inf\_ito=round(exp(mu\_star-sigma\*sigma/2/alfa),2)

mtext(cex=0.75,paste0(

"sigma2= ",sigma\_ito,

" alfa= ",alfa\_ito),3,1)

mtext(cex=0.75,paste0(

" tau= ",tau\_ito,

" s visc\_inf= ",eta\_inf\_ito,

" Pa.s"),3,0)

Film\_ito\_tau=c(Film\_ito\_tau,tau\_ito)

Film\_ito\_alfa=c(Film\_ito\_alfa,alfa\_ito)

Film\_sigma\_sigma\_ito=c(Film\_sigma\_sigma\_ito,sigma\_ito)

Film\_eta\_ito=c(Film\_eta\_ito,eta\_inf\_ito)

meanlog=coef(fitdistr(resid\_fit,"normal"))[1]

sdlog=coef(fitdistr(resid\_fit,"normal"))[2]

plot(ecdf(resid\_fit),cex.lab=1.5,xlim=c(min(resid\_fit),max(resid\_fit)\*1.1),xlab=expression(paste(sigma)))

plot(ecdf(rnorm(1000,(meanlog),(sdlog))),cex.lab=1.5,col="red",add=TRUE,do.points=FALSE)

#

lognormal="true"

Distribution\_check(aeta,lognormal)

mtext(cex=0.75,paste0("Equidistant original data #"),0.5)

log\_aeta=log(aeta)

# plot for publication

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

paste(film,"PACF\_cleaned\_detr\_eq.txt",sep="")

plot\_pacf=pacf(log\_aeta,lag.max=Dimdata,plot=FALSE)

df\_pacf=data.frame(plot\_pacf$lag,sprintf("%.6s",plot\_pacf$acf))

write.table(df\_pacf,file=paste(film,"PACF\_cleaned\_detr\_eq.txt",sep=""),quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

plot\_acf=acf(log\_aeta,lag.max=Dimdata,plot=FALSE)

df\_acf=data.frame(plot\_acf$lag,sprintf("%.6s",plot\_acf$acf))

write.table(df\_acf,file=paste(film,"ACF\_cleaned\_detr\_eq.txt",sep="") ,quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

pacf(log\_aeta,lag.max=Dimdata/2)

analysis\_acf=acf(log\_aeta,lag.max=(Dimdata/2))

lag\_acf=analysis\_acf$lag

T\_acf=2\*averagetime\*lag\_acf[which.min(analysis\_acf$acf[1:(0.8\*max(lag\_acf))])]

small\_lag\_range=28

analysis\_acf=acf(log\_aeta,lag.max=(small\_lag\_range),plot=FALSE)

lag\_acf=averagetime\*analysis\_acf$lag

acf\_value=analysis\_acf$acf

plot(lag\_acf,acf\_value,cex.lab=1.5,pch=20,col="grey",xlab="lag\_acf",xaxs="i",ylim=c(-0.4,1.2))

#

Model\_acf=function(lag\_acf,tau){exp(-lag\_acf/tau)}

acf\_fit=nlsLM(acf\_value~Model\_acf(lag\_acf,tau),

start=list(tau=1),

lower=c(0),upper=c(100))

tau =as.numeric(coef(acf\_fit)[1])

summary(acf\_fit)

Film\_acf\_tau=c(Film\_acf\_tau,tau)

# plot of fitted curve

plot\_acf\_fit=data.frame(lag\_acf,predict(acf\_fit))

lines(plot\_acf\_fit,lwd=2,col="red")

mtext(cex=0.75,paste0("tau = ",round(tau,2)," T= ",round(T\_acf,1)),3,1)

#

#mean calculation of viscosity

#ylim=c(trunc(min(log\_aeta),(trunc(max(log\_aeta)))+1))

xmax=max(at)

# mean+sd calculation (4) assuming log-normal distr (mean,sd) + tracing

meaneta=mean(log\_aeta)

sdeta=sd(log\_aeta)

trace\_meaneta=c(trace\_meaneta,meaneta)

trace\_sdeta=c(trace\_sdeta,sdeta)

skewnesseta=myskewness(log\_aeta)

kurtosiseta=mykurtosis(log\_aeta)

trace\_skewnesseta=c(trace\_skewnesseta,skewnesseta)

trace\_kurtosiseta=c(trace\_kurtosiseta,kurtosiseta)

#

meanetamin=meaneta-sdeta

meanetamax=meaneta+sdeta

ymin=trunc(min(log\_aeta))

ymax=trunc(max(log\_aeta))+1

if(characteristic\_string=="run"){ymax=max(log\_aeta)}

ylim=c(ymin,ymax)

plot(at,log\_aeta,cex.lab=1.5,pch=20,col="grey",xlab="t (s)",ylab=expression(paste(eta,"(t)")),ylim=ylim,xlim=c(0,xmax))

abline(lwd=2,h=meaneta,col="red")

residuals\_eta=log\_aeta-meaneta

}else{

# make linear for lognormal/cauchy/gaussian distribution checks on equidistant points

eta=exp(log\_eta)

lognormal="true"

Distribution\_check(eta,lognormal)

mtext(cex=0.75,paste0("Equidistant initial data # ",3,0.75))

log\_eta=log(eta)

pacf(eta,lag.max=Dimdata,main=film)

acf(eta,lag.max=Dimdata,main=film)

#fit and summary

ymin=0.9\*min(log\_eta)

ymax=1.1\*max(log\_eta)

xmax=max(t)

# mean+sd calculation (4) assuming log-normal distr (mean,sd) + tracing

meaneta=mean(log\_eta)

sdeta=sd(log\_eta)

trace\_meaneta=c(trace\_meaneta,meaneta)

trace\_sdeta=c(trace\_sdeta,sdeta)

skewnesseta=myskewness(log\_eta)

kurtosiseta=mykurtosis(log\_eta)

trace\_skewnesseta=c(trace\_skewnesseta,skewnesseta)

trace\_kurtosiseta=c(trace\_kurtosiseta,kurtosiseta)

meanetamin=meaneta-sdeta

meanetamax=meaneta+sdeta

plot(t,log\_eta,cex.lab=1.5,type="l",ylim=c(ymin,ymax),xlim=c(0,xmax))

mean\_log\_eta=function(t,meaneta){t\*0+meaneta}

residuals\_eta=log\_eta-meaneta

x=1:Dimdata

lines(x,mean\_log\_eta(x,meaneta),type="l",col="red",lwd="1")

}# End of make equidistant routine

mtext(cex=0.75,paste0("<eta> ",round(exp(meaneta),2)," st dev+ ",round(exp(meanetamax)-exp(meaneta),2)," st dev- ",round(exp(meaneta)-exp(meanetamin),2),3,0))

hist(residuals\_eta)

}# End of simulation yes or no

#

# Next part can be run independently when fit an variogram analysis has been done

# Same goes for scaling on all data

#

# variogram / plotting / modelling

#

# range\_fraction < 1 for e.g. 50% of data: 0.5; if range\_fraction = integer > 1 = number of first lags

# simulation - given in main

# Dimdata - given in main

# time series data log\_eta and log\_aeta - given in main

#

Film\_ID=c(1:length(Film\_series))

#Film\_series

#Film\_frequencies

Film\_eta\_rawdata=c(Film\_eta\_rawdata,exp(trace\_meaneta[1]))

Film\_eta\_no\_outliers=c(Film\_eta\_no\_outliers,exp(trace\_meaneta[2]))

Film\_eta\_detrended=c(Film\_eta\_detrended,exp(trace\_meaneta[3]))

Film\_eta=c(Film\_eta,exp(trace\_meaneta[4]))

#

Film\_sdetamin\_rawdata=c(Film\_sdetamin\_rawdata,exp(trace\_meaneta[1]-trace\_sdeta[1]))

Film\_sdetamin\_no\_outliers=c(Film\_sdetamin\_no\_outliers,exp(trace\_meaneta[2]-trace\_sdeta[2]))

Film\_sdetamin\_detrended=c(Film\_sdetamin\_detrended,exp(trace\_meaneta[3]-trace\_sdeta[3]))

Film\_sdetamin=c(Film\_sdetamin,exp(trace\_meaneta[4]-trace\_sdeta[4]))

#

Film\_sdetaplus\_rawdata=c(Film\_sdetaplus\_rawdata,exp(trace\_meaneta[1]+trace\_sdeta[1]))

Film\_sdetaplus\_no\_outliers=c(Film\_sdetaplus\_no\_outliers,exp(trace\_meaneta[2]+trace\_sdeta[2]))

Film\_sdetaplus\_detrended=c(Film\_sdetaplus\_detrended,exp(trace\_meaneta[3]+trace\_sdeta[3]))

Film\_sdetaplus=c(Film\_sdetaplus,exp(trace\_meaneta[4]+trace\_sdeta[4]))

#

Film\_skewness\_rawdata=c(Film\_skewness\_rawdata,trace\_skewnesseta[1])

Film\_skewness\_no\_outliers=c(Film\_skewness\_no\_outliers,trace\_skewnesseta[2])

Film\_skewness\_detrended=c(Film\_skewness\_detrended,trace\_skewnesseta[3])

Film\_skewness=c(Film\_skewness,trace\_skewnesseta[4])

#

Film\_kurtosis\_rawdata=c(Film\_kurtosis\_rawdata,trace\_kurtosiseta[1])

Film\_kurtosis\_no\_outliers=c(Film\_kurtosis\_no\_outliers,trace\_kurtosiseta[2])

Film\_kurtosis\_detrended=c(Film\_kurtosis\_detrended,trace\_skewnesseta[3])

Film\_kurtosis=c(Film\_kurtosis,trace\_kurtosiseta[4])

#

Film\_averagetime=c(Film\_averagetime,averagetime)

#

if(make\_equidistant=="no"){log\_aeta=log\_eta}# log\_aeta is main input for next steps

#

#check if variogram (and lag12) has been calculated already

#setwd("/Users/macbookair/Desktop/BerretMicroRheology/Research/DataAnalysis/Variogramoutput/")

#if(file.access(paste(film,"variogram.xlsx",sep=""),mode=0)==-1){

Variogram\_experimental(film,log\_aeta,Dimdata,simulation)

# for each film the variogram has been written to respective variogram excel file, including the lag12 outcome

# lag12 data are taken from the lage12 excel file

#setwd("/Users/macbookair/Desktop/BerretMicroRheology/Research/DataAnalysis/Variogramoutput/")

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

variogram\_lag12=read\_xlsx(paste(film,"variogram\_lag12.xlsx",sep=""),n\_max = 4,col\_names = TRUE)

# Variogram fits from variogram excel files

variogram=read\_xlsx(paste(film,"variogram.xlsx",sep=""),n\_max=Dimdata-1,col\_names = TRUE)

#lag12 analysis vectorisation

alfa=unlist((variogram\_lag12[,1]))

sigma\_sigma\_alfa=unlist(variogram\_lag12[,2])

epsilon=unlist((variogram\_lag12[,3]))

sigma\_sigma\_epsilon=unlist(variogram\_lag12[,4])

Film\_alfa=c(Film\_alfa,alfa)

Film\_sigma\_sigma\_alfa=c(Film\_sigma\_sigma\_alfa,sigma\_sigma\_alfa)

Film\_epsilon=c(Film\_epsilon,epsilon)

Film\_tau=c(Film\_tau,1/epsilon)

Film\_sigma\_sigma\_epsilon=c(Film\_sigma\_sigma\_epsilon,sigma\_sigma\_epsilon)

Film\_sigma\_alfa\_eta=c(Film\_sigma\_alfa\_eta,sqrt(sigma\_sigma\_alfa)/trace\_meaneta[4])

Film\_sigma\_epsilon\_eta=c(Film\_sigma\_epsilon\_eta,sqrt(sigma\_sigma\_epsilon)/trace\_meaneta[4])

#

# plotting of experimental variogram

lag=unlist((variogram[,1]))

variogram\_lag=unlist((variogram[,2]))

xlab=expression(italic(Lag~l))

ylab=expression(italic(Variogram))

plot(lag,variogram\_lag,xlab=xlab,ylab="",type="l",ylim=c(0,max(variogram\_lag)),xlim=c(0,max(lag)),cex.lab=1.5)

title(ylab=ylab,line=2.5,cex.lab=1.5)

mtext(cex=0.75,paste0(film),3,0)

#

# fit with 50% of the data GRW and OUP (FIT=0 and 1)

range=0.8\*length(lag)

fitlag=head(lag,range)

fitvariogram=head(variogram\_lag,range)

ymin=0.9\*min(fitvariogram)

ymax=1.1\*max(fitvariogram)

pos\_line=1

#

#testing for fit choice

#look at lag where you get the maximum then check average value for 1->Dimdata/10 further

max\_lag=fitlag[which.max(fitvariogram)]

value\_above\_max1=variogram\_lag[(max\_lag+trunc(Dimdata/12))]

test="KO"

