Tutorial for interfacing *hydroPSO* with SWAT-2005 and MODFLOW-2005

Rodrigo Rojas, PhD
Mauricio Zambrano-Bigiarini, PhD

March 19, 2012

Contents

List of Tables

List of Figures

Preface

The next tutorial shows how to use the *hydroPSO* R package to calibrate different model codes. Two of the main properties of *hydroPSO* are the independence from the model to be calibrated and the simple interfacing between the model code and the calibration-engine, i.e. *hydroPSO*. These two properties are illustrated through the calibration of real-world study cases, which should be the starting point for a user planning to implement *hydroPSO* for her/his own model calibration exercise.

We show here how to use *hydroPSO* to calibrate a semi-distributed hydrological model and a steady-state groundwater flow model. The first is fully implemented in SWAT-2005 (*Neitsch et al.*, 2005) whereas the second is implemented in MODFLOW-2005 (MF2005) (*Harbaugh*, 2005). Both model codes are widely-used programs to solve surface water and groundwater flow problems. At the same time, for illustrative purposes we use the program ZONEBUDGET (ZB) (*Harbaugh*, 1990) to obtain groundwater flows at particular aquifer cross sections. The purpose of including ZB is to illustrate how to interface *hydroPSO* with a sequential modelling application. In this case, however, only MF2005 is calibrated against observed data and including other observations can be easily implemented in a similar way as described here.

Both programs SWAT-2005 and MF2005/ZB are public domain and accessible from `http://swatmodel.tamu.edu/software/swat-model/` and `http://water.usgs.gov/nrp/gwsoftware/modflow.html`, respectively. For simplicity, we must assume some basic knowledge from part of the reader about the file naming convention used in SWAT-2005, MF2005/ZB, as well as the setting up of the internal options for these programs.

As an integral part of this tutorial, the reader can download from `http://www.rforge.net/hydroPSO/` two files (`SWAT2005.zip` and `MF2005.zip`) including all the required files to run SWAT-2005 and MF2005, sample R scripts to interface hydroPSO with SWAT-2005 and MF2005, as well as several auxiliary files.

For assistance, bugs report, comments, and suggestions please contact the authors of the hydroPSO package: Mauricio.Zambrano@jrc.ec.europa.eu and Rodrigo.RojasMujica@gmail.com.

1 Introduction

hydroPSO is an R package implementing an enhanced version of the canonical Particle Swarm Optimisation (PSO) algorithm (*Kennedy and Eberhart*, 1995; *Eberhart and Kennedy*, 1995). PSO is a population-based stochastic optimisation technique inspired by social behaviour of bird flocking, which shares few similarities with other evolutionary optimisation techniques such as Genetic Algorithms (GA) (*Poli et al.*, 2007). In PSO, however, the solution-space is explored on the basis of individual and global best-known "particle positions" with no presence of evolution operators (e.g. mutation or crossover) as in GA. PSO has recently received a surge of attention in literature given its flexibility, simplicity of implementation (programming), low memory and computational requirements, low number of adjustable parameters, and efficiency (see, e.g., *Eberhart and Shi*, 1998; *Shi and Eberhart*, 1999; *Eberhart and Shi*, 2001; *Poli et al.*, 2007).

The main motivations for developing the *hydroPSO* package are:

1. bring this powerful optimisation/calibration technique to the attention of practitioners and scientists working on environmental modelling,

2. provide a model-independent software package allowing the user to calibrate (different) environmental models without having to invest large efforts in customizing the calibration engine to the environmental model, and

3. allow the PSO research community to explore alternative advancements and configurations of the standard PSO using a versatile single-package software.

hydroPSO is also capable of performing sensitivity analysis using the Latin Hypercube One-factor-At-a-Time (LH-OAT) method (see *van Griensven et al.*, 2006) and provides detailed information about the evolution of PSO's performance. In addition, advanced plotting capabilities and a complete family of built-in functions contained in the R language (*R Development Core Team*, 2011) are available to visualize and summarize the calibration results, respectively. At the same time, *hydroPSO* features a suite of controlling options and PSO variants for fine-tuning and improving the performance of the canonical PSO algorithm, thus, allowing the user to adapt the calibration/optimisation engine to different modelling problems. For a full description of the PSO enhancements included in the *hydroPSO* package the reader is referred to *Zambrano-Bigiarini and Rojas* (2012).

In principle, *hydroPSO* only needs to know "which" model parameters need to be calibrated and "where" they need to be written. Then, it will take

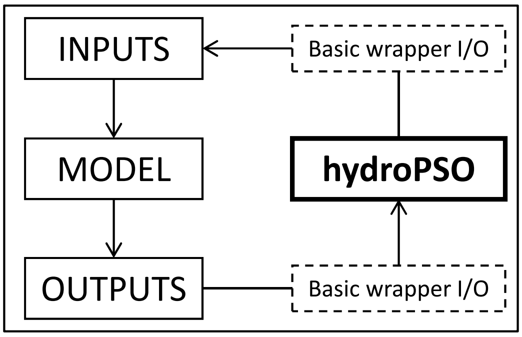

Figure 1: Flow chart of the implementation/interaction between *hydroPSO* and the model code to be calibrated. Dashed-line boxes represent basic I/O functions to read/write model files (After *Zambrano-Bigiarini and Rojas*, 2012).

control over the model(s) to be calibrated until either a maximum number of iterations or an error tolerance in the objective function are reached: these two being problem-specific and user-defined. The basic interaction between *hydroPSO* and the model to be calibrated is shown in Figure 1. For models with numerous input and output files or cascading models (i.e. serial modelling applications), several I/O functions can be combined to interface *hydroPSO* and the model.

In this tutorial we show first how to interface *hydroPSO* with SWAT-2005 (*Neitsch et al.*, 2005) to calibrate a semi-distributed hydrological model for the Ega River basin in Spain. That section illustrates three main points: 1) basic interfacing between *hydroPSO* and the model through the definition of files *ParamRanges.txt* and *ParamFiles.txt*, 2) advanced sensitivity analysis using LH-OAT and the use of *hydroPSO* pre-defined goodness-of-fit measures as objective functions, and 3) the set up for calibrating a model output variable in time and static in space (transient application). Second, we show how to interface *hydroPSO* with MF2005 (*Harbaugh*, 2005) to calibrate a groundwater flow model for the regional aquifer of the Pampa del Tamarugal basin in Chile. That section, in turn, illustrates: 1) advanced interfacing between *hydroPSO* and the model through the implementation of user-defined I/O wrapper functions written in R, 2) how to define/use a customized goodness-of-fit measure as objective function, and 3) how to set up *hydroPSO* for a model output fixed in time but variable in space (steady-state application). We therefore believe that this tutorial cover two standard-problems commonly found by the modelling community. We must note, however, that these tutorials deal with surface water and groundwater modelling applications as these two are the authors' areas of expertise.

However, based on the flexibility of the *hydroPSO* package and the benefits added by programming it in R, we believe this package can be implemented to a wider range of environmental models requiring some form of parameter estimation.

In Section 2 we present a brief description of PSO, including the main algorithm, discussion on topologies and PSO variants. An introductory application of *hydroPSO* to optimise test functions commonly used to assess the performance of optimisation algorithms is described in Section 3. The procedure to interface *hydroPSO* with SWAT-2005 and MF2005 as well as the corresponding calibration results, are reported in Sections 4 and 5, respectively.

2 Particle Swarm Optimisation (PSO)

2.1 Canonical PSO Algorithm

Particle Swarm Optimisation (PSO) is a population-based search algorithm developed by *Kennedy and Eberhart* (1995) and *Eberhart and Kennedy* (1995). A feature distinguishing PSO from evolutionary algorithms is the lack of genetic operators, instead each individual of the population, termed *particles* in PSO terminology, adjusts its *flying trajectory* around the multi-dimensional search-space according to its own flying experience (*local search*) and the one of all particles in the swarm (*global search*) (*Eberhart and Shi*, 1998).

Considering a D-dimensional search-space, *position* and *velocity* for the i-th particle are represented by $\vec{X}_i = x_{i1}, x_{i2}, \ldots, x_{iD}$ and $\vec{V}_i = v_{i1}, v_{i2}, \ldots, v_{iD}$, respectively. The performance of each particle is measured through a problem-specific "goodness-of-fit" measure, which is the basis for calculating $\vec{X}_i$, thus, reflecting the quality of the particle's position. The best-known position of the i-th particle (known as *personal best*) is represented by $\vec{P}_i = p_{i1}, p_{i2}, \ldots, p_{iD}$, whereas the best-known position for the whole swarm (known as *global best*) is represented by $\vec{G} = g_1, g_2, \ldots, g_D$. Velocity and position of the i-th particle are updated according to the following equations,

$$\vec{V}_i^{t+1} = \omega \vec{V}_i^t + c_1 \vec{U}_1^t \otimes (\vec{P}_i^t - \vec{X}_i^t) + c_2 \vec{U}_2^t \otimes (\vec{G}^t - \vec{X}_i^t) \tag{1}$$

or (*Clerc and Kennedy*, 2002),

$$\vec{V}_i^{t+1} = \chi \left[\vec{V}_i^t + c_1 \vec{U}_1^t \otimes (\vec{P}_i^t - \vec{X}_i^t) + c_2 \vec{U}_2^t \otimes (\vec{G}^t - \vec{X}_i^t) \right] \tag{2}$$

$$\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{V}_i^{t+1} \tag{3}$$

where $i = 1, 2, \ldots, N$ is the number of particles in the swarm, $t = 1, 2, \ldots, T$ is the number of iterations, ω is the inertia weight, c_1 and c_2 are the *cognitive* and *social* acceleration coefficients, $\vec{U}_1$ and $\vec{U}_2$ are independent and uniformly distributed random vectors in the range $[0, 1]$ (note that $\otimes$ denotes element-wise vector multiplication), and χ is the constriction factor defined as

$$\chi = \frac{2\kappa}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \tag{4}$$

where $\kappa \in [0, 1]$ and $\varphi = c_1 + c_2 > 4$. It should be noted that equations (1) and (2) are mathematically equivalent for appropriate values of the coefficients ω, c_1, and c_2. Finally, the PSO algorithm can be summarized as follows:

Algorithm 1 Canonical PSO algorithm.

```
for $i = 1$ to $N$ do {for each particle in the swarm}
    Initialize particles' position ($\vec{X}_i$) and velocity ($\vec{V}_i$)
    Initialize personal best $\vec{P}_i$ and global best $\vec{G}$
repeat
    for $i = 1$ to $N$ do
        Pick random vectors $\vec{U}_1$ and $\vec{U}_2$
        Update particle's velocity using equations 1 or 2
        Update particle's position ($\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{V}_i^{t+1}$)
        if $f(\vec{X}_i) < f(\vec{P}_i)$ then {minimization of $f$}
            Update particle's best-known position $\vec{P}_i = \vec{X}_i$
            if $f(\vec{P}_i) < f(\vec{G})$ then {minimization of $f$}
                Update the swarm's best-known position $\vec{G} = \vec{P}_i$
until [nr. of iterations ($T$) or tolerance error are met]
```

2.2 Topologies

In terms of neighbourhood topologies connecting individual particles, two of the most common correspond to the so-called *gbest* (global best) and *lbest* (local best). In the *gbest* topology all particles are connected to each other and, thus, the global best influences all particles in the swarm. In the *lbest* topology, in turn, each particle is connected to $k = 2$ immediate neighbours only and, thus, exchange of information from the global best is restricted only to the particle's corresponding neighbourhood. *lbest* shows the advantage of allowing parallel search on different regions of the search-space (*Poli et al.*, 2007), thus, being less sensitive to sub-optimal solutions but resulting in a slower convergence compared to *gbest*. From these definitions, it follows that the *gbest* topology is a special case of the *lbest* topology when the whole swarm is used as neighbourhood.

