Scalar field theory on the lattice

Johannes Schmidt and Till Fohrmann

This document is intended as a brief showcase of some of the analysis we did in R to produce the results presented in our report.

```
library(Hmisc)
library(boot)
```

Function definitions

```
eff_mass <- function(corr){
  #Calculates the effective mass for all time steps.

  N <- length(corr)
  return(log(corr[1:N-1]/tail(corr,N-1)))
}

eff_mass_sym <- function(corr){
  #Calculates the effective mass for all time steps.

  N <- length(corr)
  return(acosh((corr[3:N]+corr[1:(N-2)]) / (2*corr[2:(N-1)])))
}

mean.boot <- function(x, ii){
  return(apply(X=x[ii,], MARGIN=2, FUN=function(x) mean(x[is.finite(x)])))
}

meff.boot <- function(x, ii){
  return(eff_mass_sym(apply(X=x[ii,], MARGIN=2, FUN=mean)))
}
```

Read correlators

```
correlator <- simplify2array(read.csv("./long_m04f0.txt", header=F))
correlator <- correlator[1:10000,]
dim(correlator)
```

```
## [1] 10000    17
```

Bootstrap

```
boot.res <- boot(data=correlator, statistic=meff.boot, R=4*length(correlator[,1]))
print(boot.res)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = correlator, statistic = meff.boot, R = 4 * length(correlator[,
##     1]))
##
##
## Bootstrap Statistics :
##       original        bias     std. error
## t1*  1.7398884  3.888531e-07 0.0001865416
## t2*  1.4794406 -5.891469e-06 0.0007961055
## t3*  1.2055863 -2.368560e-06 0.0026752526
## t4*  0.9805232 -1.016453e-05 0.0074082635
## t5*  0.8268869 -9.261343e-05 0.0184131023
## t6*  0.7550981  5.507320e-05 0.0415977849
## t7*  0.8256905 -1.136457e-03 0.0825465135
## t8*  0.2658187  1.203409e-01 0.1610097019
## t9*  0.6871113  2.198219e-02 0.2531039107
## t10* 0.4287127  2.799236e-01 0.3735831766
## t11*       NaN           NaN 0.7027674810
## t12*       NaN           NaN 1.1479454069
## t13* 0.7389620  6.898703e-01 1.0029816107
## t14*       NaN           NaN 1.1018189332
## t15*       NaN           NaN 0.6540149579
```

```
print("Number of bootstrap samples for each time step")
```

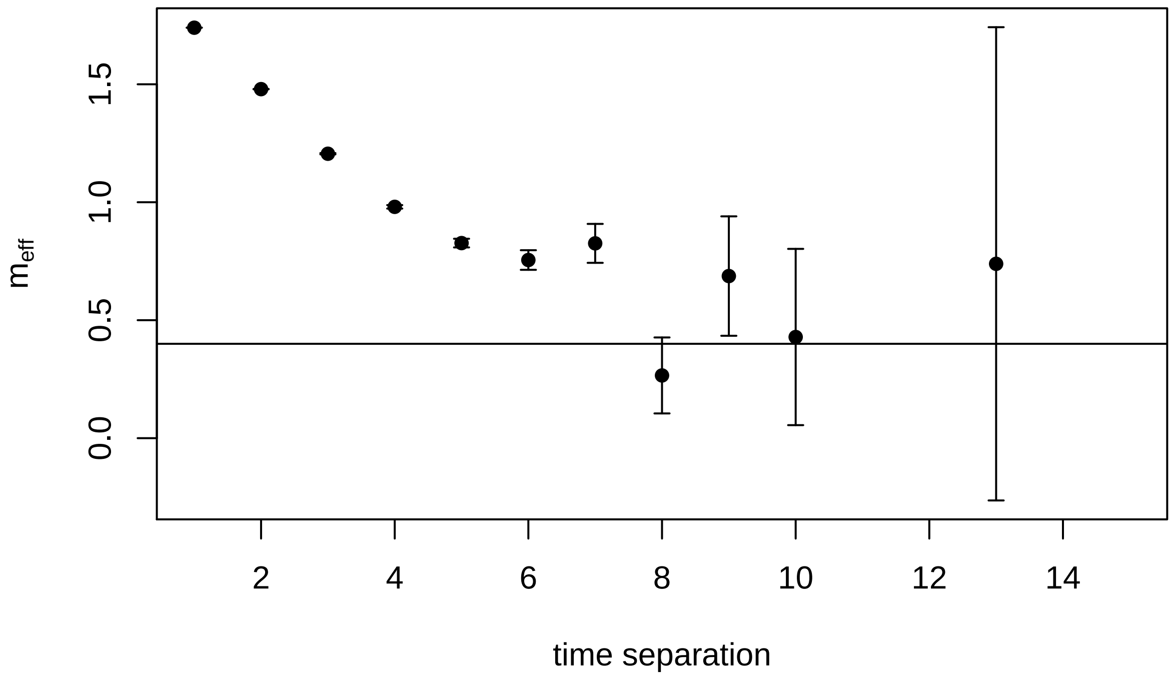
```
## [1] "Number of bootstrap samples for each time step"
```

```
print(apply(boot.res$t, 2, FUN=function(x) length(x[is.finite(x)])))
```

```
##  [1] 40000 40000 40000 40000 40000 40000 40000 26578 37434 25807  5743  7482
## [13] 11608 12251 10036
```

