Optimal Heavy Tail Estimation

Heavy tail index estimation (Hill or Pickands estimator) on time series data by means of a brute force order selector (RMSE measure), including preprocessing routines.

Version 1.0 (May 2017)

— User Manual

Climate Risk Analysis – Manfred Mudelsee
http://www.climate-risk-analysis.com

ht Version 1.0
May 2017

Climate Risk Analysis – Manfred Mudelsee e. K.
Kreuzstraße 27, Heckenbeck, 37581 Bad Gandersheim, Germany
HRA 20 13 94 (Amtsgericht Hannover)
http://www.climate-risk-analysis.com

Getting Started

Notation

We use the symbol ↵ to denote a stroke of the "Enter" key.

Files	Short description
cmd2.lnk	Windows console (command line)
ht.exe	Windows executable
ht-estimate.cfg	Configuration file example (estimation)
ht-generate.cfg	Configuration file example (generation)
mzranseed.dat	Seed file (random numbers)
HT-Manual.pdf	Manual
ht.f90	Fortran 90 source
nrtype.f90	Numerical Recipes file
rng.f90	Fortran 90 source (random numbers)
Example.dat	Example time series file

Explanation of files

The Windows console is explained under "Adaptation (cmd.exe)" on this page.

The Windows executable and the random number seed file are required for using the executable. The configuration files are optional; they are used for an automatic work modus.

The *.f90 files are required for compiling a new version of ht.exe (e.g., for another operating system than Windows, such as Linux).

The example time series file serves for illustration and explanation of the required input data format.

Installation (ht)

(1) Make an ht folder of your choice, let us say C:\ht.
(2) Copy the above mentioned files into that folder.

Installation (cmd.exe)

This is the command-line or console software. It is a part of the Windows system package, residing typically at C:\Windows\system32\.

You may also download cmd.exe at the Microsoft internet site, www.microsoft.com.

Adaptation (cmd.exe)

This step is optional but recommended to achieve a convenient work flow.
(1) Make a shortcut on the desktop:
> right-click on an open area on the desktop
> New > Shortcut > (browse to locate cmd.exe) > Next > Finish
(2) Adapt the fonts:
> right-click on the shortcut symbol for cmd.exe > Properties > Font > (make your choice; for example, on my 1280 x 1024 screen, I am using the font "Lucida Console bold 20 pt")
(3) Adapt the console window layout:
> right-click on the shortcut symbol for cmd.exe > Properties > Layout > Window Size (width x height) 146 x 56 and Windows Buffer Size 146 x 9000
(4) Adapt the colours:
> right-click on the shortcut symbol for cmd.exe > Properties > Colours > (make your choice; for example, I am using a black screen text on a grey background)

The shortcut to cmd.exe provided by CRA (cmd2.lnk), to be copied to your ht folder and then double-clicked, includes an adaptation.

```
Optimal Heavy Tail Estimation
=============================

Heavy tail index estimation (Hill or Pickands estimator)
on time series data by means of a brute force order selector
(RMSE measure), including preprocessing routines.

Version 1.0 (April 2017)

Copyright (C) Climate Risk Analysis - Manfred Mudelsee
http://www.climate-risk-analysis.com

Please see source code (ht.f90) for disclaimer and permissions.

PRESS ENTER TO CONTINUE
```

Screenshot 1 ht welcome screen.

Configuration files

The content of these files, the description of the used parameters is contained in these files (in the commented part at the bottom). They are preset before program start and cannot be changed while the program runs.

Executing the software

It is possible to run ht by double-clicking (e.g., in Windows Explorer) on "ht.exe", but it is more convenient to use the command-line window with the keyboard from the beginning.
(1) Double-click the cmd.exe shortcut on your computer desktop: the command-line window ("ht window") opens.
(2) Change into the ht directory by typing on the keyboard:
> cd C:\ht (followed by pressing the "Enter" key: ↵)
In case the ht directory is on another hard-drive than C:, say F:, you may have to switch to the harddrive first:
> F: ↵
> cd F:\ht ↵

Then, run ht by one of the two following:
(1) > ht.exe ↵
This is modus 'per-hand'.
(2) > ht.exe [configuration file name] ↵
The modus is set in the configuration file.

In the modus 'per-hand', you see first the ht welcome screen (**Screenshot 1**).

Work modus

ht can analyse time series in three modes.
(1) 'per-hand'
Here you type everything into the keyboard that is required for the processing (e.g., the estimation type).
(2) 'generate'
Here everything is read automatically from the configuration file; and the analysis consists in the automatic generation of new, artificial time series data.
(3) 'estimate'
Here everything is read automatically from the configuration file; and the analysis consists in the automatic estimation of the heavy tail index.

Many pairs of 'generate'–'estimate' calls within a batch file can be used for performing external (i.e., outside of ht) Monte Carlo experiments.

The three modes are described in the following Sections.

Input data format

Input data is a time series $\{t(i), x(i)\}_{i=1,\dots,n}$.
● no headers
● exactly two entries ($t(i)$, $x(i)$) per line, that is, no missing values
● ASCII format
● no decimal comma
● decimal point possible, but not required
● no thousands comma
● $t(i)$ strictly monotonically increasing (more precisely, time can also be strictly monotonically decreasing; since for the analysis a strictly monotonically increasing series is required, in the preprocessing the time direction has then to be reversed)

See Example.dat (**Screenshot 2**) for an illustration.

Data size

ht can process virtually unlimited volumes of data. Owing to dynamic memory allocation in Fortran 95, the only limit is set in principle by the memory (RAM) of your computer.

The maximum allowed data size (n) is given by the parameter *Nobsmax*, which currently (Version 1.0) is set equal to a value of 250000 (see ht.f90).

Screenshot 2 Example.dat (excerpt).

Section 1: Modus 'per-hand'

Continuing (↵) from the welcome screen (**Screenshot 1**) brings you to the main menu (**Screenshot 3**).

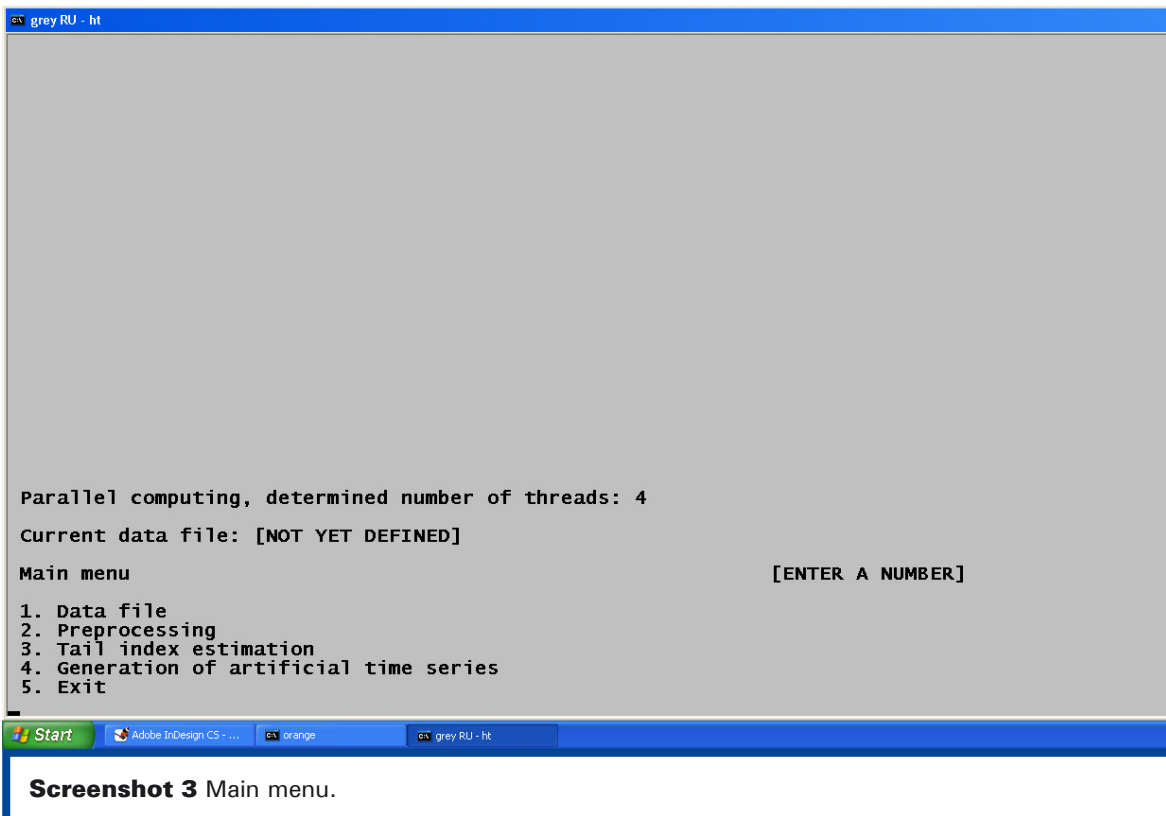
Displayed are the number of threads on the computer available for parallel computing and the current data file (not yet defined).