#Run first time to see where clear periodicities/maxima are visible

#Put those films on test=OK (as below) or KO if test is not ok

#test: if max is found=OK

if(value\_above\_max1<0.90\*max(fitvariogram)){test="OK"}

#film series 1

if(TRUE){

if(film=="film18"|film=="film8"){test="OK"}#max found > fit parameters 1 and 2 in function fitgeneral`

if(film=="film17"){test="KO"}

}

#film series 2

if(FALSE){

if(film=="film12"|film=="film13"){test="OK"}#max found so periodicity: 1 and 2 as fit parameters

if(film=="film16"|film=="film14"){test="KO"}

}

#film series 3

if(FALSE){

if(film=="film13"|film=="film17"|film=="film19"){test="OK"}

}

#series 4

if(T){test="OK"}

#

if(test=="KO" | rand=="OK"){#no maximum found

#if not evaluated > NA (not available)

Film\_acf\_T=c(Film\_acf\_T,NA)

Film\_maxV\_T=c(Film\_maxV\_T,NA)

#GRW

Film\_GRW\_constant\_cosinus\_all=c(Film\_GRW\_constant\_cosinus\_all,NA)

Film\_GRW\_T\_all=c(Film\_GRW\_T\_all,NA)

#Film\_GRW\_constant\_all=c(Film\_GRW\_constant\_all,NA)

#Film\_GRW\_alfa\_all=c(Film\_GRW\_alfa\_all,NA)

Film\_GRW\_amplitude\_sigma=c(Film\_GRW\_amplitude\_sigma,NA)

#Film\_GRW\_sigma\_eta\_all=c(Film\_GRW\_sigma\_eta\_all,NA)

#OUP

Film\_OU\_constant\_cosinus\_all=c(Film\_OU\_constant\_cosinus\_all,NA)

Film\_OU\_T\_all=c(Film\_OU\_T\_all,NA)

#Film\_OU\_constant\_all=c(Film\_OU\_constant\_all,NA)

#Film\_OU\_epsilon\_all=c(Film\_OU\_epsilon\_all,NA)

#Film\_OU\_tau\_all=c(Film\_OU\_tau\_all,NA)

Film\_OU\_amplitude\_sigma=c(Film\_OU\_amplitude\_sigma,NA)

#Film\_OU\_sigma\_eta\_all=c(Film\_OU\_sigma\_eta\_all,NA)

#simulation yes or no

if(simulation=="yes"){

WriteXLS(df, paste(amplitude,nu,trend,"simulationvariogram.xlsx",sep=""), SheetNames = "variogram")

mtext(cex=0.75,paste0("amplitude= ",amplitude,", period= ",round(1/nu,1)," s, trend= ",round(trend,1)),3,0)

pos\_line=-1

}

# FIT=0 hole + mean reverting GRW; 1 hole + mean reverting (Ornstein Uhlenbeck)

# 2 mean reverting GRW; 3 mean reverting OU mechanism; 4 cosinus; 5 constant

FIT=2

plot(fitlag,fitvariogram,xlab=xlab,ylab="",cex.lab=1.5,type="l",ylim=c(ymin,ymax), xlim=c(0,range))

title(ylab=ylab,line=2.5,cex.lab=1.5)

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

df\_fitoutput=read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

# print(df\_fitoutput)

Film\_GRW\_constant\_all=c(Film\_GRW\_constant\_all,df\_fitoutput$estimate[1])

Film\_GRW\_alfa\_all=c(Film\_GRW\_alfa\_all,df\_fitoutput$estimate[2])

Film\_GRW\_sigma\_eta\_all=c(Film\_GRW\_sigma\_eta\_all,sqrt(df\_fitoutput$estimate[1])/trace\_meaneta[4])

FIT=3

plot(fitlag,fitvariogram,xlab=xlab,ylab="",cex.lab=1.5,type="l",ylim=c(ymin,ymax), xlim=c(0,range))

title(ylab=ylab,line=2.5,cex.lab=1.5)

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

df\_fitoutput=read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

#print(df\_fitoutput)

Film\_OU\_constant\_all=c(Film\_OU\_constant\_all,df\_fitoutput$estimate[1])

Film\_OU\_epsilon\_all=c(Film\_OU\_epsilon\_all,1/df\_fitoutput$estimate[2])

Film\_OU\_tau\_all=c(Film\_OU\_tau\_all,df\_fitoutput$estimate[2])

Film\_OU\_sigma\_eta\_all=c(Film\_OU\_sigma\_eta\_all,sqrt(df\_fitoutput$estimate[1])/trace\_meaneta[4])

}else{

# only variograms with a significant signal will be sampled for periodicity investigation (if TRUE)

Film\_acf\_T=c(Film\_acf\_T,T\_acf)

T\_maxV=2\*averagetime\*fitlag[which.max(fitvariogram[1:(0.8\*max(fitlag))])]

Film\_maxV\_T=c(Film\_maxV\_T,T\_maxV)

FIT=0

plot(fitlag,fitvariogram,xlab=xlab,ylab="",cex.lab=1.5,type="l",ylim=c(ymin,ymax), xlim=c(0,range))

title(ylab=ylab,line=2.5,cex.lab=1.5)

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

df\_fitoutput=read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

# print(df\_fitoutput$estimate[1])

Film\_GRW\_constant\_cosinus\_all=c(Film\_GRW\_constant\_cosinus\_all,2\*df\_fitoutput$estimate[1])

Film\_GRW\_T\_all=c(Film\_GRW\_T\_all,df\_fitoutput$estimate[2])

Film\_GRW\_constant\_all=c(Film\_GRW\_constant\_all,df\_fitoutput$estimate[3])

Film\_GRW\_alfa\_all=c(Film\_GRW\_alfa\_all,df\_fitoutput$estimate[4])

# 2 times the amplitude vs the limit value for large l of the variogram of the GRW - value 2 for clarity

Film\_GRW\_amplitude\_sigma=c(Film\_GRW\_amplitude\_sigma,((2\*df\_fitoutput$estimate[3])/(1-df\_fitoutput$estimate[4]^2)))

Film\_GRW\_sigma\_eta\_all=c(Film\_GRW\_sigma\_eta\_all,sqrt(df\_fitoutput$estimate[3])/trace\_meaneta[4])

FIT=1

plot(fitlag,fitvariogram,xlab=xlab,ylab="",cex.lab=1.5,type="l",ylim=c(ymin,ymax), xlim=c(0,range))

title(ylab=ylab,line=2.5,cex.lab=1.5)

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

df\_fitoutput=read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

#print(df\_fitoutput)

Film\_OU\_constant\_cosinus\_all=c(Film\_OU\_constant\_cosinus\_all,2\*df\_fitoutput$estimate[1])

Film\_OU\_T\_all=c(Film\_OU\_T\_all,df\_fitoutput$estimate[2])

Film\_OU\_constant\_all=c(Film\_OU\_constant\_all,df\_fitoutput$estimate[3])

Film\_OU\_epsilon\_all=c(Film\_OU\_epsilon\_all,1/(df\_fitoutput$estimate[4]/averagetime))

Film\_OU\_tau\_all=c(Film\_OU\_tau\_all,df\_fitoutput$estimate[4])

Film\_OU\_amplitude\_sigma=c(Film\_OU\_amplitude\_sigma,2\*df\_fitoutput$estimate[1]/(df\_fitoutput$estimate[3]\*df\_fitoutput$estimate[4]))

Film\_OU\_sigma\_eta\_all=c(Film\_OU\_sigma\_eta\_all,sqrt(df\_fitoutput$estimate[3])/trace\_meaneta[4])

}

#

# first 7 lag fit (range\_fraction=integer>1) GRW and OUP (FIT=2 and3)

#

range=small\_lag\_range

range\_plot=range

fitlag=head(lag,range)

fitvariogram=head(variogram\_lag,range)

ymin=0.9\*min(fitvariogram)

ymax=1.1\*max(fitvariogram)

FIT=2

#

plot(fitlag,fitvariogram,xlab=xlab,ylab="",cex.lab=1.5,type="l",ylim=c(ymin,ymax), xlim=c(0,range))

title(ylab=ylab,line=2.5,cex.lab=1.5)

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

df\_fitoutput=read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

#

# print(df\_fitoutput)

# Arrays for frequency plots

Film\_GRW\_constant\_short\_time=c(Film\_GRW\_constant\_short\_time,df\_fitoutput$estimate[1])

Film\_GRW\_alfa\_short\_time=c(Film\_GRW\_alfa\_short\_time,df\_fitoutput$estimate[2])

Film\_GRW\_sigma\_eta\_short\_time=c(Film\_GRW\_sigma\_eta\_short\_time,sqrt(df\_fitoutput$estimate[1])/trace\_meaneta[4])

##master curve 7 lags GRW

for (j in 1:range){

y\_GRW\_master\_plot=c(y\_GRW\_master\_plot,(fitvariogram[j]\*(1-(df\_fitoutput$estimate[2]^2))/2/df\_fitoutput$estimate[1]))

x\_GRW\_master\_plot=c(x\_GRW\_master\_plot,df\_fitoutput$estimate[2]^j)

}

#

FIT=3

plot(fitlag,fitvariogram,xlab=xlab,ylab="",cex.lab=1.5,type="l",ylim=c(ymin,ymax), xlim=c(0,range))

title(ylab=ylab,line=2.5,cex.lab=1.5)

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

df\_fitoutput=read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

#

#print(df\_fitoutput)

# Arrays for frequency plots

transform\_to\_lag=df\_fitoutput$estimate[2]/averagetime

Film\_OU\_constant\_short\_time=c(Film\_OU\_constant\_short\_time,df\_fitoutput$estimate[1])

Film\_OU\_epsilon\_short\_time=c(Film\_OU\_epsilon\_short\_time,1/transform\_to\_lag)

Film\_OU\_tau\_short\_time=c(Film\_OU\_tau\_short\_time,df\_fitoutput$estimate[2])

Film\_OU\_sigma\_eta\_short\_time=c(Film\_OU\_sigma\_eta\_short\_time,sqrt(df\_fitoutput$estimate[1])/trace\_meaneta[4])

##master curve 7 lags OUP

#print(df\_fitoutput$estimate[2])

for (j in 1:range){

y\_OUP\_master\_plot=c(y\_OUP\_master\_plot,(fitvariogram[j]/transform\_to\_lag/df\_fitoutput$estimate[1]))

x\_OUP\_master\_plot=c(x\_OUP\_master\_plot,j/transform\_to\_lag)

}

# volatility clustering

# calculation of variogram of delta log eta - volatility clustering? and fit

film=paste(film,"\_VolatilityClustering",sep="")

Dimdata=Dimdata-1

log\_aeta\_volatility=abs(diff(log\_aeta))

plot(tail(at,Dimdata),log\_aeta\_volatility,type="l",cex.lab=1.5,xlab="t",ylab="Volatility proxy")

Variogram\_experimental(film,log\_aeta\_volatility,Dimdata,simulation)

# fit with the complete model on 50% of the data

FIT=5

print(film)

if(film=="Lehigh\_data\_VolatilityClustering"){FIT=0}

variogram=read\_xlsx(paste(film,"variogram.xlsx",sep=""),n\_max=Dimdata,col\_names = TRUE)

lag=unlist((variogram[,1]))

variogram\_lag=unlist((variogram[,2]))

range=0.5\*length(lag)

fitlag=head(lag,range)

fitvariogram=head(variogram\_lag,range)

ymin=0.9\*min(fitvariogram)

ymax=1.1\*max(fitvariogram)

plot(fitlag,fitvariogram,xlab=xlab,ylab="",cex.lab=1.5,type="l",ylim=c(ymin,ymax), xlim=c(0,range))

mtext(paste0("Volatility Clustering"),3,2)

title(ylab=ylab,line=2.5,cex.lab=1.5)

fit\_general(film,fitlag,fitvariogram,averagetime,FIT,pos\_line)

df\_fitoutput=read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

#

# names = LETTERS[1:26] ## Gives a sequence of the letters of the alphabet

# beta1 = rnorm(26, 5, 2) ## A vector of slopes (one for each letter)

# beta0 = 10 ## A common intercept

# for(i in 1:26){

# x = rnorm(500, 105, 10)

# y = beta0 + beta1[i]\*x + 15\*rnorm(500)

# mypath <- file.path("C:","R","SAVEHERE",paste("myplot\_", names[i], ".jpg", sep = ""))

# jpg(file=mypath)

# mytitle = paste("my title is", names[i])

# plot(x,y, cex.lab=1.5,type="l",main = mytitle)

# dev.off()

# } # End of loop

# ref. for plots http://www.medstat.nl/grafieken.html

#

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles")