Generalizations of the *lbest* topology for $k > 2$ where investigated in *Kennedy and Mendes* (2002). One of their main findings was the relative superiority of the *von Neumann* structured neighbourhood, i.e. when $k = 4$. This topology is more densely connected than *lbest* but less densely than *gbest*, thus, showing some parallelism with *lbest* but benefiting from a bigger neighbourhood (*Poli et al.*, 2007). Finally, the so-called *random* topology as defined in the "Standard PSO 2007" (*Clerc*, 2007; *Clerc et al.*, 2011) is implemented in *hydroPSO*. In the *random* topology the number of "informants" remains fixed but the "connections" among particles randomly change when the global optimum shows no improvement.

2.3 PSO Variants

In equations (1) and (2) only two effective sources of influence are considered for updating the velocity of the i-th particle, namely, the position of its personal best, $\vec{P}_i$, and that of the global best, $\vec{G}$. *Kennedy and Mendes* (2002) and *Mendes et al.* (2004) propose a Fully Informed Particle Swarm (FIPS), where information drawn from all the neighbours contributes to adjust the particle's velocity using the following equation,

$$\vec{V}_i^{t+1} = \chi \left[\vec{V}_i^t + \frac{1}{K_i} \sum_{n=1}^{K_i} \vec{U}_1^t(0, \varphi) \otimes (\vec{P}_{nbr_n}^t - \vec{X}_i^t) \right] \tag{5}$$

where K_i is the number of neighbours for particle i, and $\vec{P}_{nbr_n}$ is the personal best position of the particle's n-th neighbour.

An alternative version of the FIPS variant, called weighted FIPS (wFIPS), is also implemented in *hydroPSO*. In wFIPS the contribution of each particle to update other's particle velocities is weighted by its corresponding goodness-of-fit value, thus, better performing particles receive a higher weight. Following *Mendes et al.* (2004), FIPS shows some superiority compared to the canonical PSO algorithm, nonetheless, it is particularly sensitive to the topology implemented.

Additionally, *Zhao* (2006) proposes the Improved Particle Swarm Optimisation (IPSO) where instead of having a single term representing the social influence $(\vec{G}^t - \vec{X}_i^t)$, particles' velocities are adjusted using information drawn from a subset containing the n_{gb} best performing particles through the following equation,

$$\vec{V}_i^{t+1} = \omega \vec{V}_i^t + c_1 \vec{U}_1^t \otimes (\vec{P}_i^t - \vec{X}_i^t) + \sum_{j=1}^{n_{gb}} c_{2,j} \vec{U}_{2,j}^t \otimes (\vec{G}_j^t - \vec{X}_i^t) \tag{6}$$

where the coefficient $c_{2,j}$ is defined by

$$c_{2,j} = c_2 \hat{f}_j \tag{7}$$

where $\hat{f}_j$ is a weighting factor based on the values of the objective function for the n_{gb} particles.

According to *Zhao* (2006), IPSO follows the analogy where a group of leaders, i.e. the best performing particles for an iteration, could influence positively better decisions for society (swarm) compared to the case when a single leader is followed.

3 Step-by-Step *hydroPSO* Application

3.1 Setting Up the Environment

1. Use next commands to retrieve current working directory, set a new working directory, and printing files included in the current directory, respectively.

   ```
   > getwd()
   > setwd("~/tmp")
   > list.files(".")
   ```

2. Installing *hydroPSO*.

   ```
   > install.packages("hydroPSO")
   ```

3. Loading the *hydroPSO* library containing data and functions used in this analysis.

   ```
   > library(hydroPSO)
   ```

3.2 Basic *hydroPSO* Application

This section illustrates the implementation of the *hydroPSO* package to optimise two (highly) multi-modal and multi-dimensional test functions with a considerable number of local sub-optimal solutions. These functions are commonly used as benchmarking for the performance of optimisation algorithms. Otherwise stated, the optimal solution for these functions should be zero.

In this section we do not aim at finding an "optimum" configuration for the *hydroPSO* parameters, which in itself can be a massive task. Instead, we illustrate a few controlling options to handle problems such as premature convergence for (highly) multi-modal objective functions. We refer the reader to *Zambrano-Bigiarini and Rojas* (2012) for the results of a validation of *hydroPSO* against the Standard PSO 2007 (*Clerc et al.*, 2011).

3.2.1 Optimisation of Ackley Function

1. The Ackley function (Equation 8) includes an exponential term, which originates numerous local minima. Complexity of the Ackley function is moderated and algorithms based on gradient steepest descent will be most likely trapped in a local optima.

$$f(x) = 20 + \exp(1) - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{2}\sum_{i=1}^{n}\cos(2\pi x_i)\right) ;$$
$$-32.7 \leq x_i \leq 32.7 \ ; \ i = 1, 2, \ldots, n. \tag{8}$$

For optimisation of the Ackley function we define upper and lower limits [-32;32], dimensionality (d=10), 20 particles, and 1000 iterations:

```
> lower <- rep(-32,10)
> upper <- rep(32,10)
> set.seed(1111)
> hydroPSO(fn="ackley",lower=lower,upper=upper,
+          control=list(npart=20,maxit=1000,REPORT=100))

$par
       Param1        Param2        Param3        Param4
-6.199517e-14 -2.844921e-13  4.124145e-14  1.631265e-14
       Param5        Param6        Param7        Param8
 5.796993e-13 -4.527180e-13  1.687279e-13 -6.296430e-14
       Param9       Param10
-4.149708e-13  1.085725e-12

$value
[1] 1.794564e-12

$best.particle
[1] 14

$counts
function.calls     iterations    regroupings
         20000           1000              0

$convergence
[1] 3

$message
[1] "Maximum number of iterations reached"
```

2. Adding a linear time-variant definition for the acceleration coefficient c_1 in the range [1.28;1.05]:

```
> set.seed(1111)
> hydroPSO(fn="ackley",lower=lower,upper=upper,
+          control=list(npart=20,maxit=1000,REPORT=100,
+          use.TVc1=TRUE,TVc1.type="linear",TVc1.rng=c(1.28,1.05)))

$par
       Param1        Param2        Param3        Param4
-2.874931e-14 -4.753088e-15  4.681556e-15  2.283547e-14
       Param5        Param6        Param7        Param8
-4.239557e-15 -1.604158e-14 -2.052563e-14  1.348402e-15
       Param9       Param10
 1.699499e-14 -1.620151e-14

$value
[1] 6.439294e-14

$best.particle
[1] 11

$counts
function.calls     iterations    regroupings
         20000           1000              0

$convergence
[1] 3

$message
[1] "Maximum number of iterations reached"
```

3. Plotting the results:

```
> plot_results(do.png=TRUE)
```

Here we report results directly to the R graphical device. Using the option `do.png=TRUE`, however, it will direct the graphical output to `png` files stored in `./PSO.out/pngs/`. Results produced are shown in Figures 2-11.

In Figure 2, Gbest shows the evolution of the global optimum whereas the Normalized Swarm Radius (NSR) indicates the convergence of the swarm to the (optimum) attraction zone. Assessment of both Gbest and NSR is particularly useful when the regrouping strategy used to tackle premature convergence is activated (`use.RG=TRUE`).

Figure 3 shows dotty-plots for each of the 10 parameters, where each dot represents an evaluation of the function (model) being optimised. By default, dotty-plots are shown for all parameter sets, however, and as seen from Figure 3, not all of them may show a good performance. At the same time, Figure 4 shows boxplots summarizing each parameter. Using the options `beh.thr` and `MinMax`, a "behavioural" threshold is defined to select a sub-set of the best performing parameters. By defining `beh.thr` we discard parameters showing a poor performance and, as such, this shows some parallelism with the specification of the rejection threshold in Generalized Likelihood Uncertainty Estimation (GLUE) (see *Beven*, 2006) or the definition of the burn-in samples in Markov Chain Monte Carlo (MCMC) simulation (see *Gelman et al.*, 2004).

Interaction between parameters is shown in Figure 5. This figure shows the interaction between the first 5 parameters (selected by default by *hydroPSO*) as a function of the goodness-of-fit. Using `dp3D.names` the user can select specific parameters to plot. If the option `beh.thr` is specified, this figure shows the sub-set representing only the best-performance particles.

Figure 6 shows a (matrix) summary of the interaction among parameters. Here, the statistical significance of the correlation as well as dispersion-like and histogram plot summaries are obtained.

Empirical Cumulative Distribution Functions (ECDFs) for each parameter are shown in Figure 7. ECDFs are created for all parameter sets evaluated unless the `beh.thr` option is specified. In the last case, ECDFs are built from the sub-set representing the best performance.

Figure 8 shows the histograms of parameters. As in Figures 5 and 7, these histograms are calculated for all parameter sets retained unless the `beh.thr` option is specified.

Finally, Figures 9, 10 and 11 provide detailed information about the evolution of the goodness-of-fit function per iteration for each particle defined in the swarm, and particles' positions and velocities versus function (model) evaluation, respectively.

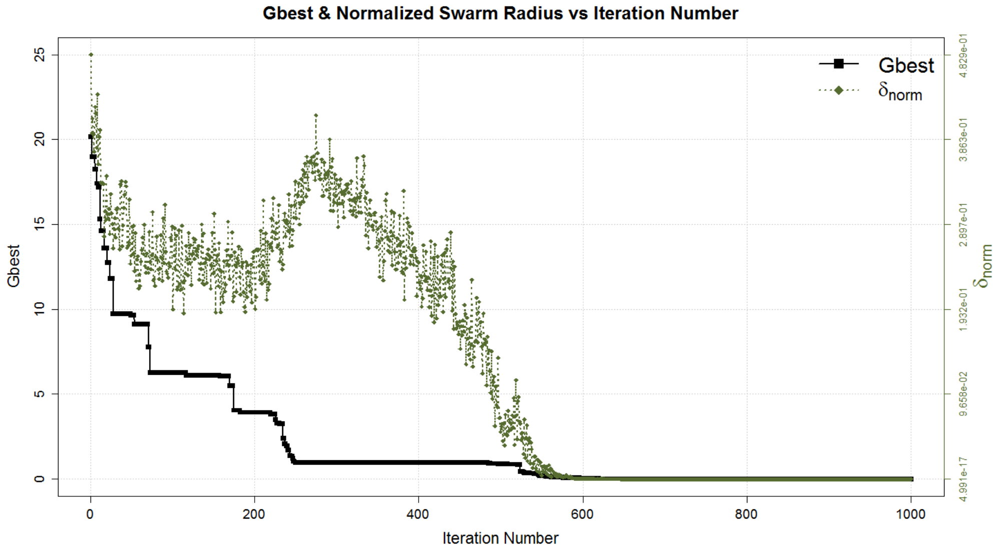

Figure 2: Global best (Gbest) and the Normalized Swarm Radius (NSR) versus iteration number.

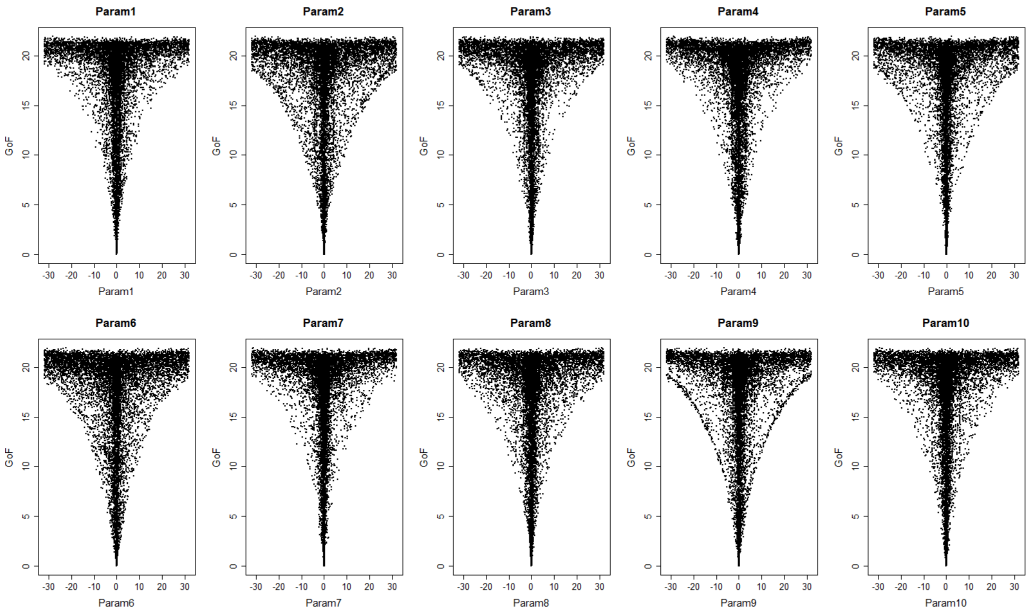

Figure 3: Dotty-plots for each parameter.

