171

Tool to determine the coefficient of variation of DegT50 values of plant protection products in water-sediment systems for different values of the sorption coefficient

M.M.S. ter Horst
J.G. Groenwold

werkdocumenten

WOT
Wettelijke Onderzoekstaken Natuur & Milieu

WAGENINGEN UR
For quality of life

Tool to determine the coefficient of variation of DegT50 values of plant protection products in water-sediment systems for different values of the sorption coefficient

The 'Working Documents' series presents interim results of research commissioned by the Statutory Research Tasks Unit for Nature & the Environment (WOT Natuur & Milieu) from various external agencies. The series is intended as an internal channel of communication and is not being distributed outside the WOT Unit. The content of this document is mainly intended as a reference for other researchers engaged in projects commissioned by the Unit. As soon as final research results become available, these are published through other channels. The present series includes documents reporting research findings as well as documents relating to research management issues.

This document was produced in accordance with the Quality Manual of the Statutory Research Tasks Unit for Nature & the Environment (WOT Natuur & Milieu) and has been accepted by Paul Hinssen, programme coordinator of WOT Natuur & Milieu.

WOt Working Document **171** presents the findings of a research project funded by the Dutch Ministry of Agriculture, Nature and Food Quality (LNV).

Tool to determine the coefficient of variation of DegT50 values of plant protection products in water-sediment systems for different values of the sorption coefficient

M.M.S. ter Horst

J.G. Groenwold

Werkdocument 171

Wettelijke Onderzoekstaken Natuur & Milieu

Wageningen, December 2009

Postbus 47, 6700 AA Wageningen
Tel: (0317) 48 07 00; fax: (0317) 41 90 00; e-mail: info.alterra@wur.nl

The Working Documents series is published by the Statutory Research Tasks Unit for Nature & the Environment (WOT Natuur & Milieu), part of Wageningen UR. **This document is available from the secretary's office, and can be downloaded from 7Hwww.wotnatuurenmilieu.wur.nl.**

Statutory Research Tasks Unit for Nature & the Environment, P.O. Box 47, NL-6700 AA Wageningen, The Netherlands
Phone: +31 317 48 54 71; Fax: +31 317 41 90 00; e-mail: 8Hinfo.wnm@wur.nl;
Internet: 9Hwww.wotnatuurenmilieu.wur.nl

F-0008 vs. 1.6 [2009] Project WOT-04-003-006 – 5233523.01 [Werkdocument 171 – december 2009]

Contents

Summary

In the present Dutch registration procedure and the EU procedure for inclusion of pesticides in a positive Community list of pesticide substances (“Annex1”; Directive 91/414) inverse modeling of water-sediment experiments is used to estimate degradation rates in both the water layer and the sediment. It is questionable whether water-sediment experiments are suitable for estimating the degradation half lives in the water layer and the sediment separately. This question will be tested using simulations with a model describing pesticide behaviour in a water layer and its underlying sediment. The method and results of this test will be described in a scientific publication (to be submitted in 2011). The intended method requires large numbers of model simulations and calibration runs. Manually operation of these processes is unworkable. Therefore a tool was developed to automate these processes. The functioning of this tool is described in this document. This means that this WOT work-document does not give an answer to the research question but merely presents a tool necessary to execute the method for answering the research question.

1 Introduction

In the present Dutch registration procedure and the EU procedure for inclusion of pesticides in a positive Community list of pesticide substances ("Annex1"; Directive 91/414) inverse modeling of water-sediment experiments is used to estimate degradation rates in both the water layer and the sediment. However the original aim of a water-sediment experiment was to get a picture of the products formed for the pesticide and their persistence in water-sediment systems. Till now it is still unclear whether water-sediment experiments are suitable for estimating of the degradation half lives in the water layer and the sediment separately. This question will be tested using simulations with a model describing pesticide behaviour in a water layer and its underlying sediment, in order to exclude experimental errors and peculiarities (Adriaanse *et al.*, 2002).

Therefore two theoretical water-sediment experiments will be defined. For each theoretical experiment Monte Carlo simulations using the model, TOXSWA (Adriaanse, 1996, Beltman *et al.*, 2006), will be done to generate hypothetical datasets using different combinations of values of the degradation half lives in water and sediment and the sorption coefficient. Per simulation concentrations in water and sediment at appropriate sampling times will be selected. These true concentrations will be replaced by random values of the concentration to take into account the uncertainty in the measurements observed in true water-sediment experiments. Per hypothetical dataset the degradation half lives will be optimized using the automatic calibration program PEST (Doherty, 2005) in combination with the model. Next the coefficient of variation of the degradation half lives in water and sediment will be determined per hypothetical dataset.

Results of the uncertainty in DegT50 values estimated from water-sediment experiments will be presented in contour plots. The method and results of this test will be described in a scientific publication (to be submitted in 2010).

The intended method briefly described above requires:
i. large number of simulations with the model to generate the hypothetical datasets and
ii. large number of calibration runs.

Manually making input files and processing the results of both the simulations to generate the hypothetical datasets and the calibrations runs is unworkable. Therefore a tool was developed to automate the two processes described above. The functioning of this tool is described in Chapter 2 of this document.

Furthermore a procedure was developed for using the tool on the Alterra gridcomputer system. This procedure is described in Chapter 3 of this document.

2 Description of the automatic procedure (DegT50 tool)

The tool was developed to automatically make input files and process output of (i) simulations with the model, TOXSWA, to generate the hypothetical datasets and (ii) calibration runs of water-sediments studies simulated with TOXSWA is a script in R. R is a language and environment for statistical computing and graphics (http://www.r-project.org/).

The functioning of the R-script will be explained step by step.

2.1 Working of the R-script Part_2.R; brief description

For each pesticide property combination (n=400) and draw (n=25), two hypothetical datasets of water-sediment studies are generated, PEST input files are automatically generated, PEST is run and needed output is extracted from a PEST output file and written in a 'summary spreadsheet'.

2.2 Needed files and folder structure

The used folder structure is depicted in Figure 1.

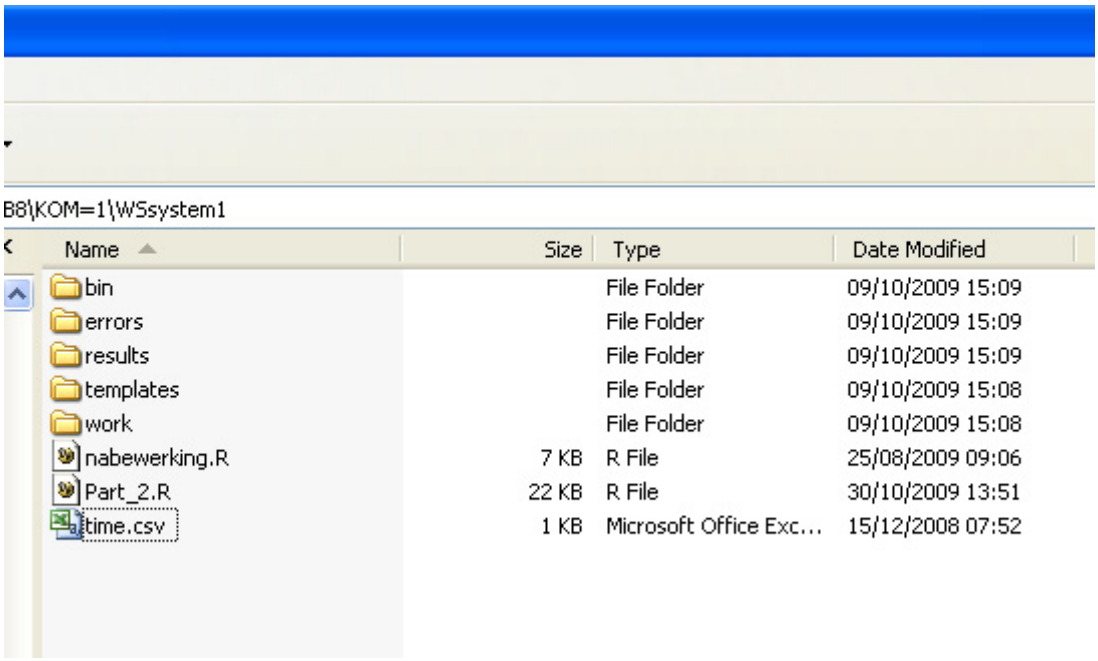


Figure 1. Used folder structure for running the R-script for aged sorption

The file 'Part_2.R' contains the used R-script.
The file 'time.csv' (Annex 1) specifies the sampling times of the measurements.
Csv file: time.csv is read by the R-script.

Other 'input files' are found in the folder templates/PEST (Figure 2). During the procedure the entire content of the folder template/PEST is copied to the folder work. Template PEST input file 'Fit_TOXSWA.pst' is adapted. i.e. values of for instance measurements are automatically written in this template files.

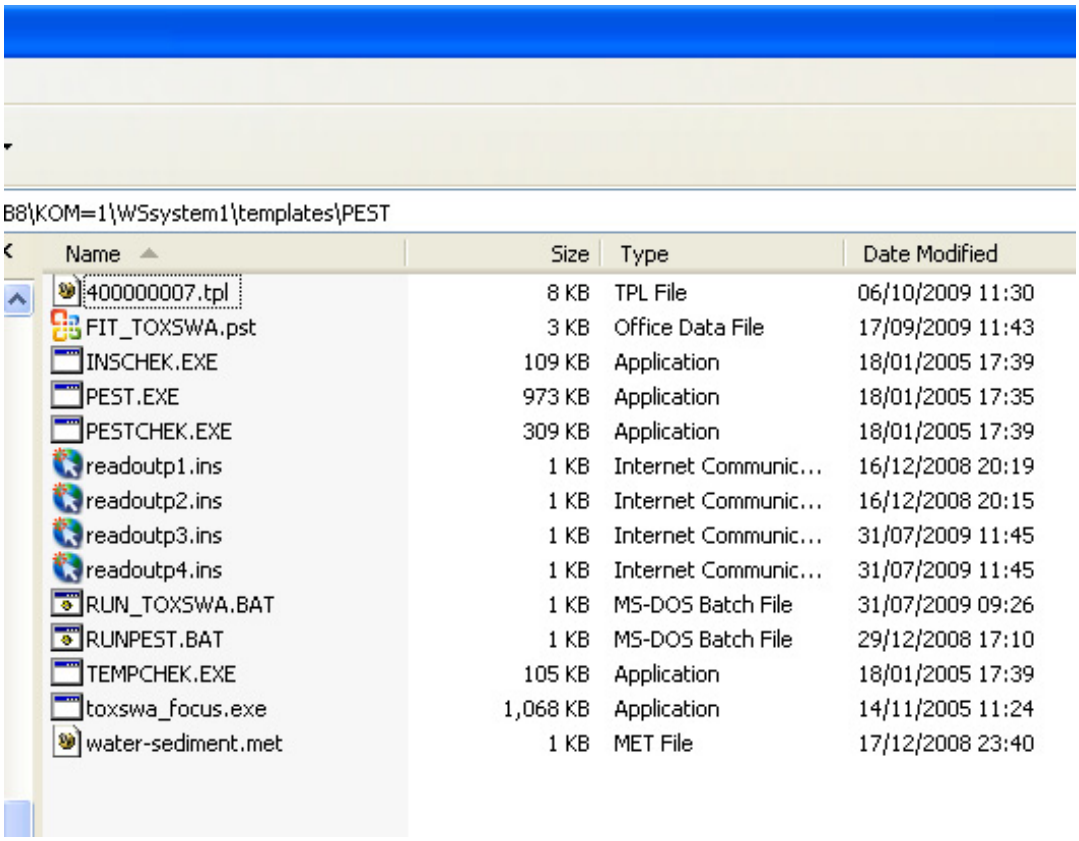


Figure 2. Content of the folder template/PEST

All calculations are performed in the folder 'work' and results are written to the folder 'results'.
Folder 'bin' contains the executables:
lognormal_WS_sediment.exe' and 'lognormal_WS_waterlayer.exe'.
Folder 'errors' is not used.

2.3 Data needed for the input files of PEST

Data needed for the input file for the parameter estimation procedure using PEST are:
- observations (hypothetical datasets; 2 in total)
- initial values DegT50water and DegT50sediment

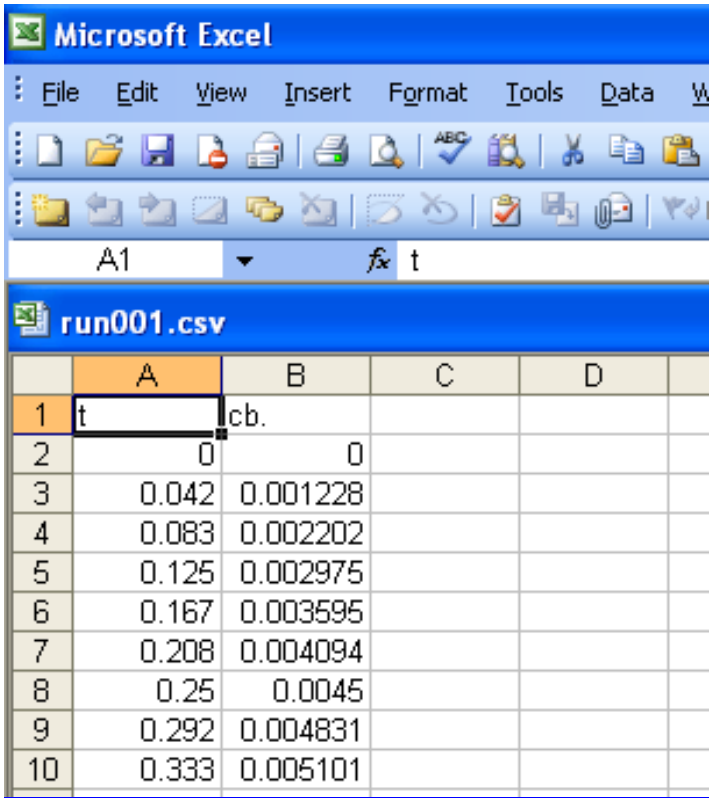
2.4 Data needed for making the two hypothetical datasets and calculation of the initial values of $DegT50_{water}$ and $DegT50_{sediment}$

Data needed for making the 2 hypothetical datasets are:

- observations
    - pre described sampling times → time.csv
    - error structure/relative errors → ..\templates\cv_wat.txt and ..\templates\cv_sed.txt
    - per pesticide property combination (n =400): true values →
        - results\TOXSWA\water\run001.csv – run400.csv
        - results\TOXSWA\sediment\run001.csv – run400.csv (see Figure 3)

Data needed for calculation of the initial values of $DegT50_{water}$ and $DegT50_{sediment}$ are:

- per pesticide property combination (n = 400) and draw (n=25) the 2 hypothetical datasets



	A	B	C	D
1	t	cb.		
2	0	0		
3	0.042	0.001228		
4	0.083	0.002202		
5	0.125	0.002975		
6	0.167	0.003595		
7	0.208	0.004094		
8	0.25	0.0045		
9	0.292	0.004831		
10	0.333	0.005101		

Figure 3. Content of 'results\TOXSWA\sediment\run001.csv '

2.5 Actions needed for making the two hypothetical datasets and calculation of the initial value of DegT50water and DegT50 sediment

2.5.1 Hypothetical datasets

a. For each combination of pesticide properties (n=400; see data frame 'key' in Figure 5) get the true values on the pre described sampling times → R-script (Figure 4) → merge time.csv with results\TOXSWA\water\run001.csv – run400.csv → result is data frame 'tmp1' and merge time.csv with results\TOXSWA\sediment\run001.csv – run400.csv → result is data frame 'tmp2' (see Figure 6; note for loop over number of rows in data frame key = number of combinations of pesticide properties)

b. Content of data frame 'tmp1' is written to the text file 'inputwl.txt' (Figure 4). Content of data frame 'tmp2' is written to the text file 'inputsed.txt' (Figure 4).

c. make 2 hypothetical data sets using data frames 'tmp1' and 'tmp2'.
   i. Content of 'tmp1' was written to 'inputwl.txt' and 'inputwl.txt' is used as input file in the Fortran program 'lognormal_WS_waterlayer.exe' this program is called in the R-script and run (Figure 7a)
   ii. Content of 'tmp2' was written to 'inputsed.txt' and 'inputsed.txt' is used as input file in the Fortran program 'lognormal_WS_sediment.exe' this program is called in the R-script and run (Figure 7b)

Fortran programs 'lognormal_WS_waterlayer.exe' and 'lognormal_WS_sediment.exe' creates hypothetical datasets by adding a random error to the 'true values of the observations' (See Annex 2 for the code of these programs). The error is specified in the files 'cv_wat.txt' and 'cv_sed.txt' (Figure 8). These files are initially located in the folder 'templates' but copied to the folder 'work' (see code in Figure 4). The Fortran programs 'lognormal_WS_waterlayer.exe' and 'lognormal_WS_sediment.exe' are run in the folder 'work'.

