Manual for generating synthetic series of precipitation and temperature for use in spring flood scenarios

Norges vassdrags- og energidirektorat

2004

Report no 2, 2004

Manual for simulation of precipitation and temperature series for use in spring flood scenarios

Utgitt av:	Norges vassdrags- og energidirektorat
Redaktør:	
Forfatter:	Jens Erik Rindal og Thomas Skaugen
Trykk:	NVEs hustrykkeri
Opplag:	30 + 16
Forsidefoto:	
ISSN:	1502-3540
ISBN	82-410-0502-4
Emneord:	Precipitation, Temperature, Simulation, Scenarios

Norges vassdrags- og energidirektorat
Middelthuns gate 29
Postboks 5091 Majorstua
0301 OSLO

Telefon: 22 95 95 95
Telefaks: 22 95 90 00
Internett: www.nve.no

Mars 2004

Table of contents

Preface

This report is part of the documentation of the project Flomrelatert FoU, financed by the Norwegian Water Resources and Energy Directorate. This report describes the procedure and programs needed to generate synthetic time series of precipitation temperature and discharge. The report is intended to be a manual for generating the time series describing the necessary procedures step by step and thus in line with the department's quality assurance project.

Oslo, March, 2004

Kjell Repp

head of Department

Sverre Husebye

head of section

Summary

This report gives a detailed description of procedures and programs necessary for generating long (1000 years) synthetic timer series of temperature, precipitation and discharge. The precipitation generator is the British Bartlett-Lewis Rectangular Pulse Model, whereas the temperature generator is originally devised by Christian Onof at the Imperial college in London, UK and implemented for use in the NVE by Jens Erik Rindal at NVE. Discharge series are simulated by the Nordic rainfall-runoff model HBV. The time series are originally intended to be used for scenarios, but the usefulness of long time series in, for example, design work is easily seen.

1 Introduction

The flood forecasting service provides each spring flood-scenarios for the melting season. The customary way to compute the scenarios is to use the most updated state of the HBV model, updated in the way of snow coverage and snow water equivalent, for a certain catchment and then run the model forward as many times, *N* as you have years of historical data of precipitation and temperature. This provides us with *N* outcomes, scenarios, of the future melting season. From the *N* outcomes, we can assess the probability of exceedance of certain critical discharge values like the mean annual- or ten year flood.

The problem with this method, and the motivation for this project, is that the length of the historical series of precipitation and temperature at hand is, as a rule, of insufficient length to make reliable interference on probabilities of exceedance. To estimate such probabilities one should ideally have several hundred years of data, but the length of the data periods at hand is from one to a few decades. To remedy this situation it was decided to use models which can generate time series of precipitation and temperature. In this way time series of sufficient length could be obtained to make reliable assessments of the probabilities of exceedance of discharge values.

The model chosen to simulate precipitation time series is the Bartlett-Lewis Rectangular Pulse Model, or more specifically the Randomised Bartlett-Lewis Rectangular Pulse Model (RBLRPM), which is the current tool in spatial-temporal rainfall modelling developments (Onof, 2000). Details of the RBLRPM can be found in Onof (2000) and references therein.

For the simulation of temperature time series, a standard approach of autoregressive modelling was chosen (Rindal and Skaugen, 2004; Onof, 2003, Onof, 2000). From the different data pre-processing programs presented below, a user of the simulation programs will be guided towards the proper strategy for generating scenarios and validate the results.

The current limit of the length of simulated time series is thousand years, which is considered sufficient to reasonably assess the probabilities of exceedance for critical discharge values. One should bear in mind that this approach is not just appropriate for melting season scenarios, but could also be useful of the estimation of other design values such as low flows, design floods etc.

All the S-plus scripts referred to can be found at ***//atna/home/flom/PTsimulering/SPlus_script/.***Note that SPlus has to be installed on the pc used.

2 Overview of the simulation process and scenario runs

The simulation of precipitation and temperature use models presented in Onof (2000), Onof (2003) and Rindal and Skaugen (2004). Observed data are pre-processed before they are used in the appropriate optimalisation program to estimate model parameters. Excel, Splus and UNIX are all used in this purpose. When the parameters have been estimated, the FORTRAN program **PTsimulation** is used to simulate precipitation and temperature data. The N years of simulated daily temperature and precipitation data are then combined into an input file for a **HBV-model** updated with the latest state of the catchment of interest. Along with a catchment-specific parameter file, this model will generate N years of estimates of daily discharge (scenarios) which is used to calculate probabilities of exceedance of the mean, 5 years, 10 years, 20 years, 50 years and 100 years return flood.

In the following a detailed description of the procedure for simulating temperature and precipitation data, validating the simulated data and running the scenarios are presented. The program **PTsimulation** is menu-driven. The notation 'menu-3-2' refers to entering '3' and then '2' from the main menu.

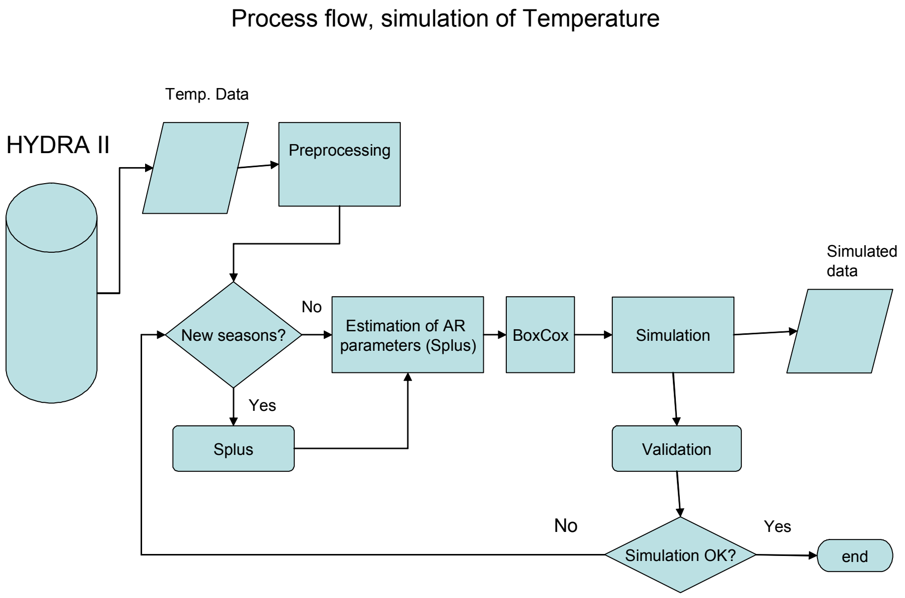

Figure 1 Overview of the temperature simulation process.