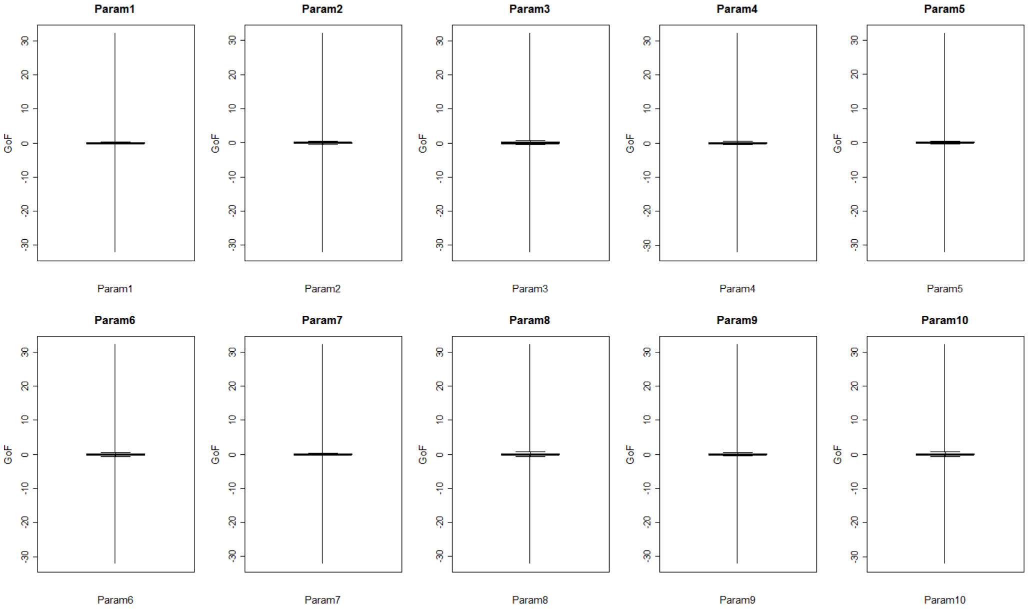

Figure 4: Box-plots for each parameter.

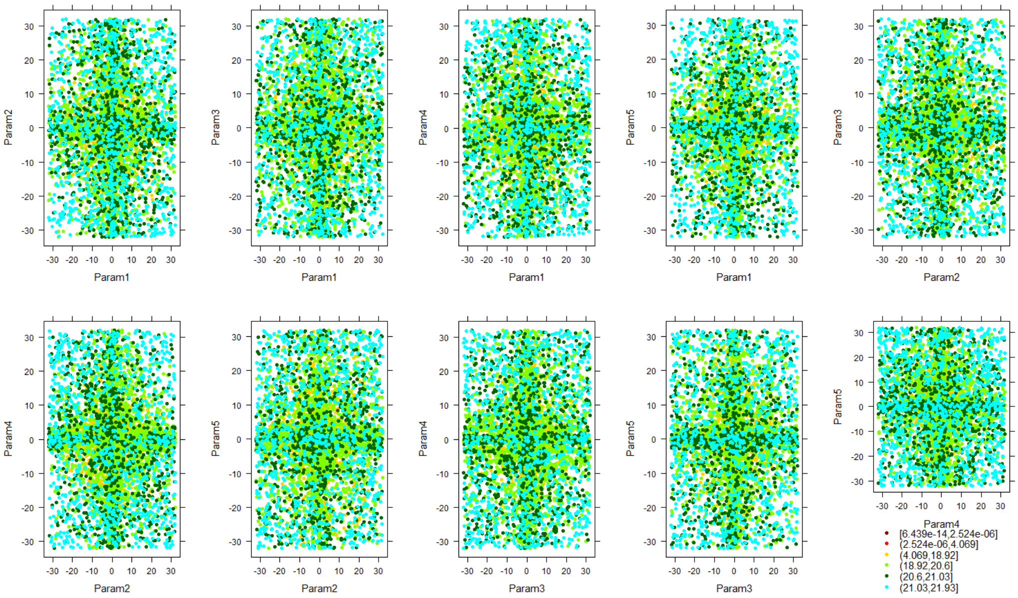

Figure 5: 2-dimensional projected dotty-plots highlighting the interaction between parameters.

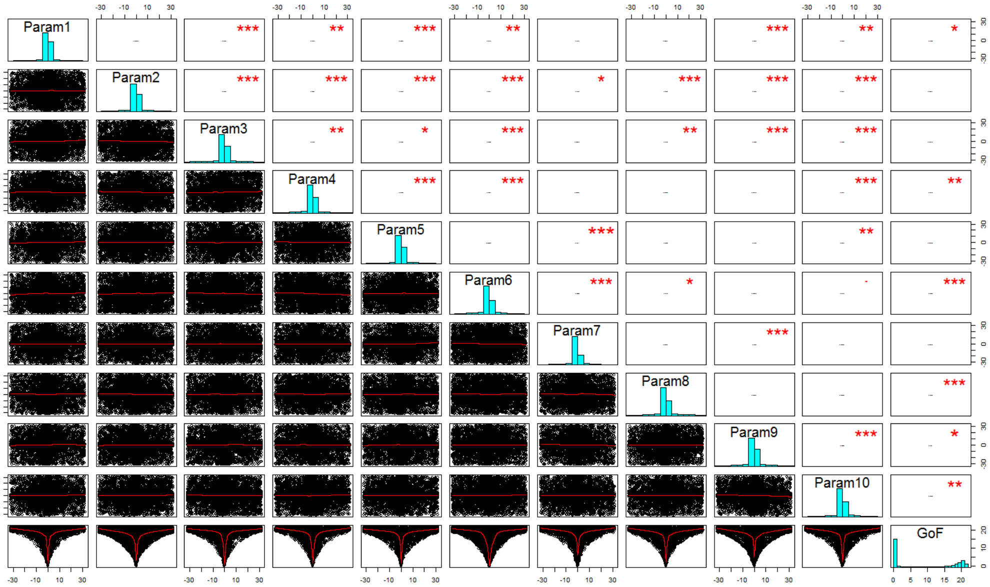

Figure 6: Matrix summarizing the interaction among parameters (e.g. cross correlation, histograms, and statistical significance of the correlation).

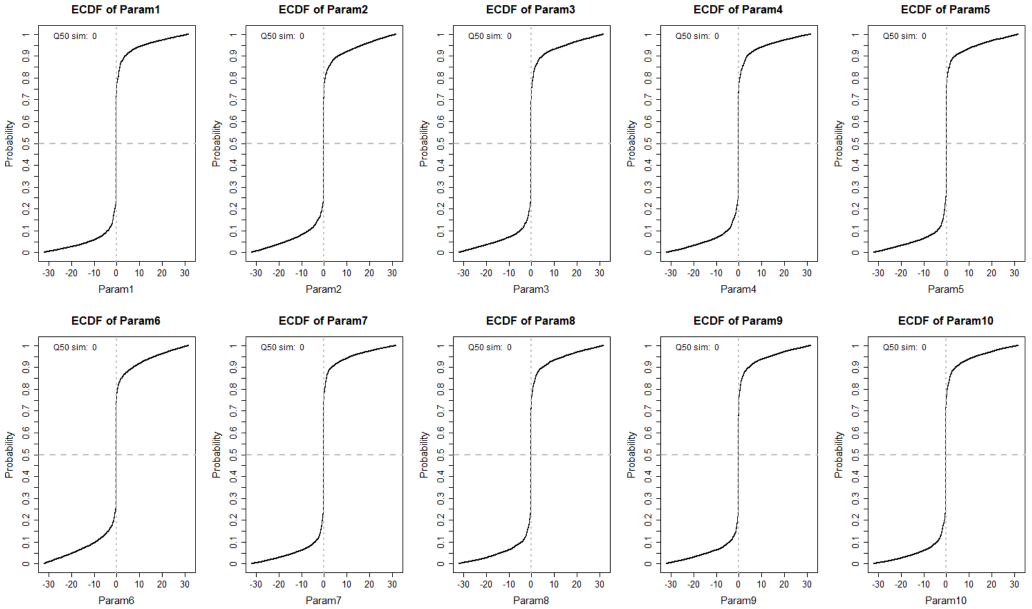

Figure 7: Empirical Cumulative Distribution Functions (ECDFs) for parameters.

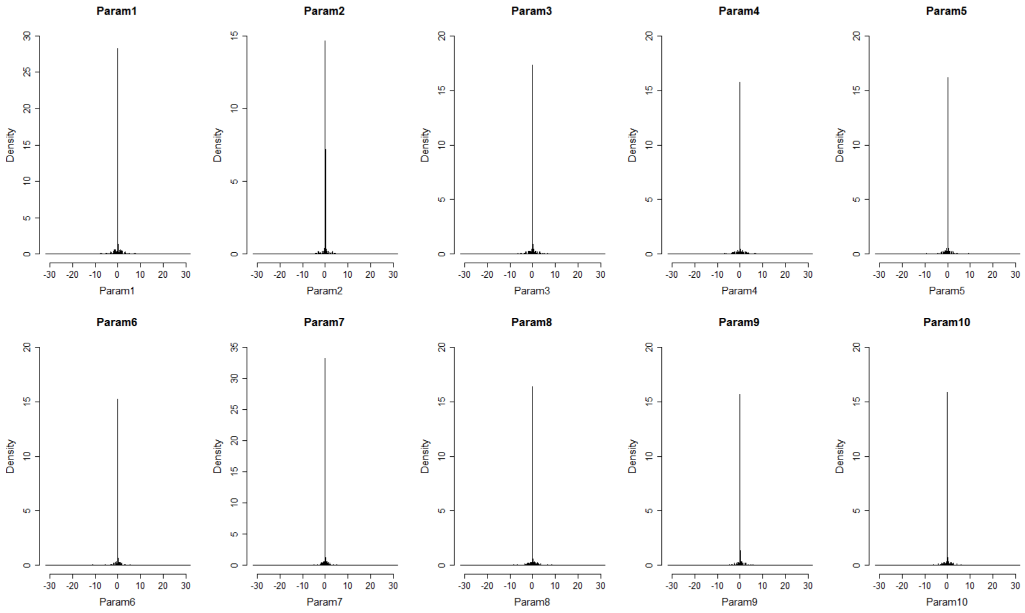

Figure 8: Histograms for parameters.