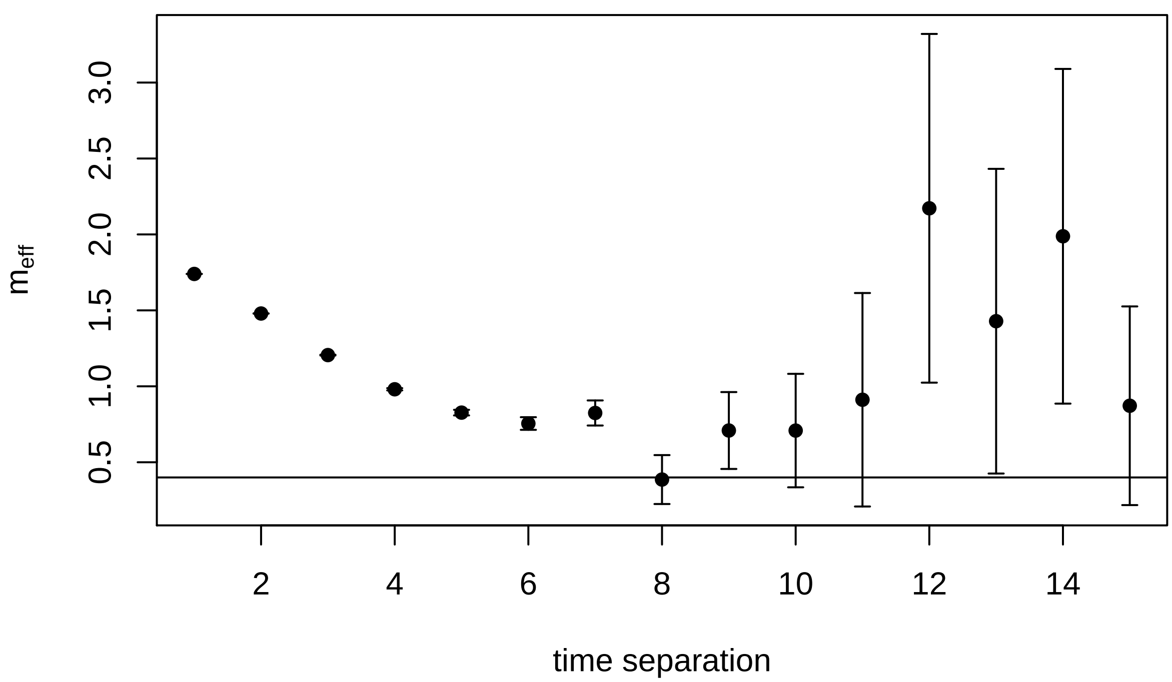
Evaluation as the library function would do it

```
boot_val <- boot.res$t0
boot_sd <- apply(X=boot.res$t, MARGIN=2, FUN=function(x) sd(x, na.rm=TRUE))
errbar(seq(1,length(boot_val)),boot_val, boot_val+boot_sd, boot_val-boot_sd, ann=F)
title(xlab="time separation", ylab=expression(paste("m"[eff])))
abline(h=0.4)
```