The output of the Fortran program 'lognormal_WS_waterlayer.exe' is written to the data frame 'tmp3' (Figure 9) and the output of the Fortran program 'lognormal_WS_sediment.exe' is written to the data frame 'tmp4' (Figure 9).

```
# copy all templates to work directory
files <- list.files(path = "../templates", full.names = TRUE)
file.copy(from = files, to = ".", overwrite = TRUE)

tmp <- expand.grid(
       dt50wl =
c(1,500^(1/19),500^(2/19),500^(3/19),500^(4/19),500^(5/19),500^(6/19),500^(7/19),500^(8/19),500^(9/19),500^(10/19),500^(11/19),500^(12/19)
,500^(13/19),500^(14/19),500^(15/19),500^(16/19),500^(17/19),500^(18/19),500),
       dt50sed =
c(1,500^(1/19),500^(2/19),500^(3/19),500^(4/19),500^(5/19),500^(6/19),500^(7/19),500^(8/19),500^(9/19),500^(10/19),500^(11/19),500^(12/19)
,500^(13/19),500^(14/19),500^(15/19),500^(16/19),500^(17/19),500^(18/19),500)
 )
key <- data.frame(filename = 1:nrow(tmp), tmp)
.
# read time of 'measurements'
time <- read.csv(file = "../time.csv", as.is = TRUE)
.
# specificeer het aantal trekkingen op zo'n manier dat je ndraw entries krijgt
Ndraws = 25
.
for(i in 1:nrow(key)){

# bepalen van de hypothetical data set with the true concentrations
#-------------waterlaag ----------------------------
# 1. mergen van uitvoer toxswa met time.csv
    result1 <- read.csv(file = paste("../results/TOXSWA/waterlayer/", key$filename[i], sep = ""), as.is = TRUE)
    tmp1 <- merge(x = time, y = result1, by = "t", all = FALSE)

   # schrijf weg als invoerfile for lognormal fortranprogramma
    write.table(x = tmp1, file = paste("inputwl.txt"), quote = FALSE, sep = "     ", row.names = FALSE, col.names = TRUE)

   # schrijf weg voor archivering
   filename3 <- paste("../results/hypothetical_observations/waterlayer/outputTOXSWA",run$filename[i], sep = "/")
   write.table(x = tmp1, file = filename3, quote = FALSE, sep = "     ", row.names = FALSE, col.names = TRUE)
```

Copy the contents of folder 'templates' to folder 'work'

Make dataframe (key) with 400 combinations of pesticide properties

Loop over the number of rows in dataframe key = loop over number of parametercombinations --> 20*20 = 400 combinations of DegT50water en DegT50sediment

```
# bepalen van de hypothetical data set with the true concentrations
#------------sediment-----------------------------------------------
# 1. mergen van uitvoer toxswa met time.csv
    result2 <- read.csv(file = paste("../results/TOXSWA/sediment/", key$filename[i], sep = ""), as.is = TRUE)
    tmp2 <- merge(x = time, y = result2, by = "t", all = FALSE)

   # schrijf weg als invoerfile for lognormal fortranprogramma
    write.table(x = tmp2, file = paste("inputsed.txt"), quote = FALSE, sep = "     ", row.names = FALSE, col.names = TRUE)

   # schrijf weg voor archivering
   filename4 <- paste("../results/hypothetical_observations/sediment/outputTOXSWA",run$filename[i], sep = "/")
   write.table(x = tmp2, file = filename4, quote = FALSE, sep = "     ", row.names = FALSE, col.names = TRUE)
```

Figure 4. Part of the R-scripts used to get the true values on the pre described sampling times. # in R-script code = comment rule

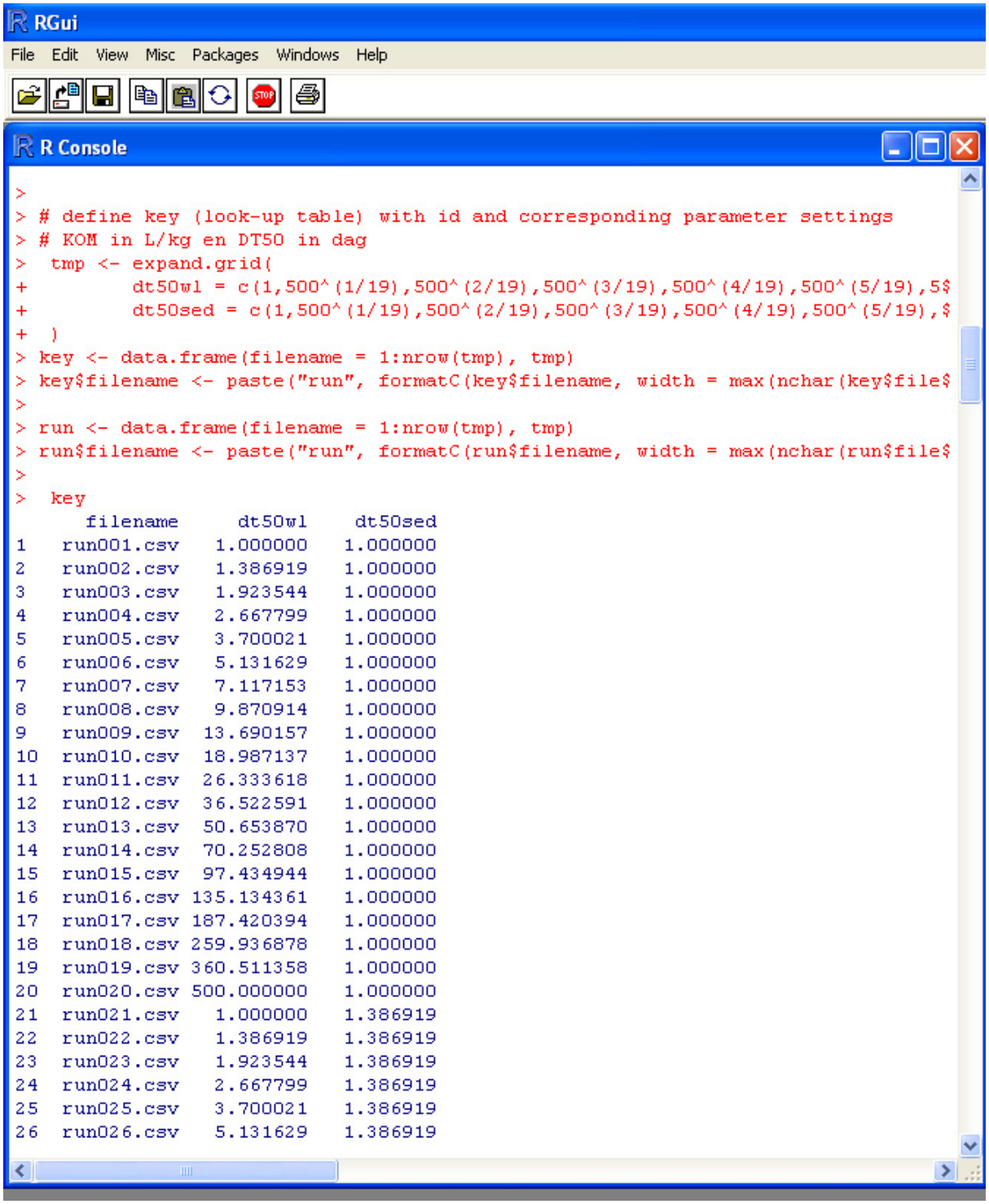


Figure 5. Data frame 'key'→ contains the true values of the parameters to be fitted

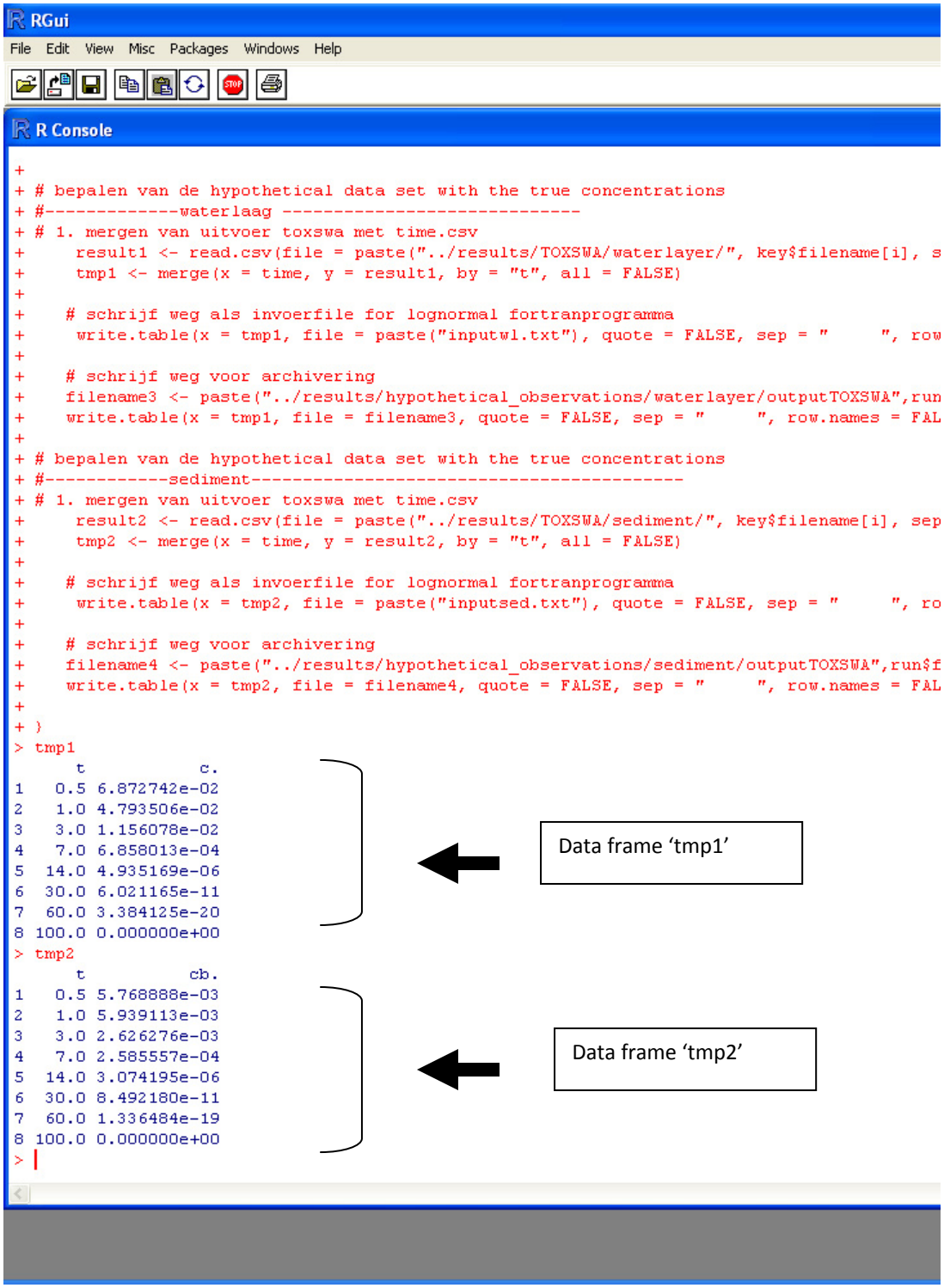


Figure 6. Data frames 'tmp1' and ' tmp2'→ contain the true values of concentration in water (tmp1) and sediment (tmp2) on the pre-described sampling times (in time.csv)

```
for (j in 1:Ndraws){

#-------------waterlaag -----------------------------

# 2. LogNormal fortran programma aanroepen. inputwl.txt is invoer voor dit programma
# output.txt is uitvoer van dit programma
# Let op dit moet je 2x keer doen, omdat je 2 replicates setjes wil! Zie paper
    # execute LogNormal_WS_waterlayer.exe
    # let op de outputfile (output.txt) wordt naar de work directory geschreven.
   system(command = "../bin/lognormal_WS_waterlayer.exe inputwl.txt", show.output.on.console = FALSE,invisible = FALSE)

# lees de uitvoer van LogNormal.exe
LogNormalOutput <- read.table(file = paste("outputwl.txt"), as.is = TRUE,  header = TRUE, sep = "", quote = "", dec = ".")

# stop de uitvoer van LogNormal.exe in een dataframe tmp4 dat je later gebruikt om PEST invoer te maken
   tmp3 <- data.frame(
   t = LogNormalOutput$t,
   ConcWlNew = LogNormalOutput$c_hyp.obs,
   WwlNew = LogNormalOutput$weegfactor
   )

# archiveer de uitvoer van LogNormal.exe
   runnr <- paste("run",formatC(i,width =max(nchar(nrow(key))), flag = "0") , sep="")
   dir.create(paste("../results/hypothetical_observations/waterlayer/outputLogNormal/",runnr,sep = ""))
   dir =  paste("../results/hypothetical_observations/waterlayer/outputLogNormal/",runnr,sep = "")
   tmp =  paste(ndraws$filenameNdraw[j],".txt",sep = "")
   filename5 = paste(dir,tmp,sep = "/")
   write.table(x = tmp3, file = filename5, append = FALSE, quote = FALSE, sep = "     ", row.names = FALSE, col.names = TRUE)
```

Loop over the number of draws (25)

Figure 7a. Part of the R-scripts used to get a hypothetical dataset containing concentrations in the water layer. # in R-script code = comment rule

Code following the code given in Figure 7a, so inside the for loop over nrow(key)and Ndraws

```
#------------sediment-----------------------------------------
# 2. LogNormal fortran programma aanroepen. ../LogNormal/sediment/input.txt is invoer voor dit programma
# ../LogNormal/sediment/output.txt is uitvoer van dit programma
# Let op dit moet je 2x keer doen, omdat je 2 replicates setjes wil! Zie paper
    # execute LogNormal_WS_sediment.exe
    # let op de outputfile (output.txt) wordt naar de work directory geschreven.
   system(command = "../bin/lognormal_WS_sediment.exe inputsed.txt", show.output.on.console = FALSE,invisible = FALSE)

# lees de uitvoer van LogNormal.exe
LogNormalOutput <- read.table(file = paste("outputsed.txt"), as.is = TRUE,  header = TRUE, sep = "", quote = "", dec = ".")

# stop de uitvoer van LogNormal.exe in een dataframe tmp4 dat je later gebruikt om PEST invoer te maken
   tmp4 <- data.frame(
   t = LogNormalOutput$t,
   ConcSedNew = LogNormalOutput$c_hyp.obs,
   WsedNew = LogNormalOutput$weegfactor
   )

# archiveer de uitvoer van LogNormal.exe
   runnr <- paste("run",formatC(i,width =max(nchar(nrow(key))), flag = "0") , sep="")
   dir.create(paste("../results/hypothetical_observations/sediment/outputLogNormal/",runnr,sep = ""))
   dir =  paste("../results/hypothetical_observations/sediment/outputLogNormal/",runnr,sep = "")
   tmp =  paste(ndraws$filenameNdraw[j],".txt",sep = "")
   filename6 = paste(dir,tmp,sep = "/")
      write.table(x = tmp4, file = filename6, append = FALSE, quote = FALSE, sep = "     ", row.names = FALSE, col.names = TRUE)
```

Figure 7b. Part of the R-scripts used to get a hypothetical dataset containing concentrations in the sediment. # in R-script code = comment rule

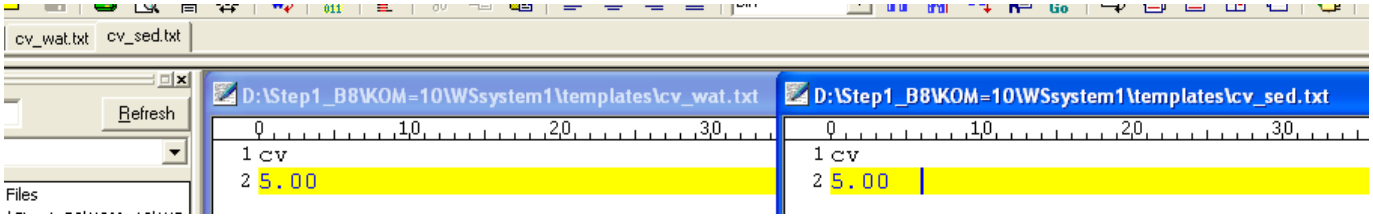
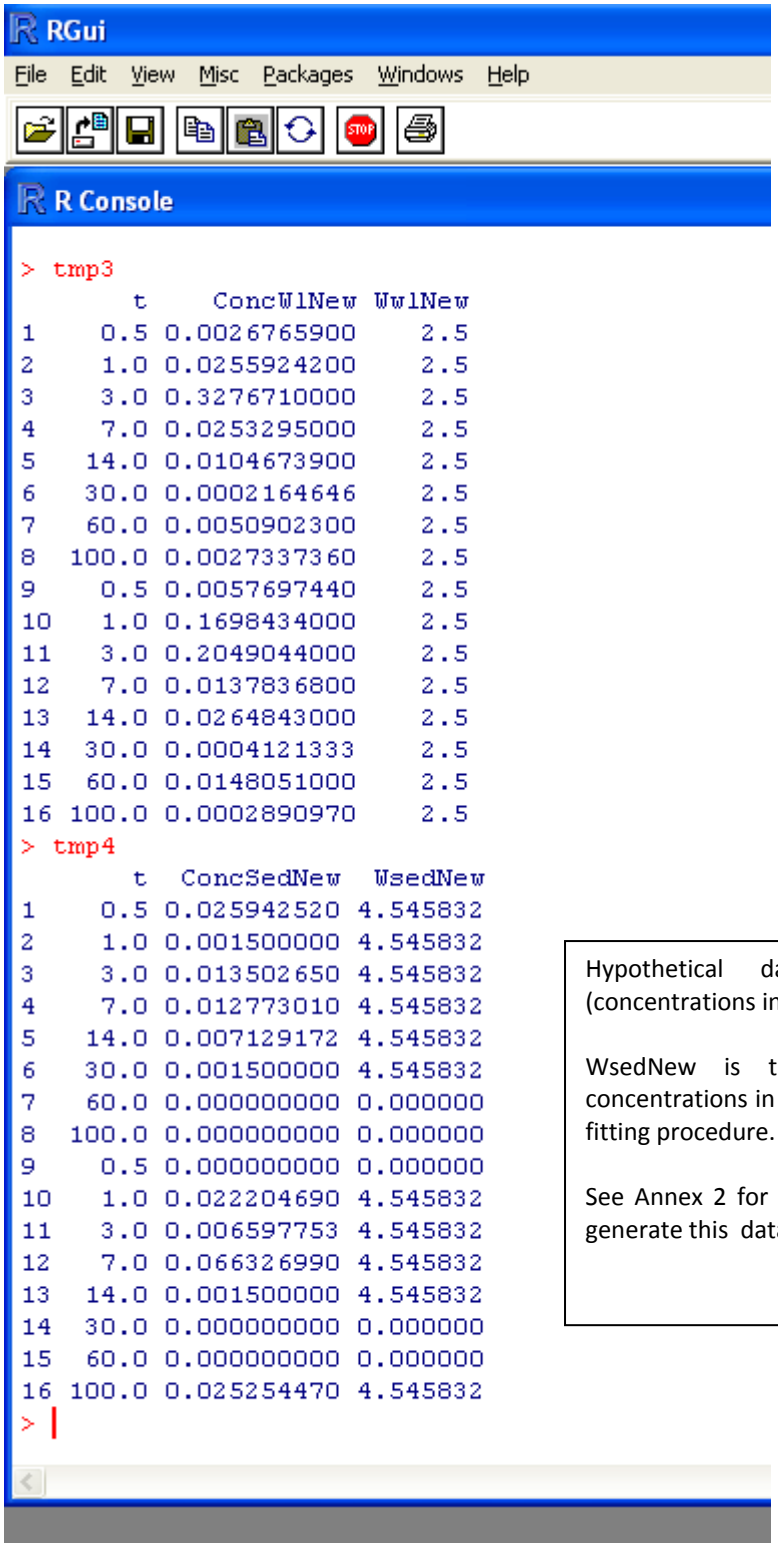


Figure 8. Content of files 'cv_wat.txt' and 'cv_sed.txt'



Hypothetical dataset for the sediment (concentrations in sediment).

WsedNew is the weight factor for the concentrations in the sediment to be used in the fitting procedure.

See Annex 2 for code Fortran program used to generate this dataset

Figure 9. Output of the Fortran programs 'lognormal_WS_waterlayer.exe' and 'lognormal_WS_sediment.exe' (example of nrow(key) = 253 → DegT50water (true value) = 50.65 and DegT50sediment (true value) = 50.65).

2.5.2 Initial values $DegT50_{water}$ and $DegT50_{sediment}$

Initial values of $DegT50_{water}$ and $DegT50_{sediment}$ are needed per pesticide property combination (n = 400) and draw (n=25).

a. 'tmp3' and 'tmp4' contain both 2 replicate datasets. First step (a. in Figure 10) is to make for each of the compartments (water, sediment) one dataset. This is done by
   i. i. making 4 separate data frames (2 for water and 2 for sediment, see Figure 11) and
   ii. calculating per sampling time the average of the concentrations on that sampling time (data frame: 'hyp_dataset_avgConc'; see Figure 12).

b. Determine the sampling time with the maximum concentration of the dataset (see Figure 13 for result of running the code in the R-script for this item) from the dataset of the average concentration in the sediment (column ConcSedAvg in data frame 'hyp_dataset_avgConc'.

c. Calculate the mass in water layer, sediment en total water-sediment system on sampling time t=100 days and the sampling time where the maximum sediment concentration is found (see Figure 14 for result).

d. Calculate the total degradation rate (ktot) according:
   *m= m0 e^(-ktot*t) and t = t100d - tb*
   where tb = breakpoint (time at which decline starts)

Note that log in R is the natural logarithm (ln).

e. The result is written to data frame 'afbraaksnelheid' (see Figure 15 for result).

f. Archive data frame 'afbraaksnelheid' by writing its contents to a text file.

g. Set for each value of k a set of starting values for fitting with TOXSWA-PEST. k = 3, so 3 fits (each fit with a different set of starting values) are done per draw. → 400 parameter combinations * 25 draws * 3 sets of starting values = 30 000 fits with TOXSWA-PEST.