Below you see the choices of the main menu:
(1) Data file
(2) Preprocessing
(3) Tail index estimation
(4) Generation of artificial time series
(5) Exit

Data file

First, a data file has to be selected. This is done by typing the number and pressing Enter (1↵). You are promted to supply the name (possible: path and name). Then (↵) the time series is read and the data size is given. Then (↵) the main menu re-appears.

```
 grey RU - ht

 Parallel computing, determined number of threads: 4

 Current data file: [NOT YET DEFINED]

 Main menu                                              [ENTER A NUMBER]

 1. Data file
 2. Preprocessing
 3. Tail index estimation
 4. Generation of artificial time series
 5. Exit

 Start      Adobe InDesign CS - ...    orange         grey RU - ht
```

Screenshot 3 Main menu.

Preprocessing

Preprocessing is only possible if a data file has been selected.

There exist various preprocessing types.

Reversing the time direction is necessary for transforming a series with strictly monotonically decreasing time (no screenshot shown).

The other preprocessing types (↵) are displayed in **Screenshot 4**. Some types require further input (e.g., linear transformation).

Note that some preprocessing types reduce the data size of the series on which the heavy tail index is estimated.

After preprocessing, you may go back to the previous menu(s).

```
Original                         time series: Torig(1:Norig), Xorig(1:Norig)
Current (possibly preprocessed) time series: T(1:N),         X(1:N)

Select transformation for X                              [ENTER A NUMBER]

1. Natural log transformation
   X = log(X)

2. Linear transformation
   X = a + b * X

3. First differences
   X(1:N-1) = X(2:N) - X(1:N-1)
   Note: data size reduced by one

4. Returns
   X(1:N-1) = [X(2:N) - X(1:N-1)] / [X(1:N-1)]
   Note: data size reduced by one

5. Natural log returns
   X(1:N-1) = log[X(2:N)] - log[X(1:N-1)]
   Note: data size reduced by one

6. Retain only values above a selectable threshold
   X(1:N) > [threshold]
   Note: data size reduced

7. Previous menu
```

Screenshot 4 Preprocessing types.

Tail index estimation

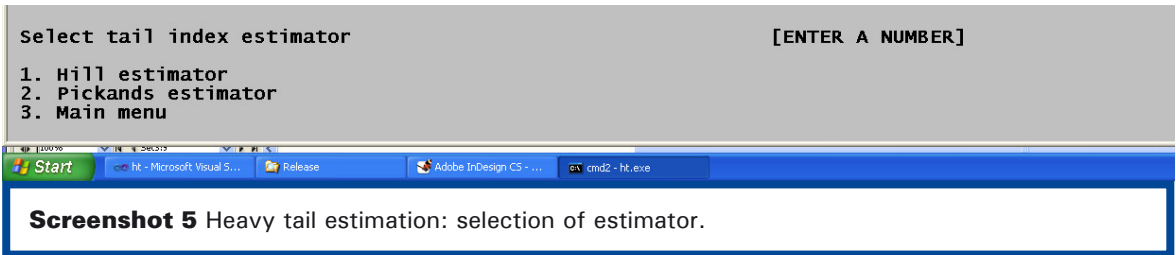
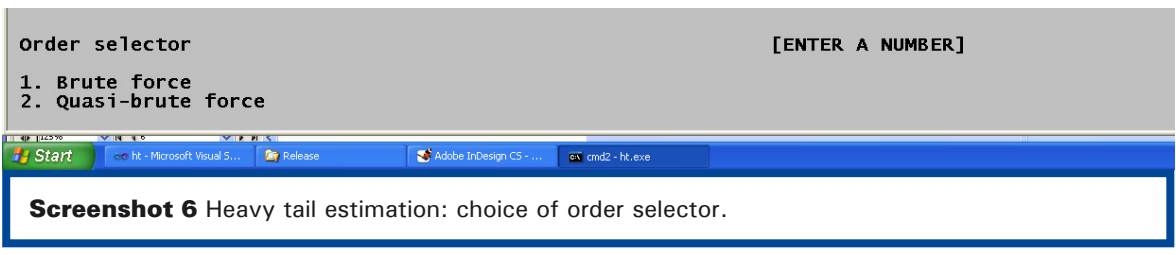
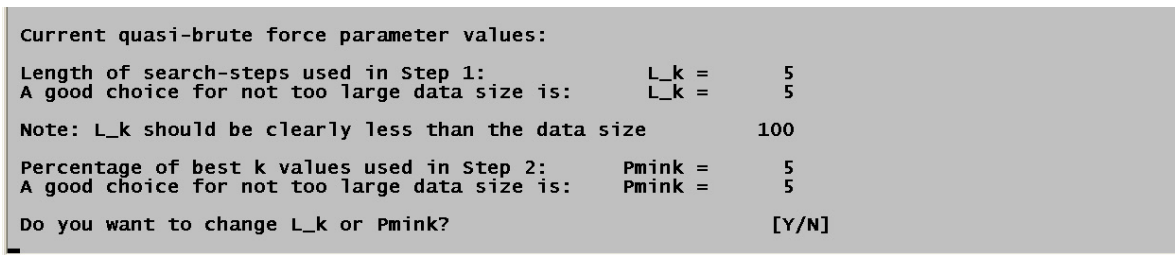
Tail index estimation is only possible if a data file has been selected. The index is estimated on the preprocessed time series. If no preprocessing has been selected, then the index is estimated on the original time series.

If from the main menu (**Screenshot 3**) tail index estimation is taken, then the first selection regards the estimator type (Hill or Pickands) (**Screenshot 5**).

Then next choice is on order selection: brute force or quasi-brute force (**Screenshot 6**).

Brute force means that all possible order values (e.g., all positive values minus one in case of the Hill estimator) are tried. The best order value (in terms of an RMSE measure calculated by ht) is then taken for calculating the tail index parameter.

The RMSE measure itself is calculated over a number N_inner of loops; this parameter can only be set within ht.f90; the preselected value of $N\_inner = 1000$ has been found in Monte Carlo experiments (not shown) to work well.

Quasi-brute force means a reduction of the search (at the risk of missing the optimum). This reduction is accomplished in two steps (**Screenshot 7**), first a coarse search, and then a fine search. See Section 4 and ht.f90 for details.

Screenshot 5 Heavy tail estimation: selection of estimator.

Screenshot 6 Heavy tail estimation: choice of order selector.

Screenshot 7 Heavy tail estimation: choice of parameters for quasi-brute force order selector.

The last selection before the estimation is the number of internal (i.e., within ht) Monte Carlo simulations to be performed (**Screenshot 8**) for determining the estimation uncertainty.

The uncertainty is determined as empirical root mean squared error (RMSE). The data generating process for this is an AR(1) process on an possibly unevenly spaced time grid with innovations from a stable distribution; the prescribed parameters (persistence time τ for the AR(1) process, heavy tail index α for the stable distribution) are overtaken from the estimation.

The number of simulations should be at least *nsim* = 10 to obtain uncertainty measures that help as a rough guide. Own Monte Carlo experiments (Section 4) indicate that the selection of *nsim* = 100 or higher gives more acceptable results. Note that the choice of *nsim* dictates the computing time. It is possible (e.g., in preliminary analyses) to circumvent RMSE calculation by selecting a negative *nsim* value.

Note that RMSE calculation can also be selected (after typing in *nsim* ↵) for the persistence time estimate.

During the calculations (estimation and Monte Carlo simulations), counters inform on the screen about the progress.

For computational details on the estimation, the implementation of a random number generator for parallel computing, the choice to calculate via $\gamma = 1/\alpha$ (for Pickands estimator) and other aspects of the uncertainty determination, see ht.f90.

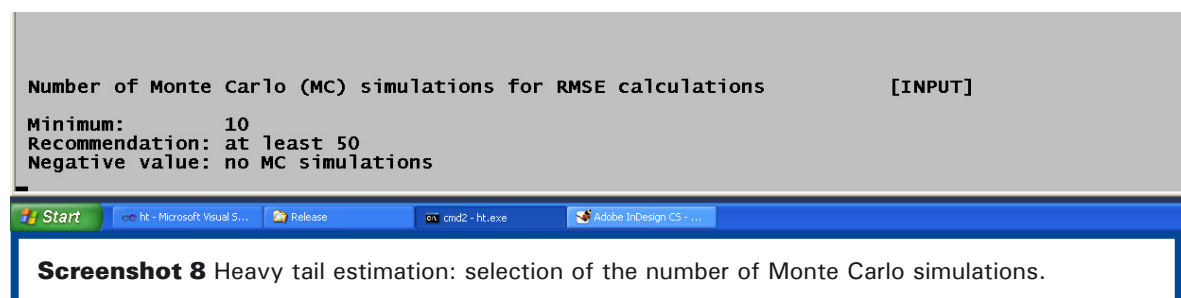
For details on stable distributions, see NOLAN (1997, 2003), ht.f90 and references cited therein.