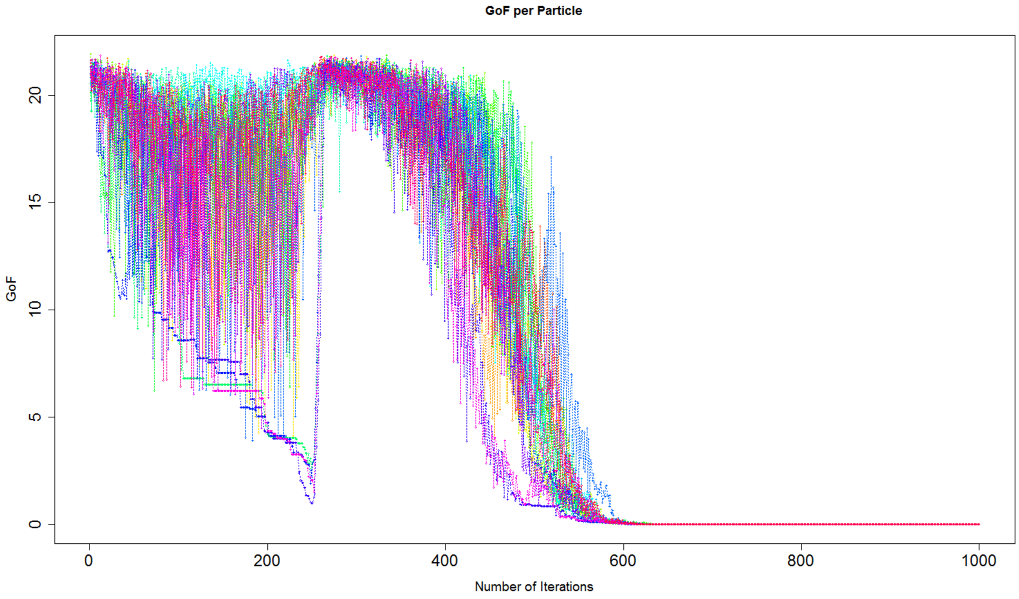

Figure 9: Goodness-of-fit measure for particles per iteration.

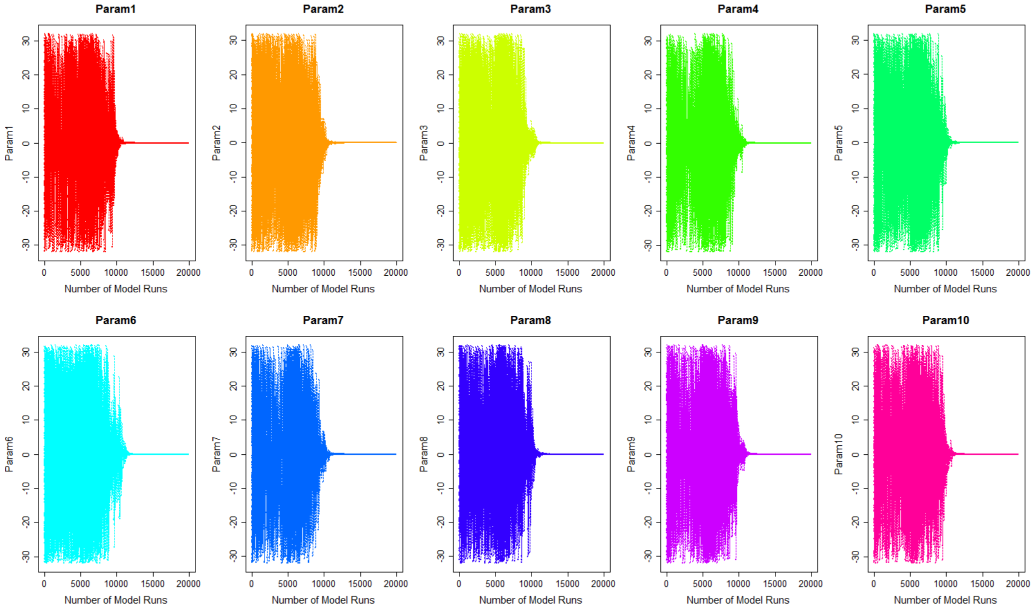

Figure 10: Parameter value per run.

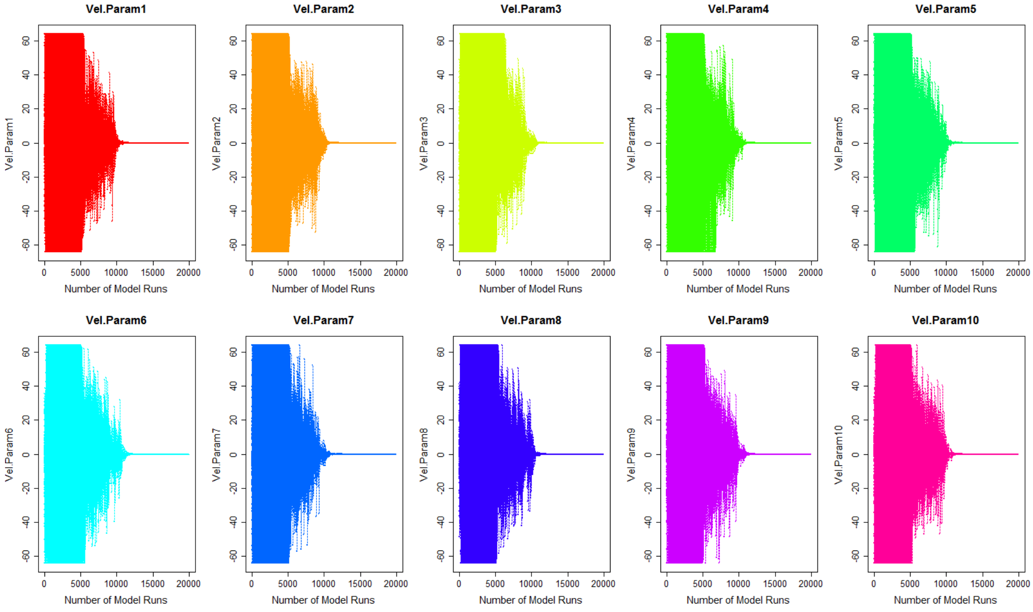

Figure 11: Particle velocities per run.

3.2.2 Optimisation of Griewank Function

Another commonly used benchmark is the Griewank function (Equation 9). This is similar to the Rastrigin function and shows a series of regularly distributed local optima, which makes the optimisation extremely challenging.

$$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \ ; \ -600 \leq x_i \leq 600 \ ; \ i = 1, 2, \ldots, n. \tag{9}$$

1. For the Griewank function we define upper and lower limits in [-600;600] and dimensionality d=10. For the optimisation we use 20 particles, 4000 iterations and the gbest topology:

```
> lower <- rep(-600,10)
> upper <- rep(600,10)
> set.seed(1111)
> hydroPSO(fn="griewank",lower=lower,upper=upper,
+          control=list(npart=20,maxit=4000,topology=
+          "gbest",REPORT=100))

$par
       Param1        Param2        Param3        Param4
 1.007925e-08 -6.167518e-09  4.463111e-09 -2.376661e-08
       Param5        Param6        Param7        Param8
-5.843686e-09 -3.135225e-08  1.800036e-08  7.965387e-09
       Param9       Param10
-1.675298e-08 -8.341296e-09

$value
[1] 2.220446e-16

$best.particle
[1] 3

$counts
function.calls     iterations    regroupings
         44040           2202              0

$convergence
[1] 1
```

```
$message
[1] "Converged ('reltol' criterion)"
```

For this example, the `reltol` criterion for convergence, which depends on the numerical characteristics of the machine where R is running, is met. `reltol` is defined as `sqrt(.Machine$$double.eps)`, i.e. the smallest positive floating-point number. Solution for the optimisation is reached at iteration 2202.

2. Using the `vonNeumann` topology:

```
> set.seed(1111)
> hydroPSO(fn="griewank",lower=lower,upper=upper,
+          control=list(npart=20,maxit=4000,topology=
+          "vonNeumann",REPORT=100))

$par
       Param1        Param2        Param3        Param4
 8.954073e-09 -1.567106e-08  9.036053e-09  3.216312e-09
       Param5        Param6        Param7        Param8
-9.106404e-09  2.246054e-08  5.931520e-09  6.022555e-09
       Param9       Param10
 4.727386e-09  2.657290e-08

$value
[1] 1.110223e-16

$best.particle
[1] 5

$counts
function.calls     iterations    regroupings
         45740           2287              0

$convergence
[1] 1

$message
[1] "Converged ('reltol' criterion)"
```

Again for this case, the `reltol` criterion for convergence is achieved at iteration 2287.

3. Defining a time-variant inertia weight between [1.2;0.4] and a non-linear (exp=1.5) time-variant c_1 coefficient between [1.28;1.05]:

   ```
   > set.seed(1111)
   > hydroPSO(fn="griewank",lower=lower,upper=upper,
   +          control=list(npart=20,maxit=4000,topology=
   +          "vonNeumann",use.IW=TRUE,IW.type="linear",
   +          IW.w=c(1.2,0.4),use.TVc1=TRUE,TVc1.type=
   +          "non-linear",TVc1.rng=c(1.28,1.05),
   +          TVc1.exp=1.5,REPORT=100))

   $par
          Param1        Param2        Param3        Param4
    1.608403e-09  5.432056e-09 -9.291102e-10  1.795820e-08
          Param5        Param6        Param7        Param8
    1.038647e-09  3.092320e-08  2.993544e-09 -2.248305e-08
          Param9       Param10
    1.747462e-08 -2.864754e-09

   $value
   [1] 1.110223e-16

   $best.particle
   [1] 18

   $counts
   function.calls     iterations    regroupings
            45360           2268              0

   $convergence
   [1] 1

   $message
   [1] "Converged ('reltol' criterion)"
   ```

   Again for this case, the reltol criterion for convergence is achieved at iteration 2268.

4. We use here the fips PSO variant with a gbest topology and a velocity limiting factor (lambda) of 0.5:

   ```
   > set.seed(1111)
   > hydroPSO(fn="griewank",lower=lower,upper=upper,
   ```

```
+           method="fips",control=list(npart=20,maxit=4000,
+           topology="gbest",use.IW=TRUE,IW.type="linear",
+           IW.w=c(1.2,0.4),use.TVc1=TRUE,TVc1.type=
+           "non-linear",TVc1.rng=c(1.28,1.05),
+           TVc1.exp=1.5,lambda=0.5,REPORT=100))

$par
     Param1      Param2      Param3      Param4      Param5
 6.17381768 -8.97080283 -0.21508208  6.38061372 -0.06824539
     Param6      Param7      Param8      Param9     Param10
 0.35711934  0.30051437 -0.18409341 -8.57713021 -1.09774674

$value
[1] 0.1878415

$best.particle
[1] 10

$counts
function.calls     iterations    regroupings
         80000           4000              0

$convergence
[1] 3

$message
[1] "Maximum number of iterations reached"
```

5. From the R console output we see premature convergence around iteration 1800 for a NSR ca. 10^{-9}. One option implemented in *hydroPSO* to tackle this problem corresponds to the “regrouping strategy” developed by *Evers and Ghalia* (2009). For this case we active the regrouping strategy (`use.RG`) when the NSR is smaller than a threshold (`RG.thr`) defined as 10^{-8}:

```
> set.seed(1111)
> hydroPSO(fn="griewank",lower=lower,upper=upper,
+           method="fips",control=list(npart=20,maxit=4000,
+           topology="gbest",use.IW=TRUE,IW.type="linear",
+           IW.w=c(1.2,0.4),use.TVc1=TRUE,TVc1.type=
+           "non-linear",TVc1.rng=c(2.2,1.8),TVc1.exp=1.5,
+           use.RG=TRUE,RG.thr=1e-8,lambda=0.5,REPORT=100))
```

```
$par
       Param1        Param2        Param3        Param4 
-3.1401659883 -0.0001477691 -5.4336512313  0.0003557016 
       Param5        Param6        Param7        Param8 
-0.0001809594 -0.0001946605  0.0012117211  0.0007441769 
       Param9       Param10 
 0.0010566101 -0.0022655021 

$value
[1] 0.00985781

$best.particle
[1] 14

$counts
function.calls     iterations    regroupings 
         80000           4000              5 

$convergence
[1] 3

$message
[1] "Maximum number of iterations reached"
```

From the results we see that the regrouping strategy allows particles escaping from stagnation and finding a new optimum (9.9×10^{-3}), which is better than the optimization without regrouping (2.7×10^{-2}) for the same number of iterations (`maxit=4000`).