Code following the code given in Figure 7b, so inside the for loop over nrow(key)and Ndraws

```
#----------DT50 overall----------------------------------------
#2. Berekenen van de overall DT50 uit de hypothetische dataset!!!
#a. uit elkaar halen van de 2 replicate sets
   repl <- data.frame(
   n = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
   )
   repl_wl <- cbind(tmp3, repl)
   repl_sed <- cbind(tmp4, repl)
   set1 <- data.frame(
      n = c(1,2,3,4,5,6,7,8)
   )
   set2 <- data.frame(
      n = c(9,10,11,12,13,14,15,16)
   )
   repl_wl_1 <- merge(x = repl_wl, y = set1, by = "n", all = FALSE)
   repl_wl_2 <- merge(x = repl_wl, y = set2, by = "n", all = FALSE)
   repl_sed_1 <- merge(x = repl_sed, y = set1, by = "n", all = FALSE)
   repl_sed_2 <- merge(x = repl_sed, y = set2, by = "n", all = FALSE)
#------------ replicate dataset with average concentrations --------------
   hyp_dataset_avgConc <- data.frame(
      t = repl_wl_1$t,
      ConcWl_avg = 0.5*(repl_wl_1$ConcWlNew + repl_wl_2$ConcWlNew),
      ConcSed_avg = 0.5*(repl_sed_1$ConcSedNew + repl_sed_2$ConcSedNew)
   )
#b. bepalen van tsedmax van hypothetische dataset met average concentrations (tijdstap
waarop maxCsed)
# replicate set 1
   maxCsed <- data.frame(
   t = 1,
   ConcSed_avg = max(hyp_dataset_avgConc$ConcSed_avg)
    )
   tmp5 <- merge(x = hyp_dataset_avgConc, y = maxCsed, by = "ConcSed_avg", all = FALSE)
   t100 <- data.frame(
      t = 100
    )
   Ct100 <- merge(x = hyp_dataset_avgConc, y = t100, by = "t", all = FALSE)
   tmp6a <- data.frame(
      t = tmp5$t.x,
      ConcSed_avg = tmp5$ConcSed_avg
   )
# Let op! het komt voor dat je de max conc in het sediment niet uniek is. Kortom conc
# in sed neemt niet meer af!. Dan heb je meerdere regels in tmp6a. Dat geeft problemen.
# neem daarom de min. t uit tmp6.
   minTtmp6 <- data.frame(
   t = min(tmp6a$t)
    )
   tmp6b <- (merge(x = tmp6a, y = minTtmp6, by = "t", all = FALSE))
   tmp7 <- merge(x=tmp6b, y=hyp_dataset_avgConc, by="t", all = FALSE)
   Ctcsedmax <- data.frame(
      t =tmp7$t,
      ConcWl_avg=tmp7$ConcWl_avg,
      ConcSed_avg=tmp7$ConcSed_avg.x
   )
# c. berekenen van massa's in wl en sed op t =100 d en t= op moment max sediment conc.
   Mt100 <- data.frame(
      t = Ct100$t,
      Mwl = (Ct100$ConcWl_avg*75000*0.000001),
      Msed = (Ct100$ConcSed_avg*25000*0.000001),
      Mtot = ((Ct100$ConcWl_avg*75000*0.000001)+(Ct100$ConcSed_avg*25000*0.000001))
      )
  Mtcsedmax <- data.frame(
      t = Ctcsedmax$t,
      Mwl = (Ctcsedmax$ConcWl_avg*75000*0.000001),
      Msed = (Ctcsedmax$ConcSed_avg*25000*0.000001),
      Mtot = ((Ctcsedmax$ConcWl_avg*75000*0.000001)+(Ctcsedmax$ConcSed_avg*25000*0.000001))
      )
```

```
# d. berekenen van ktot volgens m = m0 e^(-ktot*t) en t = t100d - tb
# tb =  breakpoint (time at which decline starts)
# let op log in R is de natuurlijke logaritme (ln)
 if (Mt100$Mtot < Mtcsedmax$Mtot){
    ktot = (log(Mt100$Mtot/Mtcsedmax$Mtot))/-(Mt100$t-Mtcsedmax$t)
    DT50overall = log(2)/ktot
 }
 if (Mt100$Mtot >= Mtcsedmax$Mtot){
    ktot = (log(Mt100$Mtot/Mtcsedmax$Mtot))/-(Mt100$t-Mtcsedmax$t)
    DT50overall = 100.0
 }
 afbraaksnelheid <- data.frame(
    t_Mtcsedmax = Mtcsedmax$t,
    t_100d = Mt100$t,
    mtot_tcsedmax = Mtcsedmax$Mtot,
    mtot_t100d = Mt100$Mtot,
    ktot = ktot,
    DT50overall =DT50overall
 )

# e. archiveren van dataset waarmee je de overall DT50 hebt bepaald
   runnr <- paste("run",formatC(i,width =max(nchar(nrow(key))), flag = "0") , sep="")
   dir.create(paste("../results/DT50overall/",runnr,sep = ""))
   dir =  paste("../results/DT50overall/",runnr,sep = "")
   tmp =  paste(ndraws$filenameNdraw[j],".txt",sep = "")
   filename7 = paste(dir,tmp,sep = "/")
   write.table(x = afbraaksnelheid, file = filename7, append = FALSE, quote = FALSE, sep =
"     ", row.names = FALSE, col.names = TRUE)

# f. loop over het aantal startwaarden combinaties → per keer andere set startwaarden
for (k in 1:3){
if (k==1){
   startcat=1
   ini_dt50wl <- afbraaksnelheid$DT50overall
   ini_dt50sed <- afbraaksnelheid$DT50overall
   }
if (k==2){
   startcat=2
   ini_dt50wl <- 0.2*afbraaksnelheid$DT50overall
   ini_dt50sed <- 5*afbraaksnelheid$DT50overall
   }
if (k==3){
   startcat=3
   ini_dt50wl <- 5*afbraaksnelheid$DT50overall
   ini_dt50sed <- 0.2*afbraaksnelheid$DT50overall
   }
# ini_dt50wl en ini_dt50sed mogen niet lager dan 0.1 of hoger dan 1000 worden
if (ini_dt50wl < 1)
    ini_dt50wl <- 1
if (ini_dt50sed < 1)
    ini_dt50sed <- 1
if (ini_dt50wl > 100)
    ini_dt50wl <- 100
if (ini_dt50sed > 100)
    ini_dt50sed <- 100
```

ktot becomes negative incase Mt100$Mtot >= Mtcsedmax$Mtot.
In that case DegT50system (=DegT50overall) is set to 100 days

Loop over the number of sets of starting values used for fitting (n=3). So for each set of starting values one fit with TOXSWA-PEST

Figure 10. Code in R-script for calculating the Degt50system to be used as initial value in the fitting procedure (initial value for DegT50water and DegT50sediment).

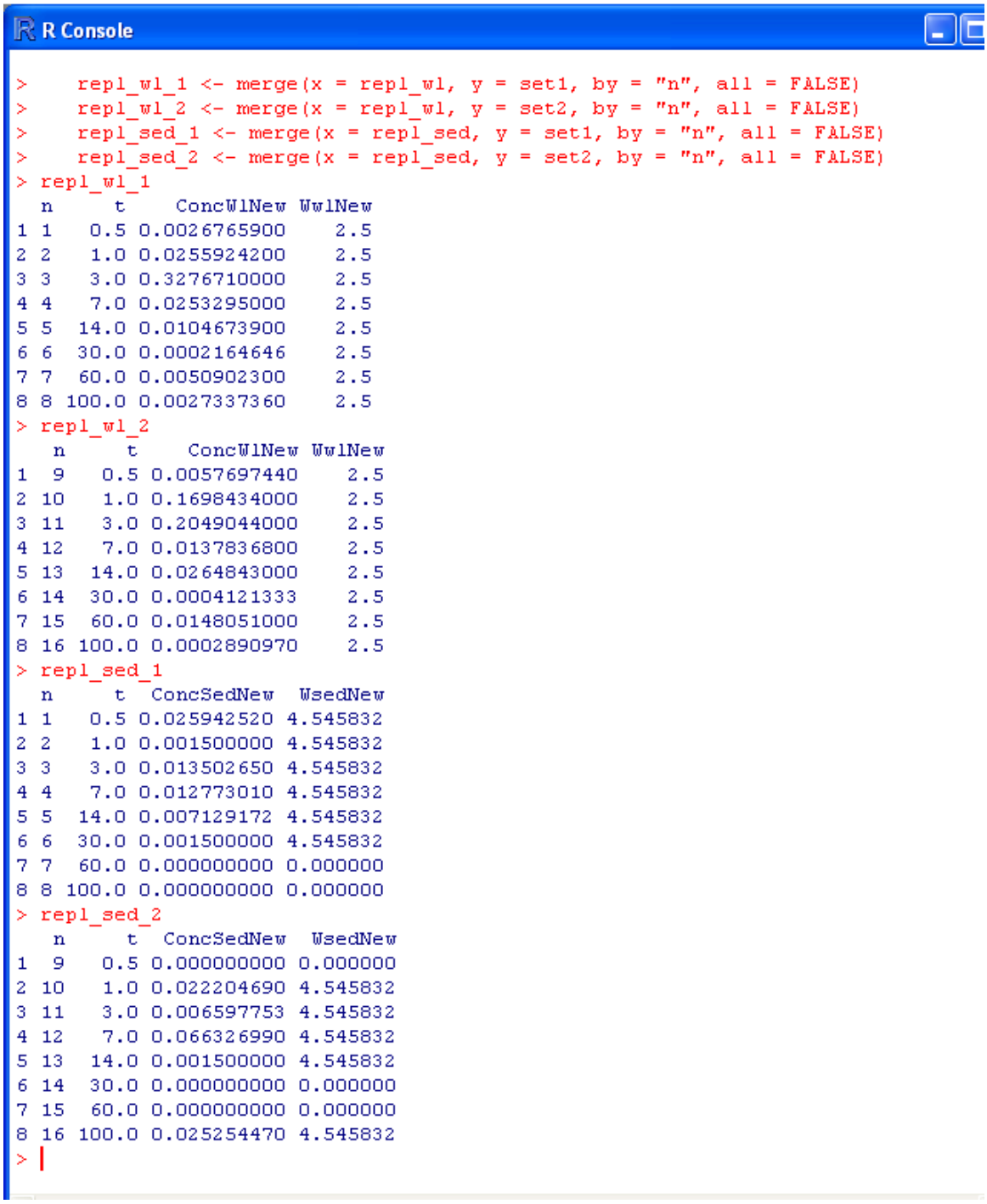


Figure 11. Data frames 'repl_wl_1', 'repl_wl_2', 'repl_sed_1', 'repl_sed_2'.

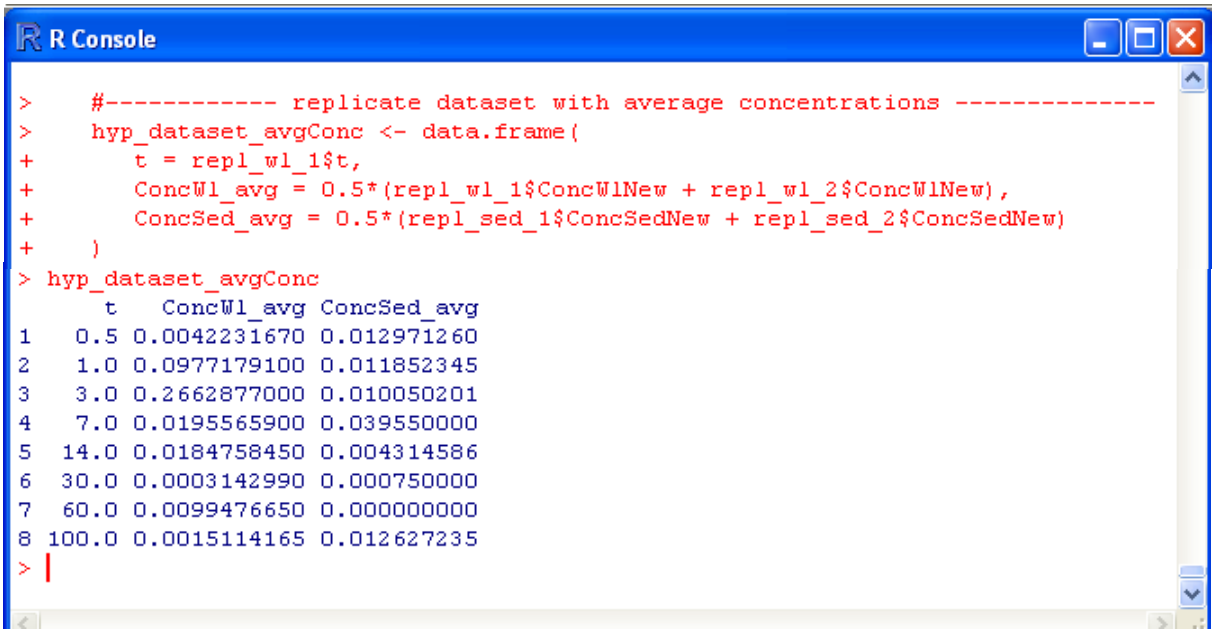


Figure 12 Data frame containing per sampling time the average of the concentrations on that sampling time

```
R Console

> #b. bepalen van tsedmax van hypothetische dataset met average concentrations (tijdstap w
> # replicate set 1
>    maxCsed <- data.frame(
+       t = 1,
+       ConcSed_avg = max(hyp_dataset_avgConc$ConcSed_avg)
+     )
>
>    tmp5 <- merge(x = hyp_dataset_avgConc, y = maxCsed, by = "ConcSed_avg", all = FALSE)
>
>    t100 <- data.frame(
+       t = 100
+     )
>    Ct100 <- merge(x = hyp_dataset_avgConc, y = t100, by = "t", all = FALSE)
>
>    tmp6a <- data.frame(
+       t = tmp5$t.x,
+       ConcSed_avg = tmp5$ConcSed_avg
+    )
> # Let op! het komt voor dat je de max conc in het sediment niet uniek is. Kortom conc
> # in sed neemt niet meer af!. Dan heb je meerdere regels in tmp6a. Dat geeft problemen.
> # neem daarom de min. t uit tmp6.
>    minTtmp6 <- data.frame(
+       t = min(tmp6a$t)
+     )
>    tmp6b <- (merge(x = tmp6a, y = minTtmp6, by = "t", all = FALSE))
>
>    tmp7 <- merge(x=tmp6b, y=hyp_dataset_avgConc, by="t", all = FALSE)
>
>    Ctcsedmax <- data.frame(
+    t =tmp7$t,
+    ConcWl_avg=tmp7$ConcWl_avg,
+    ConcSed_avg=tmp7$ConcSed_avg.x
+    )
> maxCsed
  t ConcSed_avg
1 1     0.03955
> tmp5
  ConcSed_avg t.x ConcWl_avg t.y
1     0.03955   7 0.01955659   1
> Ct100
    t  ConcWl_avg ConcSed_avg
1 100 0.001511416  0.01262724
> tmp6a
  t ConcSed_avg
1 7     0.03955
> tmp6b
  t ConcSed_avg
1 7     0.03955
> tmp7
  t ConcSed_avg.x ConcWl_avg ConcSed_avg.y
1 7       0.03955 0.01955659       0.03955
> Ctcsedmax
  t ConcWl_avg ConcSed_avg
1 7 0.01955659     0.03955
```

Figure 13. Result of determining the sampling time with the maximum concentration of the dataset from the dataset of the average concentration in the sediment (column ConcSedAvg in data frame 'hyp_dataset_avgConc'

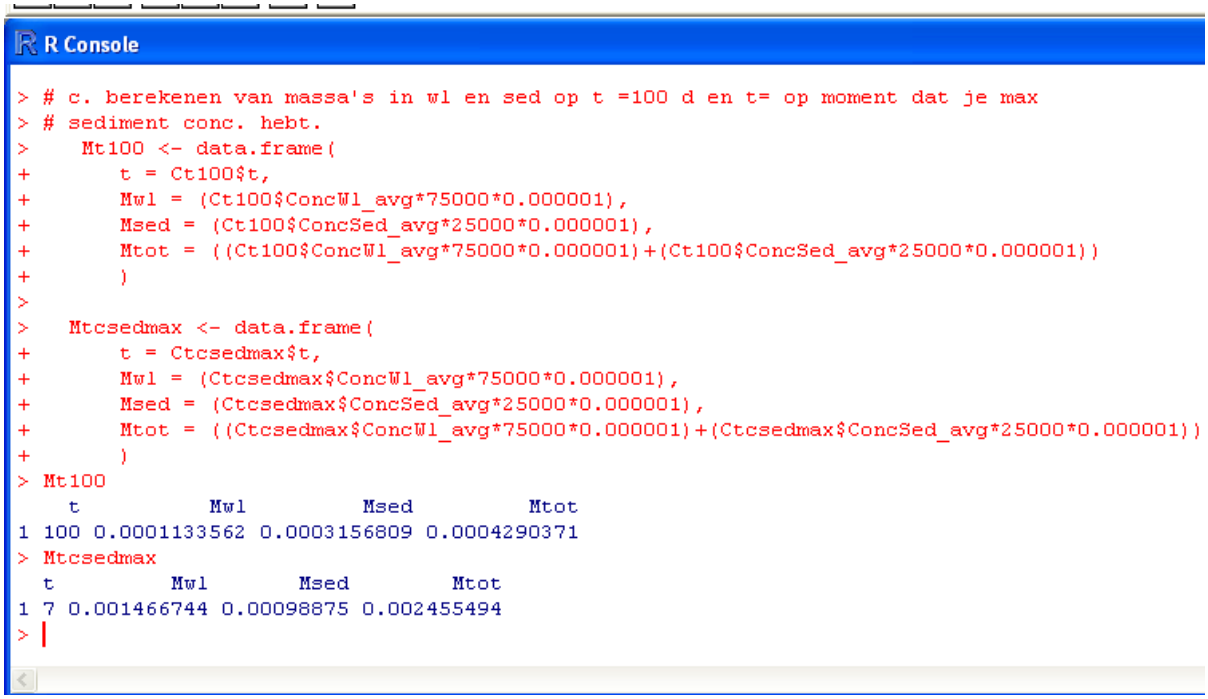


Figure 14. Result of calculating mass in water layer, sediment en total water-sediment system on sampling time t=100 days and the sampling time where the maximum sediment concentration is found.

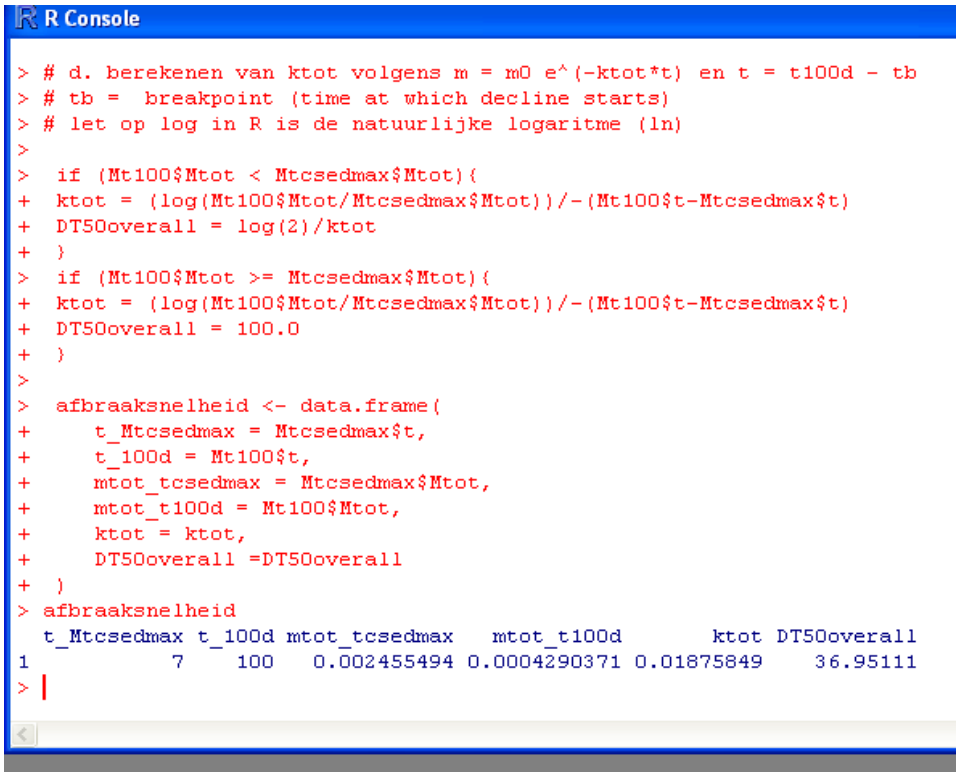


Figure 15. Result of the calculation of the DegT50system/DegT50overall

2.5.3 Making the PEST input files

Figure 10 (part f) shows the code in the R-script for setting the initial parameter values for the fitting procedure. Note that there are 3 sets of initial values. For each set in initial values (k = 3) PEST input files are made and a TOXSWA-PEST run is started.