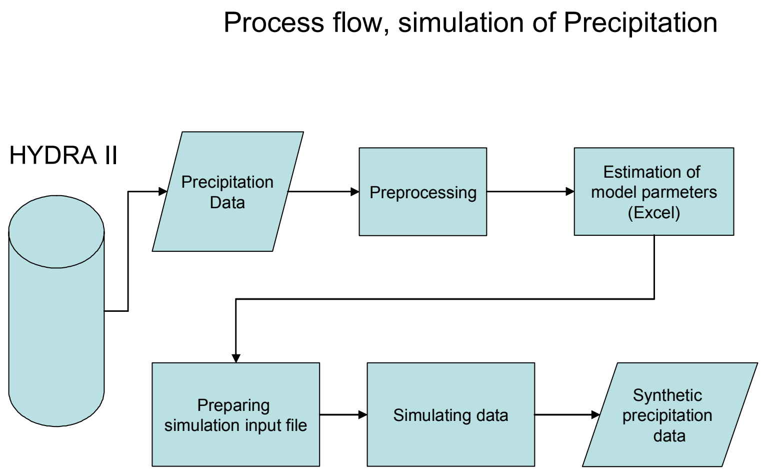
Process flow, simulation of Precipitation
HYDRA II
Precipitation Data
Preprocessing
Estimation of model parmeters (Excel)
Preparing simulation input file
Simulating data
Synthetic precipitation data

Figure 2 Overview of precipitation simulation process.

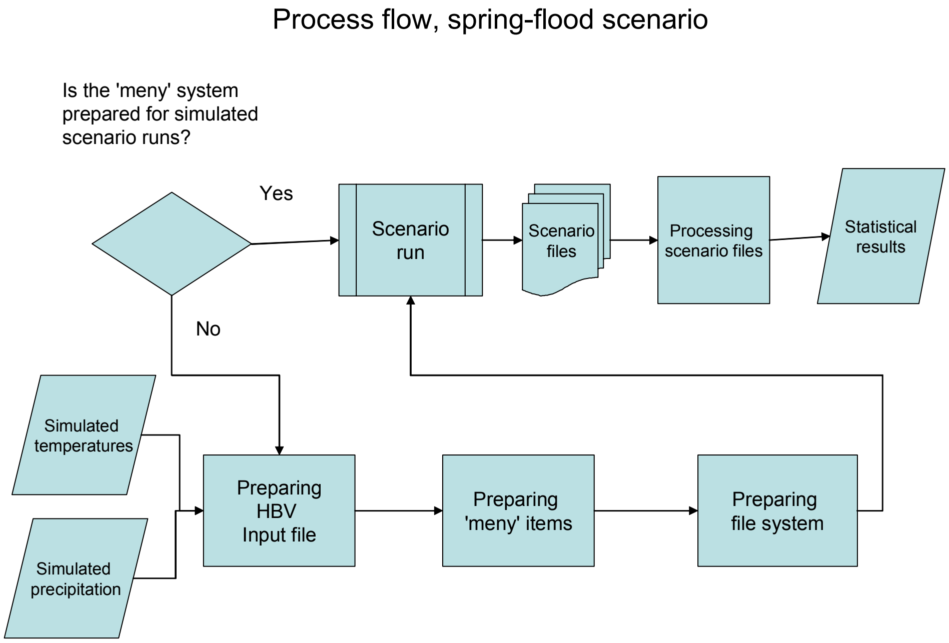

Figure 3 Overview of the scenario process.

3 Simulation of Precipitation

3.1 Source of precipitation data

From the file *hbvfelt.txt*, we find the precipitation stations which are associated with the different *hbv* catchments. Observed precipitation data can be extracted from the database using '**Dagut**', and must be saved in the '*Tid/Verdi*' format. The precipitation stations should have at least 10 complete years of records. It is necessary that the data start on 1 January, and that only complete years are present.

An example of the Tid/Verdi format can be found in appendix B figure 1.

3.2 Preparatory data analysis

To estimate the precipitation model's parameters we will use a Visual Basic program in Excel. Prior to this will must estimate the starting point for this optimization program.

1. If the hbv-catchment in question uses more than one precipitation station, you must merge these together using **PTsimulation** menu-3-5. The weights are found on the *param.dat* file for your catchment. The resulting file is named *merged-P.TV*
2. Interpolate your precipitation data using menu-6. Set the interpolation length (max NA's to interpolate over) to a low value (0-5 days). This will format the data properly.
3. Use menu-1-2 to make statistical calculations on your precipitation file. This produce 2 files: *months.txt* and *precip.txt* which consists of monthly partitioned data and calculated monthly statistics respectively.

3.3 Estimation of the precipitation model parameters

Open the file `//atna/home/flom/PTsimulering/files/norway4.xls`. This worksheet contains a Visual Basic program, so you must activate macros. Presently this program must be run on your local computer[1]. You will estimate each month's parameters separately.

1. Click input data in the lower left corner, and answer 'yes' to use autocovariance instead of autocorrelation. Then enter the values from

[1] If you get 'cannot find library'-error, stop the debugger, choose 'tools' and 'references'. Locate the SOLVER entry (which should say MISSING:SOLVER), highlight it and click 'browse'. In your Office or Excel directory, locate the file solver.xla. This will reference the correct solver file to the macro. Save the Norway4.xls to keep the changes.

precip.txt for the first month. Set the resolution to one day for all values, and the weight to 1 for all but Mean Duration Dry which should have weight 0.1. Exit the form.

2. Click Compute and answer 'no' to the question posed (You want to use 6 parameter model). This brings you to the optimizer screen.