} # End of main loop (all films have been analyzed)

#

#

# testing loop

# https://www.stat.berkeley.edu/~s133/R-4a.html

# data output & plotting

#

#sink()/

#

#### Plots of all parameters

path="/Users/macbookair/Desktop/Series/Plots4Pub/Results/"

#

# 7 lag GRW master plot

par(mfrow=c(2,2),xaxs="i",yaxs="i",mar=c(5,5,5,5))

film\_number=length(Film\_frequencies)

xlab\_GRW=expression(paste(alpha^"l"))

ylab\_GRW=expression(paste(gamma(l)(1-alpha^2)/(2\*sigma^2)))

xlab\_OUP=expression(paste(epsilon," l"))

ylab\_OUP=expression(paste(gamma(l)/(tau\*sigma^2)))

#

legend\_markers=NULL

legend\_text=NULL

legend\_colors=c("red","blue","green","orange","salmon","magenta","purple","brown","olivedrab","cyan","sea green","navy","violet","grey")

for (i in 1:film\_number){

legend\_markers=c(legend\_markers,i)

legend\_colors=legend\_colors[1:film\_number]

legend\_text=c(legend\_text,round(Film\_frequencies[i],3))

}

#print(legend\_markers)

#print(legend\_colors)

#print(legend\_text)

xlim=c(min(x\_GRW\_master\_plot,1e-3),max(x\_GRW\_master\_plot,2))

ylim=c(min(y\_GRW\_master\_plot,0.05),max(y\_GRW\_master\_plot,1.5))

#print(x\_GRW\_master\_plot)

plot(x\_GRW\_master\_plot[1:range\_plot],y\_GRW\_master\_plot[1:range\_plot],cex.lab=1.5,

log="xy",xlim=xlim,ylim=ylim,ylab="",xlab=xlab\_GRW,col=legend\_colors[1],pch=legend\_markers[1])

title(ylab=ylab\_GRW,line=2.5)

x=(1:9999)/10000

lines(lwd=2,x,(1-x))

#plot for publication

pub\_master\_x=NULL

pub\_master\_y=NULL

pub\_line=NULL

pub\_master\_x=c("alfa^l",x\_GRW\_master\_plot[1:range\_plot])

pub\_master\_y=c(Film\_frequencies[1],y\_GRW\_master\_plot[1:range\_plot],rep("0.0000",range\_plot\*(film\_number-1)))

pub\_line=c("1-alfa^l",1-x\_GRW\_master\_plot[1:range\_plot])

pub\_master\_GRW=c(sprintf("%.6s",pub\_master\_y))

#

for (i in 1:(film\_number-1)){

set\_start=i\*range\_plot+1

set\_end=i\*range\_plot+range\_plot

#plot for publication

pub\_master\_x=c(pub\_master\_x,x\_GRW\_master\_plot[set\_start:set\_end])

pub\_master\_y=c(Film\_frequencies[i+1],rep("0.0000",set\_start-1),y\_GRW\_master\_plot[set\_start:set\_end],rep("0.0000",(film\_number\*range\_plot-set\_end)))

pub\_master\_GRW=cbind(pub\_master\_GRW,sprintf("%.6s",pub\_master\_y))

pub\_line=c(pub\_line,1-x\_GRW\_master\_plot[set\_start:set\_end])

#

points(x\_GRW\_master\_plot[set\_start:set\_end],y\_GRW\_master\_plot[set\_start:set\_end],

col=legend\_colors[i+1],pch=legend\_markers[i+1])

}

pub\_master\_GRW=cbind(sprintf("%.6s",pub\_master\_x),pub\_master\_GRW,sprintf("%.6s",pub\_line))

write.table(pub\_master\_GRW,file=paste(path,"plot\_master\_GRW.txt",sep=""),quote=FALSE,na="",sep=" ",col.names=FALSE,row.names=FALSE,dec=".")

legend("bottomleft",

legend=legend\_text,cex=0.5,

pch=legend\_markers,

col=legend\_colors)

#

# 7 lag OUP master plot

xlim=c(min(x\_OUP\_master\_plot,1e-1),max(x\_OUP\_master\_plot,10))

ylim=c(min(y\_OUP\_master\_plot,0.10),max(y\_OUP\_master\_plot,1.5))

#print(x\_OUP\_master\_plot)

plot(x\_OUP\_master\_plot[1:range\_plot],y\_OUP\_master\_plot[1:range\_plot],cex.lab=1.5,

log="xy",xlim=xlim,ylim=ylim,xlab=xlab\_OUP,ylab="",col=legend\_colors[1],pch=legend\_markers[1])

title(ylab=ylab\_OUP,line=2.5)

x=(1:10000)/100

lines(lwd=2,x,(1-exp(-x)))

#plot for publication

pub\_master\_x=NULL

pub\_master\_y=NULL

pub\_line=NULL

pub\_master\_x=c("epsilon l",x\_OUP\_master\_plot[1:range\_plot])

pub\_master\_y=c(Film\_frequencies[1],y\_OUP\_master\_plot[1:range\_plot],rep("0.0000",range\_plot\*(film\_number-1)))

pub\_line=c("1-exp(- epsilon l)",1-exp(-x\_OUP\_master\_plot[1:range\_plot]))

pub\_master\_OUP=c(sprintf("%.6s",pub\_master\_y))

#

for (i in 1:(film\_number-1)){

set\_start=i\*range\_plot+1

set\_end=i\*range\_plot+range\_plot

#plot for publication

pub\_master\_x=c(pub\_master\_x,x\_OUP\_master\_plot[set\_start:set\_end])

pub\_master\_y=c(Film\_frequencies[i+1],rep("0.0000",set\_start-1),y\_OUP\_master\_plot[set\_start:set\_end],rep("0.0000",(film\_number\*range\_plot-set\_end)))

pub\_master\_OUP=cbind(pub\_master\_OUP,sprintf("%.6s",pub\_master\_y))

pub\_line=c(pub\_line,1-exp(-x\_OUP\_master\_plot[set\_start:set\_end]))

#

points(x\_OUP\_master\_plot[set\_start:set\_end],y\_GRW\_master\_plot[set\_start:set\_end],

col=legend\_colors[i+1],pch=legend\_markers[i+1])

#print(c(set\_start,set\_end))

#print(x\_GRW\_master\_plot[set\_start:set\_end])

}

pub\_master\_OUP=cbind(sprintf("%.6s",pub\_master\_x),pub\_master\_OUP,sprintf("%.6s",pub\_line))

write.table(pub\_master\_OUP,file=paste(path,"plot\_master\_OUP.txt",sep=""),quote=FALSE,na="",sep=" ",col.names=FALSE,row.names=FALSE,dec=".")

legend("bottomright",cex=0.5,

legend=legend\_text,

pch=legend\_markers,

col=legend\_colors)

# plotting of all fit parameter results

# axis labels

xlab=expression(paste(omega," ",(rad.s^-1))) # freq. NOT angular freq. (Berret date have been corrected)

y1lab=expression(paste(alpha))

y2lab=expression(paste(sigma^2))

y3lab=expression(paste(epsilon))

y4lab=expression(paste(tau (s)))

y5lab=expression(paste(sigma/ln(eta)))

y6lab=expression(paste(eta (Pa.s)))

y7lab=expression(paste("<",t,">",(s)))

y8lab=expression(paste("T",(s)))

y9lab=expression(paste(2,"A"["T"]," , ",2\*sigma^2/(1-alpha^2)))

y10lab=expression(paste("Skewness ",varsigma))

y11lab=expression(paste("Kurtosis ",kappa))

#

xlim=c(0.1,20)

# plot alfa vs frequency plus mean

plot(Film\_frequencies,Film\_alfa,cex.lab=1.5,log="xy",col="red",pch=17,xlab=xlab,ylab="",xlim=xlim)

title(ylab=y1lab,line=2.5)

# plot alfa vs frequency plus mean (removing alfa >1)

Film\_alfa[Film\_alfa>1]=NA

plot(cex.lab=1.5,Film\_frequencies,Film\_alfa,

xlab=xlab,ylab="",log="xy",xlim=xlim,ylim=c(0.2,2),pch=17,col="red")

title(ylab=y1lab,line=2.5)

points(Film\_frequencies,Film\_GRW\_alfa\_all,col="red",pch=15)

points(Film\_frequencies,Film\_GRW\_alfa\_short\_time, col="red",pch=20)

points(Film\_frequencies,Film\_ito\_alfa,pch=3)

alfa\_all=c(Film\_alfa,Film\_GRW\_alfa\_all,Film\_GRW\_alfa\_short\_time,Film\_ito\_alfa)

abline(h=mean(alfa\_all,na.rm=TRUE),col="red")

abline(h=mean(alfa\_all,na.rm=TRUE)+sd(alfa\_all,na.rm=TRUE),col="red",lwd=0.5)

abline(h=mean(alfa\_all,na.rm=TRUE)-sd(alfa\_all,na.rm=TRUE),col="red",lwd=0.5)

legend("topright",cex=0.65,

c(expression(paste("GRW lag1-2")),

expression(paste("GRW")),

expression(paste("GRW lag1-7")),

expression(paste("Ito"))),

pch=c(17,15,20,3),

col=c("red","red","red","black"))

av\_alfa=round(mean(alfa\_all,na.rm=TRUE),2)

er\_alfa=round(sd(alfa\_all,na.rm=TRUE),2)

mtext(cex=0.75,paste0("< > =",av\_alfa, " ± ",er\_alfa),3,-3,col="red",adj=0.25)

#plot for publication

Film\_alfa[Film\_alfa>1]=NA

alfa\_frequency=data.frame(Film\_frequencies,Film\_alfa,Film\_GRW\_alfa\_all,Film\_GRW\_alfa\_short\_time,Film\_ito\_alfa)

write.table(format(alfa\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_alfa\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot sigma's vs frequency

plot(Film\_frequencies,Film\_sigma\_sigma\_alfa,cex.lab=1.5,log="xy",col="red",pch=17,xlab=xlab,ylab="",ylim=c(0.01,1),xlim=xlim)

title(ylab=y2lab,line=2.5)

points(Film\_frequencies,Film\_sigma\_sigma\_epsilon,col="blue",pch=17)

points(Film\_frequencies,Film\_GRW\_constant\_all,col="red",pch=15)

points(Film\_frequencies,Film\_GRW\_constant\_short\_time,col="red",pch=20)

points(Film\_frequencies,Film\_OU\_constant\_all,col="blue",pch=15)

points(Film\_frequencies,Film\_OU\_constant\_short\_time,col="blue",pch=20)

points(Film\_frequencies,Film\_sigma\_sigma\_ito,pch=3)

#plot for publication

sigma\_sigma\_frequency=data.frame(Film\_frequencies,

Film\_sigma\_sigma\_alfa,

Film\_sigma\_sigma\_epsilon,

Film\_GRW\_constant\_all,

Film\_GRW\_constant\_short\_time,

Film\_OU\_constant\_all,

Film\_OU\_constant\_short\_time,

Film\_sigma\_sigma\_ito)

write.table(format(sigma\_sigma\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_sigma\_sigma\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot sigm alfa / ln (eta) vs frequency

plot(Film\_frequencies,Film\_sigma\_alfa\_eta, cex.lab=1.5,log="xy",col="red",pch=17,xlab=xlab,ylab="",ylim=c(0.01,1),xlim=xlim)

title(ylab=y5lab,line=2.5)

points(Film\_frequencies,Film\_sigma\_epsilon\_eta,col="blue",pch=17)

points(Film\_frequencies,Film\_GRW\_sigma\_eta\_all,col="red",pch=15)

points(Film\_frequencies,Film\_GRW\_sigma\_eta\_short\_time,col="red",pch=20)

points(Film\_frequencies,Film\_OU\_sigma\_eta\_all,col="blue",pch=15)

points(Film\_frequencies,Film\_OU\_sigma\_eta\_short\_time,col="blue",pch=20)

ln\_freq=NULL

sigma\_ln\_eta=NULL

freq=c(Film\_frequencies,

Film\_frequencies,

Film\_frequencies,

Film\_frequencies,

Film\_frequencies,

Film\_frequencies)

sigma\_eta=c(Film\_sigma\_alfa\_eta,

Film\_sigma\_epsilon\_eta,

Film\_GRW\_sigma\_eta\_all,

Film\_GRW\_sigma\_eta\_short\_time,

Film\_OU\_sigma\_eta\_all,

Film\_OU\_sigma\_eta\_short\_time)