For details on AR(1) processes and persistence time estimation, see MUDELSEE (2014: Chapter 2).

For details on heavy tail index estimation, see RESNICK (2007), ht.f90 and references cited therein.

For details on RMSE and other statistical measures, see MUDELSEE (2014: Chapter 3).

For Monte Carlo simulations and experiments, see Section 4.

Screenshot 8 Heavy tail estimation: selection of the number of Monte Carlo simulations.

After the estimation and, possibly, the Monte Carlo simulations have been carried out, the results appear on the screen (**Screenshot 9**). These contain information about the data (file name, original data size and after preprocessing), the estimation (estimator and order selector), the RMSE determination (*nsim*) and, finally, the estimation result.

A value of the estimate of the persistence time (τ) equal to –999.0 means that there are problems with the estimation of the heavy tail index (α); see below.

A value of the RMSE of the persistence time estimate equal to –999.0 means either that this option has not been selected or α-estimation problems (see below).

A value of the estimate of the heavy tail index (α) equal to –999.0 means that no estimation is possible since for all tried order values (k), the index $\alpha(k)$ is outside of the interval [0; 2] (exclusive of the endpoints). ht then states: "Suggested options for re-analysis: try other preprocessing types or an increased data size." The problem may arise for the Hill estimator, which requires at least *minNpositive* = 2 (notation in ht.f90) positive values (after mean subtraction). See ht.f90 for further details.

A value of the RMSE of the heavy tail index estimate equal to –999.0 means either that no Monte Carlo simulations have been done or α-estimation problems (see above).

In case of Pickands estimator and selected choice to calculate via $\gamma = 1/\alpha$, the results are given for γ and not α; see ht.f90 for details.

A value of the determined optimal order of –999 means α-estimation problems (see above).

These results are not only printed on the screen but also written into a file (the default name is htresult.dat). Note that an existing file is overwritten. The choice of individual result file names cannot be done in the modus 'per-hand'; it has to be made in the modus 'estimate' in the configuration file.

The output file contains additionally:
(1) original time series,
(2) preprocessed time series,
(3) sorted x-values of preprocessed time series,
(4) k,
(5) $\alpha(k)$,
(6) RMSE measure in dependence on k,
(7) $\alpha(k)$ – RMSE measure in dependence on k,
(8) $\alpha(k)$ + RMSE measure in dependence on k.

The RMSE measure, which has been calculated for order selection, should not be confused with the RMSE value of the α estimate.

```
Optimal Heavy Tail Estimation: Results
======================================

Heavy tail index estimation (Hill or Pickands estimator)
on time series data by means of a brute force order selector
(RMSE measure), including preprocessing routines.

Version 1.0 (April 2017)

Copyright (C) Climate Risk Analysis - Manfred Mudelsee
http://www.climate-risk-analysis.com

Please see source code (ht.f90) for disclaimer and permissions.

Input data file name:                                    Example.dat
Data size (original):                                    100
Data size (after preprocessing):                         100
Tail index (alpha) estimator:                            Hill
Order selector (RMSE measure):                           Brute force
Length of search-steps used in Step 1:            L_k = Not applicable (only for quasi-brute force)
Percentage of best k values used in Step 2:     Pmink = Not applicable (only for quasi-brute force)
Number of loops (order selection):                       1000
Number of Monte Carlo simulations for RMSE calculations: 100

Persistence time estimate:                          tau  = 0.268991
Error:                                        RMSE(tau) = 0.186969
Tail index estimate:                               alpha  = 1.256362
Error:                                       RMSE(alpha) = 0.327964
Optimal order:                                    k_opt = 93

Full results written into file:                          htresult.dat

PRESS ENTER TO CONTINUE
```

Screenshot 9 Results screen.

Generation of artificial time series

Generation of artificial time series is only possible if a data file has been selected. Neither preprocessing nor tail index estimation has to be done for the generation of artificial data. **Screenshot 10** shows the steps.

It is possible to overtake the original time grid, $\{t(i)\}_{i=1,\ldots,n'}$ which may show an uneven spacing. Alternatively, a grid with an even spacing (of unity) is used.

Then the data size (within bounds) is chosen. The persistence time ($\tau > 0$) for an AR(1) process with uneven spacing is selected next.

As regards the distributional shape, it is possible to select among a stable distribution ($2 > \alpha > 0$), a Cauchy distribution ($\alpha = 1$) and a Gaussian ($\alpha = 2$). **Screenshot 10** shows also the selection of α for a stable distribution.

For the generation of stable distributions, ht employs modified routines from NOLAN (1997), see ht.f90. We overtake also his recommendation to select $\alpha > 0.4$.

These artificial time series data are written into a file (the default name is hta.dat). You are alerted that an existing file is being overwritten.

Exit

You may chose to exit at the main menu (possible only if a data file has been selected).

Screenshot 10 Generation of artificial time series.

Section 2: Modus 'generate'

The modus 'generate' is set in the configuration file (**Screenshot 11**). It serves automatic generation of artificial time series data. This is useful in external Monte Carlo experiments. The modus is invoked at the command line, for example:

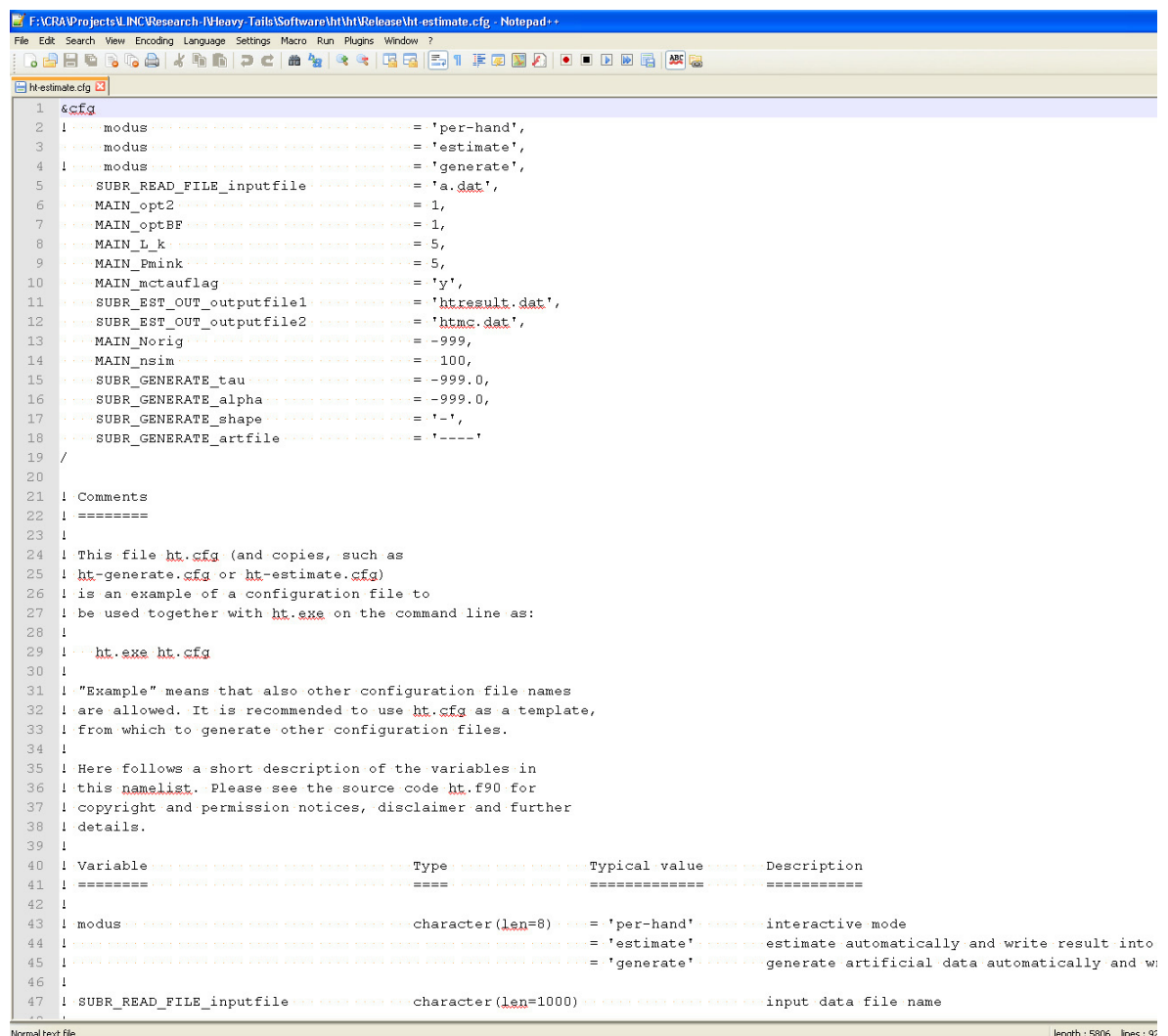
ht.exe ht-generate.cfg ↵

The steps performed are the same as in modus 'per-hand' (Section 1, Generation of artificial time series). Instead of user-interactive selection, the various parameters (n, τ, α and distributional shape) are set in the configuration file (**Screenshot 11**).