Figure 16 shows the code in the R-script for reading/defining the template PEST input file and fill in values for all variables (initial parameter values, observations and weight factors) in the template PEST input file (FIT_TOXSWA.pst; see also Annex 3 for contents of this file).

Figure 17 shows the result (list 'pst') of running the part of the code of the R-script shown in Figure 16. Figure 18 shows the FIT_TOXSWA.pst file created by running the code in the R-script shown in Figure 16. Values in the list 'pst' are automatically filled in the template file of FIT_TOXSWA.pst (see Annex 3 for the template file). All items between '<>' in the template file of FIT_TOXSWA.pst are automatically replaced by the values given in the list 'pst'.

```
#--------Pest files aanpassen-------------------

# copy all templates to directory work
files <- list.files(path = "../templates/PEST", full.names = TRUE)
file.copy(from = files, to = ".", overwrite = TRUE)

# read template(s)
Template_tpl <- readLines(con = "400000007.tpl")


# aanpassen van de pst file
# 1. initiele waarden DT50 aanpassen --> moet DT50overall worden.
# omdat pars een list moet zijn (anders kan ik de functie writeInputFile niet gebruiken),
definieer ik
# de DT50wl en DT50sed apart; echter dezelfde waarde nl DT50overall wordt gebruikt
# afspraak --> 4 significate cijfers; is meest makkelijk te bereiken via scientific notation
en 3 digits
# voorbeeld: 6.552e-02 --> 4 significate cijfers (R maakt er overigs nooit 0.655e-03 van)
     pst <- list(
         DT50wl = formatC(ini_dt50wl, format = "f"),
         DT50sed = formatC(ini_dt50sed, format = "f"),
         WwlNew1 = formatC(tmp3$WwlNew[1],digits=3, format = "e"),
         WwlNew2 = formatC(tmp3$WwlNew[2],digits=3, format = "e"),
         .
         .
         WwlNew15 = formatC(tmp3$WwlNew[15],digits=3, format = "e"),
         WwlNew16 = formatC(tmp3$WwlNew[16],digits=3, format = "e"),
         WsedNew1 = formatC(tmp4$WsedNew[1],digits=3, format = "e"),
         WsedNew2 = formatC(tmp4$WsedNew[2],digits=3, format = "e"),
         .
         .
         WsedNew15 = formatC(tmp4$WsedNew[15],digits=3, format = "e"),
         WsedNew16 = formatC(tmp4$WsedNew[16],digits=3, format = "e"),
         obs1 = formatC(tmp3$ConcWlNew[1],digits=3, format = "e"),
         obs2 = formatC(tmp3$ConcWlNew[2],digits=3, format = "e"),
         .
         .
         obs15 = formatC(tmp3$ConcWlNew[15],digits=3, format = "e"),
         obs16 = formatC(tmp3$ConcWlNew[16],digits=3, format = "e"),
         obs17 = formatC(tmp4$ConcSedNew[1],digits=3, format = "e"),
         .
         .
         obs31 = formatC(tmp4$ConcSedNew[15],digits=3, format = "e"),
         obs32 = formatC(tmp4$ConcSedNew[16],digits=3, format = "e")

    )

# read template(s)

Template_pst <- readLines(con = "FIT_TOXSWA.pst")

writeInputFile(pars = pst, template = Template_pst,  con = "FIT_TOXSWA.pst "),
```

List of initial parameter values and values of observations and weight factors.

Read/define template PEST inputfile FIT_TOXSWA.pst (Annex 3)

Values from list 'pst' are filled in in FIT_TOXSWA.pst (between < >) See Annex 3)

Figure 16. Code in R-script for reading/defining the template PEST input file and fill in values for all variables (initial parameter values, observations and weight factors) in the template PEST input file

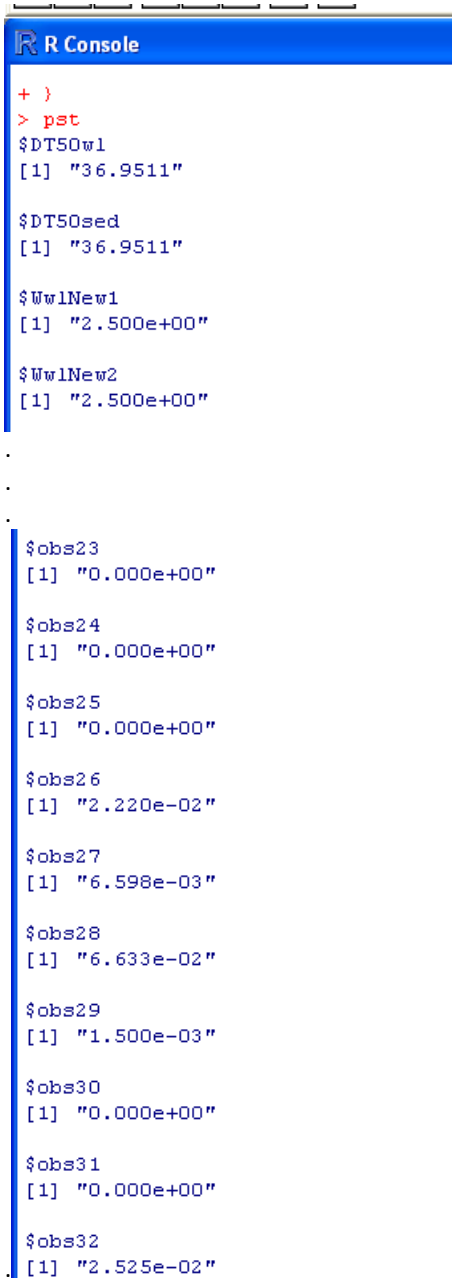
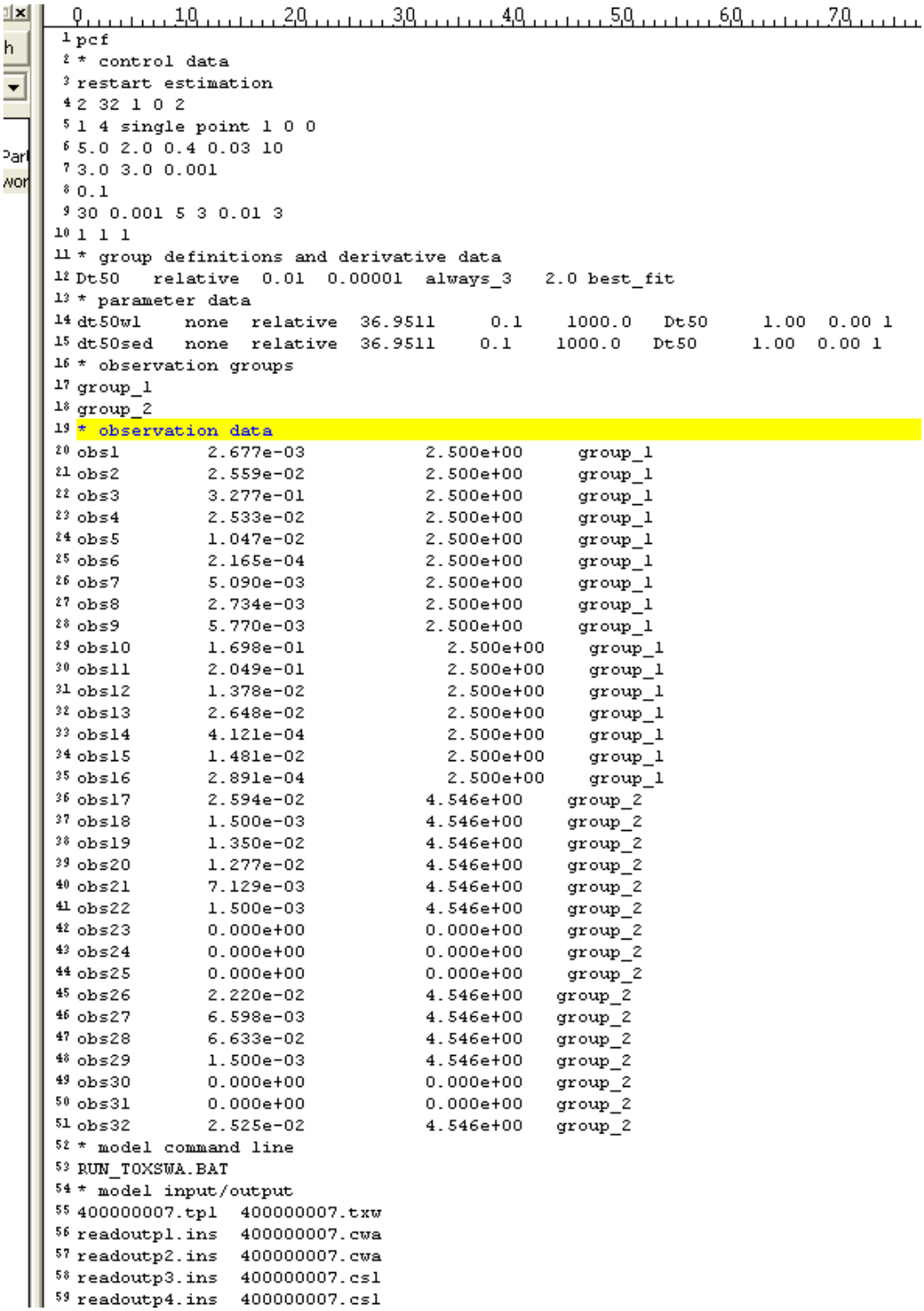


Figure 17. Result (list 'pst') of running the part of the code of the R-script shown in Figure 16 →List of initial parameter values and values of observations and weight factors



```
pcf
* control data
restart estimation
2 32 1 0 2
1 4 single point 1 0 0
5.0 2.0 0.4 0.03 10
3.0 3.0 0.001
0.1
30 0.001 5 3 0.01 3
1 1 1
* group definitions and derivative data
Dt50    relative  0.01  0.00001  always_3   2.0 best_fit
* parameter data
dt50wl     none  relative  36.9511     0.1    1000.0   Dt50     1.00  0.00 1
dt50sed    none  relative  36.9511     0.1    1000.0   Dt50     1.00  0.00 1
* observation groups
group_1
group_2
* observation data
obs1         2.677e-03            2.500e+00     group_1
obs2         2.559e-02            2.500e+00     group_1
obs3         3.277e-01            2.500e+00     group_1
obs4         2.533e-02            2.500e+00     group_1
obs5         1.047e-02            2.500e+00     group_1
obs6         2.165e-04            2.500e+00     group_1
obs7         5.090e-03            2.500e+00     group_1
obs8         2.734e-03            2.500e+00     group_1
obs9         5.770e-03            2.500e+00     group_1
obs10        1.698e-01              2.500e+00    group_1
obs11        2.049e-01              2.500e+00    group_1
obs12        1.378e-02              2.500e+00    group_1
obs13        2.648e-02              2.500e+00    group_1
obs14        4.121e-04              2.500e+00    group_1
obs15        1.481e-02              2.500e+00    group_1
obs16        2.891e-04              2.500e+00    group_1
obs17        2.594e-02            4.546e+00    group_2
obs18        1.500e-03            4.546e+00    group_2
obs19        1.350e-02            4.546e+00    group_2
obs20        1.277e-02            4.546e+00    group_2
obs21        7.129e-03            4.546e+00    group_2
obs22        1.500e-03            4.546e+00    group_2
obs23        0.000e+00            0.000e+00    group_2
obs24        0.000e+00            0.000e+00    group_2
obs25        0.000e+00            0.000e+00    group_2
obs26        2.220e-02            4.546e+00   group_2
obs27        6.598e-03            4.546e+00   group_2
obs28        6.633e-02            4.546e+00   group_2
obs29        1.500e-03            4.546e+00   group_2
obs30        0.000e+00            0.000e+00   group_2
obs31        0.000e+00            0.000e+00   group_2
obs32        2.525e-02            4.546e+00   group_2
* model command line
RUN_TOXSWA.BAT
* model input/output
400000007.tpl  400000007.txw
readoutp1.ins  400000007.cwa
readoutp2.ins  400000007.cwa
readoutp3.ins  400000007.csl
readoutp4.ins  400000007.csl
```

Figure 18. FIT_TOXSWA.pst created by running the code in the R-script shown in Figure 16. Values in the list 'pst' are automatically filled in template file of FIT_TOXSWA.pst (see Annex 3 for template file).

2.5.4 Running PEST

Figure 19 shows the code in the R-script for running PEST. The command for running PEST is 'pest.exe FIT_TOXSWA.pst'. In case of running the R-script on the Alterra Grid PEST should be called by a batch file.

```
# PEST runnen
# PEST runnen vanuit R gaat alleen als je in de batch file RunPest.bat
# Letop ! show.output.on.console altijd op FALSE zetten, want anders werkt het
niet!

system(command = "runpest.bat", wait = TRUE, show.output.on.console = FALSE,
invisible = FALSE)

#   system(command = "pest.exe FIT_TOXSWA.pst", show.output.on.console = FALSE,
invisible = TRUE)
```

Figure 19 Code in R-script for running PEST.

2.5.5 Extracting the needed output from the PEST *.rec output file

Figure 22 shows the code in the R-script for extracting needed information out of PEST output file FIT_TOXSWA.Rec.

- FIT_TOXSWA.rec is given a name: recFile

```
recFile <- file("FIT_TOXSWA.rec", "r")
```

- Content of FIT_TOXSWA.rec is read until the line:

```
"                value             lower limit       upper limit"
```

- Next 5 lines are read (= a table) and put in data frame 'dat' (see Figure 20 and Figure 21

```
dat <- read.table(file = recFile, as.is = TRUE,  header = FALSE, sep = "", quote =
"", dec = ".", nrows = 5, row.names = 1)
```

- Content of hypdat.rec is read until the line:

```
" Objective function ----->"
```

- Read an empty line

```
dummy <- readLines(con = recFile, n = 1, ok = TRUE)
```

- Read the line containing the value of the object function (phi)

```
tmp <- read.table(file = recFile, as.is = TRUE,  header = FALSE, sep = "", quote =
"",dec = ".", nrows = 1, row.names = 1)
```

- Content of hypdat.rec is read until the line:

```
" Eigenvalues ----->"
```

- Read an empty line

```
dummy <- readLines(con = recFile, n = 1, ok = TRUE)
```

- Read the line containing the values of the eigen values

```
tmpE <- read.table(file = recFile, as.is = FALSE, header = FALSE, sep = "", quote
= "", dec = ".", nrows = 1)
```

- Put all information (true, initial and fitted values of parameters, including 95% confidence interval, object function and eigen values of the fit) in a data frame and write the contents of the data frame to a csv file (see Figure 22 for code and Figures 23 and 24 for result).

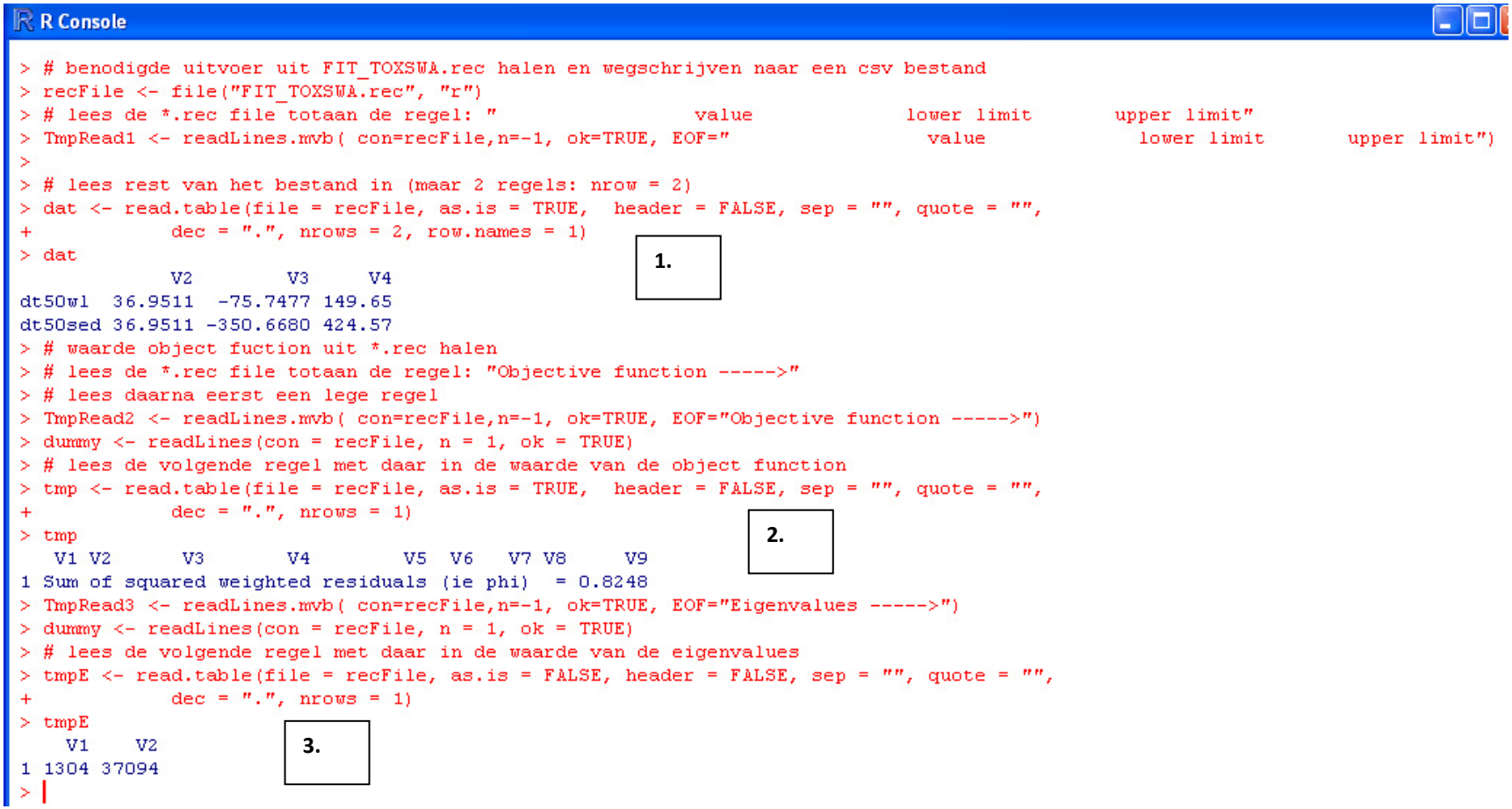
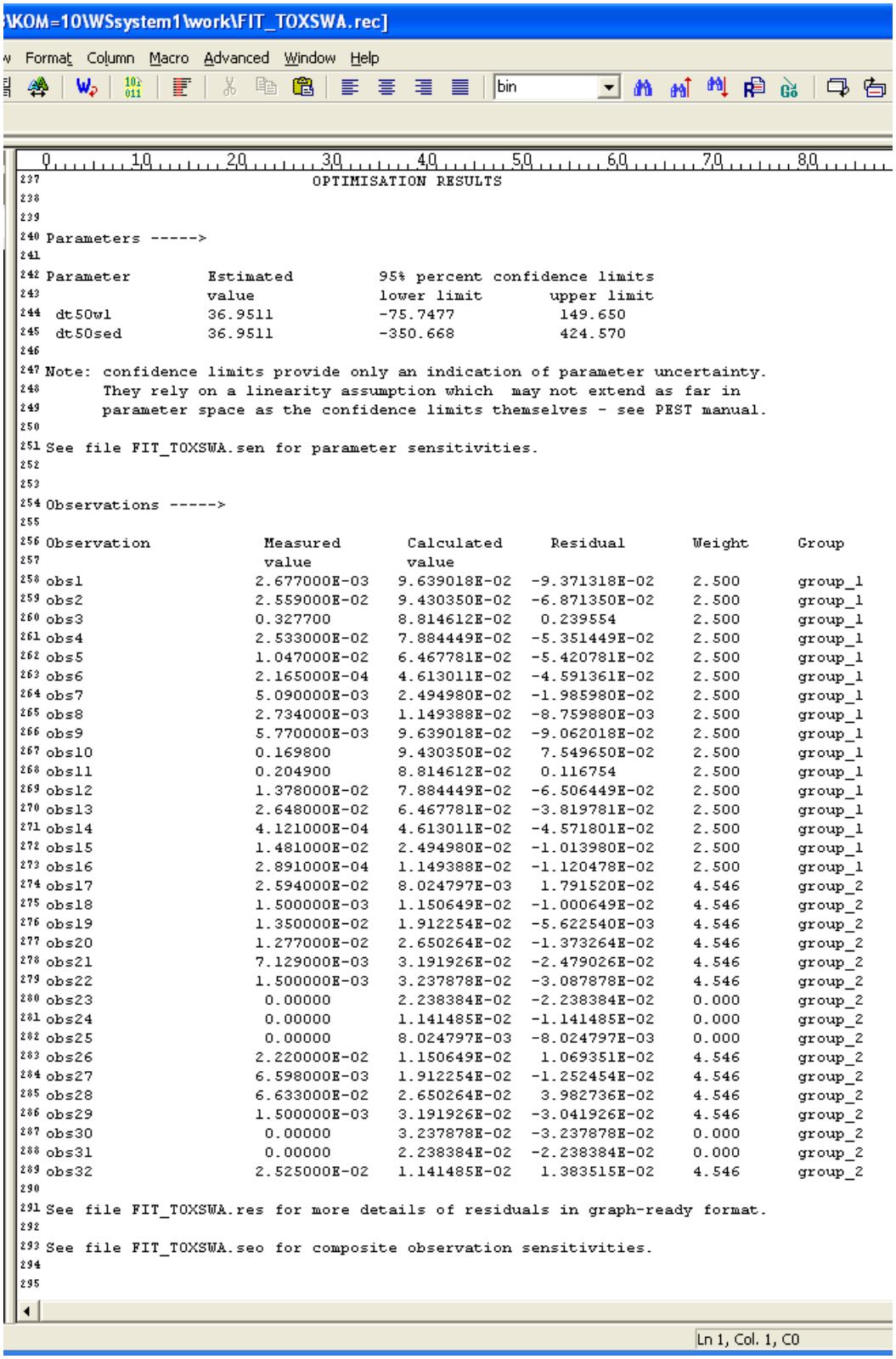