Model\_sigma\_eta=function(freq,power,cte){cte\*(freq)^power}

sigmafit=nlsLM(sigma\_eta~Model\_sigma\_eta(freq,power,cte),start=list(power=0.4,cte=0.2))

summary(sigmafit)

lines(sort(freq),sort(predict(sigmafit)),col="black")

power =coef(sigmafit)[1]

cte =coef(sigmafit)[2]

error\_power=confint(sigmafit,level=0.682)

# plot of fitted curve

mtext(cex=0.75,expression(paste(" ~ ",omega^beta)),3,-2,adj=0.1)

mtext(cex=0.75,expression(paste(beta, " = ")),3,-3.2,adj=0.1)

mtext(cex=0.75,paste0(round(power,3)," ± ",round((power-error\_power[1,1]),2)),3,-3,adj=0.3)

legend("bottomright",cex=0.65,

c(expression(paste("GRW lag1-2")),

expression(paste("OUP lag1-2")),

expression(paste("GRW")),

expression(paste("OUP")),

expression(paste("GRW lag1-7")),

expression(paste("OUP lag1-7"))),

pch=c(17,17,15,15,20,20),

col=c("red","blue","red","blue","red","blue"))

#plot for publication

sigma\_logeta\_frequency=data.frame(Film\_frequencies,

Film\_sigma\_alfa\_eta,

Film\_sigma\_epsilon\_eta,

Film\_GRW\_sigma\_eta\_all,

Film\_GRW\_sigma\_eta\_short\_time,

Film\_OU\_sigma\_eta\_all,

Film\_OU\_sigma\_eta\_short\_time)

write.table(format(sigma\_logeta\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_sigma\_logeta\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot epsilon vs frequency

plot(Film\_frequencies,Film\_epsilon,cex.lab=1.5,log="x",col="blue",pch=17,xlab=xlab,ylab="",xlim=xlim)

title(ylab=y3lab,line=2.5)

# plot with positive epsilons only

Film\_epsilon[Film\_epsilon<=0]=NA

plot(Film\_frequencies,Film\_epsilon,cex.lab=1.5,

ylab="",xlab=xlab,log="xy",ylim=c(0.02,2),xlim=xlim,pch=17,col="blue")

title(ylab=y3lab,line=2.5)

points(Film\_frequencies,Film\_OU\_epsilon\_all,col="blue",pch=15)

points(Film\_frequencies,Film\_OU\_epsilon\_short\_time, col="blue",pch=20)

legend("bottomright",cex=0.65,

c(expression(paste("OUP lag1-2")),

expression(paste("OUP")),

expression(paste("OUP lag1-7"))),

pch=c(17,15,20),

col=c("blue","blue","blue"))

epsilons=c(Film\_epsilon,Film\_OU\_epsilon\_all,Film\_OU\_epsilon\_short\_time)

abline(h=mean(epsilons,na.rm=TRUE),col="blue",lwd=2)

abline(h=(mean(epsilons,na.rm=TRUE)-sd(epsilons,na.rm=TRUE)),col="blue")

abline(h=(mean(epsilons,na.rm=TRUE)+sd(epsilons,na.rm=TRUE)),col="blue")

txt=-1

mtext(cex=0.75,paste0("< > =",round(mean(epsilons,na.rm=TRUE),2)," lags"),3,-5.5-txt,col="blue",adj=0.9)

hist(epsilons,xlab=expression(paste(epsilon(lag))))

#plot for publication

Film\_epsilon[Film\_epsilon<=0]=NA

epsilon\_frequency=data.frame(Film\_frequencies,Film\_epsilon,Film\_OU\_epsilon\_all,Film\_OU\_epsilon\_short\_time)

write.table(format(epsilon\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_epsilon\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

#File2 data

# plot tau (1/epsilon) vs frequency

Film\_tau[Film\_tau<=0]=NA

plot(cex.lab=1.5,Film\_frequencies,Film\_tau,log="x",ylim=c(min(Film\_tau,na.rm=TRUE)-5,max(Film\_tau,na.rm=TRUE)+5),col="blue",pch=17,xlab=xlab,ylab="",xlim=xlim)

title(ylab=y4lab,line=2.5)

# plot based on corrected epsilons

plot(Film\_frequencies,Film\_tau,cex.lab=1.5,

log="xy",xlab=xlab,ylab="",ylim=c(0.1,100),xlim=xlim,pch=17,col="blue")

title(ylab=y4lab,line=2.5)

points(Film\_frequencies,Film\_OU\_tau\_all,col="blue",pch=15)

points(Film\_frequencies,Film\_OU\_tau\_short\_time,col="blue",pch=20)

points(Film\_frequencies,Film\_acf\_tau,col="green",pch=15)

points(Film\_frequencies,Film\_ito\_tau,pch=3)

taus=c(Film\_tau,Film\_OU\_tau\_all,Film\_OU\_tau\_short\_time,Film\_acf\_tau,Film\_ito\_tau)

meanlog=mean(log(taus),na.rm=TRUE)

sdlog=sd(log(taus),na.rm=TRUE)

#alternative - gives same result

#meanlog=as.numeric(unlist(fitdistr(taus,"log-normal"))[1])

#sdlog=as.numeric(unlist(fitdistr(taus,"log-normal"))[2])

abline(h=exp(meanlog),col="blue")

abline(h=exp(meanlog+sdlog),col="blue",lwd=0.5)

abline(h=exp(meanlog-sdlog),col="blue",lwd=0.5)

legend("topright",cex=0.65,

c(expression(paste("OUP lag1-2")),

expression(paste("OUP")),

expression(paste("OUP lag1-7")),

expression(paste("ACF")),

expression(paste("Ito"))),

pch=c(17,15,20,15,3),

col=c("blue","blue","blue","green","black"))

mtext(cex=0.75,paste0("< > =",round(exp(meanlog),2)," s"),3,-6.5-txt,col="blue",adj=0.9)

#freq=NULL

#tau=NULL

tau\_na\_omit=na.omit(data.frame(Film\_frequencies,Film\_tau))

freq=c(tau\_na\_omit$Film\_frequencies,

Film\_frequencies,

Film\_frequencies,

Film\_frequencies,

Film\_frequencies)

tau=c(tau\_na\_omit$Film\_tau,

Film\_OU\_tau\_all,

Film\_OU\_tau\_short\_time,

Film\_acf\_tau,

Film\_ito\_tau)

Model\_tau=function(freq,cte,power){cte\*(freq)^(power)}

taufit=nlsLM(tau~Model\_tau(freq,cte,power),start=list(cte=0.35,power=-1),na.action=na.omit)

print(summary(taufit))

power =coef(taufit)[2]

cte =coef(taufit)[1]

lines(freq,(predict(taufit)),col="black")

mtext(cex=0.75,expression(paste("~ ",omega^beta)),3,-9.8-txt,adj=0.1)

mtext(cex=0.75,expression(paste(beta, " = ")),3,-11-txt,adj=0.1)

result=tryCatch({mtext(cex=0.75,paste0(" ± ",round((power-confint(taufit,level=0.683)[2]),2)),3,-10.8-txt,adj=0.3)

},error=function(e){print("Cannot determine errors")

},finally={mtext(cex=0.75,paste0(round(power,2)),3,-10.8-txt,adj=0.2)

})

hist(taus)

#plot for publication

Film\_tau[Film\_tau<0]=NA

tau\_frequency=data.frame(Film\_frequencies,

Film\_tau,

Film\_OU\_tau\_all,

Film\_OU\_tau\_short\_time,

Film\_acf\_tau,

Film\_ito\_tau)

write.table(format(tau\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_tau\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot of 2A\_T vs long term alfa sigma^2 - factor of 2 included in Film\_GRW\_constant\_cosinus\_all

plot(Film\_frequencies,Film\_GRW\_amplitude\_sigma,cex.lab=1.5,

log="xy",xlab=xlab,ylab="",ylim=c(0.02,2),xlim=xlim,pch=15,col="red")

points(Film\_frequencies,Film\_GRW\_constant\_cosinus\_all,pch=2,col="red")

title(ylab=y9lab,line=2.5)

#abline(lwd=2,h=mean(Film\_GRW\_amplitude\_sigma[Film\_GRW\_amplitude\_sigma>0]),col="red")

#points(Film\_frequencies[Film\_OU\_amplitude\_sigma>0],Film\_OU\_amplitude\_sigma[Film\_OU\_amplitude\_sigma>0],ylim=c(0.05,5),pch=20,col="blue")

#abline(lwd=2,h=mean(Film\_OU\_amplitude\_sigma[Film\_OU\_amplitude\_sigma>0]),col="blue")

#mtext(cex=0.75,paste0(" < > =",round(mean(Film\_GRW\_amplitude\_sigma[Film\_GRW\_amplitude\_sigma>0]),2)),3,-4,col="red")

#mtext(cex=0.75,paste0(" < > =",round(mean(Film\_OU\_amplitude\_sigma[Film\_OU\_amplitude\_sigma>0]),2)),3,-7,col="blue")

legend("topright",cex=0.75,

c(expression(paste(2\*sigma^2/(1-alpha^2))),expression(paste(2,"A"["T"]))),

pch=c(15,2),

col=c("red","red"))

#col=c("red","blue"))

#plot for publication

Amplitude\_Sigma\_frequency=data.frame(Film\_frequencies,Film\_GRW\_amplitude\_sigma,Film\_GRW\_constant\_cosinus\_all)

write.table(format(Amplitude\_Sigma\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_Amplitude\_Sigma\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot of Period T

plot(Film\_frequencies,Film\_GRW\_T\_all,cex.lab=1.5,

xlab=xlab,ylab="",cex=1.5,pch=1,col="red",log="xy",xlim=xlim,ylim=c(10,1000))

title(ylab=y8lab,line=2.5)

# na.rm=TRUE in mean, sd etc

abline(lwd=2,h=mean(Film\_GRW\_T\_all,na.rm=TRUE),col="red")

points(Film\_frequencies,Film\_OU\_T\_all,cex=1.5,pch=4,col="blue",xlim=xlim,ylim=ylim)

abline(lwd=2,h=mean(Film\_OU\_T\_all,na.rm=TRUE),col="blue")

points(Film\_frequencies,Film\_acf\_T,cex=1.5,col="grey",pch=1)

abline(lwd=2,h=mean(Film\_acf\_T,na.rm=TRUE),col="grey")

points(Film\_frequencies,Film\_maxV\_T,cex=1,pch=20,xlim=xlim,ylim=ylim)

abline(lwd=2,h=mean(Film\_maxV\_T,na.rm=TRUE))

mtext(cex=0.75,paste0("< > =",round(mean(Film\_GRW\_T\_all,na.rm=TRUE),0)," s"),3,-1,adj=0.7,col="red")

mtext(cex=0.75,paste0("< > =",round(mean(Film\_OU\_T\_all,na.rm=TRUE),0)," s"),3,-1.75,adj=0.7,col="blue")

mtext(cex=0.75,paste0("< > =",round(mean(Film\_acf\_T,na.rm=TRUE),0)," s"),3,-2.5,adj=0.7,col="grey")

mtext(cex=0.75,paste0("< > =",round(mean(Film\_maxV\_T,na.rm=TRUE),0)," s"),3,-3.25,adj=0.7,col="black")

legend("bottomleft",cex=0.75,

c(expression(paste("GRW")),

expression(paste("OUP")),

expression(paste("ACF (min)")),

expression(paste("Variogram (max)"))),

pch=c(1,4,1,20),

col=c("red","blue","grey","black"))

T\_all=c(Film\_GRW\_T\_all,Film\_OU\_T\_all,Film\_acf\_T,Film\_maxV\_T)

hist(T\_all,main="Histogram all periods T",xlab="T(s)")

mtext(cex=0.75,paste0("< > =",round(mean(T\_all,na.rm=TRUE),0)," s ± ",round(sd(T\_all,na.rm=TRUE),0) ),3,0.5)

#plot for publication

T\_frequency=data.frame(Film\_frequencies,Film\_GRW\_T\_all,Film\_OU\_T\_all,Film\_acf\_T,Film\_maxV\_T)

write.table(format(T\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_T\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot average time versus frequency

plot(Film\_frequencies,Film\_averagetime,cex.lab=1.5,log="xy",pch=3,xlab=xlab,ylab=y7lab,xlim=xlim,ylim=c(0.2,20))