The main difference to modus 'per-hand' is that in modus 'generate', you can set the output file name for the generated artificial series. Note that an existing file is automatically overwritten.

Another difference to modus 'per-hand' is that in modus 'generate', you cannot generate data on a predefined time grid. (However, ht.f90 may be adapted to achieve this.)

Screenshot 11 Configuration file ht-generate.cfg for modus 'generate' (excerpt).

Section 3: Modus 'estimate'

The modus 'estimate' is set in the configuration file (**Screenshot 12**). It serves automatic reading of time series data (artificial or not) and estimation of the heavy tail index. This is useful in external Monte Carlo experiments (artificial data) or the automatic processing of a large number of existing time series data files. The modus is invoked at the command line, for example:

ht.exe ht-estimate.cfg ↵

The steps performed are the same as in modus 'per-hand' (Section 1, Tail index estimation). In-stead of user-interactive selection, the various setting and parameters (input data file name, estimator, order selector and choice whether to calculate the RMSE for the estimate of the persistence time) are set in the configuration file (**Screenshot 12**).

In modus 'estimate', you can set the output file name for the estimation result (**Screenshot 12**, *SUBR_EST_OUT_outputfile1*) for each input data file. This is useful for the automatic processing of a large number of existing time series data files. Note that an existing output file is automatically overwritten.

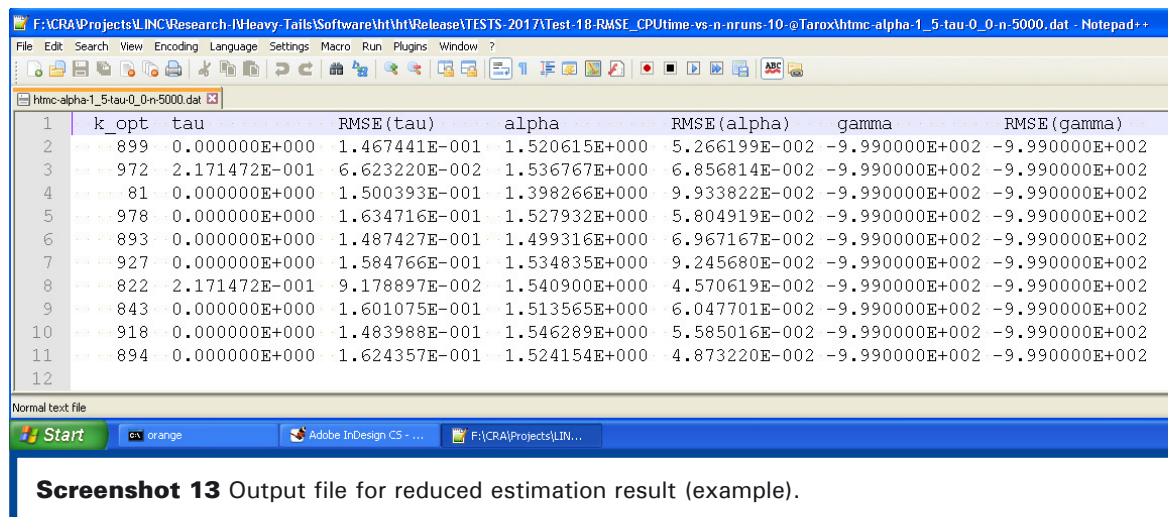
Screenshot 12 Configuration file ht-estimate.cfg for modus 'estimate' (excerpt).

In modus 'estimate', you can also set the output file name for the reduced estimation result (**Screenshot 12**, *SUBR_EST_OUT_outputfile2*).

The reduced content consists in one line of estimation result:
(1) optimal k,
(2) estimated τ,
(3) RMSE for estimated τ,
(4) estimated α,
(5) RMSE for estimated α,
(6) estimated γ,
(7) RMSE for estimated γ.
(The meaning of values equal to –999.0 is as in modus 'per-hand'.) See **Screenshot 13** for an example.

Note that an existing output file for the reduced content is not automatically overwritten; the result line is appended. This is useful in external Monte Carlo experiments (artificial data). If no output file for the reduced content exists, then a new one is started, with one header line for the estimation results (1) to (7).

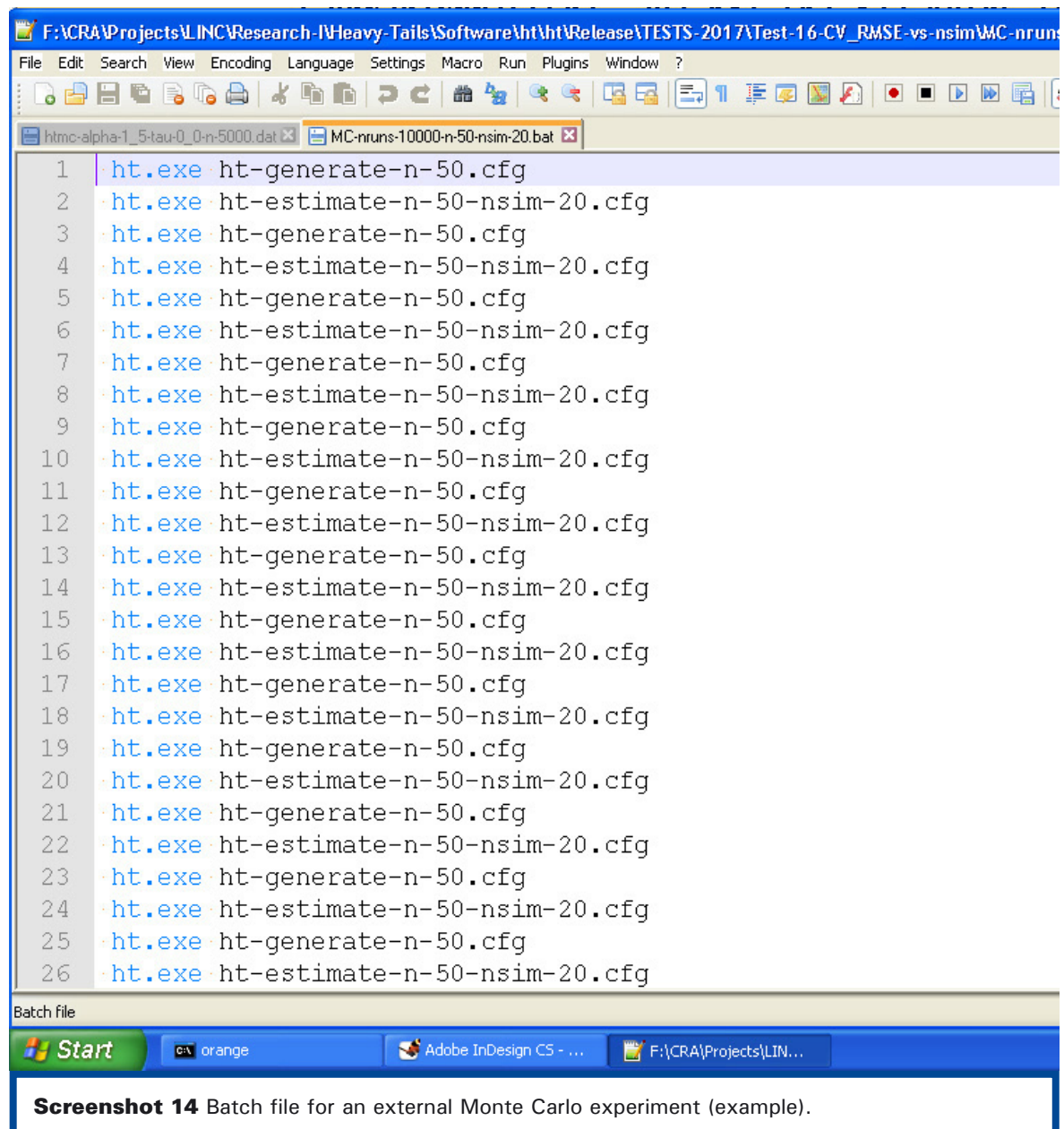
For Monte Carlo experiments, the output file of the automatically generated time series (**Screenshot 11**, ht-generate.cfg, parameter *SUBR_GENERATE_artfile*) has to be given the same name (a.dat in this case) as of the automatically read and further processed time series data (**Screenshot 12**, ht-estimate.cfg, parameter *SUBR_READ_FILE_inputfile*).

Screenshot 13 Output file for reduced estimation result (example).

Section 4: Monte Carlo Experiments

A Monte Carlo experiment is a computer experiment where artificial data are produced and analysed. It requires a random number generator. The properties of the data-generating process are prescribed. Monte Carlo methods help to study statistical distributions and the performance of statistical methods (Fishman 1996, Mudelsee 2014).

We refer to *simulations* as an internal Monte Carlo method (i.e., inside of ht); these are done

for RMSE determination (Section 1). On the contrary, we refer to *experiments* as an external Monte Carlo method (i.e., outside of ht); these serve here to assess the presented statistical method of tail index estimation and to help to tailor parameter settings for this method.