3. The top row in the optimizer screen is the model parameters as estimated by the program from your input variables. Your goal is to get an estimated weighted sum of square errors (WSSE) less than 0.0001, while at the same time retain a low μ_C value (<20), and preferably values on K1 and K2 that is close to zero (+/- 10000). Press the 'solve' button to improve the WSSE.

   The procedure is, however, not straightforward. It is likely that you must manually change some of the parameters in the first line, and re-solve with these. It may prove helpful to remember that $\mu_C = 1 + {}^{\kappa}\!/_{\varphi}$ (to retain a low μ_C value). Also the initial parameter values are important. Before each new month, press the 'initialize parameters' button and add reasonable values for the parameters. An example of this is {0.1, 0.1, 1.2, 2.1, 2.1, 0.6}, but his will change with the dataset.

4. When satisfactory estimates have been found, press the transfer the results button. Then, in the optimisation screen and input data screen, press the clear all button, and repeat the procedure for your next month until all 12 months have been solved.

5. Copy your transferred results into a text document before proceeding (only the 6 first columns are necessary. Copy the values only).

3.4 The input file *input_bartlewtempsim*

The precipitation simulation requires an input file called *input_bartlewtempsim*. The file must be edited in Emacs, and have the following format:

- 12 rows with precipitation parameters; one line for each month. The parameters on each line must be entered in the following sequence:

LAMBDA	Storm arrival
DELTA	Scale param. of the gamma cell intensity distr. ($1/\mu_x$)
ALPHA	Shape parameter of the gamma distribution for ETA
NU	Scale parameter of the gamma distribution for ETA
KAPPA	Scaling parameter for BETA
PHI	Scaling parameter for GAMMA
P	Shape parameter of the gamma cell intensity distribution. P is currently set constant at 1.0.

  Note that if an exponential distribution is used for the cell intensity, then P=1. and DELTA = $1/\mu_x$. (μ_x is obtained from the optimalisation program, norway4.xls)

Each line must conform to the FORTRAN format **7(F12.8,1x)**

- The option to eliminate data < n*0.01 **(i5)**
(n FOR YES; 0 FOR NO)
- An explanatory title for this simulation **(A40)**
- The option to output data at a different time from the hourly/daily **(F10.4)**
(0.0 FOR NO, THE TIME-SCALE IN HOURS OR MINUS THE TIME-SCALE IN MINUTES OTHERWISE)
- The four multiples of the hourly scale at which statistics will be computed **(*)**
(0.0 FOR NO ANALYSIS. IF ALL ARE 0.0, NO STATS ARE CALCULATED)
- The option of a change of parameter set as soon as the model enters a different month (so that the next storm generated on the basis if the previous month's parameters is discarded) **(i2)**
(1 IF YES)
- The option of doing the same for each day when the temperature changes from below to above the threshold or vice versa. Realistically, this option ought to be chosen only if the previous one has been. **(i2)**
(1 IF YES)

An example of this input file is given in appendix A.

(The precipitation simulation is actually carried out using a FORTRAN program named bartlewtempsim_12_jens.f, and it has many options you can alter. Please refer to the source code and Onof 2000 for this.)

3.4.1 MakeInputBartlew

Rather than entering all data into *input_bartlewtempsim* using Emacs, you can use the program **makeInputBartlew** to ensure correct format.

1. Open your saved parameter file (from 3.3) in Emacs, and remove all tabulators on the file. Make sure there is only one space between the numbers. Then replace all commas with punctuation marks. Finally, make sure all numbers use a total of 11 spaces (including the punctuation) by adding trailing zeros where needed.
2. Run **makeInputBartlew** with the edited parameter file. This produces a correct *input_bartlewtempsim* file.

3.5 Simulation of precipitation

The input file *input_bartlewtempsim* must be located in directory `//atna/home/flom/PTsimulering/files` . Run **PTsimulation**, and choose menu-5. You will be asked for number of years to simulate, with a maximum of 1000. The simulation will take considerable time to complete with 1000 years. When it is finished, 3 files will be written: *precip_sim.dta* is the file with simulated precipitation data, *drang* and *sturm* are both files with statistics of the simulated precipitation data.

4 Simulation of Temperature

4.1 Source of temperature data

From the file *hbvfelt.txt*, we find the temperature stations which are associated with the different *hbv* catchments. Observed temperature data can be extracted from the database using '**Dagut**', and must be saved in the '*Tid/Verdi*' format. The temperature stations should have at least 10 complete years of records. It is necessary that the data start on 1 January, and that only complete years are present.

An example of the Tid/Verdi format can be found in appendix B figure 1.

4.2 Preparatory data analysis

1. If the catchment has more than one meteorological station, you must merge these into one weighted series. This is done using **PTsimulation** menu-3-2. The appropriate weights can be found on the file hbvfelt.txt.

2. The data file may have missing values (NA). These are indicated as -9999.00 on the file. To retain as much data as possible for the simulation, you must interpolate over missing data in an appropriate way. Using the **PTsimulation** program, interpolate the (merged) temperature file using menu-6. You will be asked for maximum NA's to interpolate over. A maximum of 10 successive NA's are allowed. If a year has more than this, the whole year must be set as NA. You should *always* interpolate the files as the program also corrects the format.

3. In order for the simulation to be successful, you must estimate 12 optimal seasons in Splus.

    a. Prepare a Splus file using menu-7-2. This produces 2 files: *dailymean.txt* and *weekmean.txt*. For temperature data, only *dailymean.txt* is important.

    b. Import the file *dailymean.txt* into Splus. Run the script **yearanalysis** on this object. It will plot daily mean temperature, - skew, - variance and maximum/minimum. The vertical blue lines are default seasons, and you must adjust these until the variance and skew are reasonably stationary within the seasons. This is done by trial and error, using command:

    ```
    yearanalysis(<daily mean data>, dato=c(a1,a2,…,a12),
    basin="<catchment name>"])
    ```

    , where a1…ai…a12 is the end day (in day number) of season i. Special care must be taken in the period leading to melting point, as a correct start of the spring flood is critical. The more stationary the seasons are, the better the simulation. Write down your choice of seasons.

The script **yearanalysis** can be found in appendix B.