# plot average eta versus frequency + fit with power law

plot(Film\_frequencies,Film\_eta,cex.lab=1.5,log="xy",pch=20,col="red",xlab=xlab,ylab=y6lab,xlim=xlim,ylim=c(1,100))

arrows(Film\_frequencies,Film\_sdetamin, Film\_frequencies, Film\_sdetaplus, length=0.05, angle=90, code=3,col="red")

points(Film\_frequencies,Film\_eta\_ito,pch=3,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

Model\_ln\_eta=function(Film\_frequencies,cte,power){cte\*(Film\_frequencies)^(power)}

ln\_etafit=nlsLM(Film\_eta~Model\_ln\_eta(Film\_frequencies,cte,power),start=list(cte=1,power=-1))

print(summary(ln\_etafit))

power =coef(ln\_etafit)[2]

cte =coef(ln\_etafit)[1]

lines((Film\_frequencies),(predict(ln\_etafit)),col="red")

mtext(cex=0.75,expression(paste("~ ",omega^beta)),3,-8.8-txt,adj=0.1)

mtext(cex=0.75,expression(paste(beta, " = ")),3,-10-txt,adj=0.1)

result=tryCatch({mtext(cex=0.75,paste0(" ± ",round((power-confint(ln\_etafit,level=0.683)[2]),2)),3,-10-txt,adj=0.3)

},error=function(e){print("Cannot determine errors")

},finally={mtext(cex=0.75,paste0(round(power,2)),3,-10-txt,adj=0.2)

})

eta\_frequency\_error=data.frame(Film\_frequencies,Film\_eta,Film\_sdetamin,Film\_sdetaplus)

write.table(format(eta\_frequency\_error,digits=3),quote=FALSE,file=paste(path,"plot\_eta\_frequency\_errors.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot average eta after different steps versus frequency

#plot for publication

eta\_frequency=data.frame(Film\_frequencies,Film\_eta)

write.table(format(eta\_frequency,digits=3),quote=FALSE,file=paste(path,"plot\_eta\_frequency.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plot average eta after different steps versus frequency

ylim=c(1,100)

plot(Film\_frequencies,Film\_eta,cex.lab=1.5,log="xy",pch=21,cex=1.5,col="red",xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_eta\_rawdata,pch=20,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_eta\_no\_outliers,pch=1,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_eta\_detrended,pch=2,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_eta\_ito,pch=3,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

# plot skewness after different steps versus frequency

ylim=c(-2,2)

plot(Film\_frequencies,Film\_skewness,cex.lab=1.5,log="x",pch=21,cex=1.5,col="red",xlab=xlab,ylab=y10lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_skewness\_rawdata,pch=20,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_skewness\_no\_outliers,pch=1,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_skewness\_detrended,pch=2,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

# plot kurtosis after different steps versus frequency

ylim=c(0.00001,10)

plot(Film\_frequencies,Film\_kurtosis,cex.lab=1.5,log="xy",pch=21,cex=1.5,col="red",xlab=xlab,ylab=y11lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_kurtosis\_rawdata,pch=20,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_kurtosis\_no\_outliers,pch=1,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

points(Film\_frequencies,Film\_kurtosis\_detrended,pch=2,xlab=xlab,ylab=y6lab,xlim=xlim,ylim=ylim)

# plot multiple axis

# plot lag1-2 data

# with(lag\_12,plot(Film\_frequencies,Film\_alfa,cex.lab=1.5,pch=20,col="grey",xlab=expression(paste(omega)),ylab=expression(paste(alfa))))

# https://stats.idre.ucla.edu/r/codefragments/greek\_letters/

#labels x and 2 y axes

y1lab="Original Number of data"

y2lab="Fraction of Outliers"

extremes=data.frame(x=Film\_frequencies,y1=Film\_numberofdata,y2=Film\_outliers/Film\_numberofdata)

#markers and colors for 2 points

pch1=20

col1="grey"

pch2=15

col2="red"

with(extremes,plot(x,y1,cex.lab=1.5,pch=pch1,col=col1,log="xy",xlab=xlab,ylab=y1lab,ylim=c(10,1000),xlim=xlim))

par(new = TRUE)

with(extremes,plot(x,y2,cex.lab=1.5,axes=F,log="xy",cex=1,pch=pch2,col=col2,xlab=NA,ylab=NA,ylim=c(0.01,0.2),xlim=xlim))

axis(side=4)

mtext(side=4,line=3,y2lab,cex=0.8)

legend("bottomright",cex=0.75,legend=c(y1lab,y2lab),pch=c(pch1,pch2),col=c(col1,col2))

#

# Scaling of delta eta with comparison with Gaussian and Cauchy distibution (Check files for scaling!)

par(mfrow=c(2,2),xaxs="i",yaxs="i",mar=c(5,5,5,5))

Scalingdelta\_aeta()

#

# Copying files

#

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

Fitoutput\_path="/Users/macbookair/Desktop/Series/OutputAll/"

file.copy(from=list.files(pattern="Fit Output.txt",full.names = TRUE), overwrite=TRUE, to=Fitoutput\_path)

#

# for plot saving after code above has been running

# you need to go through the whole plotting history first and run code below

#

if(TRUE){

#save all plots in ScreenPlotRStudio directory

RStudio\_path="/Users/macbookair/Desktop/Series/OutputAll/ScreenPlotsRStudio/"

plots.dir.path =list.files(tempdir(), pattern="rs-graphics", full.names = TRUE)

plots.png.paths=list.files(plots.dir.path, pattern=".png", full.names = TRUE)

file.copy(from=plots.png.paths,to=RStudio\_path)

plots.png.details=file.info(plots.png.paths)

print(plots.png.details)

plots.png.details=plots.png.details[order(plots.png.details$mtime,decreasing=TRUE),]

print(plots.png.details)

sorted.png.names=gsub(plots.dir.path,RStudio\_path,

row.names(plots.png.details), fixed=TRUE)

numbered.png.names=paste0(RStudio\_path, 1:length(sorted.png.names), ".png")

print(numbered.png.names)

file.rename(from=sorted.png.names, to=numbered.png.names)

}

#

# emptying directories before new series is analyzed

#

GF\_path="/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/"

RStudio\_path="/Users/macbookair/Desktop/Series/OutputAll/ScreenPlotsRStudio/"

DAM\_path="/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/"

DAMD\_path="/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/Dataprep/"

DAR\_path="/Users/macbookair/Desktop/Series/Plots4Pub/Results/"

file.remove(list.files(GF\_path, include.dirs = F, full.names = T, recursive = T))

file.remove(list.files(DAM\_path, include.dirs = F, full.names = T, recursive = T))

file.remove(list.files(DAR\_path, include.dirs = F, full.names = T, recursive = T))

file.remove(list.files(DAMD\_path, include.dirs = F, full.names = T, recursive = T))

file.remove(list.files(RStudio\_path, include.dirs = F, full.names = T, recursive = T))

#

***DISTRIBUTION FUNCTION***

function(eta,lognormal){

#cauchy only is always on deltas therefore deltaeta =eta input / log normal :

deltaeta <- diff(eta)

#lognormal=true when you analyze individual viscosity profiles

deltaeta <- eta

meanlog <- 0

if(lognormal=="true"){

# lognormal\_check

meanlog=coef(fitdistr(eta,"log-normal"))[1]

sdlog=coef(fitdistr(eta,"log-normal"))[2]

median=meanlog-sdlog\*sdlog/2

mode=meanlog-3\*sdlog\*sdlog/2

etaecdf=exp(meanlog)

etaecdfhb=exp(meanlog+sdlog)

etaecdflb=exp(meanlog-sdlog)

# no ecdf plots

# xlab="eta"

xlab="Transmission"

plot(ecdf(eta),cex.lab=1.5,log="x",xlim=c(min(eta),max(eta)),xlab=xlab)

plot(ecdf(rlnorm(1000,(meanlog),(sdlog))),cex.lab=1.5,col="red",add=TRUE,do.points=FALSE)

mtext(cex=0.75,paste0("eta lognormal ",round(etaecdf,1)," hb ",round(etaecdfhb,1)," lb ",round(etaecdflb,1)),3,0.5)

deltaeta=diff(eta)

}

# cauchy\_check via fitdistr

#plot(ecdf(deltaeta),xlab="delta eta",pch=20)

meancauchy=coef(fitdistr(deltaeta,"cauchy"))[1]

scalecauchy=coef(fitdistr(deltaeta,"cauchy"))[2]

mean=round(meancauchy,2)

fwhm=2\*(round(scalecauchy,2))

#

#plot((ecdf(rcauchy(1000,meancauchy,scalecauchy))),col="red",add=TRUE,do.points=FALSE)

#mtext(cex=0.75,paste0("delta eta cauchy: mean ",mean," FWHM ",fwhm),3,0.5)

#

# gauss\_check via fitdistr

meannormal=coef(fitdistr(deltaeta,"normal"))[1]

sdnormal=coef(fitdistr(deltaeta,"normal"))[2]

if(lognormal=="false"){

#plot(ecdf(deltaeta),xlab="delta eta", pch=20)

mean=round(meannormal,2)

sd=round(sdnormal,2)

#plot((ecdf(rnorm(1000,meannormal,sdnormal))),col="red",add=TRUE,xlab="delta eta",do.points=FALSE)

#mtext(cex=0.75,paste0("delta eta gauss: mean ",mean," sd ",sd),3,0.5)

}

#

#plot routine of a vector of "experimental" values ( returns, deltas etc. )

#

#subtract mean (obtained from gaussian) from deltaeta - also for the master curve

delta\_eta=sort(deltaeta-meannormal,na.last = NA) #normalization

mykurtosis=function(x){(mean((x-mean(x,na.rm=TRUE))^4)/(sd(x,na.rm=TRUE)^4))-3}

myskewness=function(x){mean((x-mean(x,na.rm=TRUE))^3)/(sd(x,na.rm=TRUE)^3)}

kurtosis=mykurtosis(delta\_eta)

skewness=myskewness(delta\_eta)

# t Student

#fitdistr(delta\_eta,"t",start = list(m=mean(delta\_eta),s=sd(delta\_eta), df=3), lower=c(-1,0.01,1))

lower=c(-1,0.02,2)

# for series 1: if(film=="film9"){lower=c(-1,0.01,8)} - check

Param=fitdistr(delta\_eta,"t",

start = list(m=mean(delta\_eta),s=sd(delta\_eta), df=3),

lower=lower,

upper=c(1,40,50))

tm =coef(Param)[1]

ts =coef(Param)[2]

tdf=coef(Param)[3]

#

deltaeta\_neg =sort(abs(delta\_eta[delta\_eta<0]))

deltaeta\_pos =sort(delta\_eta[delta\_eta>=0])

count\_neg =length(deltaeta\_neg)

count\_pos =length(deltaeta\_pos)

count=count\_neg+count\_pos

xneg\_exp=(count\_neg:1)/count

xpos\_exp=(count\_pos:1)/count

ylim=c(1/(2\*count),1)

xlim\_max=max(deltaeta\_neg,deltaeta\_pos)

xlim\_min=min(deltaeta\_neg,deltaeta\_pos)

#xlim\_min=min(deltaeta\_neg,deltaeta\_pos)\*0.5

xlim=c(xlim\_min,xlim\_max)

if(lognormal=="false"){#for scaling law

mean=round(meannormal,2)

sd=round(sdnormal,2)

ylim=c(1/(2\*count),1)

if(max(deltaeta\_pos)>10){#criterion for deciding on type of x axis (normalized to sigma or not)

#xlab=expression(paste("|",Delta,"",eta,"|"))

xlab="delta T"

}else{

xlab=expression(paste("|",Delta,"",eta,"| / ",sigma,"(",Delta,"",eta,")"))

xlab="delta T/sigma(T)"

}

plot(deltaeta\_neg,xneg\_exp,cex.lab=1.5,pch=20,col="red",

log="y",ylim=ylim,xlim=xlim,xlab=xlab,ylab="CDF")

}else{#for each film

if(film=="Lehigh\_data"){

plot(deltaeta\_neg,xneg\_exp,cex.lab=1.5,cex=0.75,

log="y",ylab="CDF",xlim=xlim,ylim=ylim,

pch=20,col="red",xlab=expression(paste("|",Delta,"",G,"'| (Pa.s)")))

}else{

plot(deltaeta\_neg,xneg\_exp,cex.lab=1.5,cex=0.75,

log="y",ylab="CDF",xlim=xlim,ylim=ylim,

pch=20,col="red",xlab=expression(paste("|",Delta,"",eta,"| (Pa.s)")))

}

}

points(deltaeta\_pos,xpos\_exp,pch=20)

#for every frequency/film data are written to a seperate file when "lognormal=true"

#when Scaling function is called set "false"

#scaling is now done to sd and not the mean of the viscosity as earlier (however these are proportional to eachother)