The Monte Carlo experiments use a batch file to run ht several times by invoking the configuration files (modes 'generate' and 'estimate'). See **Screenshot 14** for an example.

Screenshot 14 Batch file for an external Monte Carlo experiment (example).

First Monte Carlo experiment: Required number of simulations

The first Monte Carlo experiment studies the number of simulations required for achieving a certain accuracy in case of the Hill estimator and the brute force order selector (Figure 1).

The prescribed properties of the data-generating AR(1) process with stable-distributed innovations are:
n = 50 or 100,
τ = 0.0,
α = 1.5.

Shown against *nsim* is the coefficient of variation (CV) of the RMSE of the estimated α. The RMSE is calculated from *nsim* internal Monte Carlo simulations (Section 1). The CV is given by the standard deviation of RMSE, which is calculated over a number of external runs, divided by the mean calculated over the runs. The CV is a well-known, handy measure of the relative uncertainty (Mudelsee 2014: Chapter 3).

The number of runs (one run consists of generating a series and estimating the tail index with RMSE) is 10000.

Figure 1 demonstrates that for a number of *nsim* ≈ 100, saturation behaviour of the CV sets in. This value seems not to depend on α or n; such an independence on n is known from "classical experiments" for the RMSE of standard deviation estimation of Gaussian white noise (Mudelsee 2014: Table 3.2).

To summarize, taking *nsim* = 100 seems sufficient to achieve a decent level of accuracy for the RMSE determination. This value also agrees roughly with the Monte Carlo findings on the minimum number of bootstrap simulations required for obtaining reliable results for the bootstrap standard error (Efron & Tibshirani 1993).

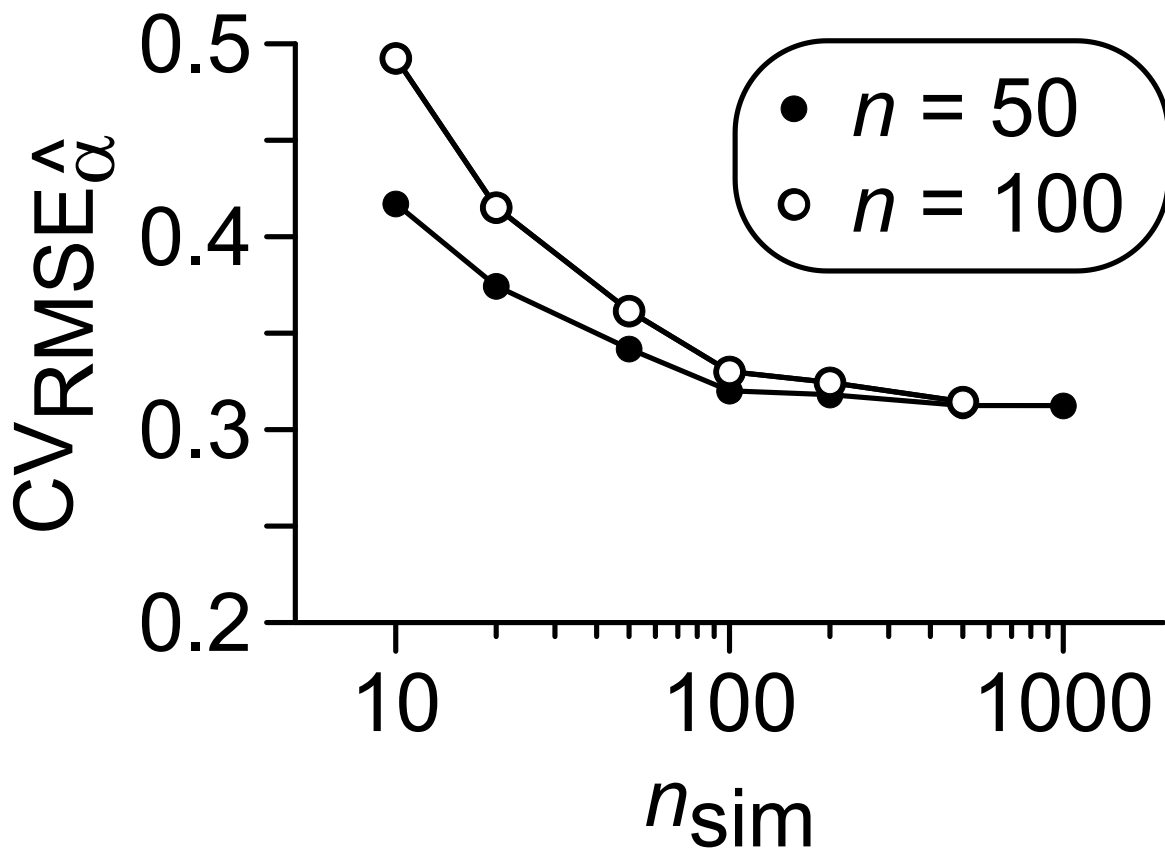



Figure 1 Monte Carlo experiment: required number of simulations. ($\hat{\alpha}$ is the estimate of α.)

Second Monte Carlo experiment: Estimation accuracy

The second Monte Carlo experiment studies the estimation accuracy in dependence on the data size in case of the Hill estimator (Figures 2 and 3).

The prescribed properties of the data-generating AR(1) process with stable-distributed innovations are:
n = 100, 200, 500, 1000, 2000 or 5000;
τ = 0.0 or 1.0;
time spacing: equidistant 1.0;
α = 1.2, 1.5 or 1.8.

The heavy tail index estimation is done with:
Hill estimator,
brute force order selector,
$nsim$ = 100.

Shown against n is the RMSE of the estimated α, which is calculated over a number of 10 external runs.

Both Figures 2 and 3 show how strongly the estimation accuracy increases (i.e., the RMSE decreases) with n.

In case of persistence in the data-generating process (here an AR(1) process with τ = 1.0), the resulting RMSE values are systematically larger than when no persistence is present (Figure 2). This is as expected from the reduced "effective data size" (MUDELSEE 2014: Chapter 2).

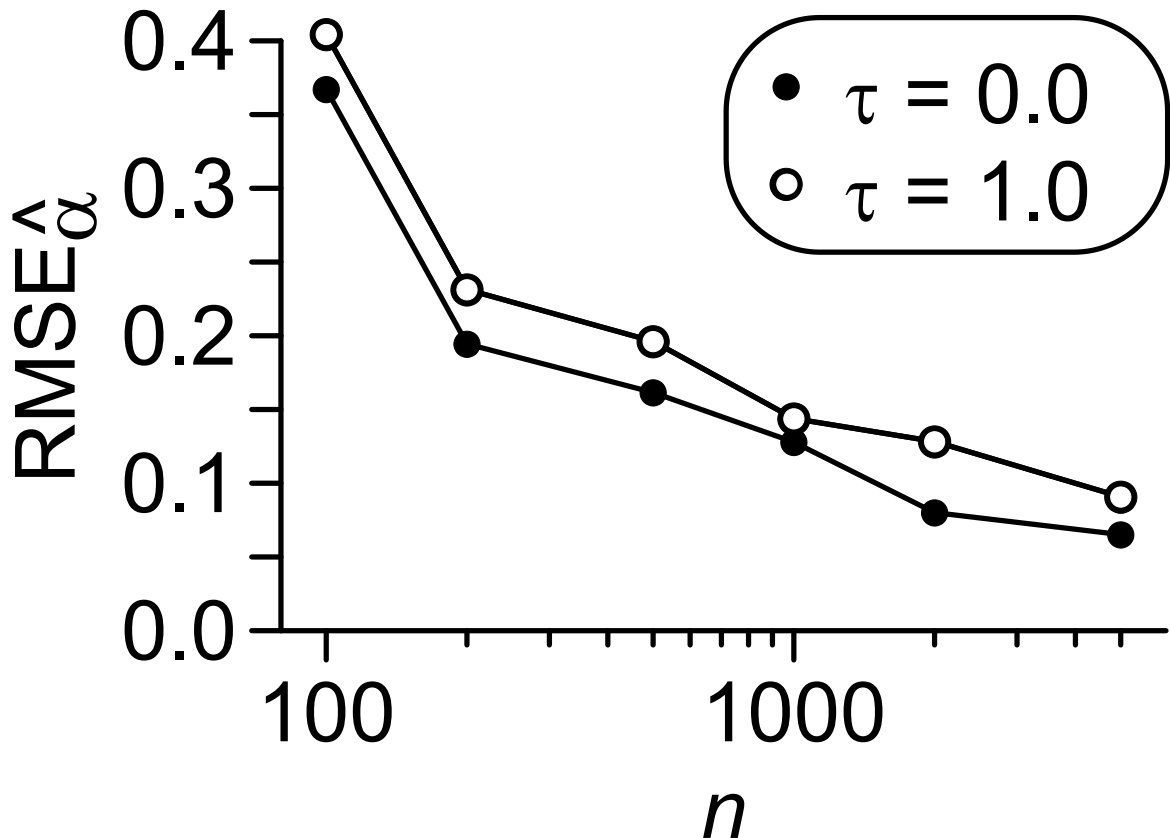



Figure 2 Monte Carlo experiment: estimation accuracy, dependence on n and τ (fixed: α = 1.5).