Figure 20. 1. Data frame 'dat' containing op each parameter the fitted value (V2), lower boundary of the 95 percentile confidence interval (V3) and the upper boundary of the 95 percentile confidence interval (V4). 2. Data frame tmp containing the value of the object function (V9). 3. Data frame tmpE containing the values of the eigen values (V1: dt50wl, V2:dt50sed).



```
:\KOM=10\WSsystem1\work\FIT_TOXSWA.rec]
Format  Column  Macro  Advanced  Window  Help
bin
                          OPTIMISATION RESULTS


Parameters ----->

Parameter        Estimated         95% percent confidence limits
                 value             lower limit       upper limit
 dt50wl          36.9511           -75.7477           149.650
 dt50sed         36.9511           -350.668           424.570

Note: confidence limits provide only an indication of parameter uncertainty.
      They rely on a linearity assumption which  may not extend as far in
      parameter space as the confidence limits themselves - see PEST manual.

See file FIT_TOXSWA.sen for parameter sensitivities.


Observations ----->

Observation             Measured       Calculated     Residual       Weight     Group
                        value          value
obs1                   2.677000E-03   9.639018E-02  -9.371318E-02    2.500      group_1
obs2                   2.559000E-02   9.430350E-02  -6.871350E-02    2.500      group_1
obs3                   0.327700       8.814612E-02   0.239554        2.500      group_1
obs4                   2.533000E-02   7.884449E-02  -5.351449E-02    2.500      group_1
obs5                   1.047000E-02   6.467781E-02  -5.420781E-02    2.500      group_1
obs6                   2.165000E-04   4.613011E-02  -4.591361E-02    2.500      group_1
obs7                   5.090000E-03   2.494980E-02  -1.985980E-02    2.500      group_1
obs8                   2.734000E-03   1.149388E-02  -8.759880E-03    2.500      group_1
obs9                   5.770000E-03   9.639018E-02  -9.062018E-02    2.500      group_1
obs10                  0.169800       9.430350E-02   7.549650E-02    2.500      group_1
obs11                  0.204900       8.814612E-02   0.116754        2.500      group_1
obs12                  1.378000E-02   7.884449E-02  -6.506449E-02    2.500      group_1
obs13                  2.648000E-02   6.467781E-02  -3.819781E-02    2.500      group_1
obs14                  4.121000E-04   4.613011E-02  -4.571801E-02    2.500      group_1
obs15                  1.481000E-02   2.494980E-02  -1.013980E-02    2.500      group_1
obs16                  2.891000E-04   1.149388E-02  -1.120478E-02    2.500      group_1
obs17                  2.594000E-02   8.024797E-03   1.791520E-02    4.546      group_2
obs18                  1.500000E-03   1.150649E-02  -1.000649E-02    4.546      group_2
obs19                  1.350000E-02   1.912254E-02  -5.622540E-03    4.546      group_2
obs20                  1.277000E-02   2.650264E-02  -1.373264E-02    4.546      group_2
obs21                  7.129000E-03   3.191926E-02  -2.479026E-02    4.546      group_2
obs22                  1.500000E-03   3.237878E-02  -3.087878E-02    4.546      group_2
obs23                   0.00000       2.238384E-02  -2.238384E-02    0.000      group_2
obs24                   0.00000       1.141485E-02  -1.141485E-02    0.000      group_2
obs25                   0.00000       8.024797E-03  -8.024797E-03    0.000      group_2
obs26                  2.220000E-02   1.150649E-02   1.069351E-02    4.546      group_2
obs27                  6.598000E-03   1.912254E-02  -1.252454E-02    4.546      group_2
obs28                  6.633000E-02   2.650264E-02   3.982736E-02    4.546      group_2
obs29                  1.500000E-03   3.191926E-02  -3.041926E-02    4.546      group_2
obs30                   0.00000       3.237878E-02  -3.237878E-02    0.000      group_2
obs31                   0.00000       2.238384E-02  -2.238384E-02    0.000      group_2
obs32                  2.525000E-02   1.141485E-02   1.383515E-02    4.546      group_2

See file FIT_TOXSWA.res for more details of residuals in graph-ready format.

See file FIT_TOXSWA.seo for composite observation sensitivities.


Ln 1, Col. 1, C0
```

Figure 21A. Part of the hypdat.rec (PEST output file). Note that the fitted values and the 95% confidence limits correspond with the values in Figure 20.

```
296 Objective function ----->
297   Sum of squared weighted residuals (ie phi)                =  0.8248
298   Contribution to phi from observation group "group_1"      =  0.7177
299   Contribution to phi from observation group "group_2"      =  0.1071
300
301 Correlation Coefficient ----->
302   Correlation coefficient                                   =  0.3846
303
304 Analysis of residuals ----->
305   All residuals:-
306      Number of residuals with non-zero weight                        =    27
307      Mean value of non-zero weighted residuals                       = -2.3790E-02
308      Maximum weighted residual [observation "obs3"]                  =  0.5989
309      Minimum weighted residual [observation "obs1"]                  = -0.2343
310      Standard variance of weighted residuals                         =  3.2991E-02
311      Standard error of weighted residuals                            =  0.1816
312      Note: the above variance was obtained by dividing the objective
313      function by the number of system degrees of freedom (ie. number of
314      observations with non-zero weight plus number of prior information
315      articles with non-zero weight minus the number of adjustable parameters.)
316      If the degrees of freedom is negative the divisor becomes
317      the number of observations with non-zero weight plus the number of
318      prior information items with non-zero weight.
319   Residuals for observation group "group_1":-
320      Number of residuals with non-zero weight                        =    16
321      Mean value of non-zero weighted residuals                       = -2.7160E-02
322      Maximum weighted residual [observation "obs3"]                  =  0.5989
323      Minimum weighted residual [observation "obs1"]                  = -0.2343
324      "Variance" of weighted residuals                                =  4.4854E-02
325      "Standard error" of weighted residuals                          =  0.2118
326      Note: the above "variance" was obtained by dividing the sum of squared
327      residuals by the number of items with non-zero weight.
328   Residuals for observation group "group_2":-
329      Number of residuals with non-zero weight                        =    11
330      Mean value of non-zero weighted residuals                       = -1.8888E-02
331      Maximum weighted residual [observation "obs28"]                 =  0.1811
332      Minimum weighted residual [observation "obs22"]                 = -0.1404
333      "Variance" of weighted residuals                                =  9.7385E-03
334      "Standard error" of weighted residuals                          =  9.8684E-02
335      Note: the above "variance" was obtained by dividing the sum of squared
336      residuals by the number of items with non-zero weight.
337
338 Parameter covariance matrix ----->
339                  dt50wl      dt50sed
340 dt50wl           2993.       -7588.
341 dt50sed         -7588.       3.5406E+04
342
343 Parameter correlation coefficient matrix ----->
344                  dt50wl      dt50sed
345 dt50wl           1.000      -0.7372
346 dt50sed        -0.7372        1.000
347
348 Normalized eigenvectors of parameter covariance matrix ----->
349                Vector_1     Vector_2
350 dt50wl          0.9761      -0.2172
351 dt50sed         0.2172       0.9761
352
353 Eigenvalues ----->
354                  1304.       3.7094E+04
```

Ln 328, Col. 1, C0

Figure 21B. Part of the FIT_TOXSWA.rec (PEST output file). Note that the value of the object function and the values of the eigen values correspond with the values in Figure 20.

```
# benodigde uitvoer uit FIT_TOXSWA.rec halen en wegschrijven naar een csv bestand
recFile <- file("FIT_TOXSWA.rec", "r")
# lees de *.rec file totaan de regel: "                    value               lower limit
upper limit"
TmpRead1  <-  readLines.mvb(  con=recFile,n=-1,  ok=TRUE,  EOF="                              value
lower limit       upper limit")

# lees rest van het bestand in (maar 2 regels: nrow = 2)
dat <- read.table(file = recFile, as.is = TRUE,  header = FALSE, sep = "", quote = "",
           dec = ".", nrows = 2, row.names = 1)

# waarde object fuction uit *.rec halen
# lees de *.rec file totaan de regel: "Objective function ----->"
# lees daarna eerst een lege regel
TmpRead2 <- readLines.mvb( con=recFile,n=-1, ok=TRUE, EOF="Objective function ----->")
dummy <- readLines(con = recFile, n = 1, ok = TRUE)
# lees de volgende regel met daar in de waarde van de object function
tmp <- read.table(file = recFile, as.is = TRUE,  header = FALSE, sep = "", quote = "",
           dec = ".", nrows = 1)

TmpRead3 <- readLines.mvb( con=recFile,n=-1, ok=TRUE, EOF="Eigenvalues ----->")
dummy <- readLines(con = recFile, n = 1, ok = TRUE)
# lees de volgende regel met daar in de waarde van de eigenvalues
tmpE <- read.table(file = recFile, as.is = FALSE, header = FALSE, sep = "", quote = "",
           dec = ".", nrows = 1)

close(recFile) # recfile sluiten want je hebt hem niet meer nodig

runnr <- paste("run",formatC(i,width =max(nchar(nrow(key))), flag = "0") , sep="")
drawnr <-  paste("draw",formatC(j,width =max(nchar(nrow(ndraws))), flag = "0") ,sep="")
# stop de data in een dataframe
   recResult <- data.frame(
        nrun= runnr,
        ndraws = drawnr,
        startcat=startcat,
        true_dt50wl = key$dt50wl[i],
        ini_dt50wl = formatC(ini_dt50wl, format = "f"),
        fitted_dt50wl = dat[1,1],
        lower_limit_dt50wl = dat[1,2],
        upper_limit_dt50wl = dat[1,3],
        true_dt50sed = key$dt50sed[i],
        ini_dt50sed = formatC(ini_dt50sed, format = "f"),
        fitted_dt50sed = dat[2,1],
        lower_limit_dt50sed = dat[2,2],
        upper_limit_dt50sed = dat[2,3],
        object_function = tmp$V9,
        EigenVal_dt50wl = tmpE$V1,
        EigenVal_dt50sed = tmpE$V2
    )



   write.table(x = recResult, file = output, append = TRUE, quote = FALSE, row.names = FALSE,
col.names=FALSE, sep=",")

.
.
}
}
}
```

Put all information (true, initial and fitted values of parameters, including 95% confidence interval, object function and eigen values of the fit) in a data frame and write the contents of the data frame to a text file

Close for loops over nrow(key) (number of parameter combinations, ndraw, and k (number of combinations with starting values)

Figure 22. Code in R-script for extracting needed information out of PEST output file FIT_TOXSWA.Rec

R Console

```
> runnr <- paste("run",formatC(i,width =max(nchar(nrow(key))), flag = "0") , sep="")
> drawnr <-  paste("draw",formatC(j,width =max(nchar(nrow(ndraws))), flag = "0") ,sep="")
> # stop de data in een dataframe
>    recResult <- data.frame(
+         nrun= runnr,
+         ndraws = drawnr,
+         startcat=startcat,
+         true_dt50wl = key$dt50wl[i],
+         ini_dt50wl = formatC(ini_dt50wl, format = "f"),
+         fitted_dt50wl = dat[1,1],
+         lower_limit_dt50wl = dat[1,2],
+         upper_limit_dt50wl = dat[1,3],
+         true_dt50sed = key$dt50sed[i],
+         ini_dt50sed = formatC(ini_dt50sed, format = "f"),
+         fitted_dt50sed = dat[2,1],
+         lower_limit_dt50sed = dat[2,2],
+         upper_limit_dt50sed = dat[2,3],
+         object_function = tmp$V9,
+         EigenVal_dt50wl = tmpE$V1,
+         EigenVal_dt50sed = tmpE$V2
+      )
>
>    write.table(x = recResult, file = output, append = TRUE, quote = FALSE, row.names = FALSE, col.names=FALSE, sep=",")
> recResult
    nrun ndraws startcat true_dt50wl ini_dt50wl fitted_dt50wl lower_limit_dt50wl upper_limit_dt50wl true_dt50sed ini_dt50sed
1 run253 draw01        1     50.65387    36.9511       36.9511          -75.7477          149.65     50.65387     36.9511
  fitted_dt50sed lower_limit_dt50sed upper_limit_dt50sed object_function EigenVal_dt50wl EigenVal_dt50sed
1        36.9511             -350.668              424.57          0.8248            1304            37094
```

Figure 23. Result of data frame recResult

RecResult.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	filename	drawnr	startcat	true_dt50w	ini_dt50wl	fitted_dt50	lower_limit	upper_limit	true_dt50s	ini_dt50se	fitted_dt50	lower_limit	upper_limit	phi	EigenVal_	EigenVal_dt50sed	
2	run253	draw01	1	50.65387	36.9511	36.9511	-75.7477	149.65	50.65387	36.9511	36.9511	-350.668	424.57	0.8248	1304	37094	
3																	

*Figure 24. Part of the *.csv file to which the information in data frame recResult is written*

3 Grid computing; batch processing for DegT50 tool

3.1 Step 1

In this step the input files for the PEST runs are prepared (Annex 4). The input files will be generated with TOXSWA for all defined runs for sediment and the water layer as well. The files will be numbered starting with 1. The numbering has a fixed format, so the first numbers are proceeded with zeros depending on the total run numbers. This means that for 400 run numbers the first number will be run001.csv. The files will be stored in a shared folder so the application can pick up the right file. The structure of the shared folder looks like:

KOM1
    WSsystem1
        Sediment
            Run001.csv
            …
            Run400.csv
        Waterlayer
            Run001.csv
            …
            Run400.csv
KOM10
    Etc.

3.2 Step 2

In this step the application for a run with PEST and TOXSWA is described. This application directory will be zipped for processing on the grid.

The file structure for the runs is shown below.
<Root>
    \ bin
    \ errors
        \ results \ DT50overall
        \ hypothetical_observations\sediment\outputLognormal
        \ hypothetical_observations\sediment\outputTOXSWA

        \ hypothetical_observations\waterlayer\outputLognormal
        \ hypothetical_observations\waterlayer\outputTOXSWA

        \ PEST

    \ templates \ PEST
    \work

The output files of TOXSWA will be generated in the root folder.
Executables are located in the bin directory.
Input files for PEST en TOXSWA are located in results\TOXSWA directory.

3.3 Step 3

The final step is the actual run on several grid computers. For this process a script for the grid software CONDOR is written to send all the essential files to the grid computers en get the final results back (Annex 4). The input file are shipped with a zip file and will be unzipped on the grid computers. After the run is completed the output files are zipped again and send back to the submit computer (Annex 5). Each zip files is numbered with a unique run number. The runs are executed with some modifications to the original Part_2.R script (Annex 6).

3.4 Step 4

In this step (Annex 7) the result files are processed to get a summary file of all the result files of the individual result files. This script search all files the numbered zip files from the grid computer and extract these to a new output location. The output files in the zip files have a numbered file like RecResultXX.csv (XX = run number), see Figure 25. After that all result files of PEST (located in the PEST subfolders) are put together to one result file: 'RecResultSummary.csv' The individual results for each draw and start category are located in 'startcat1', 'startcat 2' and 'startcat 3'.

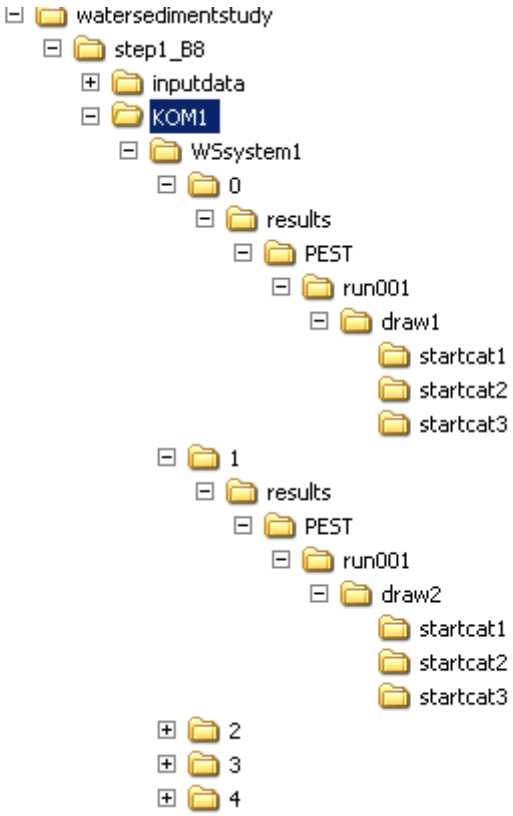


Figure 25. Folder structure used for storing the output files

Literature

Adriaanse, P.I., 1996. Fate of pesticides in field ditches: the TOXSWA simulation model. Report 90, DLO Winand Staring Centre for Integrated Land, Soil and Water Research, Wageningen, The Netherlands.