#if statement has to be changed film="filmx" first in the series of films

Format=function(number){sprintf("%3.6f",number)}#to make stardardized output

if(lognormal=="true"){

delta\_eta\_scaling=delta\_eta/sd(delta\_eta)

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

#delta\_eta\_scaling=delta\_eta/exp(meanlog)

dfscalingdeltaeta=data.frame(delta\_eta\_scaling)

WriteXLS(data.frame(delta\_eta\_scaling), paste(film,"Scalingdeltaeta.xlsx",sep=""), SheetNames = "scalinglaw")

tdistribution\_fits=data.frame(film,Format(tm),Format(ts),Format(tdf),Format(skewness),Format(kurtosis))

print(film)

if(film=="film10"){write.table(tdistribution\_fits,paste("tdistribution\_fits.txt",sep=""),col.names=TRUE, append=TRUE)

}else{

write.table(tdistribution\_fits,paste("tdistribution\_fits.txt",sep="",col.names=FALSE,append=TRUE))}

}

# plot with distributions

# cauchy

lorentz=NULL

Cauchy=function(x,meancauchy,scalecauchy){1/2-atan((x-meancauchy)/scalecauchy)/pi}

# delta\_eta with mean = 0

lines(deltaeta\_neg,Cauchy(deltaeta\_neg,0,scalecauchy),lwd=2,col="green")

#if(lognormal=="false"){

y=(1:count)/count

# gaussian

deltaeta\_gauss=qnorm(y,0,sd(delta\_eta))

deltaeta\_gauss\_neg=abs(deltaeta\_gauss[deltaeta\_gauss<0])

negcdf=y[deltaeta\_gauss<0]

deltaeta\_gauss\_pos=deltaeta\_gauss[deltaeta\_gauss>=0]

poscdf=1-y[deltaeta\_gauss>=0]

count\_gauss\_neg =length(deltaeta\_gauss\_neg)

count\_gauss\_pos =length(deltaeta\_gauss\_pos)

lines(deltaeta\_gauss\_neg,negcdf,lwd=2,col='blue')

# t student - need to develop code as qt is with scale (sigma)=1 only

# - fitdistr uses more general t student: with mean (mu), scale (sigma) and

# degrees of freedaom (nu) as output parameters)

t\_general=function(x,mu,sigma,nu){

1/sigma\*

gamma((nu+1)/2)/

sqrt(nu\*pi)/gamma(nu/2)\*

(1+((x-mu)^2)/nu/sigma/sigma)^(-(1+nu)/2)

}

min\_eta=min(delta\_eta)

max\_eta=max(delta\_eta)

if(abs(min\_eta)>max\_eta){

max\_eta=abs(min\_eta)

}else{

min\_eta=-max\_eta

}

DELTA=max\_eta-min\_eta

deltaeta\_tstudent=NULL

cfdt=NULL

tdist=NULL

cfdt[1]=t\_general(min\_eta,tm,ts,tdf)

tdist[1]=cfdt[1]

deltaeta\_tstudent[1]=min\_eta

for(i in 2:count){

deltaeta\_tstudent[i]=min\_eta+(i-1)\*DELTA/count

if(abs(deltaeta\_tstudent[i])<0.0001){deltaeta\_tstudent[i]=0.0001}

tdist[i]=t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

cfdt[i]=cfdt[i-1]+t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

}

cfdt=cfdt/cfdt[count]

deltaeta\_tstudent\_neg=abs(deltaeta\_tstudent[deltaeta\_tstudent<0])

negcdft=cfdt[deltaeta\_tstudent<0]

deltaeta\_tstudent\_pos=deltaeta\_tstudent[deltaeta\_tstudent>=0]

poscdft=1-cfdt[deltaeta\_tstudent>=0]

count\_tstudent\_neg =length(deltaeta\_tstudent\_neg)

count\_tstudent\_pos =length(deltaeta\_tstudent\_pos)

#points(log="xy",xneg,yneg,pch=17,cex=1.5,col="red")

#points(xpos,ypos,pch=20)

lines(deltaeta\_tstudent\_neg,negcdft,lwd=2,lty=2,col='red')

lines(deltaeta\_tstudent\_pos,poscdft,lwd=2)

mtext(cex=0.75,paste("df = ",round(tdf,2)," scale = ",round(ts,2)),3,2)

mtext(cex=0.75,paste("Skewn = ",round(skewness,2)," Kurt = ",round(kurtosis,2)),3,1)

#

#

if(film=="film9" | film=="film15" | film=="film10" | film=="film21"){#series 1

#if(film=="film12" | film=="film13" | film=="film17" | film=="film10"){#series 2

delta\_eta\_all =c(deltaeta\_neg,deltaeta\_pos,deltaeta\_gauss\_neg,deltaeta\_gauss\_pos,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

ecdf\_exp\_neg =c(xneg\_exp,rep(NA,(count\_pos+count+count)))

ecdf\_exp\_pos =c(rep(NA,(count\_neg)),xpos\_exp,rep(NA,count+count))

ecdf\_cauchy\_neg =c(Cauchy(deltaeta\_neg,0,scalecauchy),rep(NA,(count\_pos+count+count)))

ecdf\_cauchy\_pos =c(rep(NA,count\_neg),Cauchy(deltaeta\_pos,0,scalecauchy), rep(NA,count+count))

ecdf\_gauss\_neg =c(rep(NA,count),negcdf,rep(NA,count\_gauss\_pos+count))

ecdf\_gauss\_pos =c(rep(NA,count+count\_gauss\_neg),poscdf,rep(NA,count))

ecdf\_tstudent\_neg =c(rep(NA,count+count),negcdft,rep(NA,count\_tstudent\_pos))

ecdf\_tstudent\_pos =c(rep(NA,count+count+count\_tstudent\_neg),poscdft)

# plot for publication non-scaled

plot\_cdf\_delta\_eta=data.frame(Format(delta\_eta\_all),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/Dataprep/")

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plots for publication - plots for order of magnitude difference

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results")

#series 1: 9, 21; series 2: 10, 17

if(film=="film9"){

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

if(film=="film21"){

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

# plot for publication scaled data for each frequency with the respective distributions for films 9,10,15,21

delta\_eta\_scaled = delta\_eta\_all/sd(delta\_eta)

plot\_cdf\_delta\_eta\_scaling = data.frame(Format(delta\_eta\_scaled),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

write.table(plot\_cdf\_delta\_eta\_scaling,quote=FALSE,file=paste(film,"plot\_correct\_scaling.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

if(lognormal=="false"){

# plot for publication - old master curve for cdf -

# data have been scaled on individual basis - see above - therefore deltaeta\_neg/pos are scaled

# arrays of x (xneg and pos) and y (deltaeta\_pos and neg) values

delta\_eta\_vs\_eta=c(deltaeta\_neg,deltaeta\_pos,deltaeta\_gauss\_neg,deltaeta\_gauss\_pos,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

ecdf\_exp\_neg =c(xneg\_exp,rep(NA,(count\_pos+count+count)))

ecdf\_exp\_pos =c(rep(NA,(count\_neg)),xpos\_exp,rep(NA,count+count))

ecdf\_cauchy\_neg =c(Cauchy(deltaeta\_neg,0,scalecauchy),rep(NA,(count\_pos+count+count)))

ecdf\_cauchy\_pos =c(rep(NA,count\_neg),Cauchy(deltaeta\_pos,0,scalecauchy), rep(NA,count+count))

ecdf\_gauss\_neg =c(rep(NA,count),negcdf,rep(NA,count\_gauss\_pos+count))

ecdf\_gauss\_pos =c(rep(NA,count+count\_gauss\_neg),poscdf,rep(NA,count))

ecdf\_tstudent\_neg =c(rep(NA,count+count),negcdft,rep(NA,count\_tstudent\_pos))

ecdf\_tstudent\_pos =c(rep(NA,count+count+count\_tstudent\_neg),poscdft)

# dataframe for plotting sprintf("%.6s",t)

#Format=function(number){sprintf("%.6s",number)}

plot\_mastercurve\_delta\_eta=data.frame(Format(delta\_eta\_vs\_eta),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_mastercurve\_delta\_eta,quote=FALSE,file="plot\_mastercurve\_delta\_eta\_scaled.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

}

***SCALING DELTA***

function(){

# for Series/File selection put on TRUE

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

# All FALSE for series 3

if(FALSE){

# Film Series 1 for scaling of the selected data for publication

File=c("film10Scalingdeltaeta.xlsx",

"film15Scalingdeltaeta.xlsx",

"film21Scalingdeltaeta.xlsx" )

}

if(TRUE){

# Film Series 1

File=c("film1Scalingdeltaeta.xlsx",

"film8Scalingdeltaeta.xlsx",

"film9Scalingdeltaeta.xlsx",

"film10Scalingdeltaeta.xlsx",

"film11Scalingdeltaeta.xlsx",

"film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film14Scalingdeltaeta.xlsx",

"film15Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx",

"film18Scalingdeltaeta.xlsx",

"film19Scalingdeltaeta.xlsx",

"film20Scalingdeltaeta.xlsx",

"film21Scalingdeltaeta.xlsx" )

}

if(FALSE){

# Film Series Lehigh

File=c( "SheetScalingdeltaeta.xlsx")

}

if(FALSE){

# Film Series 2 for scaling

File=c( "film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx")

}

if(FALSE){

# Film Series 2

File=c( "film10Scalingdeltaeta.xlsx",

"film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film14Scalingdeltaeta.xlsx",

"film16Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx",

"film2Scalingdeltaeta.xlsx",

"film9Scalingdeltaeta.xlsx" )

}

dim=length(File)

combined\_deltaeta=NULL

for (i\_delta\_eta in 1:dim){

delta\_eta = read\_xlsx(File[i\_delta\_eta],"scalinglaw")

#check for end of file / determination of Dimension of data - Dimdata

for (i in 1:1000){

if (is.na(delta\_eta[i,1])){

Dimdata=i-1

break}

}

deltaeta=head(unlist((delta\_eta[,1])),Dimdata)

combined\_deltaeta=c(combined\_deltaeta,deltaeta)

#Distribution\_check(deltaeta,lognormal=FALSE)

}

#### scaling law all data irrespective of frequency - one does not see this in litarature

#### this part can be used to look at limited number files for the plot distribution data

#### (FALSE) leads to analysis of limited number of data

#### provided these have been set accordingly

df=data.frame(combined\_deltaeta)

if(FALSE){WriteXLS(df,"ScalingLawDeltaEta", SheetNames ="Alldata")}else{

WriteXLS(df,"ScalingLawDeltaEta", SheetNames ="Alldata")

}

lognormal="false"

count=length(combined\_deltaeta)

Distribution\_check(combined\_deltaeta,lognormal)

#read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

ylim=c(1/(2\*count),1)

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results")

####scaling law 1 order of magnitude apart: 0.089 Hz vs 0.847 Hz (Bouchaud et al.) for delta eta

if(TRUE){scaling\_nu8=read.table(paste("film9","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

scaling\_nu80=read.table(paste("film21","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

}

if(FALSE){scaling\_nu8=read.table(paste("film10","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

scaling\_nu80=read.table(paste("film17","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

}

deltaeta\_ecdfneg8=subset(scaling\_nu8[,c(1,2)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,2])))

deltaeta\_ecdfpos8=subset(scaling\_nu8[,c(1,3)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,3])))

deltaeta\_tdfneg8=subset(scaling\_nu8[,c(1,8)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,8])))

deltaeta\_tdfpos8=subset(scaling\_nu8[,c(1,9)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,9])))

#

deltaeta\_ecdfneg80=subset(scaling\_nu80[,c(1,2)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,2])))

deltaeta\_ecdfpos80=subset(scaling\_nu80[,c(1,3)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,3])))

deltaeta\_tdfneg80=subset(scaling\_nu80[,c(1,8)], (!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,8])))

deltaeta\_tdfpos80=subset(scaling\_nu80[,c(1,9)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,9])))

#

ylim=c(1/(2\*count),1)

ylab="CDF"

xmax=max(deltaeta\_ecdfpos80,deltaeta\_ecdfneg80,deltaeta\_tdfpos80,

deltaeta\_ecdfpos8,deltaeta\_ecdfneg8,deltaeta\_tdfpos8)

logmax=trunc(log10(xmax))+1

logmax=10^logmax

linmax=(logmax/10)\*(trunc(xmax/(logmax/10))+1)

xlab=expression(paste("|",Delta,"",eta,"| (Pa.s)"))

plot(deltaeta\_ecdfneg80,cex.lab=1.5,col="red",log="xy",ylim=ylim,xlim=c(0.0001,logmax),

ylab=ylab,

xlab=xlab)

points(deltaeta\_ecdfpos80,pch=20,cex=0.5)