4. Prepare for estimating the Autoregressive parameters

    a. Run **PTsimulation** menu-1 on your observed temperature file, and answer as follows:

        i. The number of meteorological stations in the catchment.

        ii. Filename of observed data

        iii. Catchment name

        iv. The window size delta. Use 5 or less.

        v. Interpolating limit. A number 10 or less.

        vi. 'Y' to enter the optimal seasons as found above.

    The boxcox parameters are now being automatically estimated. When this is done, you can manually change these by selecting the season number. You **must** change this if the automatic estimation failed for some seasons. This is easily seen by a large skewness, and is a result of unstable seasonal data. You may consider going back to step 3 and try finding better seasons. If the estimation is satisfactory, simply enter '0'.

The program writes 4 files: **comparestats.txt** is a file used to compare observed and simulated statistics, **intsim.txt** is the start of an input file to the simulation program, **prep-bcresid** is the input file to Splus, and **prep-stats** records some statistics concerning the boxcox estimation.

4.3 Estimation of the Autoregressive parameters

1. Import the file **prep-bcresid** into Splus

2. Run the script **ar4v22** on this object using command

```
ar4v22(<objectname>, "<catchment name>", "<filename>")
```

    The script **ar4v22** can be found in Appendix B

3. The file <*filename*> is created in your default Splus directory, and consists of optimal AR parameters for each season. Copy these into the file *intsim.txt* (at the end of the file) using Emacs, and rename the file *intsim-test*.

For each season, 4 plots are generated to help validate the model. The aim is to have stationary variance, good normal approximation and that the estimated order of the AR model is adequate. An example of such plots is provided in Appendix B, figure 1.

4.4 Simulation of temperatures

In **PTsimulation,** use menu-2. Answer the following at the questions that appear:

- Name: Catchment name
- Seed: ‘0’ (zero). On some computers zero cannot be used as random seed, and another integer must be used. Zero will get the seed from the system clock.
- Startyear: The year you want the simulated data to run from, i.e. 1000
- Total number of years in a series, i.e. 1000
- Num. of simulations: The total number of series to simulate. For our purpose only one series should be simulated.

Some comparable statistics are printed to screen, but also appended to the file *comparestats.txt*. Two files are created; *simT-all* consisting only of data, with columns being each simulated series (only useful if you simulate more than one series), and *simT-mean* which is the mean of all series on Tid/Verdi format. This file *simT-mean* is the correct file for our purpose (provided you have simulated only one series).

5 Validation of data and preparing scenario runs

Validation is mainly done by comparing plots of daily means of observed and simulated data. In addition, the file *comparestats.txt* compares some statistics between simulated and observed temperature data, and during the estimation of the AR parameters you were provided with goodness of fit plots for each season (appendix B, figure 1).

5.1 Preparing input file for the HBV-model and generating discharges for validation purposes

You now have one file consisting of 1000 years of simulated precipitation data, and one file consisting of 1000 years of simulated temperature data. These must be merged into an input file for HBV.

1. Using **PTsimulation**, choose menu-3-1. You will typically not have observed discharge data in this case, so answer no (0) to that question. If the catchment has more than 1 meteorological station, enter the correct number when asked for number of stations the file is merged from.

2. Copy the files *climcha.dat, default.dat, vegtype.dat* and *param.dat* found at ***`//atna/home/flom/vf-prognoser/<hbv-catchment>`*** to the directory you have the data. If more than one *param.dat* file exist use *param.dat_synop*.

3. Open *param.dat (param.dat_synop)* and make sure the specified meteorological stations are the ones appropriate to your data. Write down the catchment area.

4. Start the HBV-model by typing /home/flom/bin/hbv_all. Please refer to the file http://inca/hfelles/h/KS-Intranett/8.3.1_VEDLEGG_Retningslinjer%20for%20kalibrering%20av%20HBV-modellen.doc for questions concerning the running of the HBV-model.

The HBV-model will write a file call *PRTFIL.RES* which, among other things, consists of daily estimates of discharge. Save this for use in the validation.

5.2 Preparing the correct Splus input files.

To generate comparable plots of precipitation, temperature and discharge, you need 6 files altogether: simulated precipitation and temperature, observed precipitation and temperature and discharges based on observed and simulated data. To make a file of discharges based on observed data, please follow the steps indicated in 5.1 (now you should use menu-3-2 instead).

All 6 files must be prepared in PTsimulation before used in Splus. Please follow these steps:

1. Use **PTsimulation** menu-4-3 to transform the special format of simulated precipitation data into Tid/Verdi format.
2. The output file from running the hbv-model is named *PRTFIL.RES*. Transform the resulting *PRTFIL.RES* file into Tid/Verdi format using menu-4-2. You will be asked for catchment area. Consult the *param.dat* file for this.
3. All files should now be on Tid/Verdi format. Run menu-7 on each file, remembering to rename the resulting output files to avoid overwriting them. For temperature data and discharge data only *dailymean.txt* is important, while for precipitation data only *weekmean.txt* is important.

5.3 Plotting statistics in Splus.

1. Import the 6 files into Splus.
2. Run the script **PTQ** on these using command:

   ```
   PTQ(<obsP>, <simP>, <obsT>, <simT>, <Disch.obsPT>, <Disch.simPT>,
   basin="<catchment-name>")
   ```

   The script PTQ can be found in appendix B.
3. The resulting plots show how well simulated data fit observed data. An example of these plots are to be found in Appendix B, figure 2 and 3

6 Scenario run

If the simulated data are satisfactory, you may use them to run scenarios. The scenario runs are initiated from 'meny' at Atna. First you must prepare the data for this task.

1. Make sure the files *snow_median.prn* and *returnQ.prn* is present at `//atna/home/flom/PTsimulering/files/ .`
2. Copy the HBV input file of simulated data you created (in 5.1) to `//atna/home/flom/PTsimulering/files/<catchmentname>_simulation.dta`
3. Log into Atna as user 'flom', and type 'meny' in an xterm window. Choose scenario run, simulated data and then the correct catchment. If the catchment is not present, you must edit the 'meny' script (see below). The scenario run should start, and this may take some minutes, depending of how many years of simulated data you have.
4. The FORTRAN program **scenarie_stat** is then initiated. You will be asked for catchment name. Make sure you do not use any other letters than a-z (i.e Austenå will become Austenaa). The program reads some catchment-specific data, and then calculates probabilities of exceedance for some return levels. The resulting file is named `//atna/home/flom/PTsimulering/<catchmentname>_frequency.txt.`

An example of this file can be found in Appendix B. There are a number of options to run the **scenarie_stats** program with; please consult the source code for this. **Scenarie_stats** can be run on any scenario files. In these cases you should use the program **scenarie_stats_all** to avoid using the previously simulated data. Copy this program into the directory where you have the scenario files of interest before running the program.