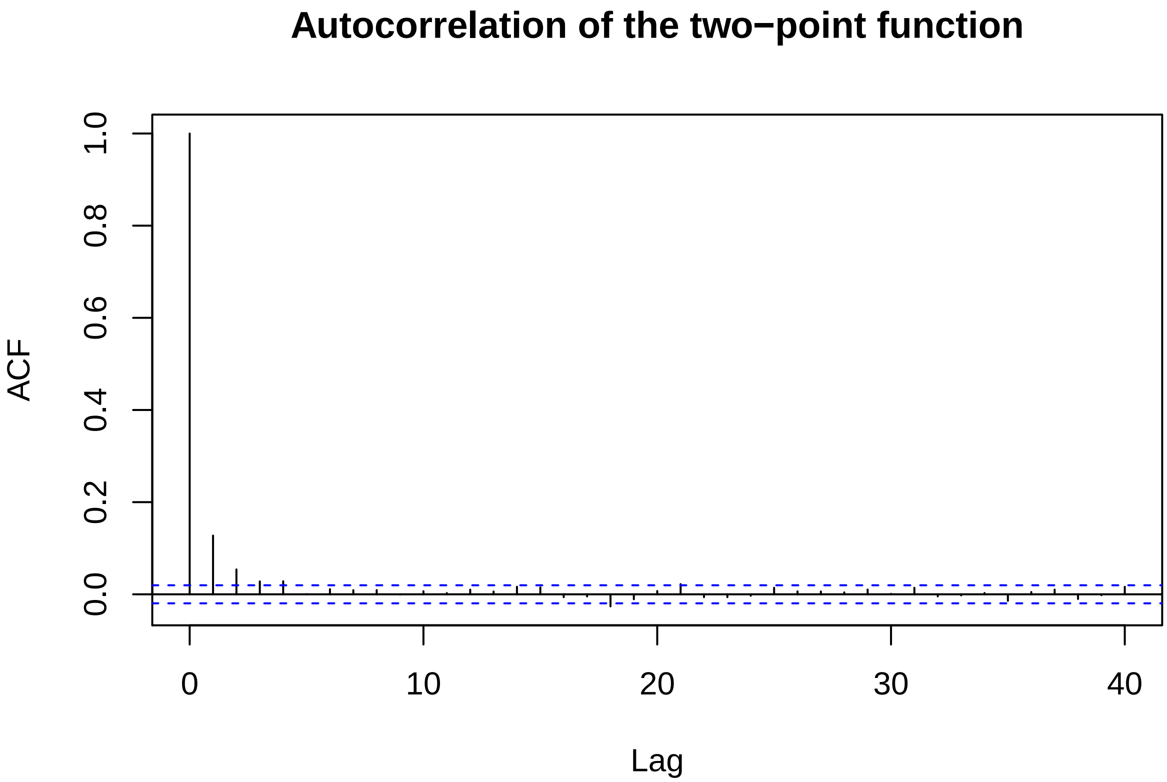
Evalutation with omitting the NaNs

```
#boot.res$t
boot_val <- apply(X=boot.res$t, MARGIN=2, FUN=function(x) sum(x[is.finite(x)]) / length(x[is.finite(x)]
boot_sd <- apply(X=boot.res$t, MARGIN=2, FUN=function(x) sd(x, na.rm=TRUE))
errbar(seq(1,length(boot_val)),boot_val, boot_val+boot_sd, boot_val-boot_sd, ann=F)
title(xlab="time separation", ylab=expression(paste("m"[eff])))
abline(h=0.4)
```



Autocorrelation of the measurment

```
acf(correlator[,3], main="Autocorrelation of the two-point function")
```




```
ac <- acf(correlator[,3], plot=FALSE, lag.max=1000)$acf
```