#lines(deltaeta\_tdfneg80,col="red")

lines(deltaeta\_tdfpos80,lty=3)

points(deltaeta\_ecdfneg8,pch=20,col="red")

points(deltaeta\_ecdfpos8)

#lines(deltaeta\_tdfneg8,col="red")

lines(deltaeta\_tdfpos8,lty=3)

legend("bottomleft",cex=1,

c(expression(paste("Negative, ",omega,"=5.3 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=0.55 ",rad.s^-1)),

expression(paste("Positive"))),

pch=c(1,20,20,1),

col=c("red","black","red","black"))

#

plot(deltaeta\_ecdfneg80,cex.lab=1.5,col="red",log="y",ylim=ylim,xlim=c(0,linmax),

ylab=ylab,

xlab=xlab)

points(deltaeta\_ecdfpos80,pch=20,cex=0.5)

#lines(deltaeta\_tdfneg80,col="red")

lines(deltaeta\_tdfpos80,lty=3)

points(deltaeta\_ecdfneg8,pch=20,col="red")

points(deltaeta\_ecdfpos8)

#lines(deltaeta\_tdfneg8,col="red")

lines(deltaeta\_tdfpos8,lty=3)

####end of scaling law 1 order of magnitude apart

####0###49#6######17################23#########28

####scaling law 0.089 Hz, 0.104 Hz, 0.207 Hz and 0.847 Hz Mantegna and Stanley (scaled to sd)

#in case of series 3 all FALSE

#series 1

if(TRUE){

films=c("film10","film15","film21")

nus=c(0.65,1.3,5.3)

}

#series 2

if(FALSE){

films=c("film12","film13","film17")

nus=c(0.76,0.94,5.7)

}

#correct\_scaling\_film=NULL

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

all\_scaled\_datax=NULL#input data of negative and positive values for fitdistr

all\_scaled\_datay=NULL

plot\_datax=NULL

#plotfile=TRUE for making one file else 4 files txt with a cdf version

plotfile=FALSE

if(plotfile){

plot\_index=list()

for (i in 1:(length(films)\*2)){plot\_index[i]=NULL}

#plot\_index=1>8 expcdfng\_film9 > \_film21

cdf\_=NULL

}

# to get the required data from the 4 film txt files and plotting (for publication)

# for series of frequencies/films make a plot

for (i in 1:length(films)){

file=paste(films[i],"plot\_correct\_scaling.txt",sep="")

correct\_scaling=read.table(file=file,header=TRUE)

#going from 3 columns - x y1 (>0) y2 (<0/made abs) - to 2, x y for fitdistr function

#uneven nr are negative deltas

nr=2\*i-1

cdf=subset(correct\_scaling[,c(1,2)],(!is.na(correct\_scaling[,1]) & !is.na(correct\_scaling[,2])))

cdf\_neg=-cdf[,1]#all absolute negative values > negative for fitdistr for uneven nr

all\_scaled\_datax=c(all\_scaled\_datax,cdf\_neg)#for fitdistr

plot\_datax=c(plot\_datax,cdf[,1])

#plot for Rstudio

col="red"

log="y"

if(nr==1){plot(cdf,cex.lab=1.5,log=log,type="p",pch=1,col=col,cex=2/nr,

ylim=c(0.001,1),

xlim=c(0.0001,10),

ylab="CDF",

#xlab=expression(paste("|",Delta,"",eta,"(",omega,")| / ",sigma,"(",omega,")"))

xlab=expression(paste("|",Delta,"",eta,"| / ",sigma,"(",Delta,"",eta,")"))

)

}else{points(cdf,col=col,pch=1,cex=2/nr)}

#

if(plotfile){

plot\_index[[nr]]=c(plot\_index[[nr]],cdf[,2])

for (index in (nr+1):8){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_neg)))}

if(nr>1){for (index in 1:(nr-1)){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_neg)))}}

}

#

#even nr are positive deltas

nr=2\*i

#all positive values remain positive to calculate fitdistr

cdf=subset(correct\_scaling[,c(1,3)],(!is.na(correct\_scaling[,1]) & !is.na(correct\_scaling[,3])))

cdf\_=cdf[,1]

all\_scaled\_datax=c(all\_scaled\_datax,cdf\_)

plot\_datax=c(plot\_datax,cdf[,1])

#plot for Rstudio

col="black"

points(cdf,col=col,pch=1,cex=2/(nr-1))

if(plotfile){

plot\_index[[nr]]=c(plot\_index[[nr]],cdf[,3])

for (index in 1:(nr-1)){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_)))}

for (index in (nr+1):8){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_)))}

}

}

#print(all\_data)

#lognormal="false"

#Distribution\_check(all\_scaled\_datax,lognormal)

Param=fitdistr(all\_scaled\_datax,"t",start = list(m=mean(all\_scaled\_datax),s=sd(all\_scaled\_datax), df=3), lower=c(-0.05,0.5,1),upper=c(0.05,50,50))

tm =coef(Param)[1]

ts =coef(Param)[2]

tdf=coef(Param)[3]

t\_general=function(x,mu,sigma,nu){

1/sigma\*

gamma((nu+1)/2)/

sqrt(nu\*pi)/gamma(nu/2)\*

(1+((x-mu)^2)/nu/sigma/sigma)^(-(1+nu)/2)

}

#delta\_eta is the key variable and equal to all\_scaled\_datax

delta\_eta=all\_scaled\_datax

count=length(delta\_eta)

min\_eta=min(delta\_eta)

max\_eta=max(delta\_eta)

if(abs(min\_eta)>max\_eta){

max\_eta=abs(min\_eta)

}else{

min\_eta=-max\_eta

}

DELTA=max\_eta-min\_eta

deltaeta\_tstudent=NULL

cdft=NULL

tdist=NULL

#negcfdt=NULL

#poscfdt=NULL

cdft[1]=t\_general(min\_eta,tm,ts,tdf)

tdist[1]=cdft[1]

deltaeta\_tstudent[1]=min\_eta

for(i in 2:count){

deltaeta\_tstudent[i]=min\_eta+(i-1)\*DELTA/count

if(abs(deltaeta\_tstudent[i])<0.0001){

deltaeta\_tstudent[i]=0.0001}

tdist[i]=t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

cdft[i]=cdft[i-1]+t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

}

cdft=cdft/cdft[count]

deltaeta\_tstudent\_neg=abs(deltaeta\_tstudent[deltaeta\_tstudent<0])

negcdft=cdft[deltaeta\_tstudent<0]

deltaeta\_tstudent\_pos=deltaeta\_tstudent[deltaeta\_tstudent>=0]

poscdft=1-cdft[deltaeta\_tstudent>=0]

count\_tstudent\_neg =length(deltaeta\_tstudent\_neg)

count\_tstudent\_pos =length(deltaeta\_tstudent\_pos)

#points(log="xy",xneg,yneg,pch=17,cex=1.5,col="red")

#points(xpos,ypos,pch=20)

lines(deltaeta\_tstudent\_neg,negcdft,lwd=2,col='red')

lines(deltaeta\_tstudent\_pos,poscdft,lwd=2,lty=2)

mtext(cex=0.75,paste("mean= ",round(tm,2)," df= ",round(tdf,2)," scale= ",round(ts,2)),3,1)

if(log=="xy"){position="bottomleft"}else{position="topright"}

#in case of series 3 all FALSE

if(TRUE){#Series 1

legend(position,cex=1,

#change frequencies when necessary series 1 0.104, 0.207, 0.847

#nus=c(0.65,1.3,5.3)

c(expression(paste("Negative, ",omega,"=0.65 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=1.3 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=5.3 ",rad.s^-1)),

expression(paste("Positive"))),

pt.cex=c(1.5,1.5,1,1,0.5,0.5),

pch=c(1,1,1,1,1,1),col=c("red","black","red","black","red","black"))

}

if(FALSE){#Series 2

legend(position,cex=1,

#change frequencies when necessary series 1 0.104, 0.207, 0.847

#nus=c(0.76,0.94,5.7)

c(expression(paste("Negative, ",omega,"=0.76 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=0.94 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=5.7 ",rad.s^-1)),

expression(paste("Positive"))),

pt.cex=c(1.5,1.5,1,1,0.5,0.5),

pch=c(1,1,1,1,1,1),col=c("red","black","red","black","red","black"))

}

Format=function(number){sprintf("%3.6f",number)}#to make stardardized Format output

plot\_cdfmasterstudent=data.frame(Format(c(deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)),

Format(c(negcdft,rep(NA,count\_tstudent\_pos))),

Format(c(rep(NA,count\_tstudent\_neg),poscdft)))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_cdfmasterstudent,quote=FALSE,file="plot\_mastercurve\_Student.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

#make a dataframe

#series 1 films 9, 10, 15, 21

#series 2 films 12, 13, 17

if(plotfile){

all\_scaled\_datax=c(all\_scaled\_datax,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

negcdft=c(rep(NA,length(all\_scaled\_datax)),negcdft,rep(NA,count\_tstudent\_pos))

poscfdt=c(rep(NA,length(all\_scaled\_datax)),rep(NA,count\_tstudent\_neg),poscdft)

#assigning the right plot\_index[[i]] (a list) to the right film

if(TRUE){

plot\_expcdfneg\_film9=plot\_index[[1]]

plot\_expcdfpos\_film9=plot\_index[[2]]

plot\_expcdfneg\_film10=plot\_index[[3]]

plot\_expcdfpos\_film10=plot\_index[[4]]

plot\_expcdfneg\_film15=plot\_index[[5]]

plot\_expcdfpos\_film15=plot\_index[[6]]

plot\_expcdfneg\_film21=plot\_index[[7]]

plot\_expcdfpos\_film21=plot\_index[[8]]

plot\_all\_data=data.frame(Format(plot\_datax),

Format(plot\_expcdfneg\_film9),

Format(plot\_expcdfpos\_film9),

Format(plot\_expcdfneg\_film10),

Format(plot\_expcdfpos\_film10),

Format(plot\_expcdfneg\_film15),

Format(plot\_expcdfpos\_film15),

Format(plot\_expcdfneg\_film21),

Format(plot\_expcdfpos\_film21),

Format(negcfdt),

Format(poscfdt))

}

if(FALSE){

plot\_expcdfneg\_film12=plot\_index[[1]]

plot\_expcdfpos\_film12=plot\_index[[2]]

plot\_expcdfneg\_film13=plot\_index[[3]]

plot\_expcdfpos\_film13=plot\_index[[4]]

plot\_expcdfneg\_film17=plot\_index[[5]]

plot\_expcdfpos\_film17=plot\_index[[6]]

plot\_all\_data=data.frame(Format(plot\_datax),

Format(plot\_expcdfneg\_film12),

Format(plot\_expcdfpos\_film12),

Format(plot\_expcdfneg\_film13),

Format(plot\_expcdfpos\_film13),

Format(plot\_expcdfneg\_film17),

Format(plot\_expcdfpos\_film17),

Format(negcfdt),

Format(poscfdt))

}

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_all\_data,quote=FALSE,file="plot\_mastercurve\_delta\_eta\_scaled.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

}

***RAW DATA READING***

function(analyserawdata){

# either testing (of raw data or reading txt files and writing them to excel files -initial setup)

# first testing (of raw data)

setwd("/Users/macbookair/Desktop/Series/InputData")

#Extensive Testing (time (etamin) etamin time (etamax) etamax)

#setwd("/Users/macbookair/Desktop/BerretMicroRheology/Berret/Data/Raw data txt files")

#if KO go to line 169

### AnalyseRawData must still be written

if(analyserawdata=="OK"){

dataset = read.delim("Recol\_film1 57mHz.txt", sep = "\t", nrows=500, header = TRUE, stringsAsFactors = FALSE)

dim(dataset)

#

# skip empty files / test the size of each file， and skip the file size of 0

#for (file in list.files(,"\*.txt")){

# if (file.size(file) == 0) next

# print(file)

#}

# In case header=true: first row (number 1 = first row of matrix ) is data

# check below with Pil=OK and row=1 / print individual lines Pil=OK and row=number

Pil="OK"

if (Pil=="OK")

{

#give row number

row=1800

if (is.na(dataset[row,3])){

print("stop due to NA")

print(row)}

firstline=c(dataset[row,1],dataset[row,2],dataset[row,3])

print (firstline)

}

for (i in 1000:3000){

if (is.na(dataset[i,3])){

print("stop due to NA")

print(i-1)

iend=i-1

print(dataset[i-1,3])

print(dataset[i,3])

break}

}

quit(save="ask")

}

dim=length(File)