Also the influence of the prescribed α-value on the resulting accuracy is clear: smaller α-values lead to a larger resulting RMSE (Figure 3).

To summarize, the data size has to be large enough (i.e., at least a few thousand) to achieve a decent level of accuracy for the heavy tail index estimation, especially if the tail index is small (i.e, clearly less than 2.0).

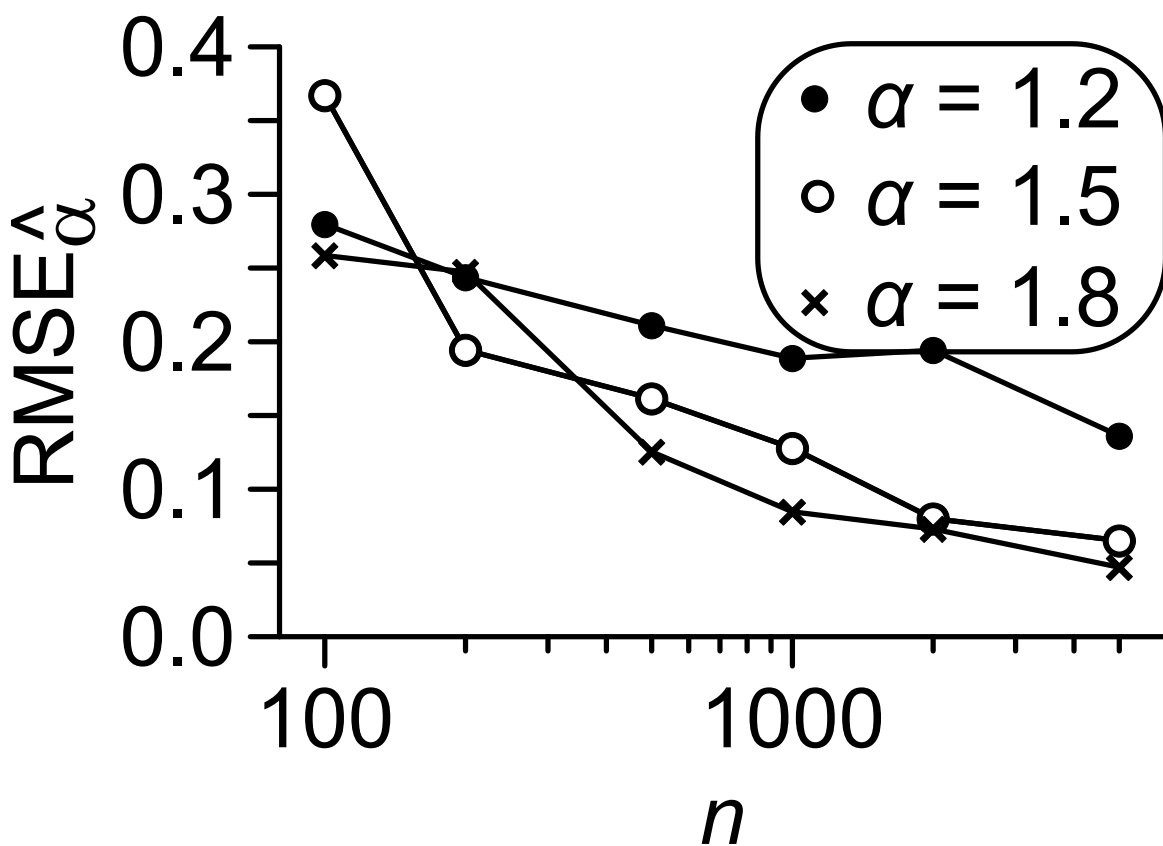



Figure 3 Monte Carlo experiment: estimation accuracy, dependence on n and α (fixed: $\tau = 0.0$).

Third Monte Carlo experiment: Computing time

We recorded also the computing time necessary to carry out the Monte Carlo simulations shown in Figures 2 and 3. The resulting plot is shown in Figure 4.

The used hardware:
Tarox Workstation 745TQ,
2 CPU Intel Xeon E5-2620 (64 GB RAM),
24 threads.

The used software:
operating system: Linux Fedora 15;
compiler: Intel(R) Visual Fortran Composer XE 2013;
compiling options: /nologo /O2 /Qipo /Qopenmp /Qdiag-disable:8290 /module:"Release\Modules" /object:"Release\Obj\ht" /Fd"Release\vc100.pdb" /traceback /check:bounds /libs:static /threads /c.

Shown against n is the computing time for the heavy tail estimation, which is calculated as the average over a number of 10 external runs. The results indicate a strong increase with n in the form of a power-law (Figure 4).

To summarize, on computing systems similar to that one used, data sizes in the order of 1000 and higher lead to sensible computing times. This may lead to consider quasi-brute force instead of brute force order selectors.

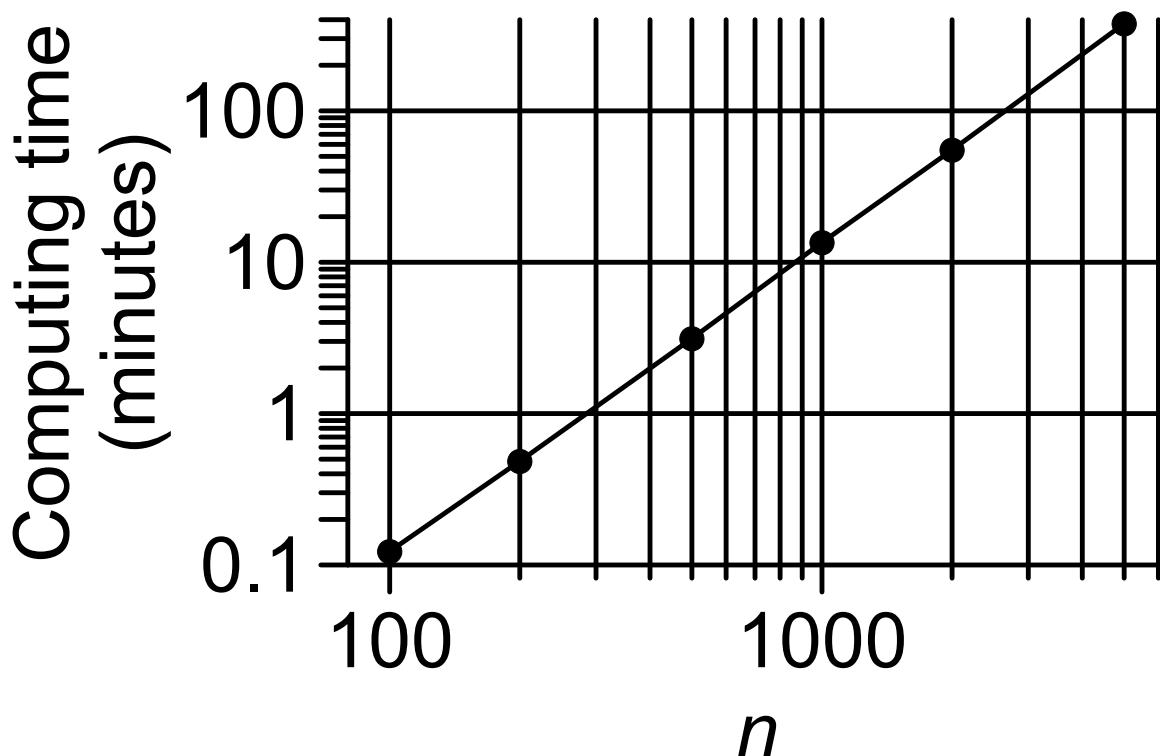



Figure 4 Monte Carlo experiment: computing time, dependence on n (fixed: $\tau = 0.0$, $\alpha = 1.5$).

Fourth Monte Carlo experiment: Quasi-brute force vs. brute force

For larger data sizes and brute force (BF) order selection, computing time may be a sensible factor (Figure 4). Quasi-brute force (QBF) selection may considerably reduce computing time. The fourth Monte Carlo experiment studies how well QBF results agree with BF results (Figures 5 and 6).

The prescribed properties of the data-generating AR(1) process with stable-distributed innovations are:
n = 200, 300, 400, 500, 1000, 2000, 5000 or 10000;
τ = 0.0;
α = 1.5.

The heavy tail index estimation is done with:
Hill estimator,
BF or QBF order selector.

The QBF selector does not search through all order values. Instead, it works in two steps. At the first step, it calculates through the order values at an increment of L_k. At the second

step, the *Pmink* percent of those order values calculated at the first step that have minimal RMSE measure, are subjected to a brute force search over the following increment of L_k.