6.1 Adding catchments to the scenario run

1. Make sure you have prepared HBV-input data file for your catchment (as in 5.1). Copy this file into directory `//atna/home/flom/PTsimulering/files/<catchmentname>_simulation.dta`
2. Edit the file `//atna/home/flom/PTsimulering/files/eksempel-felt.txt` (in Emacs) by moving the '1' in the run-column to the catchment of interest. Save this file as *<catchment>_hbv_stations.txt* .

3. Edit the files *snow_median.prn* and *returnQ.prn* to include your catchment. Information on the median snow cower can be found at `//mikke/Flomvarsling/Forvaltning/Snø/sno_hbv_03_04.xls`. Information of the return levels to include in the file *returnQ.prn* can be found at flomvarslinga. You should, however, calculate this from your simulated data by following the steps in chapter 7 below. It is imperative that *snow_median.prn* have the format [i8, n(i10)], where n=number of catchments. Edit this in Emacs.

4. Finally you must edit the shell-script *meny* to incorporate the new catchment by following the examples given by already existing catchments. Shell scripts does not have to be compiled in any way; just type the name of the file to run it.

This concludes the procedures for simulating data and running scenarios.

7 Calculating empirical return levels

This assumes you have run the HBV-model the standard way (not in scenario mode), which generated one output file (usually named *PRTFIL.RES*). Transform this file using **PTsimulation** menu-4-2.

1. Start the program **gringorten** (so named because of the method used in calculating the levels) using the transformed file as input. This generates 2 files: *gringortenData.txt* and *gringortenStat.txt*.

2. Import the file *gringortenData.txt* into Splus, and run the script **gringorten** with that data frame as input parameter. The script is given in Appendix B. It produces a graphical fit plot of your data vs. a Gumble distribution, as seen in figure 4 (Appendix B). Run the script using command:
   ```
   gringorten(datafile, "basin name")
   ```

3. The file *gringortenStat.txt* consists of calculated return levels for you data. This information can be included in the *returnQ.prn* file (needed during scenario running) by running the program **makeReturnQ**. Remember to have the file *gringortenStat*.txt in the same directory you are running the program.

References

Onof, C., 2000 A precipitation and temperature generator for the nordic climate. Consultancy report for NVE, May 22, 2000

Onof, C., 2003 Time-series modelling of temperatures. Consultancy report for NVE, March 12 2003.

Rindal, J. E. and T. Skaugen, 2004. Simulation of precipitation and temperature for generating long synthetic discharge series for us in spring flood scenarios. NVE Report March 2004.

Appendix A

Example of the file input_bartlewtempsim:

```
  0.01812949   3.24526046   1.93604050   1.14318927   0.82773218   0.06196931   1.00000000
  0.01484152   3.99826509   2.42151530   3.66853836   0.82263910   0.10183637   1.00000000
  0.01341881   4.45511559   2.49134330   2.75300222   1.19418833   0.12845393   1.00000000
  0.01514783   6.02669469   2.94696955   5.11261456   0.77474329   0.13782253   1.00000000
  0.01546467   5.16506166   4.25697313  14.05646036   0.65492738   0.33418611   1.00000000
  0.01935289   3.69565510   6.22093083  14.49089766   0.68352984   0.26255616   1.00000000
  0.02146105   1.88194032   4.39174086   3.82364008   0.45370198   0.08865367   1.00000000
  0.02030774   6.98975353  18.50432612  50.97818524   4.63562045   0.48997604   1.00000000
  0.02322441   3.27798665   2.78225745   4.73315899   0.35708837   0.09339339   1.00000000
  0.02080591   2.59490799   3.05923300   6.08926205   0.29831388   0.08950979   1.00000000
  0.01383224   3.96308717   2.27379737   7.48610454   0.26573477   0.08723624   1.00000000
  0.01813476   5.32784182   2.98333226  10.14145664   0.80812036   0.19487627   1.00000000
    0
 No Title Provided
0.0
 1 6 24 48
1
0
```

Appendix B

EXAMPLE OF TID/VERDI FORMAT

```
19570101/1200 -18.799999
19570102/1200 -16.600000
19570103/1200 -21.600000
19570104/1200 -11.000000
19570105/1200 -0.700000
19570106/1200 -2.900000
19570107/1200 -0.500000
19570108/1200 5.900000
19570109/1200 3.500000
19570110/1200 -2.200000
19570111/1200 -4.000000
19570112/1200 -6.100000
19570113/1200 -10.200000
19570114/1200 -26.900000
19570115/1200 -9.200000
19570116/1200 -0.300000
19570117/1200 -0.200000
19570118/1200 -3.700000
19570119/1200 0.700000
19570120/1200 2.100000
19570121/1200 -2.600000
```