Adriaanse, P.I., Leistra, M., Vink, J.P.M., Brouwer, W.W.M., Tas, J.W., Linders, J.B.H.J., Pol, J.W. 2002. Estimating transformation rates of pesticides, to be used in the TOXSWA model, from water-sediment studies. Alterra-report 023, Alterra, Green World Research, Wageningen, The Netherlands.

Beltman, W.H.J., M.M.S. Ter Horst, P.I. Adriaanse, A. De Jong, 2006. Manual of FOCUS_TOXSWA v2.2.1. Alterra-rapport 586. Wageningen, the Netherlands

Doherty, J., 2005. PEST. Model-Independent Parameter Estimation, user Manual, 5th edition. Watermark Numerical Computing, www.sspa.com/pest, version 9.01.

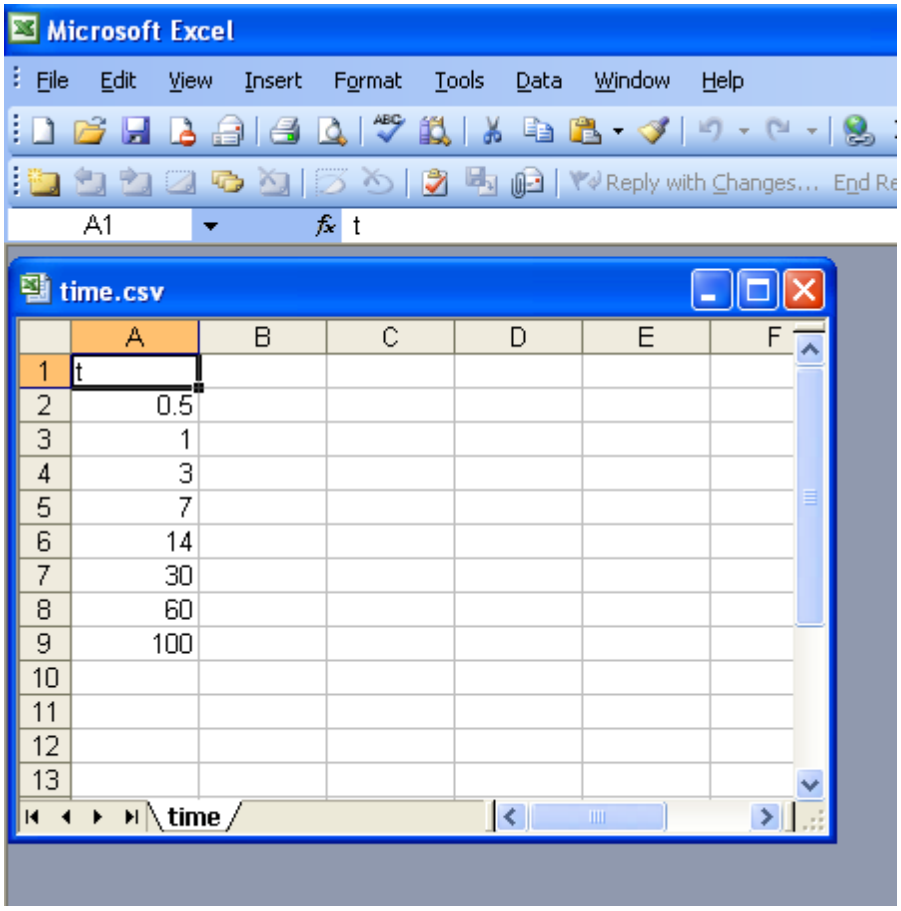
FOCUS, 2006. Guidance document on estimating persistence and degradation kinetics from environmental fate studies on pesticides in EU registration. DG SanCo/10058/version 2.0, June 2006.

http://www.pesthomepage.org/Downloads.php

http://www.r-project.org/

http://www.cs.wisc.edu/condor/

Annex 1 Contents of time.csv



Annex 2 Lognormal program in FORTRAN

Water layer

```
! Mechteld ter Horst
! Goal: Bepalen van random waarden voor c*wat en c*sed,ave op basis van
! een lognormale distributie. Nodig voor het 'water-sediment paper'
! De functies uit de volgende module van Jos Boesten zijn in dit programma
! opgenomen: 2008.09.20 module random numbers.f90
! function lognormal_av(average,cv) wordt gebruikt.
! input: average = c*wat of c*sed of t(i) = uitvoer uit TOXSWA
! CV = 5%; hier als fractie dus 0.05 (verantwoording zie paper).
! 2 versies: 1 voor de waterlaag en 1 voor het sediment (vanwege verschil
! in LOD waarde)
! version 9 Oktober 2009
!
! Let op  ongetransformeerde c*wat / c*sed,ave en CV
! gebruiken als invoer (dus niet de log van...).
! Dit programma is alleen voor de waterlaag!

PROGRAM LogNormal_WS_waterlayer

use random_numbers

implicit none

! declaraties
real    :: time, cv, weegfac,cwatini
real    :: cTOXSWA,LODwat,cTOXSWAprev, HypObsPrev, nobs
integer :: nmetingen, i,nreplicates, j, k, obstot
real, allocatable :: HypObs(:), tijd(:)
nmetingen = 8
nreplicates = 2
obstot = nmetingen*nreplicates
! cwatini in g/m3 (voor weegfactor)
cwatini = 100E-3
! LODwat = 0.1 microgram/L
! LODwat = 0.0001 g/m3
LODwat = 1.0E-4

! open de benodigde input/output files
OPEN(11,FILE='inputwl.txt',status='unknown')
OPEN(12,FILE='cv_wat.txt',status='unknown')
OPEN(13,FILE='outputwl.txt',status='unknown')

! lees header van de 2 inputfiles
read(11,*)
read(12,*)
! schrijf header van de uitvoer file
write(13, '(a)')   '     t      c_hyp-obs           weegfactor'


read (12,*) cv

! lees regel voor regel input.txt en bepaal per regel de random gegenereerde
! concentratie via trekken uit een log normale verdeling
! gebruik de functie lognormal_av(average,cv)
! inlezen ongeformatteerd doen, want format Conc is niet overal gelijk.
! voor dit paper 8 metingen dus 8x inlezen, lognormal berekenen en wegschrijven
! eerst aantal observaties (=metingen - metingen waar c=0) bepalen
! is nodig voor het berekenen van de weegfactor
! Waarden HypObs in array trappen en zo onthouden

allocate(HypObs(obstot))
allocate(tijd(obstot))
k=0
```

```
nobs = 0
do j = 1, nreplicates
   close(11)
   OPEN(11,FILE='inputwl.txt',status='unknown')
   ! lees header van input.txt
   read(11,*)
   do i = 1, nmetingen
      k = k+1
      read (11, *) time, cTOXSWA
      tijd(k) = time
! als cTOXSWA is nul dan HypObs = 0
         if (cTOXSWA .lt. 1.0E-12) then
            HypObs(k) = 0.0
         else
            HypObs(k) = lognormal_av(cTOXSWA,cv)
         end if

      if (i .eq. 1) then
         HypObsPrev = HypObs(k)
      end if

      if (HypObs(k) .lt. LODwat .and. HypObsPrev .gt. LODwat) then
         HypObs(k) = 0.5*LODwat
                nobs = nobs + 1
      else if (HypObs(k) .lt. LODwat .and. HypObsPrev .lt. LODwat) then
             HypObs(k) = 0.0
                nobs = nobs
         else
            nobs = nobs + 1
      end if
         HypObsPrev = HypObs(k)
   end do
end do

! bepalen weegfactor en wegschrijven
! let op TOXSWA schrijft uitvoer 7 significant weg.
! hypothetische metingen moet niet nauwkeuriger zijn dan
! TOXSWA uitvoer!
k=0
do j = 1, nreplicates
   close(11)
   do i = 1, nmetingen
      k = k+1
         if (i .eq. 1) then
         HypObsPrev = HypObs(k)
      end if
      if (HypObs(k) .lt. LODwat .and. HypObsPrev .gt. LODwat) then
         weegfac = 1.0/((sqrt(nobs))*cwatini)
      else if (HypObs(k) .lt. LODwat .and. HypObsPrev .lt. LODwat) then
                weegfac = 0.0
         else
            weegfac = 1.0/((sqrt(nobs))*cwatini)
      end if
! let op TOXSWA schrijft uitvoer 7 significant weg.
! hypothetische metingen moet niet nauwkeuriger zijn dan
! TOXSWA uitvoer!

   write (13, '(f6.1, 2e20.7)') tijd(k), HypObs(k), weegfac

   HypObsPrev = HypObs(k)
   end do
end do
close(11)
close(12)
close(13)

end program LogNormal_WS_waterlayer
```

Sediment

```
! Mechteld ter Horst
! Goal: Bepalen van random waarden voor c*wat en c*sed,ave op basis van
! een lognormale distributie. Nodig voor het 'water-sediment paper'
! De functies uit de volgende module van Jos Boesten zijn in dit programma
! opgenomen: 2008.09.20 module random numbers.f90
! function lognormal_av(average,cv) wordt gebruikt.
! input: average = c*wat of c*sed of t(i) = uitvoer uit TOXSWA
! CV = 5%; hier als fractie dus 0.05 (verantwoording zie paper).
! 2 versies: 1 voor de waterlaag en 1 voor het sediment (vanwege verschil
! in LOD waarde)
! version 9 Oktober 2009
!
! Let op  ongetransformeerde c*wat / c*sed,ave en CV
! gebruiken als invoer (dus niet de log van...).
! Dit programma is alleen voor het sediment!

PROGRAM LogNormal_WS_sediment

use random_numbers

implicit none

! declaraties
real    :: time, cv, weegfac, csedHypmax, nobs
real    :: cTOXSWA,LODsed,cTOXSWAPrev, HypObsPrev
integer :: nmetingen, i, j, nreplicates,obstot, k
real, allocatable :: HypObs(:), tijd(:)

nmetingen = 8
nreplicates = 2
obstot = nmetingen*nreplicates
! LODsed =3 microgram/L = 3E-3 g/m3
LODsed = 3.0E-3

! open de benodigde input/output files
OPEN(11,FILE='inputsed.txt',status='unknown')
OPEN(12,FILE='cv_sed.txt',status='unknown')
OPEN(13,FILE='outputsed.txt',status='unknown')

! lees header van de 2 inputfiles
read(11,*)
read(12,*)
! schrijf header van de uitvoer file
write(13, '(a)')   '     t      c_hyp-obs           weegfactor'

read (12,*) cv

! lees regel voor regel input.txt en bepaal per regel de random gegenereerde
! concentratie via trekken uit een log normale verdeling
! gebruik de functie lognormal_av(average,cv)
! inlezen ongeformatteerd doen, want format Conc is niet overal gelijk.
! voor dit paper 8 metingen dus 8x inlezen, lognormal berekenen en wegschrijven

! eerst aantal observaties (=metingen - metingen waar c=0) bepalen
! is nodig voor het berekenen van de weegfactor
! Waarden HypObs in array trappen en zo onthouden

allocate(HypObs(obstot))
allocate(tijd(obstot))
k=0
nobs=0
do j = 1, nreplicates
   close(11)
   OPEN(11,FILE='inputsed.txt',status='unknown')
   ! lees header van input.txt
   read(11,*)
   do i = 1, nmetingen
      k = k+1
      read (11, *) time, cTOXSWA
```

```
      tijd(k) = time
! als cTOXSWA is nul dan HypObs = 0
         if (cTOXSWA .lt. 1.0E-12) then
            HypObs(k) = 0.0
         else
            HypObs(k) = lognormal_av(cTOXSWA,cv)
         end if

      if (i .eq. 1) then
         HypObsPrev = HypObs(k)
      end if

      if (HypObs(k) .lt. LODsed .and. HypObsPrev .gt. LODsed) then
         HypObs(k) = 0.5*LODsed
                nobs = nobs + 1
      else if (HypObs(k) .lt. LODsed .and. HypObsPrev .lt. LODsed) then
             HypObs(k) = 0.0
                 nobs = nobs
         else
            nobs = nobs + 1
      end if
          HypObsPrev = HypObs(k)
   end do
end do

! bepalen van csedmax
k=0
csedHypmax = 0.0
do j = 1, nreplicates
   do i = 1, nmetingen
      k = k+1
      if (HypObs(k) .gt. csedHypmax) then
         csedHypmax = HypObs(k)
      end if
   end do
end do

! bepalen weegfactor en wegschrijven
! let op TOXSWA schrijft uitvoer 7 significant weg.
! hypothetische metingen moet niet nauwkeuriger zijn dan
! TOXSWA uitvoer!

k = 0
do j = 1, nreplicates
   do i = 1, nmetingen
    k = k+1
      if (i .eq. 1) then
         HypObsPrev = HypObs(k)
      end if

      if (HypObs(k) .lt. LODsed .and. HypObsPrev .gt. LODsed) then
         weegfac = 1.0/((sqrt(nobs))*csedHypmax)
      else if (HypObs(k) .lt. LODsed .and. HypObsPrev .lt. LODsed) then
         weegfac = 0.0
      else
         weegfac = 1.0/((sqrt(nobs))*csedHypmax)
      end if
! let op TOXSWA schrijft uitvoer 7 significant weg.
! hypothetische metingen moet niet nauwkeuriger zijn dan
! TOXSWA uitvoer!
   write (13, '(f6.1, 2e20.7)') tijd(k), HypObs(k), weegfac

   HypObsPrev = HypObs(k)
   end do
end do
close(11)
close(12)
close(13)

end program LogNormal_WS_sediment
```

module random numbers

```
   module random_numbers
   ! version of 20 Sep 2008

   contains

!=======================================================================

   real function normal(average,sd)
      ! purpose: generate random numbers of normal distribution
      ! version: 1.3
      ! date:    21 Sep 2008
      !
      ! tested against analytical cpdf 26 August 2008
      implicit none
      real, intent(in) :: average,sd
      integer idum
      logical :: first=.true.
      if (first) then
         idum=seedgenerator()
            first=.false.
      endif
      normal =  average + sd * gasdev(idum)
   end function

!=======================================================================

   real function normal_cv(average,cv)
      ! purpose: generate random numbers of normal distribution
      ! version: 1.3
      ! date:    21 Sep 2008
      !
      ! note that cv is not a percentage but a fraction of average
      !
      implicit none
      real, intent(in) :: average,cv
      integer idum
      logical :: first=.true.
      if (first) then
         idum=seedgenerator()
            first=.false.
      endif
      normal_cv =  average *  (  1. + cv * gasdev(idum)  )
   end function

!=======================================================================

   real function lognormal(median,sd)
      ! purpose: generate random numbers of a lognormal distribution
      ! version: 1.4
      ! date: 21 Sep 2008
      ! median  = median of lognormal distribution (so with normal unit)
      ! sd      = dimensionless standard deviation of lognormal distribution
      ! tested against the analytical expression of the cpdf on 13 July 2008;
      ! it is somewhat inconsistent to require the untransformed median as input
      implicit none
      real, intent(in) :: median, sd
      real ln
      integer idum
      logical :: first=.true.
      if (first) then
         idum=seedgenerator()
            first=.false.
      endif
      ln=log(median)+sd*gasdev(idum)
      lognormal=exp(ln)
   end function
```

```
!=========================================================================

   real function lognormal_av(average,cv)
      ! purpose: generate random numbers of a lognormal distribution based on
      !          average and CV of untransformed variable
      ! version: 1.3
      ! date: 21 Sep 2008
      ! average  = average of lognormal distribution (in normal units)
      ! cv         = cv of lognormal distribution (dimensionless, expressed as a
fraction of
      !            the average
      implicit none
      real, intent(in) :: average, cv
      real x, median, sd
      x=1+cv**2
      ! the standard equation for median is logarithm of the expression in next
line;
      ! here we need however, untransformed median because function 'lognormal'
needs
      ! the untransformed median as input
      ! tested on 20 August 2008: generated correct average for cv=1.1
      median=average/sqrt(x)
      sd=sqrt(log(x))
      lognormal_av=lognormal(median,sd)
   end function

!=========================================================================

   real function uni(mini,maxi)
      real mini,maxi
      integer idum
      logical :: first=.true.
      if (first) then
         idum=seedgenerator()
             first=.false.
      endif
      uni = mini + (maxi-mini) * ran1(idum)
   end function

!=========================================================================

   integer function seedgenerator()
      implicit none
      integer ihour,imin,isec,i100th
      call gettim(ihour,imin,isec,i100th)
      seedgenerator=-1.*(ihour+1)*(imin+1)*(isec+1)*(i100th+1)-3457.
   end function

!=========================================================================

      real FUNCTION GASDEV(IDUM)
      DATA ISET/0/
      IF (ISET.EQ.0) THEN
1       V1=2.*RAN1(IDUM)-1.
        V2=2.*RAN1(IDUM)-1.
        R=V1**2+V2**2
        IF(R.GE.1.)GO TO 1
        FAC=SQRT(-2.*LOG(R)/R)
        GSET=V1*FAC
        GASDEV=V2*FAC
        ISET=1
      ELSE
        GASDEV=GSET
        ISET=0
      ENDIF
      RETURN
      END function

!=========================================================================

      real FUNCTION RAN1(IDUM)
```

```
      DIMENSION R(97)
      PARAMETER (M1=259200,IA1=7141,IC1=54773,RM1=3.8580247E-6)
      PARAMETER (M2=134456,IA2=8121,IC2=28411,RM2=7.4373773E-6)
      PARAMETER (M3=243000,IA3=4561,IC3=51349)
      DATA IFF /0/
      IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
        IFF=1
        IX1=MOD(IC1-IDUM,M1)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX2=MOD(IX1,M2)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX3=MOD(IX1,M3)
        DO 11 J=1,97
          IX1=MOD(IA1*IX1+IC1,M1)
          IX2=MOD(IA2*IX2+IC2,M2)
          R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
11      CONTINUE
        IDUM=1
      ENDIF
      IX1=MOD(IA1*IX1+IC1,M1)
      IX2=MOD(IA2*IX2+IC2,M2)
      IX3=MOD(IA3*IX3+IC3,M3)
      J=1+(97*IX3)/M3
      IF(J.GT.97.OR.J.LT.1)PAUSE
      RAN1=R(J)
      R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
      RETURN
      END function
```

end module

Annex 3 Template of Fit_TOXSWA.pst

Items between '< >' are substituted by numbers (of initial values, hypothetical observations or weight factors) by the R-script.

```
pcf
* control data
restart estimation
2 32 1 0 2
1 4 single point 1 0 0
5.0 2.0 0.4 0.03 10
3.0 3.0 0.001
0.1
30 0.001 5 3 0.01 3
1 1 1
* group definitions and derivative data
Dt50   relative  0.01  0.00001  always_3   2.0 best_fit
* parameter data
dt50wl    none  relative  <DT50wl>     0.1    1000.0   Dt50     1.00  0.00 1
dt50sed   none  relative  <DT50sed>    0.1    1000.0   Dt50     1.00  0.00 1
* observation groups
group_1
group_2
* observation data
obs1        <obs1>            <WwlNew1>     group_1
obs2        <obs2>            <WwlNew2>     group_1
obs3        <obs3>            <WwlNew3>     group_1
obs4        <obs4>            <WwlNew4>     group_1
obs5        <obs5>            <WwlNew5>     group_1
obs6        <obs6>            <WwlNew6>     group_1
obs7        <obs7>            <WwlNew7>     group_1
obs8        <obs8>            <WwlNew8>     group_1
obs9        <obs9>           <WwlNew9>     group_1
obs10       <obs10>           <WwlNew10>    group_1
obs11       <obs11>           <WwlNew11>    group_1
obs12       <obs12>           <WwlNew12>    group_1
obs13       <obs13>           <WwlNew13>    group_1
obs14       <obs14>           <WwlNew14>    group_1
obs15       <obs15>           <WwlNew15>    group_1
obs16       <obs16>           <WwlNew16>    group_1
obs17       <obs17>           <WsedNew1>    group_2
obs18       <obs18>           <WsedNew2>    group_2
obs19       <obs19>           <WsedNew3>    group_2
obs20       <obs20>           <WsedNew4>    group_2
obs21       <obs21>           <WsedNew5>    group_2
obs22       <obs22>           <WsedNew6>    group_2
obs23       <obs23>           <WsedNew7>    group_2
obs24       <obs24>           <WsedNew8>    group_2
obs25       <obs25>           <WsedNew9>    group_2
obs26       <obs26>           <WsedNew10>   group_2
obs27       <obs27>           <WsedNew11>   group_2
obs28       <obs28>           <WsedNew12>   group_2
obs29       <obs29>           <WsedNew13>   group_2
obs30       <obs30>           <WsedNew14>   group_2
obs31       <obs31>           <WsedNew15>   group_2
obs32       <obs32>           <WsedNew16>   group_2
* model command line
RUN_TOXSWA.BAT
* model input/output
400000007.tpl  400000007.txw
readoutp1.ins  400000007.cwa
readoutp2.ins  400000007.cwa
readoutp3.ins  400000007.cs1
readoutp4.ins  400000007.cs1
* prior information
```