6. By setting the working directory to `PSO.out` and using the `read_convergence` *hydroPSO* function we can directly assess the results from the optimization as function of the iterations:

```
> setwd("PSO.out")
> read_convergence(beh.thr=0.01,MinMax="min",do.png=TRUE,
+             png.fname="ConvergenceMeasuresRegrouping.png")
```

Figure 12 shows the effect of the regrouping strategy for iterations with an optimised value smaller than 0.01. In this figure we observe the first stagnation occurring around iteration 1900, and the corresponding triggering of the regrouping for NSR values smaller than 10^{-8}. After the first triggering an initial exploration stage is activated until a better optimum is found (ca. 3450 it.), where again a second stagnation is observed. This whole process is repeated 5 times before reaching the maximum number of iterations.

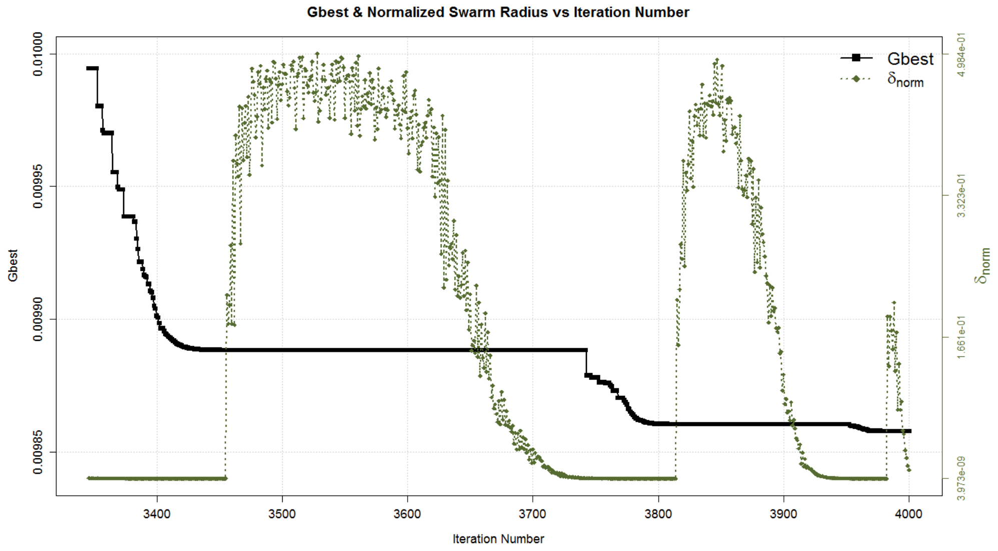

Figure 12: Effect of regrouping strategy on the global best (Gbest) and the Normalized Swarm Radius (NSR) versus iteration number.

hydroPSO has been validated against the Standard PSO 2007 algorithm developed by *Clerc et al.* (2011) employing five test functions commonly used to assess the performance of optimisation algorithms. Validation indicates that both the Standard PSO 2007 and *hydroPSO* produce comparable average results for fixed boundary condition, topology, inertia weight and number of iterations. For a detailed validation analysis we refer the reader to *Zambrano-Bigiarini and Rojas* (2012).

Finally, here we have illustrated a few options to boost the performance of the *hydroPSO* package or to adapt the optimisation engine to different problems (e.g. premature convergence). We must note, however, that a successful optimisation for a given model code is most likely a proper combination of modeller's expertise and a versatile optimisation engine.

4 Calibration of a Semi-Distributed Hydrological Model Using *hydroPSO*

4.1 Hydrological System and Conceptualization

The Ega River is a tributary of the Ebro River and originates in Álava (Cantabrian mountain range) flowing through the province of Navarra (see Figure 13). It has an area of 1445 km^2 and elevations ranging from 300 to 1400 m above sea level (a.s.l.) (*CHE*, 2000). For the implementation of the *hydroPSO* package we concentrate on the headwater of the Ega catchment. This upper catchment has an area of 808 km^2, mean annual precipitation of ca. 818 mm year^{-1} and mean daily discharge equal to 12.5 m^3 s^{-1} measured in Ega en Estella (Q071 in Figure 13) for the period 1961-1990.

Simulated daily discharges were obtained with the Soil and Water Assessment Tool (SWAT) 2005 (*Arnold et al.*, 1998; *Arnold and Fohrer*, 2005), which is a basin scale, physically-based, continuous-time hydrological model operating on a daily time step. Model components include weather, hydrology, erosion/sedimentation, and diverse components for the plant-soil-nutrients system. For a full overview the reader is referred to *Arnold et al.* (1998) and *Neitsch et al.* (2005).

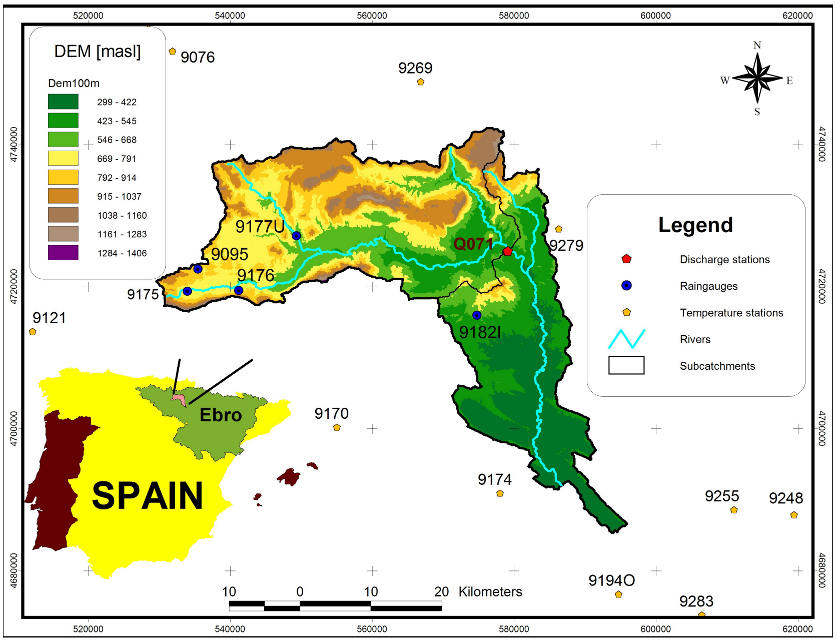

Figure 13: Location of the Ega River catchment and gauging station (Q071) used for calibration.

We set up SWAT-2005 with the modified SCS curve number for computing surface runoff, the Priestley-Taylor method for computing the evapotranspiration (ETP), and the Muskingum procedure for the channel routing (see *Neitsch et al.*, 2005). The AVSWAT-X GIS interface (see *Di Luzio et al.*, 2004) was used to prepare all the input files required by SWAT-2005. Information on weather, topography, soil properties, and land use in the study area was provided by the Confederación Hidrográfica del Ebro (CHE). Dominant soils in the catchment are marlstones, argillaceous marlstones, and breccia, whereas dominant land uses are forest (57.5%), pasture (39.0%), agriculture (2.95%), rocks (0.4%) and urban areas (0.12%). Precipitation estimates were obtained from interpolation of daily data in four rain gauges (P9175, P9176, P9095, P9177U see Figure 13).

SWAT-2005 contains numerous parameters describing processes where hydrology, water quality, and the soil-plant system interact. Table 1 shows a subset of 22 parameters (potentially) relevant for hydrological simulations only. These parameters are located in different files required for SWAT-2005 (see column Location in Table 1) and, thus, this tutorial illustrates how to interface *hydroPSO* and a model code with multiple controlling files (see Section 4.2). As explained later, only sensitive parameters were selected for calibrating the hydrological model for the Ega headwater catchment.

Table 1: Parameters of the SWAT-2005 model relevant for hydrological simulation. Range and sensitivity ranking constitute the basis for the implementation of the *hydroPSO* package to calibrate SWAT-2005 in the upper Ega catchment.

Parameter		SWAT-2005	Range		Default	Sensitivity
		File	Min	Max	Value	Ranking[1]
Baseflow alpha factor [days]	ALPHA_BF	*.gw	1.00e-01	9.90e-01	4.80e-02	1
Manning's "n" value for the main channel [-]	CH_N2	*.rte	1.60e-02	1.50e-01	1.40e-02	2
Initial SCS CN II value [-]	CN2	*.mgt	4.00e+01	9.50e+01	5.21e+02[2]	3
Saturated hydraulic conductivity [mm/hr]	SOL_K	*.sol	1.00e-03	1.00e+03	4.28e+00[2]	4
Available water capacity [mm H2O/mm soil]	SOL_AWC	*.sol	1.00e-02	3.50e-01	1.20e-01[2]	5
Effective hydraulic conductivity in main channel alluvium [mm/hr]	CH_K2	*.rte	0.00e+00	2.00e+02	0.00e+00	6
Soil evaporation compensation factor [-]	ESCO	*.hru	1.00e-02	1.00e+00	9.50e-01	7
Surface runoff lag time [days]	SURLAG	*.bsn	1.00e+00	1.20e+01	4.00e+00	8
Snowfall temperature [°C]	SFTMP	*.bsn	-5.00e+00	5.00e+00	1.00e+00	9
Snowmelt base temperature [°C]	SMTMP	*.bsn	-5.00e+00	5.00e+00	5.00e-01	10[3]
Minimum melt factor for snow [°C]	SMFMN	*.bsn	1.40e+00	6.90e+00	4.50e+00	11[3]
Snowpack temperature lag factor [-]	TIMP	*.bsn	1.00e-02	1.00e+00	1.00e+00	12[3]
Maximum melt factor for snow [°C]	SMFMX	*.bsn	1.40e+00	6.90e+00	4.50e+00	13[3]
Manning's "n" value for overland flow [-]	OV_N	*.hru	8.00e-03	6.00e-01	1.00e-01	14[3]
Deep aquifer percolation factor [-]	RCHRG_DP	*.gw	0.00e+00	1.00e+00	5.00e-02	15[3]
Threshold water depth in the shallow aquifer for flow [mm]	GWQMN	*.gw	0.00e+00	5.00e+03	0.00e+00	16[3]
Groundwater "*revap*" coefficient [-]	GW_REVAP	*.gw	0.00e+00	2.00e-01	2.00e-02	17[3]
Groundwater delay time [days]	GW_DELAY	*.gw	1.00e+00	1.00e+02	3.10e+01	18[3]
Moist soil albedo	SOL_ALB	*.sol	0.00e+00	1.00e-01	1.00e-02[2]	19[3]
Threshold water depth in the shallow aquifer for "*revap*" [mm]	REVAVMN	*.gw	1.00e+00	5.00e+02	1.00e+00	22[3]
Plant uptake compensation factor [-]	EPCO	*.bsn	1.00e-02	1.00e+00	1.00e+00	22[3]
Maximum canopy storage [mm H_2O]	CANMX	*.hru	0.00e+00	1.00e+01	0.00e+00	22[3]

[1] Sensitivity analysis based on LH-OAT (see Section 4.4.1).

[2] Default values based on the study area information.

[3] Insensitive parameters obtained from the LH-OAT analysis.

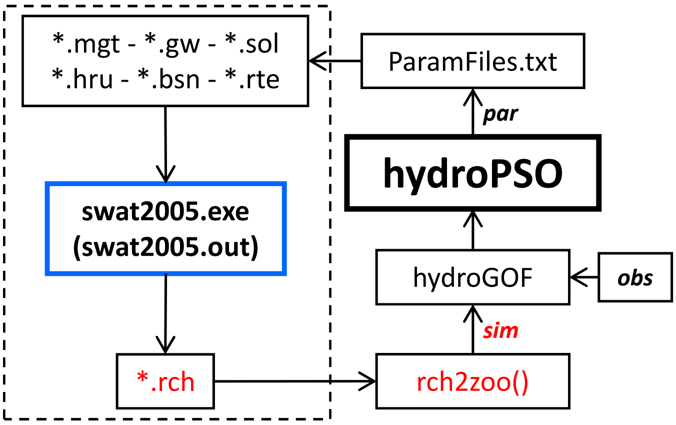

Figure 14: Interaction of *hydroPSO* with SWAT-2005 and main I/O wrapper functions defined.