THE SPLUS SCRIPT YEARANALYSIS

```
"yearanalysis" <- function(data,
dato=c(31,65,90,105,120,151,190,240,273,304,334), type = "Temperature", unit =
"Degree Celcius", sdate = "01/01/1980")
{
# Function to plot data made from PTsimulation.
# defseason <- c(31,65,90,105,120,151,190,240,273,304,334)
      defseason <- dato

      par(mfrow = c(2, 2))
      minvalue <- min(data[, 7])
      maxvalue <- max(data[, 8])
      plot(seq(1:365), data[, 2], type = "lines", ylim = c(minvalue, maxvalue),
col = 1, main = paste("Daily mean", type, "and 0.05, 0.95 quantiles"), xlab =
"Julian days", ylab = unit)
      abline(h = 0, lty = 2)
      lines(seq(1:365), data[, 7], col = 2)
      lines(seq(1:365), data[, 8], col = 2)
      abline(v = defseason, col = 9)

      plot(seq(1:365), data[, 4], type = "n", main = "Daily skew", xlab =
"Julian days", ylab = "Skew")
      points(seq(1:365),data[,4],pch=14,mkh=0,cex=0.2,col=8)
      lines(ksmooth(seq(1:365),data[,4],"normal",bandwidth=3),col=1)

      abline(v = defseason, col = 9)
      abline(h = 0)

      plot(seq(1:365), data[, 3],type="n", main = "Daily variance", xlab =
"Julian days", ylab = "Variance")
      points(seq(1:365),data[,3],pch=14,mkh=0,cex=0.2,col=8)
      abline(v = defseason, col = 9)
      lines(ksmooth(seq(1:365),data[,3],"normal",bandwidth=3),col=1)
      minvalue <- min(data[, 5])
      maxvalue <- max(data[, 6])

      plot(seq(1:365), data[, 2], type = "lines", ylim = c(minvalue, maxvalue),
col = 1, main = paste("Daily mean", type, "and daily max/min"), xlab = "Julian
days",ylab = unit)
      abline(h = 0, lty = 2)
      lines(seq(1:365), data[, 5], type = "lines", col = 2)
      lines(seq(1:365), data[, 6], type = "lines", col = 2)
      abline(v = defseason, col = 9)
      par(mfrow = c(1, 1))
}
```

THE SPLUS SCRIPT AR4V22

```
"ar4v22" <- function(data, station = "NA", filename = "ar-noname.txt")
{
# This function takes data (standardized), fits AR(1-4) models to it and choose
# the best model from AIC criteria.
# It works with a n*m matrix of unknown dimensional span. It returns diagnostic
# plots; one for each coloums of data (season), and writes the AR parameters to
# a file.
# Num rows and cols:
      b <- dim(data)[1]
      c <- dim(data)[2]
      print(dim(data))
      d <- matrix(data = NA, nrow = c, ncol = 1)
      filedata <- matrix(data = NA, ncol = 5, nrow = c)
      for(i in 1:c) {
            for(j in 1:5) {
                  filedata[i, j] <- 0
            }
      }
      i <- 1
      j <- 1
      for(i in 1:c) {
            j <- 1
            print(j)
            print(data[j, i])
            while(data[j, i] >= -9990 && j != b) {
                  j <- j + 1
            }
            d[i, 1] <- j - 1
            print(d[,  ])
      }
      par(mfrow = c(2, 2))
      for(j in 1:c) {
            a <- ar(data[1:d[j, 1], j], aic = T, order.max = 4)
            aic <- a$aic      # Finds the order of the best fitted model
            teller <- 1
            for(i in 2:5) {
                  if(aic[i] < aic[teller]) {
                        teller <- i
                  }
            }
            print(aic)
            print(a$ar)
            print("standard deviation     :")
            print(a$var.pred) # Prepares the output data.
            for(i in 1:(teller-1)) {
                  if(a$ar[i] == "NA") {
                        filedata[j, i] <- 0
                  }
                  else {
                        filedata[j, i] <- a$ar[i]
                  }
            }
            filedata[j, 5] <- a$var.pred[1]
            order <- as.character(teller - 1)   # GRAPHICS FOR A COLOUMN
            plot(a$resid, main = paste("Resid.plot, AR(", order, "), station",
station,
                  "s", as.character(j)), ylab = "Residuals")
      # tsplot(a$resid, main=paste("Residuals for an AR(",order,") for
station",station))
            abline(h = 0, col = 8)  # acf(a$resid)
            hist(a$resid, nclass = 25, main = "Histogram", xlab = "Residuals",
ylab =
                  "Density", col = 4, probability = T)
      # lines(seq(-3, 3, 0.1), dnorm(seq(-3, 3, 0.1)))
```

```
            b <- qqnorm((a$resid[5:length(a$resid)] -
mean(a$resid[5:length(a$resid)]))/
                  sqrt(var(a$resid[5:length(a$resid)])), plot = T, main =
                  "QQ normal plot", ylab = "Residuals")
            qqline((a$resid[5:length(a$resid)] -
mean(a$resid[5:length(a$resid)]))/sqrt(
                  var(a$resid[5:length(a$resid)])))   #  abline(0,1,col=8)
            c <- acf(data[1:d[j, 1], j], type = "partial")
      # Linjen under gir feilmeldingen, men hindrer unødig utskrift. La stå!
#  plot(c,main="Partial autocorrelation function")
      }
# end column-wise iteration
      write(format(t(filedata),  , digits = 5), filename, ncol = 5)
      par(mfrow = c(1, 1))
}
```

THE SPLUS SCRIPT GRINGORTEN

```
"gringorten" <- function(data,basin="NA") {


      length <- dim(data)[1]-2
      width <- min(-log(-log(data[1:length,3])))
      width2 <- max(-log(-log(data[1:length,3])))

      height <- max(data[1:length,1])
      low <- min(data[1:length,1])

      u <- data[length+2,3]
      alpha <- data[length+2,4]
      r10 <- data[length+1,1]
      r20 <- data[length+1,2]
      r50 <- data[length+1,3]
      r100 <- data[length+1,4]


      if(length>99) {
            index =
c(1,length%*%0.01,length%*%0.05,length%*%0.1,length%*%0.25,length%*%0.5,length%
*%0.75,length%*%0.9,length%*%0.95,length%*%0.99,length%*%0.998,length)
      }
      if(length<100) {
            index =
c(1,length%*%0.1,length%*%0.25,length%*%0.5,length%*%0.75,length%*%0.9,length%*
%0.98,length)
      }

      distance <- c(-log(-log(data[index,3])))            # prepare axis
      xprob <- c(data[index,3])
      xreturn <- c(data[index,4])

      plot(-log(-
log(data[1:length,3])),data[1:length,1],xaxt="n",xlab="",ylab="Discharge",type=
"n")
      points(-log(-log(data[1:length,3])),data[1:length,1],pch=16, mkh=0,
cex=0.4)
      lines(-log(-log(data[1:length,3])), data[1:length,2] )
      abline(v=(-log(-log(1-1/10))),lty=7,col=4)
      abline(v=(-log(-log(1-1/20))),lty=7,col=4)
      abline(v=(-log(-log(1-1/50))),lty=7,col=4)
      abline(v=(-log(-log(1-1/100))),lty=7,col=4)

      axis(1,at=distance,lab=xprob)
      axis(3,at=distance, lab=xreturn)
      title(paste("Empirical Gumble fit ",basin))
      text(width2,height*1.15,"Return Period T",adj=1)
      text(width2,low,"P[X < x]",adj=1)
      text(width,height*0.9,paste("Num. of data:",length),adj=0,cex=0.8)
      text(width,height*0.86,paste("Gumble location parameter u
:",u),adj=0,cex=0.8)
      text(width,height*0.83,paste("Gumble shape parameter alpha :
",alpha),adj=0,cex=0.8)
      text(width,height*0.78,paste("10  yr returnlvl (est):
",r10),cex=0.8,adj=0,lty=4)
      text(width,height*0.75,paste("20  yr returnlvl (est):
",r20),cex=0.8,adj=0,lty=3)
      text(width,height*0.72,paste("50  yr returnlvl (est):
",r50),cex=0.8,adj=0,lty=2)
      text(width,height*0.69,paste("100 yr returnlvl (est):
",r100),cex=0.8,adj=0,lty=6)

}     # END SCRIPT
```