As an example, consider an BF search through 1, 2, …, 300. Let L_k = 5 and *Pmink* = 5. Then, the first step of QBF calculates through 1, 6, 11, 16, …, 296 (60 values). Let the 5% best (in terms of RMSE measure) values be: 16, 111, 266,. The second step of QBF then studies 16, 17, 18, 19, 20, 111, 112, 113, 114, 115, 266, 267, 268, 269, 270. The general formula is: If the computing time for BF is ~ n, then the computing time for QBF is ~ n $(1/L_k + Pmink/100)$. These are approximations. See also ht.f90.

Shown against n is the percentage of agreement between BF and QBF (L_k = 5, *Pmink* = 5) for the detected optimal order (Figure 5), which is calculated over a number of 100 external runs.

To summarize, for the studied design (Figure 5), $n \geq 1000$ yields almost perfect agreement.

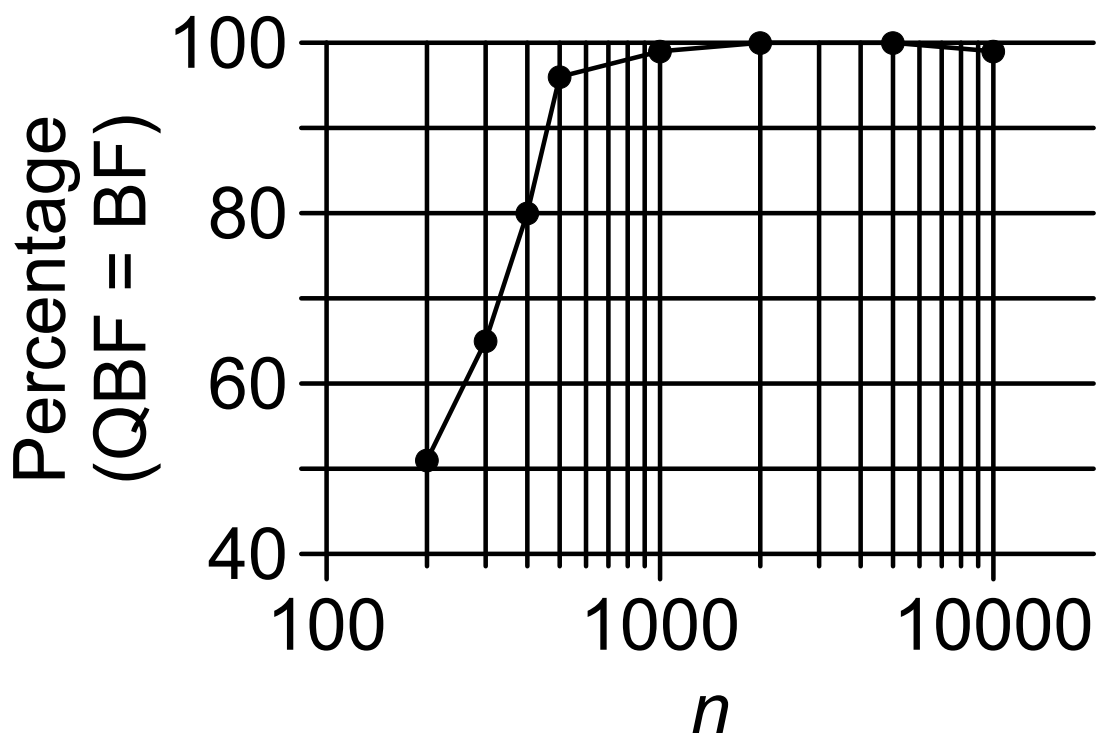



Figure 5 Monte Carlo experiment: QBF vs. BF, dependence on n (fixed: τ = 0.0, α = 1.5).

Another feature explored in the Monte Carlo experiment of QBF vs. BF is the dimension of L_k (Figure 6).

The prescribed properties of the data-generating AR(1) process with stable-distributed innovations are:
$n = 5000$,
$\tau = 0.0$,
$\alpha = 1.5$.

The heavy tail index estimation is done with:
Hill estimator,
BF or QBF order selector.

The QBF parameters are:
L_k = 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 120, 130 or 140;
$Pmink$ = 5.

Shown against n is the percentage of agreement between BF and QBF for the detected optimal order (Figure 6), which is calculated over a number of 100 external runs.

To summarize, for the studied design (Figure 6), $L_k \leq 40$ yields almost perfect agreement.

Further dimensions in the space of estimation parameters (e.g., *Pmink*) can be studied with Monte Carlo experiments.

The practical conclusions we draw from the four Monte Carlo experiments shown so far in Section 4 are the following:

(1) A number of simulations of *nsim* = 100 is sufficient to achieve a decent accuracy for the RMSE determination.

(2) A data size of n = 5000 is sufficient to achieve a decent accuracy for the havy tail index estimation (caveat: if α is clearly less than 2.0, close to 1.0, then larger data sizes may be required).

(3) For $n \geq 5000$, the QBF order selector with L_k = 5, 10 or 20 and *Pmink* = 5 may yield accurate estimations at clearly reduced computing times.

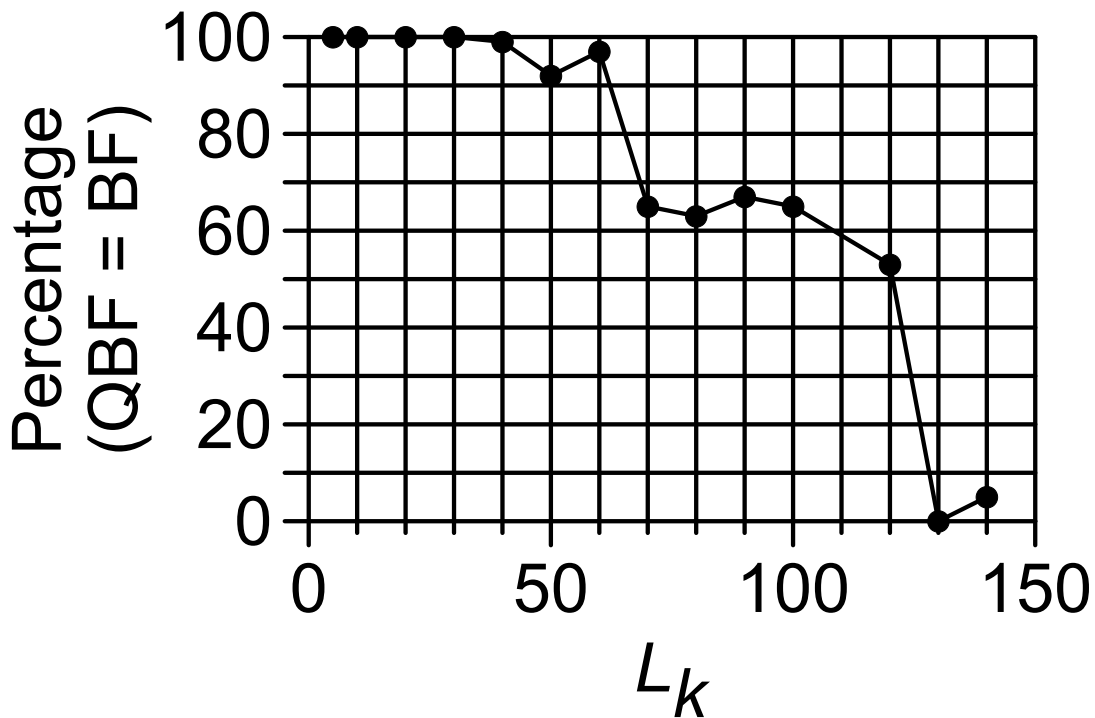



Figure 6 Monte Carlo experiment: QBF vs. BF, dependence on L_k (fixed: n = 5000, τ = 0.0, α = 1.5).

Section 5: Example Analyses

Two example analyses on artificially generated time series serve to illustrate the work with ht. Both times, the heavy tail index estimation is done with Hill estimator, BF order selector and *nsim* = 100.

The first example (Figure 7) prescribes the properties of the data-generating AR(1) process as:
n = 5000,
τ = 1.5,
α = 1.75.

The order selection yields a clear result: The curve of the RMSE measure has a minimum at k_{opt} = 1071; note that this does not occur in a "plateau region" of the curve $\alpha(k)$.

The Hill estimator analyses only the positive extremes (RESNICK 2007). k_{opt} corresponds to a value of 1.04 of the sorted $\{x(i)\}$.