4.2 Interfacing *hydroPSO* and SWAT-2005

The interaction between SWAT-2005 and *hydroPSO* is shown in Figure 14. Here, SWAT-2005 is executed by `swat2005.exe` (or `swat2005.out` under GNU/Linux), which reads several input files (e.g. *.mgt, *.gw, *.sol, *.hru, *.bsn, and *.rte) containing the parameters to be calibrated. `swat2005.exe` executes the SWAT-2005 model code and produces the *.rch file, which contains simulated discharges for defined river reaches. River discharges simulated by SWAT-2005 are then read by the *hydroPSO* function `rch2zoo`, which transforms the time series of discharge values into a `zoo` R object (***sim***). Subsequently, the *hydroGOF* R package (see `http://cran.r-project.org/web/packages/hydroGOF/`) is used to calculate a Nash-Sutcliffe Efficiency (NSE) (see *Nash and Sutcliffe*, 1970) as goodness-of-fit measure (Note that loading the *hydroGOF* R package gives the user a full suite of goodness-of-fit measures such as: root mean square error (rms), normalized rms (nrms), Pearson correlation coefficient, (r), coefficient of determination (R2), modified NSE (mNSE), index of agreement (d), coefficient of persistence (cp), percent bias (pbias), Kling-Gupta Efficiency (KGE) among others). NSE is used to assess the quality of the current parameter set (particles' positions) by *hydroPSO*. Then, *hydroPSO* updates the current particles' positions (parameter set) on the basis of the current NSE and the new updated parameter values (best particles' positions) are written into the corresponding files defined in *ParamFiles.txt*.

4.3 Definition of *ParamFiles.txt* and *ParamRanges.txt* files

The basic interaction between *hydroPSO* and SWAT-2005 is defined through the specification of, first, the names and location of the parameters to be calibrated and, second, the definition of meaningful parameter ranges. This information is provided by two (problem-specific) ASCII files, namely, *ParamFiles.txt* and *ParamRanges.txt*, which must be contained in the `./PSO.in` folder. In general, if a sensitivity analysis is implemented resulting in the identification of (in)sensitive parameters, the *hydroPSO* user will have to define two sets of interfacing files, one for the sensitivity analysis (most likely including all parameters considered) and other for the calibration stage (most likely including only sensitive parameters). In this study case, *ParamFiles-Sens.txt* and *ParamRanges-Sens.txt* are defined as the interfacing files used for the sensitivity analysis, which correspond to the full set of hydrology-related parameters listed in Table 1. Note that care must be taken in numbering and naming the parameters as they require to be consistent in both files.

ParamFiles-Sens.txt

```
ParameterNmbr  ParameterName  Filename       Row.Number Col.Start Col.End DecimalPlaces
1              CN2            000010001.mgt  11          4        16      5
1              CN2            000020001.mgt  11          4        16      5
2              ESCO           basins.bsn     13          4        16      3
3              SURLAG         basins.bsn     20          4        16      3
4              RCHRG_DP       000010001.gw    9          1        16      7
4              RCHRG_DP       000020001.gw    9          1        16      7
5              GWQMN          000010001.gw    6          1        16      7
5              GWQMN          000020001.gw    6          1        16      7
6              GW_REVAP       000010001.gw    7          1        16      7
6              GW_REVAP       000020001.gw    7          1        16      7
7              REVAPMN        000010001.gw    8          1        16      7
7              REVAPMN        000020001.gw    8          1        16      7
8              GW_DELAY       000010001.gw    4          1        16      7
8              GW_DELAY       000020001.gw    4          1        16      7
9              ALPHA_BF       000010001.gw    5          1        16      7
9              ALPHA_BF       000020001.gw    5          1        16      7
10             SOL_K          000010001.sol  11         28        39      5
10             SOL_K          000010001.sol  11         40        51      5
10             SOL_K          000010001.sol  11         52        63      5
10             SOL_K          000020001.sol  11         28        39      5
10             SOL_K          000020001.sol  11         40        51      5
10             SOL_K          000020001.sol  11         52        63      5
11             SOL_AWC        000010001.sol  10         28        39      5
11             SOL_AWC        000010001.sol  10         40        51      5
11             SOL_AWC        000010001.sol  10         52        63      5
11             SOL_AWC        000020001.sol  10         28        39      5
11             SOL_AWC        000020001.sol  10         40        51      5
11             SOL_AWC        000020001.sol  10         52        63      5
12             CH_N2          000010000.rte   6          4        16      3
12             CH_N2          000020000.rte   6          4        16      3
13             CH_K2          000010000.rte   7          4        16      3
13             CH_K2          000020000.rte   7          4        16      3
14             OV_N           000010001.hru   5          4        16      3
```

```
14             OV_N           000020001.hru   5          4      16      3
15             SFTMP          basins.bsn      4          4      16      3
16             SMTMP          basins.bsn      5          4      16      3
17             SMFMX          basins.bsn      6          4      16      3
18             SMFMN          basins.bsn      7          4      16      3
19             TIMP           basins.bsn      8          4      16      3
20             EPCO           basins.bsn     14          4      16      3
21             CANMX          000010001.hru   9          4      16      3
21             CANMX          000020001.hru   9          4      16      3
22             SOL_ALB        000010001.sol  17         28      39      5
22             SOL_ALB        000010001.sol  17         40      51      5
22             SOL_ALB        000010001.sol  17         52      63      5
22             SOL_ALB        000020001.sol  17         28      39      5
22             SOL_ALB        000020001.sol  17         40      51      5
22             SOL_ALB        000020001.sol  17         52      63      5
```

ParamRanges-Sens.txt

```
ParameterNmbr  ParameterName  MinValue     MaxValue
1              CN2            40           95
2              ESCO           0.01         1
3              SURLAG         1            12
4              RCHRG_DP       0            1.0
5              GWQMN          0            5000
6              GW_REVAP       0            0.2
7              REVAPMN        1            500
8              GW_DELAY       1            100
9              ALPHA_BF       0.01         0.99
10             SOL_K          0.001        1000
11             SOL_AWC        0.01         0.35
12             CH_N2          0.016        0.150
13             CH_K2          0            200
14             OV_N           0.008        0.600
15             SFTMP          -5           5
16             SMTMP          -5           5
17             SMFMX          1.4          6.9
18             SMFMN          1.4          6.9
19             TIMP           0.01         1
20             EPCO           0.01         1
21             CANMX          0            10
22             SOL_ALB        0            0.1
```

4.4 Implementation Details and Results of the Calibration

4.4.1 Sensitivity Analysis

As first step, a sensitivity analysis is performed on the 22 parameters listed in Table 1. The `lhoat()` *hydroPSO* function allows the user to rank relevant parameters according to their impact on model predictions or performance using the Latin Hypercube One-factor-At-a-Time (LH-OAT) method developed by *van Griensven et al.* (2006). LH-OAT works by taking M LH sampling points (M strata for each parameter) and then varying by a fraction s each LH sampling point D times, where D is the number of parameters (i.e. problem dimensionality). For each LH sampling point a partial effect for each parameter is calculated and a final effect (impact on model predictions) is calculated by averaging these partial effects for each parameter. The method is very efficient requiring a total of $M(D+1)$ runs. Details of the sensitivity analysis implemented for SWAT-2005 are as follows:

1. Nash-Sutcliffe Efficiency (NSE) is used as a goodness-of-fit measure (*hydroGOF* R package).

2. Period for the analysis corresponds to 01-Jan-1962 to 31-Dec-1970 using a daily time step.

3. The number of strata for the LH sampling was defined as $M = 300$, whereas the fraction of variation was 10%, i.e. $s = 0.1$.

4. Observations used to assess the model performance are stored in the auxiliary file `SWAT_obs.txt`, whereas the auxiliary file `LHOAT-SWAT2005.R` implements the sensitivity analysis as follows:

LHOAT-SWAT2005.R

```
###Loading required libraries
library(hydroPSO)
library(hydroGOF)
library(hydroTSM)
library(SWAT2R)

###Definition of working directory: input, output and model files paths
model.drty <- "~/SWAT2005"
setwd(model.drty)
param.ranges <- paste(model.drty,"/PSO.in/ParamRanges-Sens.txt",sep="")

###Period of analysis (see "file.cio" SWAT file)
Sim.Ini="1962-01-01"
Sim.Fin="1970-12-31"
gof.Ini="1962-01-01"
gof.Fin="1970-12-31"

###Goodness-of-fit function, either customized or pre-defined from hydroGOF
gof.FUN <- "NSE"
gof.FUN.args <- list()

###Getting the OBSERVATIONS
q.obs <- read.zoo("SWAT_obs.txt")

###Arguments for the model to be assessed
model.FUN.args=list(
   model.drty=model.drty,
   param.files=paste(model.drty,"/PSO.in/ParamFiles-Sens.txt",sep=""),
   exe.fname="./swat2005.out",
   verbose=FALSE,
   stdout=FALSE,
   stderr=FALSE,
   ###Function for reading the simulated equivalents
   out.FUN="rch2zoo",
   out.FUN.args=list(
      file="output.rch",
      col.name="FLOW_OUTcms",
      out.type="Q",
      rchID=1,
      Date.Ini=Sim.Ini,
      Date.Fin=Sim.Fin,
      tstep="daily"), ###END out.FUN.args
   ###Function assessing the simulated equivalents against the observations
   gof.FUN=gof.FUN,
   gof.FUN.args=gof.FUN.args
   gof.Ini=gof.Ini,
   gof.Fin=gof.Fin,
   obs=q.obs,
) ###END model.FUN.args

###Main Latin-Hypercube One-factor-At-a-Time Sensitivity Analysis
lhoat(
   fn="hydromod",
   model.FUN="hydromod",
   model.FUN.args=model.FUN.args,
   control=list(
      N=300,
      f=0.1,
      drty.out="LH_OAT",
      param.ranges=param.ranges,
      gof.name="GoF",
      do.plots=FALSE,
      write2disk=TRUE,
      verbose= TRUE) ###END control options
) ###END lhoat
```

5. This sensitivity analysis requires a total of 6900 iterations ($M = 300$ and $D = 22$). In `LHOAT-SWAT2005.R` we define the keyword `NSE` from *hydroGOF* as goodness-of-fit measure as well as `SWAT_obs.txt` as the file containing the daily discharge observations. Within the arguments for the model code to be assessed, we must define the location of the `ParamFiles.txt` and the name of the executable file (`swat2005.out`). In addition, we specify the function for reading the simulated discharges (`out.FUN`) as `rch2zoo`, whereas the assessment of each simulation is done through `gof.FUN="NSE"`. For the `lhoat()` function we first define the keyword `hydromod`, which indicates that an external model code (i.e. not a pre-defined test function coded in *hydroPSO*) will be analysed. Arguments `N=300`, `f=0.1`, and `drty.out=LH_OAT` are used to define the number of strata, the fraction of variation for each parameter, and the name of the folder where results will be saved, respectively. `lhoat()` produces the file `LH_OAT-Ranking.txt`, which contains a ranking of parameters according to their relative importance.