EXAMPLE OF FREQUENCY.TXT

```
This is estimated probabilities of exceedance
 for the polmak          catchment
 *********************************************
 Total num of outcomes from scenario run:
      995
 (Removed top   0.499999989%)

 Start of scenario run:  4 March
 End of scenario run  :  10 October

 State of catchment  4 March
          Snow       Q    slw    gdw   Prec   Temp
 State  225.54   37.23  49.37  17.41   0.00 -17.93
 Mean   193.00



 Probabilities of exceedance:
               Q  num.above       Prob
   mean  1311.65     844.00      0.848
  5 yrs  1662.16     530.00      0.533
 10 yrs  1947.21     296.00      0.297
 20 yrs  2220.64     130.00      0.131
 50 yrs  2574.57      44.00      0.044
100 yrs  2839.78      18.00      0.018

 Most likely days the peaks will occur
Peaks above mean.....  29   May      (daynum 149) +/-   11 days (st.dev)
Peaks above 5 yrs....  30   May      (daynum 150) +/-   10 days (st.dev)
Peaks above 10 yrs...  30   May      (daynum 150) +/-   10 days (st.dev)
Peaks above 20 yrs...  31   May      (daynum 151) +/-   10 days (st.dev)
Peaks above 50 yrs...   1   June     (daynum 152) +/-   10 days (st.dev)
Peaks above 100 yrs..   4   June     (daynum 155) +/-   13 days (st.dev)
```

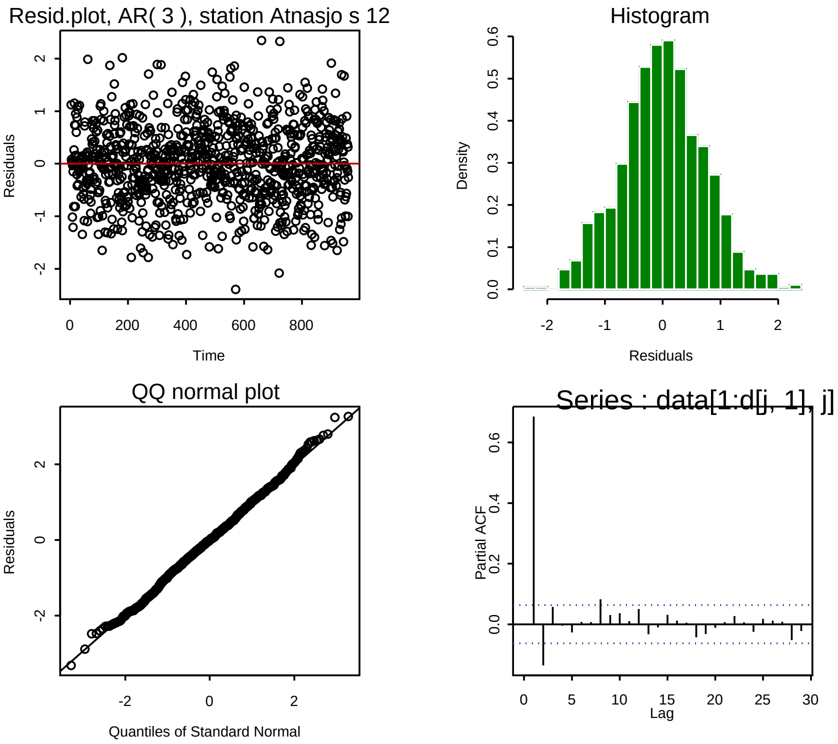

Figure 1 Example of validation plot produced by ar4v22. This is for Atnasjo season 12, and an AR model of order 3 has been found to be optimal.