Also shown in Figure 7 are frequency plots: histograms in comparison with scaled heavy-tailed densities, $f(x)$; the scaling ensures that the number of events for $x \geq 1.04$ agree.

```
Optimal Heavy Tail Estimation: Results
======================================

Input data file name:                                    hta.dat
Data size (original):                                    5000
Data size (after preprocessing):                         5000
Tail index (alpha) estimator:                            Hill
Order selector (RMSE measure):                           Brute force
Length of search-steps used in Step 1:          L_k = Not applicable (only for quasi-brute force)
Percentage of best k values used in Step 2:     Pmink = Not applicable (only for quasi-brute force)
Number of loops (order selection):                       1000
Number of Monte Carlo simulations for RMSE calculations: 100

Persistence time estimate:                         tau  = 1.458698
Error:                                        RMSE(tau) = 0.043132
Tail index estimate:                               alpha = 1.757687
Error:                                      RMSE(alpha) = 0.059838
Optimal order:                                     k_opt = 1071
```

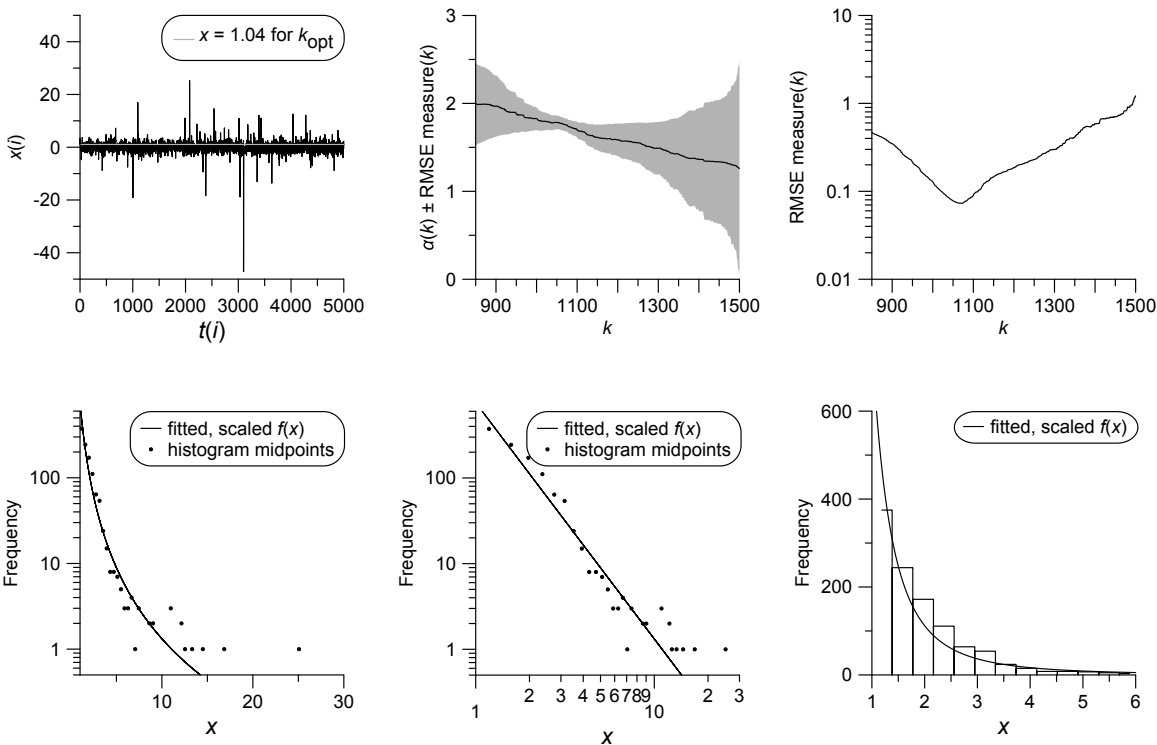



Figure 7 Example Analysis: n = 5000, τ = 1.5, α = 1.75, Hill estimator (BF), *nsim* = 100.

The second example (Figure 8) prescribes the properties of the data-generating AR(1) process as:

$n = 2000$,
$\tau = 0.25$,
$\alpha = 1.25$.

The order selection yields a less clear result (compared with the first example): still, the curve of the RMSE measure has a minimum at $k_{opt} = 372$; also this minimum does not occur in a "plateau region" of the curve $\alpha(k)$.

k_{opt} corresponds to a value of 1.52 of the sorted $\{x(i)\}$. This larger value at smaller n (compared with the first example) is owing to a smaller α.

Both examples (Figures 7 and 8) attest that the presented methodology (software ht) does a good job at selecting the order and estimating the heavy tail index parameter.

Both examples further show a good agreement (i.e., taking into account the RMSE error bars) between prescribed and estimated values for τ and α (Figures 7 and 8).

On one hand, the good performance of ht is owing to the absence of model mis-specification: both order selection and data generation impose a stable distribution. On the other, stable distributions form a large class. More analyses on that can be done using ht.

```
Optimal Heavy Tail Estimation: Results
==============================

Input data file name:                                              hta.dat
Data size (original):                                              2000
Data size (after preprocessing):                                  2000
Tail index (alpha) estimator:                                     Hill
Order selector (RMSE measure):                                    Brute force
Length of search-steps used in Step 1:             L_k = Not applicable (only for quasi-brute force)
Percentage of best k values used in Step 2:      Pmink = Not applicable (only for quasi-brute force)
Number of loops (order selection):                                1000
Number of Monte Carlo simulations for RMSE calculations:          100

Persistence time estimate:                             tau  = 0.252084
Error:                                            RMSE(tau) = 0.057732
Tail index estimate:                                  alpha  = 1.386193
Error:                                          RMSE(alpha) = 0.106587
Optimal order:                                        k_opt = 372
```

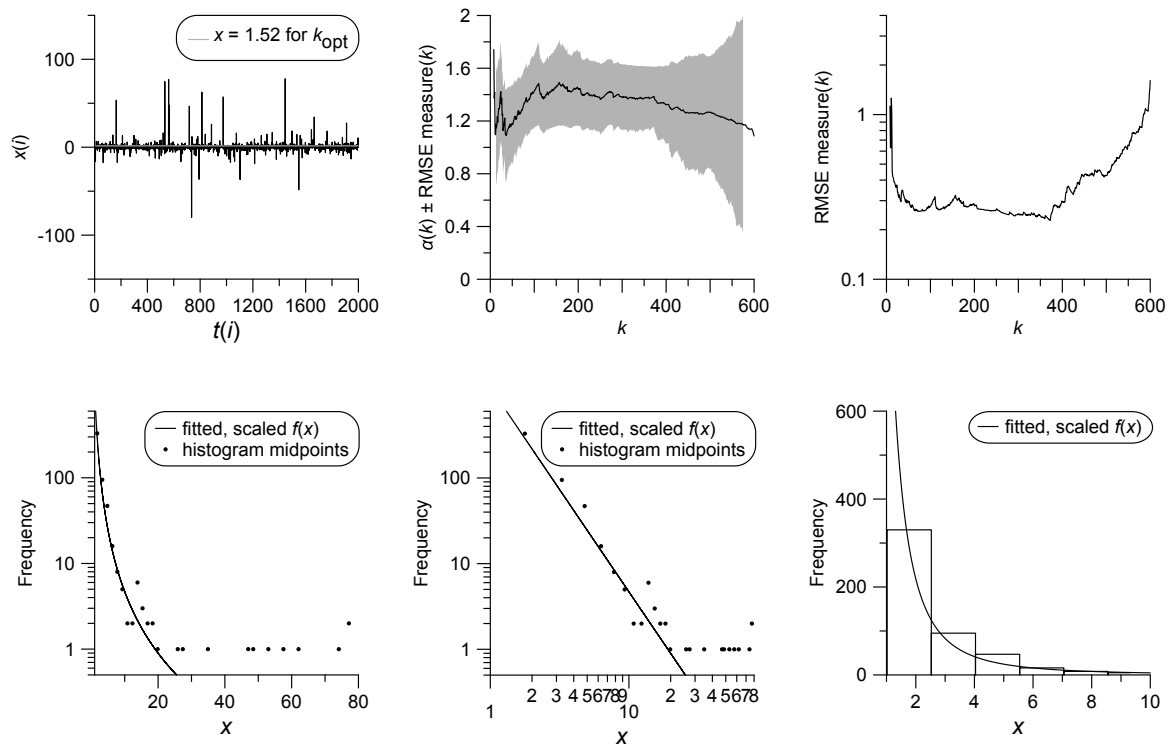



Figure 8 Example Analysis: $n = 2000$, $\tau = 0.25$, $\alpha = 1.25$, Hill estimator (BF), *nsim* = 100.

References

Efron B, Tibshirani RJ (1993) *An Introduction to the Bootstrap.* Chapman and Hall, New York, 436 pp.

Fishman GS (1996) *Monte Carlo: Concepts, Algorithms, and Applications.* Springer, New York, 698 pp.

Mudelsee M (2014) *Climate Time Series Analysis: Classical Statistical and Bootstrap Methods.* Second edition. Springer, Cham, 454 pp.

Nolan JP (1997) Numerical calculation of stable densities and distribution functions. *Communications in Statistics—Stochastic Models* 13:759–774.

Nolan JP (2003) Modeling financial data with stable distributions. In: Rachev ST (Ed.) *Handbook of Heavy Tailed Distributions in Finance.* Elsevier, Amsterdam, 106–130.

Resnick SI (2007) *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling.* Springer, New York, 404 pp.

Internet Links

Climate Risk Analysis website
http://www.climate-risk-analysis.com

ht: manual and software
http://www.climate-risk-analysis.com/soft/ht

Microsoft website
http://www.microsoft.com

MUDELSEE (2014): sample PDF, links to data
and further software
http://www.manfredmudelsee.com/book

ht