`LH_OAT-Ranking.txt`

```
RankingNmbr ParameterName RelativeImportance
1           ALPHA_BF      5.707630e+02
2           CH_N2         2.238171e+02
3           CN2           1.874606e+02
4           SOL_K         1.433732e+02
5           SOL_AWC       1.298079e+02
```

```
6           CH_K2         9.871332e+01
7           ESCO          9.403977e+01
8           SURLAG        6.890757e+01
9           SFTMP         5.144755e+01
10          SMTMP         2.285713e+01
11          SMFMN         1.619080e+01
12          TIMP          6.051893e+00
13          SMFMX         3.735775e+00
14          OV_N          2.607991e+00
15          RCHRG_DP      4.295568e-01
16          GWQMN         3.301034e-01
17          GW_REVAP      1.705301e-01
18          GW_DELAY      6.303209e-02
19          SOL_ALB       1.192648e-02
20          REVAPMN       4.581639e-03
22          EPCO          0.000000e+00
22          CANMX         0.000000e+00
```

6. The ranking obtained from the sensitivity analysis is included in the last column of Table 1. We see from this table that 9 parameters are identified as sensitive using a NSE as goodness-of-fit measure for daily discharge simulations in the period 01-Jan-1962 to 31-Dec-1970 (several trials showed that the tenth parameter, `SMTMP`, is relatively insensitive, and for sake of clarity, we have excluded it from the subsequent analysis). In general, this ranking is in agreement with previous research (see e.g. *Holvoet et al.*, 2005; *Muleta and Nicklow*, 2005; *van Liew et al.*, 2005; *van Griensven et al.*, 2006; *Kannan et al.*, 2007; *van Liew et al.*, 2007), and this subset of parameters constitute the basis for the calibration of the SWAT-2005 model. The resulting files (`ParamFiles.txt` and `ParamRanges.txt`) used to interface *hydroPSO* and SWAT-2005 in the calibration stage are as follows:

ParamFiles.txt

```
ParameterNmbr ParameterName Filename      Row.Number Col.Start Col.End DecimalPlaces
1             CN2           000010001.mgt 11          4        16      5
1             CN2           000020001.mgt 11          4        16      5
2             ESCO          basins.bsn    13          4        16      3
3             SURLAG        basins.bsn    20          4        16      3
4             ALPHA_BF      000010001.gw   5          1        16      7
4             ALPHA_BF      000020001.gw   5          1        16      7
5             SOL_K         000010001.sol 11         28        39      5
5             SOL_K         000010001.sol 11         40        51      5
5             SOL_K         000010001.sol 11         52        63      5
5             SOL_K         000020001.sol 11         28        39      5
5             SOL_K         000020001.sol 11         40        51      5
5             SOL_K         000020001.sol 11         52        63      5
6             SOL_AWC       000010001.sol 10         28        39      5
6             SOL_AWC       000010001.sol 10         40        51      5
6             SOL_AWC       000010001.sol 10         52        63      5
6             SOL_AWC       000020001.sol 10         28        39      5
6             SOL_AWC       000020001.sol 10         40        51      5
6             SOL_AWC       000020001.sol 10         52        63      5
7             CH_N2         000010000.rte  6          4        16      3
7             CH_N2         000020000.rte  6          4        16      3
```

```
8             CH_K2          000010000.rte   7          4        16      3
8             CH_K2          000020000.rte   7          4        16      3
9             SFTMP          basins.bsn      4          4        16      3
```

```
ParamRanges.txt
ParameterNmbr  ParameterName  MinValue     MaxValue
1              CN2            40           95
2              ESCO           0.01         1
3              SURLAG         1            12
4              ALPHA_BF       0.01         0.99
5              SOL_K          0.001        1000
6              SOL_AWC        0.01         0.35
7              CH_N2          0.016        0.150
8              CH_K2          0            200
9              SFTMP          -5           5
```

4.4.2 Calibration

After obtaining a ranking with the most sensitive parameters, the calibration of the river discharges for the EGA headwater catchment proceeded as follows:

1. Files interfacing *hydroPSO* and SWAT-2005 (i.e. ParamFiles.txt and ParamRanges.txt) are stored in the ./PSO.in folder, within the directory containing all the required files to run SWAT-2005, which for this tutorial is ./SWAT2005.

2. Several auxiliary files (described below) are included in ./SWAT2005:

   ```
   Auxiliary Files in SWAT2005
   hydroPSO-SWAT2005.R -> Main R script to run hydroPSO
   SWAT_obs.txt -> ASCII file with the discharge observations
   ```

   As previously explained (see Section 4.4.1), the file LH_OAT-Ranking.txt is saved in the folder ./SWAT2005/LH_OAT.

3. The setup for the calibration of the Ega headwater catchment is defined in the hydroPSO-SWAT2005.R script. By default all the results from hydroPSO are saved into the PSO.out folder, however, this can be redefined by using the drty.out argument.

   ```
   hydroPSO-SWAT2005.R
   ####################################################################################
   ## Example to interface SWAT-2005 with hydroPSO. This script allows hydroPSO to    #
   ## take control over the execution of SWAT-2005 through the definition of a batch  #
   ## file (run_me.bat) and a series of simple I/O R scripts                          #
   ##                                                                                 #
   ## Part of the hydroPSO R package                                                  #
   ## http://www.rforge.net/hydroPSO/ http://cran.r-project.org/web/packages/hydroPSO #
   ## Copyright 2011-2012 Mauricio Zambrano-Bigiarini & Rodrigo Rojas                 #
   ## Distributed under GPL 2 or later                                                #
   ```

```
##                                                                                    #
## Created by Mauricio Zambrano-Bigiarini and Rodrigo Rojas. 26-Oct-2011            #
## Last saved: 13-Feb-2012                                                          #
#######################################################################################

###Loading required libraries
library(hydroPSO)
library(hydroGOF)
library(hydroTSM)
library(SWAT2R)

###Definition of working directory: input, output and model files paths
model.drty <- "~/SWAT2005"
setwd(model.drty)

###Period of analysis
Sim.Ini="1962-01-01"
Sim.Fin="1970-12-31"
gof.Ini="1962-01-01"
gof.Fin="1970-12-31"

###Goodness-of-fit function, either customized or pre-defined from hydroGOF
gof.FUN <- "NSE"
gof.FUN.args <- list()

###Getting the OBSERVATIONS
q.obs <- read.zoo("SWAT_obs.txt")

###MAIN model function
model.FUN.args=list(
   model.drty=model.drty,
   param.files=paste(model.drty,"/PSO.in/ParamFiles.txt",sep=""),
   exe.fname="./swat2005.out",
   verbose=FALSE,
   stderr=FALSE,
   ###Function for reading the simulated equivalents
   out.FUN="rch2zoo",
   out.FUN.args=list(
      file="output.rch",
      col.name="FLOW_OUTcms",
      out.type="Q",
      rchID=1,
      Date.Ini=Sim.Ini,
      Date.Fin=Sim.Fin,
      tstep="daily",
      verbose=FALSE), ###END out.FUN.args
   ###Function for assessing the simulated equivalents against the observations
   gof.FUN=gof.FUN,
   gof.FUN.args=gof.FUN.args,
   gof.Ini=gof.Ini,
   gof.Fin=gof.Fin,
   obs=q.obs,
) ###END model.FUN.args

###MAIN PSO ALGORITHM
###For hydroPSO fine-tuning parameters, see Zambrano-Bigiarini and Rojas,2012
set.seed(100)
hydroPSO(
   fn="hydromod",
   model.FUN="hydromod",
   model.FUN.args=model.FUN.args,
   control=list(
      param.ranges="ParamRanges-09params-sub090.txt",
      MinMax="max",
      npart=20,
      maxit=2000,
      lambda=1,
      c2=0.5+log(2),
      use.IW=TRUE,IW.type="linear",IW.w=1/(2*log(2)),IW.exp=1,
      use.TVc1=TRUE,TVc1.type="non-linear",TVc1.rng=c(1.28,1.05),TVc1.exp=1.5,
      use.TVlambda=TRUE,TVlambda.type="linear",TVlambda.rng=c(1.0,0.5),TVlambda.exp=
      1,
      topology="random",K=3,
      boundary.wall="reflecting",
      write2disk=TRUE,
      REPORT=5,
      verbose=TRUE
   ) ###END control options
) ###END MAIN hydroPSO ALGORITHM
```

4. In the `hydroPSO-SWAT2005.R` script we first load all the required libraries, setup the working directory, and define the simulation period as well as the period for calculating the goodness-of-fit measure (these two could be different depending on the nature of the application). To assess the performance of each particle in *hydroPSO* we use the Nash-Sutcliffe Efficiency (NSE) implemented in the *hydroGOF* package as goodness-of-fit measure. Discharge observations used to assess the corresponding simulated equivalents are read as a `zoo` object in R (to take advantage of time attributes) from the `SWAT_obs.txt` ASCII file. Then, we define all arguments related to the main model code to be calibrated. Note that these arguments are identical as for the sensitivity analysis (`LHOAT-SWAT2005.R`) and, so, they will not be repeated here. For the main *hydroPSO* algorithm, the keyword `hydromod` **must** be used for the argument `fn` when a function different from a predefined test function is calibrated. This will indicate *hydroPSO* to expect for external files containing the model executable(s), input(s), an ouput(s) files. For this example, we use the traditional `pso` (default) algorithm to maximize (`MinMax="max"`) the NSE calculated by *hydroGOF* (`gof.FUN="NSE"`). A swarm of 20 particles (`npart=20`) considering 2000 iterations (`maxit=2000`) for the algorithm is defined. A constant inertia weight equal to `1/(2*log(2))` and a non-linear (`TVc1.type="non-linear"`) time-variant cognition coefficient (c_1) (`use.TVc1=TRUE`) in the range [1.28,1.05] with exponent 1.5 (`TVc1.exp=1.5`) are defined. In addition, we improve on the definition of the factor clamping the velocities (`lambda`) by using a linear variation between [1.0,0.5]. Particles interact following the `random` topology with 3 informants. Finally, results are saved by default in the folder `./PSO.out`.

5. Plotting the results (Figures 15 to 26). Using the `plot_results()` function results are saved directly to the folder `./PSO.out/pngs`:

   ```
   > plot_results(do.png=TRUE)
   ```

   Although it is not the aim of this tutorial to provide an extensive analysis of the hydrological calibration for the Ega headwater catchment, we briefly discuss the results obtained from *hydroPSO*. Figure 15 shows the evolution of the the Gbest, i.e. the Nash-Sutcliffe Efficiency, as a function of the iteration number. We see an initial exploration phase (up to ca. iter=120) that stabilizes after iteration 200 around a value of NSE=0.77. At the same time, the Normalized Swarm Radius indicates a clear convergence to the attraction zone around this optimum. Although not shown here, several trials (including different number of iterations and particles for the swarm) indicate that this solution is most likely the global optimum for the calibration.

   A dotty-plot for the different (sensitive) parameters listed in Table 1 is shown in Figure 16. From this figure we see a good identification for all parameters, except ESCO and SURLAG, which show a relatively flat response surface. Figure 17 complements the previous one by showing a 2-Dimensional projection of the goodness-of-fit surface response (NSE) for different parameter pairs. Figure 18, in turn, summarizes the interaction among all (sensitive) parameters listed in Table 1. Here a clear correlation between the NSE and CH_N2, CH_K2 is observed, whereas a statistically significant cross-correlation is observed between CH_K2 (CH_N2) and ALPHA_BF. Empirical Cumulative Distribution Functions (ECDFs) (Figure 19) and histograms (Figure 20) for the parameters show a good specification with reduced uncertainty around the median and some for the extreme quantiles. Figures 21, 22, and 23 provide, in turn, useful information to assess the performance and convergence of the NSE, parameters, and velocities per iteration number. By analysing these figures we conclude that the *hydroPSO* package converged to a suitable attraction zone.

   Figure 24 provides the daily and monthly time-series summarizing the calibration together with a series of performance indicators. In general, we see a slight underestimation of recession and peak-flows (PBIAS=-2.8%) with a good agreement of average flows (MAE=4.67) and a final optimised Nash-Sutcliffe Efficiency of 0.78. At the same time, Figure 25 shows a scatter plot of simulated versus observed daily discharges. Finally, Figure 26 shows different user-defined quantiles including an estimation of the percentage bias for the specified quantiles.