#combined\_eta\_all=NULL

for (i\_t\_eta in 1:dim){

#print(File[i\_t\_eta])

t\_eta = read.delim(File[i\_t\_eta], sep = "\t", nrows=500, header = TRUE, stringsAsFactors = FALSE)

print(t\_eta)

#check for end of file / determination of Dimension of data - Dimdata

for (i in 1:500){

if (is.na(t\_eta[i,5])){

Dimdata=i-1

break}

}

t=head(unlist((t\_eta[,1])),Dimdata)

eta=head(unlist((t\_eta[,5])),Dimdata)

df\_t\_eta=data.frame(t,eta)

WriteXLS(df\_t\_eta, paste((File[i\_t\_eta]),".xlsx",sep=""), SheetNames = "Checkdata")

#combined\_eta\_all=c(combined\_eta\_all,eta)

}

}

***FIT GENERAL FUNTION***

function(film,fitlag,fitvariogram,averagetime,FIT,pos\_line){

lwd=2

if (FIT==0){

Modelhole=function(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_GRW,C3\_GRW)

{C0\_constant\_cosinus\*(1-cos(2\*pi\*averagetime\*fitlag/C1\_T))+

2\*C2\_constant\_GRW\*(1-C3\_GRW^fitlag)/(1-C3\_GRW^2)}

#in case of damped cosine

#/(6.28\*fitlag/C2))

maxvar=max(fitvariogram)

#0.5 range covered by periodic function on Variogram axis

fraction\_period=0.5

C0l=fraction\_period\*maxvar/5

C0m=C0l\*5

#period=2\*averagetime\*fitlag[which.max(fitvariogram)]

period=150

int=100

# series 1

if(FALSE){

if(film=="film20"|film=="film18"|film=="film21"){

period=150

int=30

}

if(film=="film1"|film=="film19"){

period=80

int=40

}

if(film=="film14"){

period=40

int=20

}

}

# series 2

if(FALSE){

if(film=="film17"){

period=50

int=25

}

}

# series 3

if(FALSE){

if(film=="film20"|film=="film18"){

period=80

int=20

}

}

# Lehigh\_data

if(film=="Lehigh\_data"|film=="Lehigh\_data\_VolatilityClustering"){

period=20

int=10

}

# series 4

if(T){

if(film=="run1-Curo"){

#C0l=0.00005

#C0m=0.00005

period=250

int=25

C3l=0.97

C3m=0.97

#C2l=0.00002

#C2m=0.00002

}

if(film=="run2-Curo+silica"){

#C0l=0.00001

#C0m=0.00001

period=400

int=100

C3l=0.95

C3m=0.95

#C2l=0.00002

#C2m=0.00002

}

if(film=="run3-Curo+alumina"){

period=150

int=50

C3l=0.479

C3m=0.479

}

C1l=period-int

C1m=period+int

C2l=0.1\*C0l

C2m=C0m

# constant C2/C3= fixed

if(F){

C2l=0.036

C2m=C2l

C3l=0.658

C3m=C3l

}

#

C0s=(C0l+C0m)/2

C1s=(C1l+C1m)/2

C2s=(C2l+C2m)/2

C3s=(C3l+C3m)/2

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_GRW,C3\_GRW),

start=list(C0\_constant\_cosinus=C0s,C1\_T=C1s,C2\_constant\_GRW=C2s,C3\_GRW=C3s),

lower=c(C0l,C1l,C2l,C3l),

#upper=c(C0l,C1l,C2l,C3l))

upper=c(C0m,C1m,C2m,C3m))

# C0 amplitude C1 period C2 alfa C3 sigma^2

C0\_constant\_cosinus =coef(variogramfit)[1]

C1\_T =coef(variogramfit)[2]

C2\_constant\_GRW =coef(variogramfit)[3]

C3\_GRW =coef(variogramfit)[4]

#lines(lwd=lwd,predict(variogramfit),col='red')

x=(10:(10\*length(fitlag)))/10

lines(lwd=lwd,predict(variogramfit,list(x)),col='red')

lines(lwd=lwd,x,2\*C2\_constant\_GRW\*(1-C3\_GRW^x)/(1-C3\_GRW^2),col="blue")

mtext(cex=0.75,paste0("T(s)=",round(C1\_T,1)," alfa=",round(C3\_GRW,3)," sigma^2=",round(C2\_constant\_GRW,3)),3,pos\_line)

# plot for publication

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

GRW=2\*C2\_constant\_GRW\*(1-C3\_GRW^fitlag)/(1-C3\_GRW^2)

Periodic\_plus\_GRW=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,Periodic\_plus\_GRW,GRW)

file=paste(film,"plot\_fit\_variogram\_GRW.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,na="",quote=FALSE,sep=" ",col.names=TRUE,row.names=FALSE)

}

}

}

#(damped) cosinus with Ornstein-Uhlenbeck contribution (Mean Reverting Mechanism)

if (FIT==1){

Modelhole=function(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_OU,C3\_correlationtime)

{C0\_constant\_cosinus\*(1-cos(2\*pi\*averagetime\*fitlag/C1\_T))+

C2\_constant\_OU\*(C3\_correlationtime/averagetime)\*(1-exp(-fitlag\*averagetime/C3\_correlationtime))}

#in case of damped cosine

#/(6.28\*fitlag/C2))

maxvar=max(fitvariogram)

#0.5 range covered by periodic function on Variogram axis

fraction\_period=0.5

C0l=fraction\_period\*maxvar/5

C0m=C0l\*5

#C1=6.28/estimated period = 100 lags > 0.0628

#period=2\*fitlag[which.max(fitvariogram)]

#period=2\*averagetime\*fitlag[which.max(fitvariogram)]

period=150

int=100

# all FALSE for series 3

# series 1

if(TRUE){

if(film=="film20"|film=="film18"|film=="film21"){

period=120

int=30

}

if(film=="film1"|film=="film19"){

period=80

int=40

}

if(film=="film14"){

period=40

int=20

}

}

# series 2

if(FALSE){

if(film=="film17"){

period=50

int=25

}

}

# series 3

if(FALSE){

if(film=="film20"|film=="film18"){

period=80

int=20

}

}

if(FALSE){

# Lehigh\_data

if(film=="Lehigh\_data"|film=="Lehigh\_data\_VolatilityClustering"){

period=20

int=10

}

}

# series 4

if(FALSE){

if(film=="run1-Curo"){

#C0l=0.00005

#C0m=0.00005

period=250

int=25

C3l=22

C3m=22

#C2l=0.00002

#C2m=0.00002

}

if(film=="run2-Curo+silica"){

#C0l=0.00001

#C0m=0.00001

period=400

int=100

C3l=22

C3m=22

#C2l=0.00002

#C2m=0.00002

}

if(film=="run3-Curo+alumina"){

period=150

int=50

C3l=0.68

C3m=0.68

}

}

C1l=period-int

C1m=period+int

C2l=0.1\*C0l

C2m=C0m

#C3l=0.01

##C3m=0.5\*fitlag[which.max(fitvariogram)]

#C3m=5\*averagetime

# constant C2/C3= fixed for the master curve

if(FALSE){

C2l=0.058

C2m=C2l

C3l=2.386

C3m=C3l

}

#

C0s=(C0l+C0m)/2

C1s=(C1l+C1m)/2

C2s=(C2l+C2m)/2

C3s=(C3l+C3m)/2

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_cosinus,C1\_T,C2\_constant\_OU,C3\_correlationtime),

start=list(C0\_constant\_cosinus=C0s,C1\_T=C1s,C2\_constant\_OU=C2s,C3\_correlationtime=C3s),

lower=c(C0l,C1l,C2l,C3l),

#upper=c(C0l,C1l,C2l,C3l))

upper=c(C0m,C1m,C2m,C3m))

# C0 amplitude C1 period C2 UO prefactor sigma^2 C3 1/relaxation lag

C0\_constant\_cosinus =coef(variogramfit)[1]

C1\_T =coef(variogramfit)[2]

C2\_constant\_OU =coef(variogramfit)[3]

C3\_correlationtime =coef(variogramfit)[4]

#lines(lwd=lwd,predict(variogramfit),col='red')

x=(10:(10\*length(fitlag)))/10

lines(lwd=lwd,predict(variogramfit,list(x)),col='red')

lines(lwd=lwd,x,C2\_constant\_OU\*(C3\_correlationtime/averagetime)\*(1-exp(-x\*averagetime/C3\_correlationtime)),col="blue")

mtext(cex=0.75,paste0("T(s)=",round(C1\_T,1)," tau(s)=",round(C3\_correlationtime,3)," sigma^2=",round(C2\_constant\_OU,3)),3,pos\_line)

# plot for publication - dending on the series

if(film=="film15" | film=="film10" | film=="film21"){

#if(film=="film12" | film=="film13" | film=="film17"){

OUP=C2\_constant\_OU\*(C3\_correlationtime/averagetime)\*(1-exp(-fitlag\*averagetime/C3\_correlationtime))

Periodic\_plus\_OUP=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,Periodic\_plus\_OUP,OUP)

file=paste(film,"plot\_fit\_variogram\_OUP.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE)

}

}

#GRW mean reverting mechanism

if (FIT==2){

Modelhole=function(fitlag,C0\_constant\_GRW,C1\_GRW){2\*C0\_constant\_GRW\*(1-C1\_GRW^fitlag)/(1-C1\_GRW^2)}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_GRW,C1\_GRW),start=list(C0\_constant\_GRW=0.05,C1\_GRW=0.5))

C0\_constant\_GRW =coef(variogramfit)[1]

C1\_GRW =coef(variogramfit)[2]

lines(lwd=lwd,predict(variogramfit),col='blue')

mtext(cex=0.75,paste0("alfa=",round(C1\_GRW,3)," sigma^2=",round(C0\_constant\_GRW,3)),3,pos\_line)

# plot for publication

if(length(fitlag)>7){

GRW=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,GRW)

file=paste(film,"plot\_fit\_variogram\_GRW\_only.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,na="",quote=FALSE,sep=" ",col.names=TRUE,row.names=FALSE)

}

}

#Ornstein Uhlenbeck mean reverting mechanism

if (FIT==3){

Modelhole=function(fitlag,C0\_constant\_OUP,C1\_correlationtime){C0\_constant\_OUP\*(C1\_correlationtime/averagetime)\*(1-exp(-fitlag\*averagetime/C1\_correlationtime))}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0\_constant\_OUP,C1\_correlationtime),start=list(C0\_constant\_OUP=0.1,C1\_correlationtime=5))

C0\_constant\_OUP =coef(variogramfit)[1]

C1\_correlationtime =coef(variogramfit)[2]

lines(lwd=lwd,predict(variogramfit),col='blue')

mtext(cex=0.75,paste0("tau=",round(C1\_correlationtime,3)," sigma^2=",round(C0\_constant\_OUP,3)),3,pos\_line)

# plot for publication

if(length(fitlag)>7){

OUP=predict(variogramfit,list(fitlag))

plot\_fit=data.frame(fitlag,fitvariogram,OUP)

file=paste(film,"plot\_fit\_variogram\_OUP\_only.txt",sep="")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/")

write.table(format(plot\_fit,digits=3),file=file,quote=FALSE,na="",sep=" ",col.names=TRUE,row.names=FALSE)

}

}

#(damped) cosinus - hast to be rewritten (constant names constant in terms of seconds)

if (FIT==4){

Modelhole=function(fitlag,C0,C1,C2){C0+C1\*(1-cos(C2\*fitlag))}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0,C1,C2),start=list(C0=0.32,C1=0.1,C2=0.05),

lower=c(0.04,0.01,0.01),

upper=c(0.4,0.15,0.5))

C0 =coef(variogramfit)[1]

C1 =coef(variogramfit)[2]

C2 =coef(variogramfit)[3]

lines(predict(variogramfit),col='red')

mtext(cex=0.75,paste0("T= ",round(2\*pi\*averagetime/C2,1)),3,pos\_line)

}

#constant

if (FIT==5){

Modelhole=function(fitlag,C0){C0+0\*fitlag}

variogramfit=nlsLM(fitvariogram~Modelhole(fitlag,C0),start=list(C0=1))

C0 =coef(variogramfit)[1]

lines(predict(variogramfit),col='red')

mtext(cex=0.75,paste0("cte= ",round(C0,3)),3,pos\_line)

}

print(summary(variogramfit))

#csv file> , to ; >. to , then text to columns

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

write.table(tidy(variogramfit),paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),append=FALSE)

#print(coef(variogramfit))

#print(confint(variogramfit))

#plot(fitlag,residuals(variogramfit),pch=20,col="grey")

if(length(fitlag)>10){hist(residuals(variogramfit))}

Variogram\_fit=list(summary(variogramfit))

}