Annex 4 Submitting a job with _rstart.sub

(calls the command file _rstart.cmd)

```
# RScript.sub

# submitfile for distributed gridcomputing by Condor

executable = Rstart.cmd

universe = vanilla
output = RStart.out
#output = RStart$(Process).out
error = RStart$(Process).error
log = RStart$(Process).log

# werkt vanaf Condor versie 7.3
#cron_hour = 0-5,19-23

should_transfer_files = YES
when_to_transfer_output = ON_EXIT_OR_EVICT


requirements = (Arch == "INTEL" && OpSys == "WINNT50") || \
               (Arch == "INTEL" && OpSys == "WINNT51") || \
               (Arch == "INTEL" && OpSys == "WINNT52") && \
               (Disk >= 20000) && \
               ((Memory * 1024) >= ImageSize) && \
               (HasFileTransfer)


rank = MIPS

arguments = $(Process), KOM1, WSSystem1
# Tranfer the Inputfiles

transfer_input_files   =   ..\..\..\zip.exe,   ..\..\..\unzip.exe,
..\Part_2.R, ..\inversemodellingKOM1WSSystem1.zip

# Total Runnumbers
queue 10000
```

Annex 5 Command file _rstart.cmd

```
Paramaters coming from _rstart.sub (appendix 1):
%1 = JobNr
%2 = KOM1
%3 = WSsystem1

rem Start.cmd

rem Start script for distributed computing

rem Start argument is the RunNumber processed through the Condor
process

rem set share open for condor
net use \\scomp1081\condor_job$ /user:anonymous

rem Logging the computername of this run
rem set computername >%computername%_%1.log

rem Unzip the RunFiles

unzip -o inversemodelling%2%3.zip

dir *.* /s

rem opstarten Rscript
rem par1=runnr, par2=KOMdir (KOM1), par3 = WSdir (WS1)
"\\scomp1081\condor_job$\R\R-2.9.1\bin\Rscript" --no-restore --no-
save Part_2.R %1 %2 %3

rem dir *.* /s > dir.log

rem Zip all files for this process with a unique Filename
zip  -r RBatchOutput_%1.zip  results\*.* -x *.exe

rem remove garbage from the root so that is not being copied to
the submitter
del *.ins *.exe FIT_TOXSWA.* 200000007.* *.inp *.met *.txt /q
```

Annex 6 Modified R-script (in bold) for grid computing

Note that the entire R-script is not given. All code indicated in bold is added to the R-script, so this script was used for grid computing.

```
## load packages
library(mvbutils) # nodig voor readLines.mvb


# set work directory
#setwd("H:/WS/step0_Inverse_modelling/KOM=1/WSsystem1/work")
setwd("work")

## Input parameter: RunNumber [char]

## load packages
library(mvbutils) # nodig voor readLines.mvb

## arguments = number of inputfiles
Args = commandArgs(trailingOnly = TRUE)

#Teller als parameter voor besturing programma
TxtNr <- Args[1]
#Directory voor lezen van juiste files per Run
FDir  <- Args[2]
#Directory voor lezen van WS1 of SW2
FWSDir  <- Args[3]

#Jobnr kan hier worden verhoogd
#Standaard begint Jobnr met 0 dus eerste runnr = jobnr + 1
Teller <- as.numeric(TxtNr)+1

#inlezen file met Teller en DrawNr
JobTeller <- read.csv(file="//scomp1081/condor_job$/groen033/JobTeller400-25.csv")

# i als RunNr, j = DrawNr
i <- JobTeller$RunNr[Teller]
j <- JobTeller$DrawNr[Teller]

# standaard functie: write parameters in `pars' to a connection
writeInputFile <-
function(pars, template, con)
{
.
.
.
run <- data.frame(filename = 1:nrow(tmp), tmp)
run$filename      <-      paste("run",      formatC(run$filename,      width      =
max(nchar(run$filename)), flag = "0"), ".txt", sep = "")

# Tel aantal scenarios
ZeroCount <- nrow(tmp)

# Pas lengte runnummer aan aan het max. aantal runnummers: max<100 = 01 .. 99
if ( ZeroCount < 10 )
FNaam <- paste("run", i, ".csv", sep="") else
        if ( ZeroCount < 100 )
                if (i < 10)
                        FNaam <- paste("run0", i, ".csv", sep="") else
                        FNaam <- paste("run", i, ".csv", sep="") else
        if ( ZeroCount < 1000 )
                if (i < 10)
                        FNaam <- paste("run00", i, ".csv", sep="") else
                        if (i < 100)
                                FNaam <- paste("run0", i, ".csv", sep="") else
```

```
                        FNaam <- paste("run", i, ".csv", sep="") else
        if ( ZeroCount < 10000 )
                if (i < 10)
                        FNaam <- paste("run000", i, ".csv", sep="") else
                        if (i < 100)
                                FNaam <- paste("run00", i, ".csv", sep="") else
                                if (i < 1000)
                                        FNaam  <-  paste("run0",  i,  ".csv",  sep="")
else
                                        FNaam <- paste("run", i, ".csv", sep="")


print (FNaam)


#------------TOXSWA berekeningen zijn al gedaan in Part_1-------------------------

# read time of 'measurements'
time <- read.csv(file = "../time.csv", as.is = TRUE)

.
.
.

# specificeer het aantal trekkingen op zo'n manier dat je ndraw entries krijgt
Ndraws = 25
ndraws <- data.frame(filenameNdraw = 1:25)
ndraws$filenameNdraw   <-   paste("draw",   formatC(ndraws$filenameNdraw,   width   =
max(nchar(ndraws$filenameNdraw)), flag = "0"), sep = "")

#voor gridcomputing uitgeschakeld; i wordt ingelezen
#for(i in 1:nrow(key)){

# bepalen van de hypothetical data set with the true concentrations
#-------------waterlaag -----------------------------
# 1. mergen van uitvoer toxswa met time.csv
#          result1   <-   read.csv(file   =   paste("../results/TOXSWA/waterlayer/",
key$filename[i], sep = ""), as.is = TRUE)
# voor gridcomputing lezen vanaf share
result1 <- read.csv(file = paste("//scomp1081/condor_job$/groen033/", FDir, "/",
FWSDir, "/waterlayer/", FNaam, sep = ""), as.is = TRUE)

    tmp1 <- merge(x = time, y = result1, by = "t", all = FALSE)
.
.
.
#------------sediment-----------------------------------------
# 1. mergen van uitvoer toxswa met time.csv
#           result2   <-   read.csv(file   =   paste("../results/TOXSWA/sediment/",
key$filename[i], sep = ""), as.is = TRUE)
# voor gridcomputing lezen vanaf share
result2                   <-                read.csv(file                   =
paste("//scomp1081/condor_job$/groen033/",FDir,"/",FWSDir,"/sediment/",FNaam,sep =
""), as.is = TRUE)

    tmp2 <- merge(x = time, y = result2, by = "t", all = FALSE)

.
.
.

# bepalen van de weegfactoren gebeurd in het logNormal programma

# loop over het aantal trekkingen

#for (j in 1:Ndraws){

#-------------waterlaag -----------------------------

# 2. LogNormal fortran programma aanroepen. inputwl.txt is invoer voor dit
programma
```

```
.
.
.

#------------sediment-----------------------------------------

#  2.  LogNormal  fortran  programma  aanroepen.  ../LogNormal/sediment/input.txt  is
invoer voor dit programma
.
.

#----------DT50 overall----------------------------------------
#2. Berekenen van de overall DT50 uit de hypothetische dataset!!!
#a. uit elkaar halen van de 2 replicate sets

.
.

#b.  bepalen  van  tsedmax  van  hypothetische  dataset  met  average  concentrations
(tijdstap waarop maxCsed)
.
.
.
# c. berekenen van massa's in wl en sed op t =100 d en t= op moment dat je max
# sediment conc. hebt.
.
.
.
# d. berekenen van ktot volgens m = m0 e^(-ktot*t) en t = t100d - tb
# tb =  breakpoint (time at which decline starts)
# let op log in R is de natuurlijke logaritme (ln)

.
.
.

# e. archiveren van dataset waarmee je de overall DT50 hebt bepaald
.
.
.

# loop over het aantal startwaarden combinaties
for (k in 1:3){

if (k==1){
.
.
.
if (ini_dt50sed > 1000)
    ini_dt50sed <- 1000
#--------Pest files aanpassen-------------------

# copy all templates to directory work
files <- list.files(path = "../templates/PEST", full.names = TRUE)
file.copy(from = files, to = ".", overwrite = TRUE)

# read template(s)

Template_tpl <- readLines(con = "400000007.tpl")

# aanpassen van de pst file
.
.
.     pst <- list(
         DT50wl = formatC(ini_dt50wl, format = "f"),
         DT50sed = formatC(ini_dt50sed, format = "f"),
         WwlNew1 = formatC(tmp3$WwlNew[1],digits=3, format = "e"),
         WwlNew2 = formatC(tmp3$WwlNew[2],digits=3, format = "e"),
.
.
.
```

```
         obs31 = formatC(tmp4$ConcSedNew[15],digits=3, format = "e"),
         obs32 = formatC(tmp4$ConcSedNew[16],digits=3, format = "e")

    )
# read template(s)

Template_pst <- readLines(con = "FIT_TOXSWA.pst")

writeInputFile(pars = pst, template = Template_pst,  con = "FIT_TOXSWA.pst")

# PEST runnen
# PEST runnen vanuit R gaat alleen als je in de batch file RunPest.bat
# Letop ! show.output.on.console altijd op FALSE zetten, want anders werkt het
niet!
   system(command = "runpest.bat", wait = TRUE, show.output.on.console = FALSE,
invisible = FALSE)

#     system(command = "pest.exe FIT_TOXSWA.pst", show.output.on.console = FALSE,
invisible = TRUE)

# benodigde uitvoer uit FIT_TOXSWA.rec halen en wegschrijven naar een csv bestand
recFile <- file("FIT_TOXSWA.rec", "r")

#}
#}
}

#-per run is nu alle pest invoer en uitvoer gemaakt en weggeschreven naar een
aparte folder-
# de loop over nrows in key.csv kan nu worden afgesloten
# close output, want anders kun je recResult.csv niet openen

close(output)
close(objectfunction)
```

Annex 7 Post processing zip files using a Python script

```
# Postprocessing
# Step 1: After the zipfiles come back from the gridcomputers, the files of all
processes have to be unzipped
# Step 2: Make summary of outputfile RecResult and ObjectFunctionResult

import os, zipfile, sys

KOMvar = ["KOM=1","KOM=10","KOM=100","KOM=1000","KOM=10000","KOM=100000"]
WSvar = ["WSsystem1","WSsystem2"]

Scenario = "step1_B8"
TotalJobs = 10000

InputDrive = "//scomp1082/Condor_Data$/groen033/"
OutputDrive = "//nesg0101/era$/gridruns/watersedimentstudy/"

def step1():
    for komvar in KOMvar:
        for wsvar in WSvar:
            if not os.path.exists(OutputDrive+Scenario+"/"+komvar+"/"+wsvar):
                os.makedirs(OutputDrive+Scenario+"/"+komvar+"/"+wsvar)

            fullpathToZip = InputDrive+komvar+"/"+wsvar+"/"
            #destinationPath = "E:/temp/wstudy/"+KOMvar+"/"+WSvar+"/"
            destinationPath = OutputDrive+Scenario+"/"+komvar+"/"+wsvar+"/"
            #err_file = destinationPath + "zip.err"
            err_file = destinationPath+Scenario+"zip.err"
            outp = open (err_file, "w")

            for nummer in range (0,TotalJobs):
                Zipfile = "/RBatchOutput_" + str(nummer) + ".zip"

                sourceZip = zipfile.ZipFile(fullpathToZip + Zipfile, 'r')
                if  os.path.exists(fullpathToZip + Zipfile ) :
                    sourceZip = zipfile.ZipFile(fullpathToZip + Zipfile, 'r')
                    print sourceZip.filename
                    print nummer
                    for name in sourceZip.namelist():
                        if ( (name.find('RecResult.csv')!= -1 ) or
                             (name.find('objectfunction.txt')!=-1 ) or \
                             (name.find('FIT_TOXSWA.rec')!=-1 ) ):
                                 sourceZip.extract(name,     destinationPath+'/'+
str(nummer) )

                    sourceZip.close()

                else:
                    print sourceZip.filename + " does not exist"
                    outp.write ( sourceZip.filename + " does not exist " + "\n")

            outp.close()

def step2():
    # Write header once for 2 outputfiles
    WriteOnce1 = True
    WriteOnce2 = True

    for komvar in KOMvar:
        for wsvar in WSvar:
            sourcePath =  OutputDrive+ Scenario + "/" + komvar + "/" + wsvar + "/"
            destinationPath = OutputDrive+Scenario+"/results/"
            if not os.path.exists(    OutputDrive+Scenario+"/results"):
                os.makedirs(OutputDrive+Scenario+"/results")
```

```
            ObjectOutputFile = open (destinationPath + "ObjectFunctionResults"+
komvar + wsvar + ".txt", "w")
            ObjectOutputFile.write('')
            ObjectOutputFile.close()
            ObjectOutputFile = open (destinationPath + "ObjectFunctionResults"+
komvar + wsvar + ".txt", "a")

            RecResultOutputFile = open (destinationPath + "RecResult"+ komvar +
wsvar + ".csv", "w")
            RecResultOutputFile.write('')
            RecResultOutputFile.close()
            RecResultOutputFile = open (destinationPath + "RecResult"+ komvar +
wsvar + ".csv", "a")

            for nummer in range (0,TotalJobs):
                foldername = sourcePath + str(nummer) +  '/results/PEST/'
                print foldername

                InputFile = open (foldername + 'objectfunction.txt', 'r')
                for regel in InputFile.readlines ():
                    if (regel[0:8] <> 'filename') or WriteOnce1:
                        ObjectOutputFile.write(regel)
                        WriteOnce1 = False
                InputFile.close()

                InputFile = open (foldername + 'RecResult.csv', 'r')
                for regel in InputFile.readlines ():
                   if (regel[0:8] <> 'filename') or WriteOnce2:
                       RecResultOutputFile.write(regel)
                       WriteOnce2 = False
                InputFile.close()

            ObjectOutputFile.close()
            RecResultOutputFile.close()

step1()
step2()
```

Verschenen documenten in de reeks Werkdocumenten van de Wettelijke Onderzoekstaken Natuur & Milieu vanaf 2007

Werkdocumenten zijn verkrijgbaar bij het secretariaat van Unit Wettelijke Onderzoekstaken Natuur & Milieu, te Wageningen. T 0317 – 48 54 71; F 0317 – 41 90 00; E info.wnm@wur.nl
De werkdocumenten zijn ook te downloaden via de WOt-website www.wotnatuurenmilieu.wur.nl

2007

47 *Ten Berge, H.F.M., A.M. van Dam, B.H. Janssen & G.L. Velthof.* Mestbeleid en bodemvruchtbaarheid in de Duin- en Bollenstreek; Advies van de CDM-werkgroep Mestbeleid en Bodemvruchtbaarheid in de Duin- en Bollenstreek

48 *Kruit, J. & I.E. Salverda.* Spiegeltje, spiegeltje aan de muur, valt er iets te leren van een andere planningscultuur?

49 *Rijk, P.J., E.J. Bos & E.S. van Leeuwen.* Nieuwe activiteiten in het landelijk gebied. Een verkennende studie naar natuur en landschap als vestigingsfactor

50 *Ligthart, S.S.H.* Natuurbeleid met kwaliteit. Het Milieu- en Natuurplanbureau en natuurbeleidsevaluatie in de periode 1998-2006

51 *Kennismarkt 22 maart 2007; van onderbouwend onderzoek Wageningen UR naar producten MNP in 27 posters*

52 *Kuindersma, W., R.I. van Dam & J. Vreke.* Sturen op niveau. Perversies tussen nationaal natuurbeleid en besluitvorming op gebiedsniveau.

53.1 *Reijnen, M.J.S.M.* Indicators for the 'Convention on Biodiversity 2010'. National Capital Index version 2.0

53.3 *Windig, J.J., M.G.P. van Veller & S.J. Hiemstra.* Indicatoren voor 'Convention on Biodiversity 2010'. Biodiversiteit Nederlandse landbouwhuisdieren en gewassen

53.4 *Melman, Th.C.P. & J.P.M. Willemen.* Indicators for the 'Convention on Biodiversity 2010'. Coverage protected areas.

53.6 *Weijden, W.J. van der, R. Leewis & P. Bol.* Indicatoren voor 'Convention on Biodiversity 2010'. Indicatoren voor het invasieproces van exotische organismen in Nederland

53.7a *Nijhof, B.S.J., C.C. Vos & A.J. van Strien.* Indicators for the 'Convention on Biodiversity 2010'. Influence of climate change on biodiversity.

53.7b *Moraal, L.G.* Indicatoren voor 'Convention on Biodiversity 2010'. Effecten van klimaatverandering op insectenplagen bij bomen.

53.8 *Fey-Hofstede, F.E. & H.W.G. Meesters.* Indicators for the 'Convention on Biodiversity 2010'. Exploration of the usefulness of the Marine Trophic Index (MTI) as an indicator for sustainability of marine fisheries in the Dutch part of the North Sea.

53.9 *Reijnen, M.J.S.M.* Indicators for the 'Convention on Biodiversity 2010'. Connectivity/fragmentation of ecosystems: spatial conditions for sustainable biodiversity

53.11 *Gaaff, A. & R.W. Verburg.* Indicators for the 'Convention on Biodiversity 2010' Government expenditure on land acquisition and nature development for the National Ecological Network (EHS) and expenditure for international biodiversity projects

53.12 *Elands, B.H.M. & C.S.A. van Koppen.* Indicators for the 'Convention on Biodiversity 2010'. Public awareness and participation

54 *Broekmeyer, M.E.A. & E.P.A.G. Schouwenberg & M.E. Sanders & R. Pouwels.* Synergie Ecologische Hoofdstructuur en Natura 2000-gebieden. Wat stuurt het beheer?