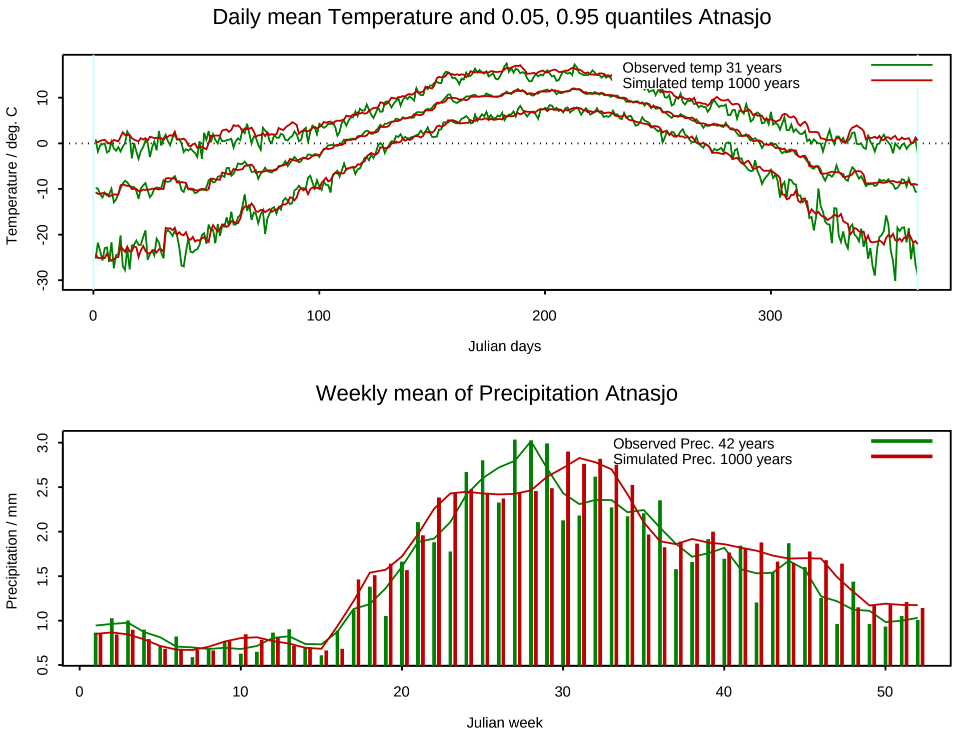

Figure 2 Plots of simulated and observed precipitation and temperature for use in validation.

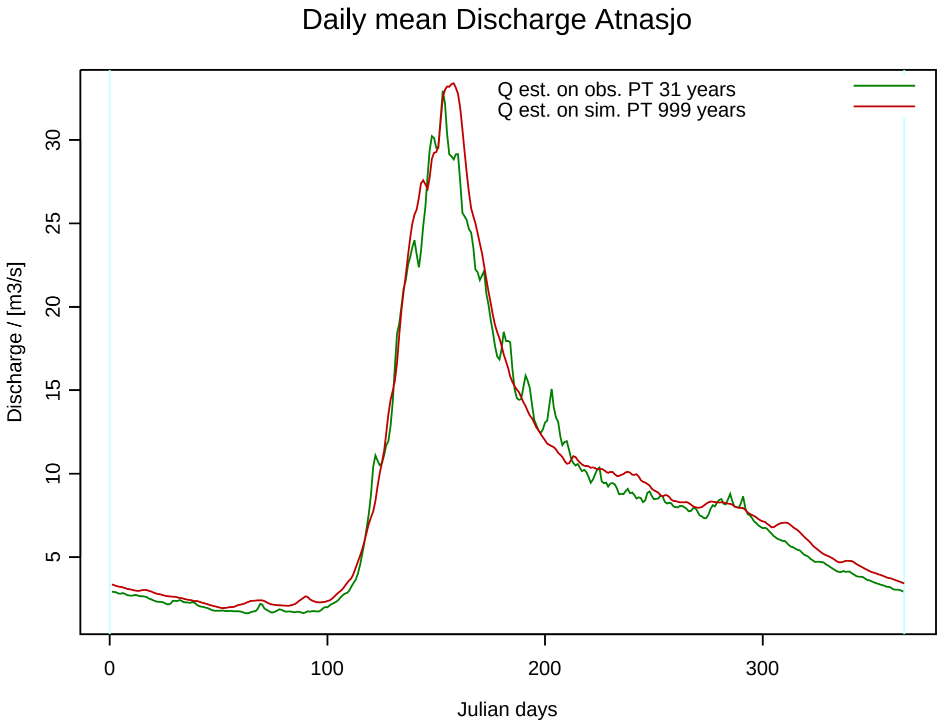

Figure 3 Plot of discharge estimation based on simulated and observed precipitation and temperature data.

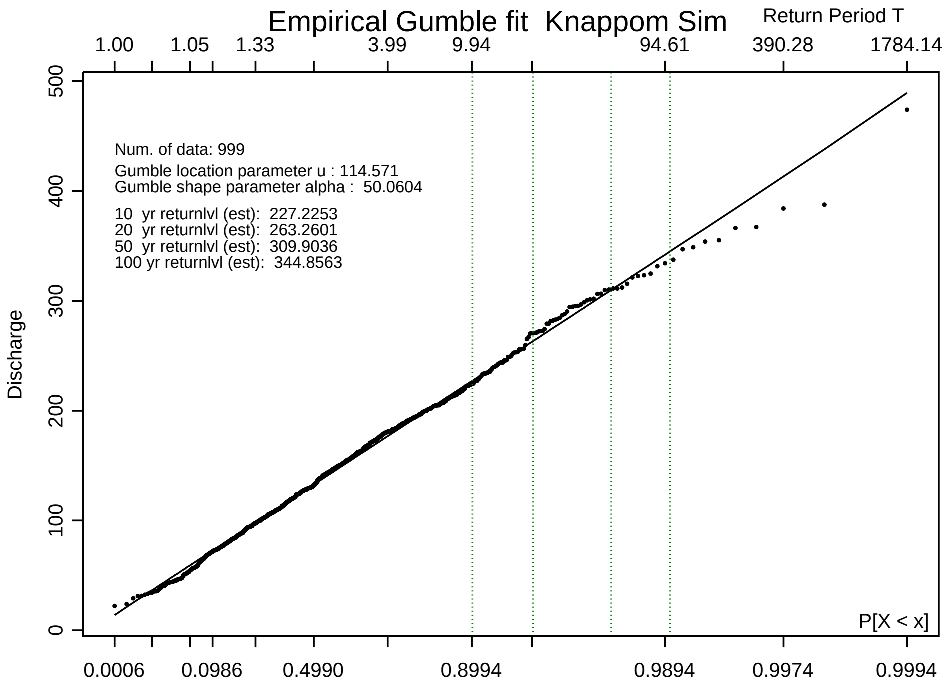

Figure 4 Example of return level plot based on simulated discharge data.

This series is published by Norwegian Water Resources and Energy Directorate (NVE)

Published in the Report series 2004

No. 1 Jens Erik Rindal, Thomas Skaugen: Simulation of precipitation and temperature for generating long synthetic discharge series for use in spring flood scenarios. (22 pp.)

No. 2 Jens Erik Rindal, Thomas Skaugen: Manual for generating synthetic series of precipitation and temperature for use in spring flood scenarios. (22 pp.)