55 *Bosch, F.J.P. van den.* Draagvlak voor het Natura 2000-gebiedenbeleid. Onder relevante betrokkenen op regionaal niveau

56 *Jong, J.J. & M.N. van Wijk, I.M. Bouwma.* Beheerskosten van Natura 2000-gebieden

57 *Pouwels, R. & M.J.S.M. Reijnen & M. van Adrichem & H. Kuipers.* Ruimtelijke condities voor VHR-soorten

58 Niet verschenen/ vervallen

59 *Schouwenberg, E.P.A.G.* Huidige en toekomstige stikstofbelasting op Natura 2000-gebieden

60 Niet verschenen/ vervallen

61 *Jaarrapportage 2006.* WOT-04-001 – ME-AVP

62 *Jaarrapportage 2006.* WOT-04-002 – Onderbouwend Onderzoek

63 *Jaarrapportage 2006.* WOT-04-003 – Advisering Natuur & Milieu

64 *Jaarrapportage 2006.* WOT-04-385 – Milieuplanbureaufunctie

65 *Jaarrapportage 2006.* WOT-04-394 – Natuurplanbureaufunctie

66 *Brasser E.A., M.F. van de Kerkhof, A.M.E. Groot, L. Bos-Gorter, M.H. Borgstein, H. Leneman* Verslag van de Dialogen over Duurzame Landbouw in 2006

67 *Hinssen, P.J.W.* Wettelijke Onderzoekstaken Natuur & Milieu. Werkplan 2007

68 *Nieuwenhuizen, W. & J. Roos Klein Lankhorst.* Landschap in Natuurbalans 2006; Landschap in verandering tussen 1990 en 2005; Achtergronddocument bij Natuurbalans 2006.

69 *Geelen, J. & H. Leneman.* Belangstelling, motieven en knelpunten van natuuraanleg door grondeigenaren. Uitkomsten van een marktonderzoek.

70 *Didderen, K., P.F.M. Verdonschot, M. Bleeker.* Basiskaart Natuur aquatisch. Deel 1: Beleidskaarten en prototype

71 *Boesten, J.J.T.I, A. Tiktak & R.C. van Leerdam.* Manual of PEARLNEQ v4

72 *Grashof-Bokdam, C.J., J. Frissel, H.A.M. Meeuwsen & M.J.S.M. Reijnen.* Aanpassing graadmeter natuurwaarde voor het agrarisch gebied

73 *Bosch, F.J.P. van den.* Functionele agrobiodiversiteit. Inventarisatie van nut, noodzaak en haalbaarheid van het ontwikkelen van een indicator voor het MNP

74 *Kistenkas, F.H. en M.E.A. Broekmeyer.* Natuur, landschap en de Wet algemene bepalingen omgevingsrecht

75 *Luttik, J., F.R. Veeneklaas, J. Vreke, T.A. de Boer, L.M. van den Berg & P. Luttik.* Investeren in landschapskwaliteit; De toekomstige vraag naar landschappen om in te wonen, te werken en te ontspannen

76 *Vreke, J.* Evaluatie van natuurbeleidsprocessen

77 *Apeldoorn, R.C. van,* Working with biodiversity goals in European directives. A comparison of the implementation of the Birds and Habitats Directives and the Water Framework Directive in the Netherlands, Belgium, France and Germany

78 *Hinssen, P.J.W.* Werkprogramma 2008; Unit Wettelijke Onderzoekstaken Natuur & Milieu (WOT-04). Onderdeel Planbureaufuncties Natuur en Milieu.

79 *Custers, M.H.G.* Betekenissen van Landschap in onderzoek voor het Milieu- en Natuurplanbureau; een bibliografisch overzicht

80 *Vreke,J., J.L.M. Donders, B.H.M. Elands, C.M. Goossen, F. Langers, R. de Niet & S. de Vries.* Natuur en landschap voor mensen Achtergronddocument bij Natuurbalans 2007

81 *Bakel, P.J.T. van, T. Kroon, J.G. Kroes, J. Hoogewoud, R. Pastoors, H.Th.L. Massop, D.J.J. Walvoort.* Reparatie Hydrologie voor STONE 2.1. Beschrijving reparatie-acties, analyse resultaten en beoordeling plausibiliteit.

2008

82 *Kistenkas, F.H. & W. Kuindersma..* Jurisprudentie-monitor natuur 2005-2007; Rechtsontwikkelingen Natura 2000 en Ecologische Hoofdstructuur

83 *Berg, F. van den, P.I. Adriaanse, J. A. te Roller, V.C. Vulto & J.G. Groenwold..* SWASH Manual 2.1; User's Guide version 2

84 *Smits, M.J., M.J. Bogaardt, D. Eaton, P. Roza & T. Selnes..* Tussen de bomen het geld zien. Programma Beheer en vergelijkbare regelingen in het buitenland (een quick-scan)

85 *Dijk, T.A. van, J.J.M. Driessen, P.A.I. Ehlert, P.H. Hotsma, M.H.M.M. Montforts, S.F. Plessius & O. Oenema..* Protocol beoordeling stoffen Meststoffenwet; versie 1.0

86 *Goossen, C.M., H.A.M. Meeuwsen, G.J. Franke & M.C. Kuyper. V*erkenning Europese versie van de website www.daarmoetikzijn.nl.

87 *Helming, J.F.M. & R.A.M. Schrijver.* Economische effecten van inzet van landbouwsubsidies voor milieu, natuur en landschap in Nederland; Achtergrond bij het MNP-rapport 'Opties voor Europese landbouw-subsidies

88 *Hinssen, P.J.W.* Werkprogramma 2008; Unit Wettelijke Onderzoekstaken Natuur & Milieu (WOT-04). Programma 001/003/005

90 *Kramer, H.* Geografisch Informatiesysteem Bestaande Natuur; Beschrijving IBN1990t en pilot ontwikkeling BN2004

92 *Jaarrapportage 2007.* WOT-04-001 – Koepel

93 *Jaarrapportage 2007.* WOT-04-002 – Onderbouwend Onderzoek

94 *Jaarrapportage 2007.* WOT-04-003 – Advisering Natuur & Milieu

95 *Jaarrapportage 2007.* WOT-04-005 – M-AVP

96 *Jaarrapportage 2007.* WOT-04-006 – Natuurplanbureaufunctie

97 *Jaarrapportage 2007.* WOT-04-007 – Milieuplanbureaufunctie

98 *Wamelink, G.W.W.* Gevoeligheids- en onzekerheids-analyse van SUMO

99 *Hoogeveen, M.W., H.H. Luesink, L.J. Mokveld & J.H. Wisman.* Ammoniakemissies uit de landbouw in Milieubalans 2006: uitgangspunten en berekeningen

100 *Kennismarkt 3 april 2008; Van onderbouwend onderzoek Wageningen UR naar producten MNP*

101 *Mansfeld, M.J.M. van & J.A. Klijn.* "Balansen op de weegschaal". Terugblik op acht jaar Natuurbalansen (1996-2005)

102 *Sollart, K.M. & J. Vreke.* Het faciliteren van natuur- en milieueducatie in het basisonderwijs; NME-ondersteuning in de provincies

103 *Berg, F. van den, A. Tiktak, J.G. Groenwold, D.W.G. van Kraalingen, A.M.A. van der Linden & J.J.T.I. Boesten*, Documentation update for GeoPEARL 3.3.3

104 *Wijk, M.N., van (redactie).* Aansturing en kosten van het natuurbeheer. Ecologische effectiviteit regelingen natuurbeheer

105 *Selnes, T. & P. van der Wielen.* Tot elkaar veroordeeld? Het belang van gebiedsprocessen voor de natuur

106 *Annual reports for 2007; Programme WOT-04*

107 *Pouwels, R. J.G.M. van der Greft, M.H.C. van Adrichem, H. Kuiper, R. Jochem & M.J.S.M. Reijnen.* LARCH Status A

108 *Wamelink, G.W.W.* Technical Documentation for SUMO2 v. 3.2.1,

109 *Wamelink, G.W.W., J.P. Mol-Dijkstra & G.J. Reinds.* Herprogrammeren van SUMO2. Verbetering in het kader van de modelkwaliteitsslag

110 *Salm, C. van der, T. Hoogland & D.J.J. Walvoort.* Verkenning van de mogelijkheden voor de ontwikkeling van een metamodel voor de uitspoeling van stikstof uit landbouwgronden

111 *Dobben H.F. van & R.M.A. Wegman.* Relatie tussen bodem, atmosfeer en vegetatie in het Landelijk Meetnet Flora (LMF)

112 *Smits, M.J.W. & M.J. Bogaardt.* Kennis over de effecten van EU-beleid op natuur en landschap

113 *Maas,G.J. & H. van Reuler.* Boomkwekerij en aardkunde in Nederland,

114 *Lindeboom, H.J., R. Witbaard, O.G. Bos & H.W.G. Meesters.* Gebiedsbescherming Noordzee, habitattypen, instandhoudingdoelen en beheermaatregelen

115 *Leneman, H., J. Vader, L.H.G. Slangen, K.H.M. Bommel, N.B.P. Polman, M.W.M. van der Elst & C. Mijnders.* Groene diensten in Nationale Landschappen- Potenties bij een veranderende landbouw,

116 *Groeneveld, R.A. & D.P. Rudrum.* Habitat Allocation to Maximize BiOdiversity, A technical description of the HAMBO model

117 *Kruit, J., M. Brinkhuijzen & H. van Blerck.* Ontwikkelen met kwaliteit. Indicatoren voor culturele vernieuwing en architectonische vormgeving

118 *Roos-Klein Lankhorst, J.* Beheers- en Ontwikkelingsplan 2007: Kennismodel Effecten Landschap Kwaliteit; Monitoring Schaal; BelevingsGIS

119 *Henkens, R.J.H.G.* Kwalitatieve analyse van knelpunten tussen Natura 2000-gebieden en waterrecreatie

120 *Verburg, R.W., I.M. Jorritsma & G.H.P. Dirkx.* Quick scan naar de processen bij het opstellen van beheerplannen van Natura 2000-gebieden. Een eerste verkenning bij provincies, Rijkswaterstaat en Dienst Landelijk Gebied

121 *Daamen, W.P.* Kaart van de oudste bossen in Nederland; Kansen op hot spots voor biodiversiteit

122 *Lange de, H.J., G.H.P. Arts & W.C.E.P. Verberk.* Verkenning CBD 2010-indicatoren zoetwater. Inventarisatie en uitwerking relevante indicatoren voor Nederland

123 *Vreke, J., N.Y. van der Wulp, J.L.M. Donders, C.M. Goossen, T.A. de Boer & R. Henkens.* Recreatief gebruik van water. Achtergronddocument Natuurbalans 2008

124 *Oenema, O. & J.W.H. van der Kolk*. Moet het eenvoudiger? Een essay over de complexiteit van het milieubeleid

125 *Oenema, O. & A. Tiktak.* Niets is zonder grond; Een essay over de manier waarop samenlevingen met hun grond omgaan

2009

126 *Kamphorst, D.A.* Keuzes in het internationale biodiversiteitsbeleid; Verkenning van de beleidstheorie achter de internationale aspecten van het Beleidsprogramma Biodiversiteit (2008-2011)

127 *Dirkx, G.H.P. & F.J.P. van den Bosch.* Quick scan gebruik Catalogus groenblauwe diensten

128 *Loeb, R. & P.F.M. Verdonschot.* Complexiteit van nutriëntenlimitaties in oppervlaktewateren

129 *Kruit, J. & P.M. Veer.* Herfotografie van landschappen; Landschapsfoto's van de 'Collectie de Boer' als uitgangspunt voor het in beeld brengen van ontwikkelingen in het landschap in de periode 1976-2008

130 *Oenema, O., A. Smit & J.W.H. van der Kolk.* Indicatoren Landelijk Gebied; werkwijze en eerste resultaten

131 *Agricola, H.J.A.J. van Strien, J.A. Boone, M.A. Dolman, C.M. Goossen, S. de Vries, N.Y. van der Wulp, L.M.G. Groenemeijer, W.F. Lukey & R.J. van Til.* Achtergrond-document Nulmeting Effectindicatoren Monitor Agenda Vitaal Platteland

132 *Jaarrapportage 2008.* WOT-04-001 – Koepel

133 *Jaarrapportage 2008.* WOT-04-002 – Onderbouwend Onderzoek

134 *Jaarrapportage 2008.* WOT-04-003 – Advisering Natuur & Milieu

135 *Jaarrapportage 2008.* WOT-04-005 – M-AVP

136 *Jaarrapportage 2008.* WOT-04-006 – Natuurplanbureaufunctie

137 *Jaarrapportage 2008.* WOT-04-007 – Milieuplanbureaufunctie

138 *Jong de, J.J., J. van Os & R.A. Smidt.* Inventarisatie en beheerskosten van landschapselementen

139 *Dirkx, G.H.P., R.W. Verburg & P. van der Wielen.* Tegenkrachten Natuur. Korte verkenning van de weerstand tegen aankopen van landbouwgrond voor natuur

140 *Annual reports for 2008; Programme WOT-04*

141 *Vullings, L.A.E., C. Blok, G. Vonk, M. van Heusden, A. Huisman, J.M. van Linge, S. Keijzer, J. Oldengarm & J.D. Bulens.* Omgaan met digitale nationale beleidskaarten

142 *Vreke, J.,A.L. Gerritsen, R.P. Kranendonk, M. Pleijte, P.H. Kersten & F.J.P. van den Bosch.* Maatlat Government – Governance

143 *Gerritsen, A.L., R.P. Kranendonk, J. Vreke, F.J.P. van den Bosch & M. Pleijte.* Verdrogingsbestrijding in het tijdperk van het Investeringsbudget Landelijk Gebied. Een verslag van casusonderzoek in de provincies Drenthe, Noord-Brabant en Noord-Holland.

144 *Luesink, H.H., P.W. Blokland, M.W. Hoogeveen & J.H. Wisman.* Ammoniakemissie uit de landbouw in 2006 en 2007

145 *Bakker de, H.C.M. & C.S.A. van Koppen.* Draagvlakonderzoek in de steigers. Een voorstudie naar indicatoren om maatschappelijk draagvlak voor natuur en landschap te meten

146 *Goossen, C.M.,* Monitoring recreatiegedrag van Nederlanders in landelijke gebieden. Jaar 2006/2007

147 *Hoefs, R.M.A., J. van Os & T.J.A. Gies.* Kavelruil en Landschap. Een korte verkenning naar ruimtelijke effecten van kavelruil.

148 *Klok, T.L., R. Hille Ris Lambers, P. de Vries, J.E. Tamis & J.W.M. Wijsman.* Quick scan model instruments for marine biodiversity policy.

149 *Spruijt, J., P. Spoorenberg & R. Schreuder.* Milieueffectiviteit en kosten van maatregelen gewasbescherming.

150 *Ehlert, P.A.I. (rapporteur).* Advies Bemonstering bodem voor differentiatie van fosfaatgebruiksnormen.

151 *Wulp van der, N.Y.* Storende elementen in het landschap: welke, waar en voor wie? Bijlage bij WOt-paper 1 – Krassen op het landschap

152 *Oltmer, K., K.H.M. van Bommel, J. Clement, J.J. de Jong, D.P. Rudrum & E.P.A.G. Schouwenberg.* Kosten voor habitattypen in Natura 2000-gebieden. Toepassing van de methode Kosteneffectiviteit natuurbeleid.

153 *Adrichem van, M.H.C., F.G. Wortelboer & G.W.W. Wamelink.* MOVE. Model for terrestrial Vegetation. Version 4.0

154 *Wamelink, G.W.W., R.M. Winkler & F.G. Wortelboer.* User documentation MOVE4 v 1.0

155 *Gies de, T.J.A., L.J.J. Jeurissen, I. Staritsky & A. Bleeker.* Leefomgevingsindicatoren Landelijk gebied. Inventarisatie naar stand van zaken over geurhinder, lichthinder en fijn stof.

156 *Tamminga, S., A.W. Jongbloed, P. Bikker, L. Sebek, C. van Bruggen & O. Oenema.* Actualisatie excretiecijfers landbouwhuisdieren voor forfaits regeling Meststoffenwet

157 *Van der Salm, C., L. .M. Boumans, G.B.M. Heuvelink & T.C. van Leeuwen.* Protocol voor validatie van het nutriëntenemissiemodel STONE op meetgegevens uit het Landelijk Meetnet effecten Mestbeleid

158 *Bouwma, I.M.* Quickscan Natura 2000 en Programma Beheer. Een vergelijking van Programma Beheer met de soorten en habitats van Natura 2000

159 *Gerritsen, A.L., D.A. Kamphorst, T.A. Selnes, M. van Veen, F.J.P.van den Bosch, L. van den Broek, M.E.A. Broekmeyer, J.L.M. Donders, R.J. Fontein, S. van Tol, G.W.W. Wamelink & P. van der Wielen.* Dilemma's en barrières in de praktijk van het natuur- en landschapsbeleid; Achtergronddocument bij Natuurbalans 2009.

160 *Fontein R.J, T.A. de Boer, B. Breman, C.M. Goossen, R.J.H.G. Henkens, J. Luttik & S. de Vries*. Relatie recreatie en natuur; Achtergronddocument bij Natuurbalans 2009

161 *Deneer, J.W. & R. Kruijne.* Atmosferische depositie van gewasbeschermingsmiddelen. Een verkenning van de literatuur verschenen na 2003.

162 *Verburg, R.W., M.E. Sanders, G.H.P. Dirkx, B. de Knegt & J.W. Kuhlman.* Natuur, landschap en landelijk gebied. Achtergronddocument bij Natuurbalans 2009.

163 *Doorn van, A.M. & M.P.C.P. Paulissen.* Natuurgericht milieubeleid voor Natura 2000-gebieden in Europees perspectief: een verkenning.

164 *Smidt, R.A., J. van Os & I. Staritsky.* Samenstellen van landelijke kaarten met landschapselementen, grondeigendom en beheer. Technisch achtergronddocument bij de opgeleverde bestanden.

165 *Pouwels, R., R.P.B. Foppen, M.F. Wallis de Vries, R. Jochem, M.J.S.M. Reijnen & A. van Kleunen,* Verkenning LARCH: omgaan met kwaliteit binnen ecologische netwerken.

166 *Born van den, G.J., H.H. Luesink, H.A.C. Verkerk, H.J. Mulder, J.N. Bosma, M.J.C. de Bode & O. Oenema,* Protocol voor monitoring landelijke mestmarkt onder een stelsel van gebruiksnormen, versie 2009.

167 *Dijk, T.A. van, J.J.M. Driessen, P.A.I. Ehlert, P.H. Hotsma, M.H.M.M. Montforts, S.F. Plessius & O. Oenema.* Protocol beoordeling stoffen Meststoffenwet- Versie 2.1

168 *Smits, M.J., M.J. Bogaardt, D. Eaton, A. Karbauskas & P. Roza.* De vermaatschappelijking van het Gemeenschappelijk Landbouwbeleid. Een inventarisatie van visies in Brussel en diverse EU-lidstaten.

169 *Vreke, J. & I.E. Salverda.* Kwaliteit leefomgeving en stedelijk groen.

170 *Hengsdijk, H. & J.W.A. Langeveld.* Yield trends and yield gap analysis of major crops in the World.

171 *Horst, M.M.S. ter & J.G. Groenwold,* Tool to determine the coefficient of variation of DegT50 values of plant protection products in water-sediment systems for different values of the sorption coefficient.

172 *Boons-Prins, E., P. Leffelaar, L. Bouman & E. Stehfest.* Grassland simulation with the LPJmL model.

173 *Smit, A., O. Oenema & J.W.H. van der Kolk.* Indicatoren Kwaliteit Landelijk Gebied.

174 *Boer de, S., M.J. Bogaardt, P.H. Kersten, F.H. Kistenkas, M.G.G. Neven & M. van der Zouwen.* De zoektocht naar nationale beleidsruimte in de EU-richtlijnen voor het milieu- en natuurbeleid

175 *Jaarrapportage 2009.* WOT-04-001 – Koepel

176 *Jaarrapportage 2009.* WOT-04-002 – Onderbouwend Onderzoek

177 *Jaarrapportage 2009.* WOT-04-003 – Advisering Natuur & Milieu

178 *Jaarrapportage 2009.* WOT-04-005 – M-AVP

179 *Jaarrapportage 2009.* WOT-04-006 – Natuurplanbureaufunctie

180 *Jaarrapportage 2009.* WOT-04-007 – Milieuplanbureaufunctie

181 *Annual reports for 2009; Programme WOT-04*