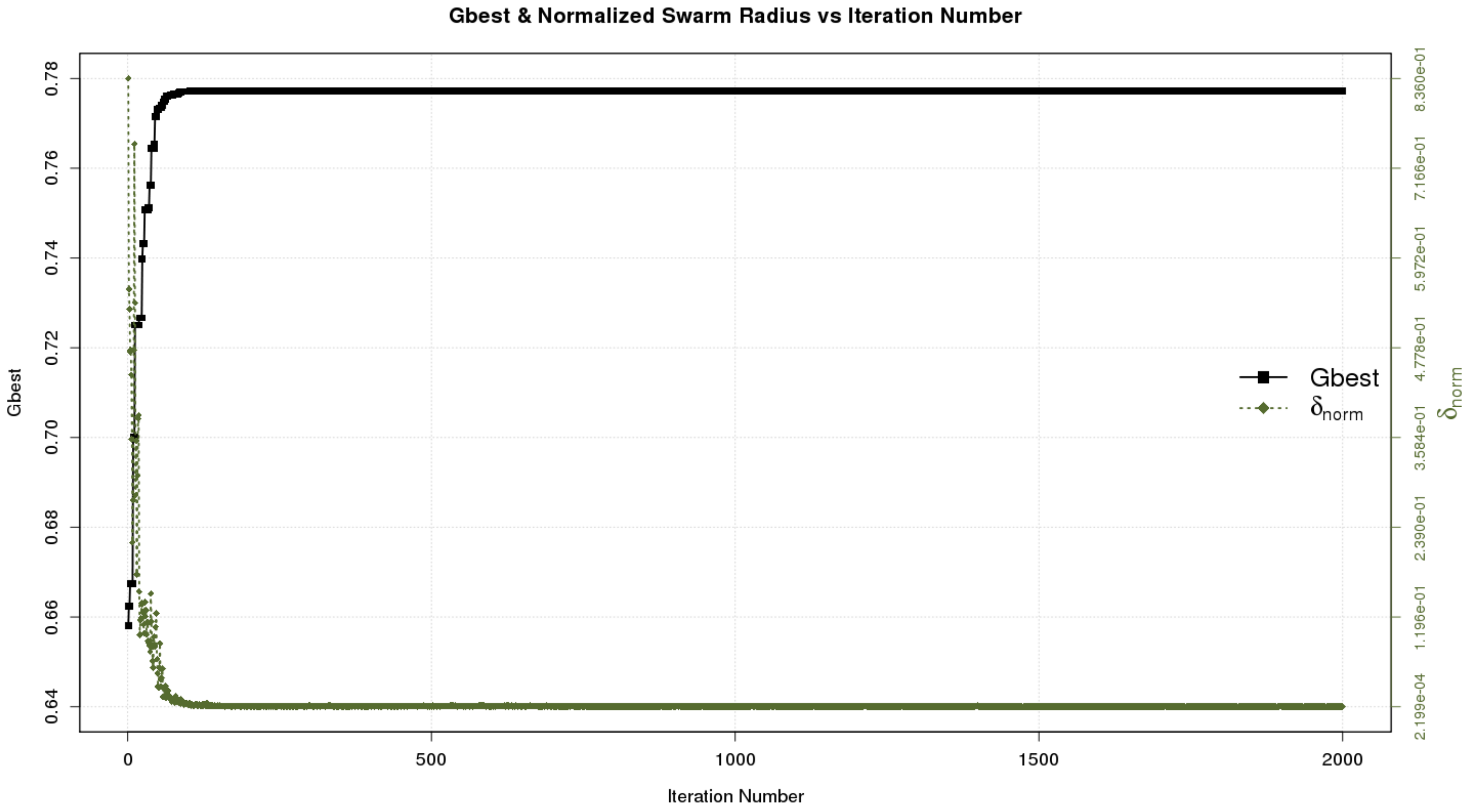

Figure 15: Evolution of the global best (Gbest) and the Normalized Swarm Radius (NSR) versus iteration number for the calibration of the Ega headwater catchment.

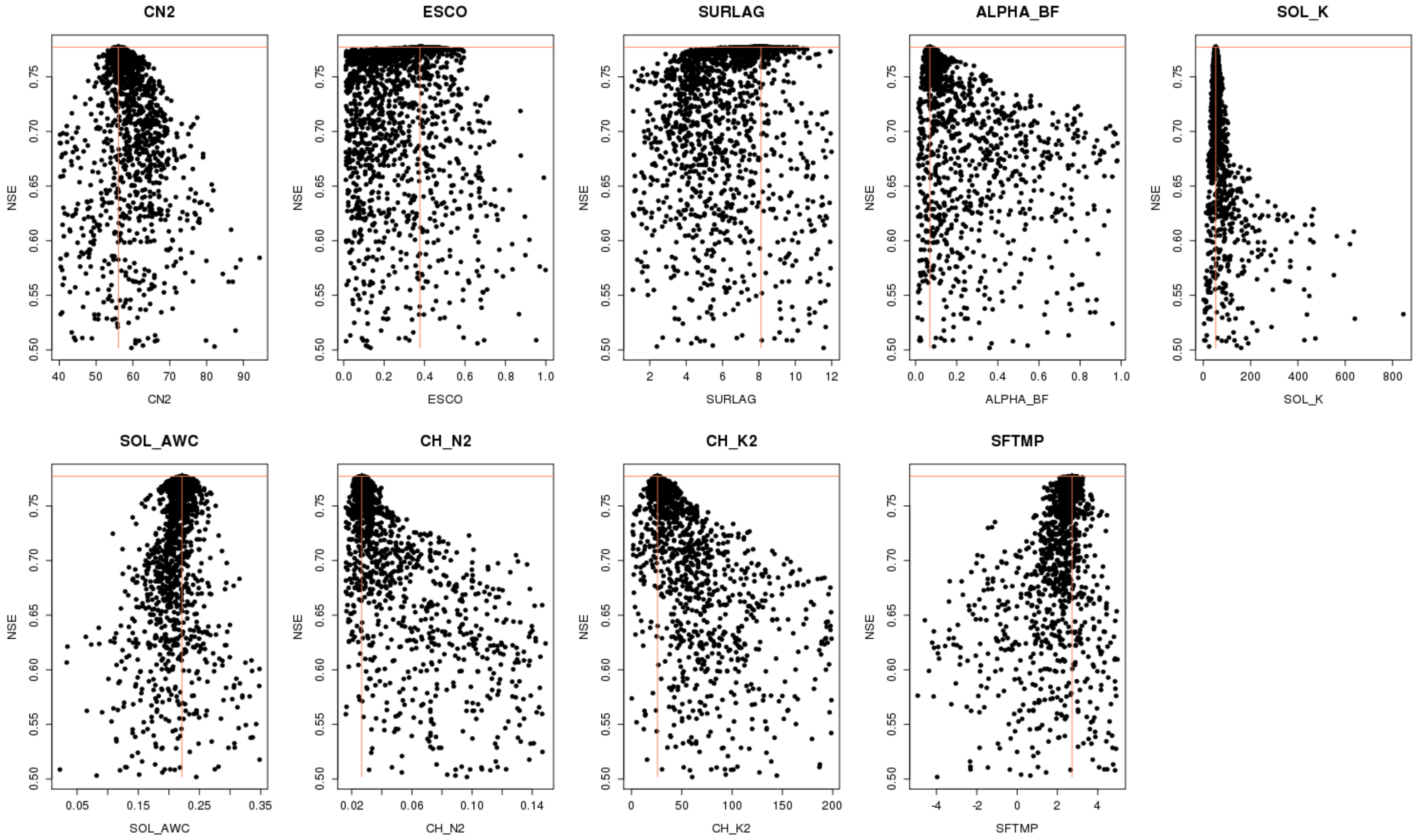

Figure 16: Dotty-plots for the (sensitive) parameters listed in Table 1. Vertical and horizontal coloured lines show the location of the highest goodness-of-fit value (Nash-Sutcliffe Efficiency).

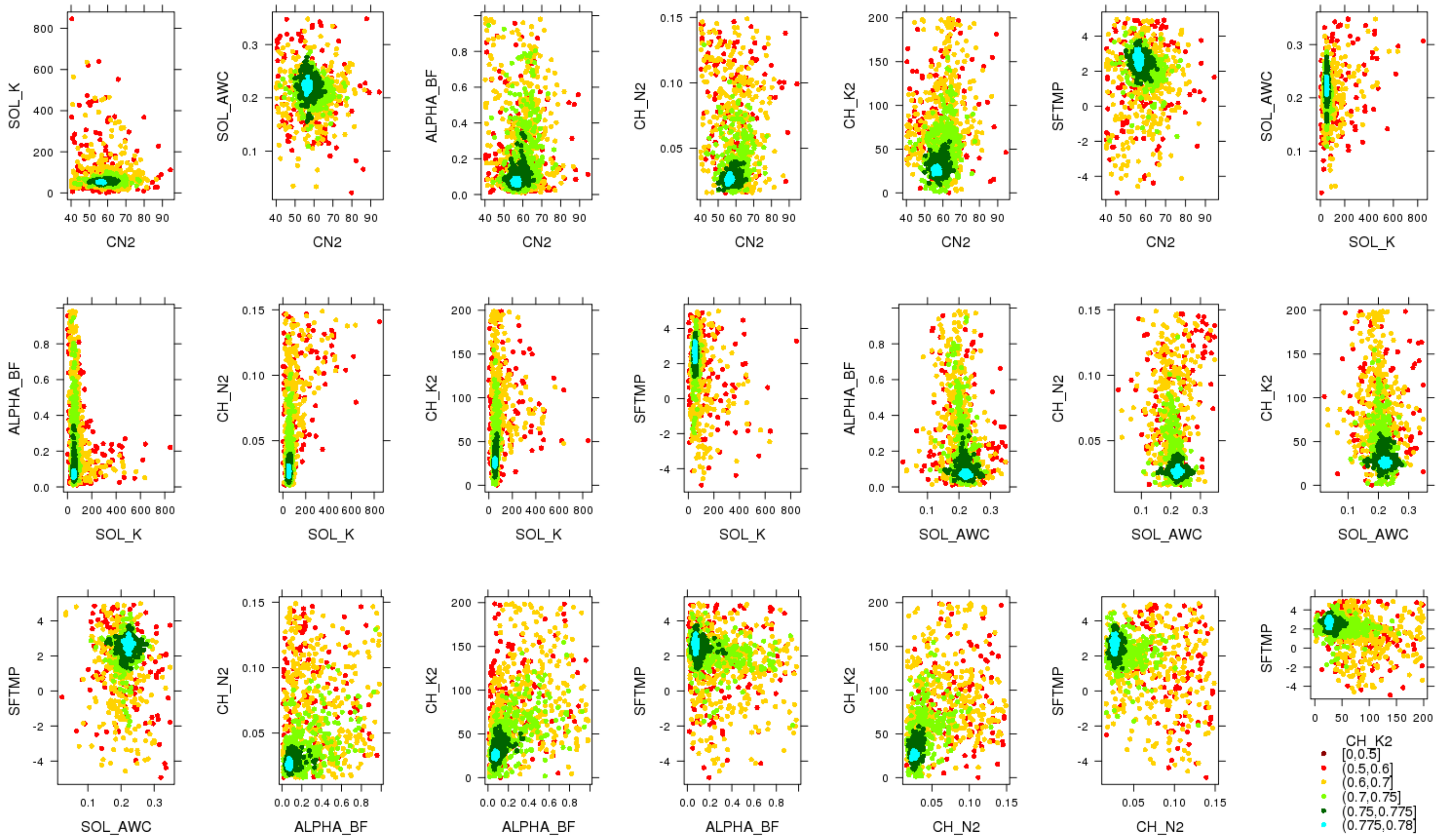

Figure 17: 2-dimensional projected dotty-plots highlighting the interaction among the first seven (sensitive) parameters listed in Table 1.

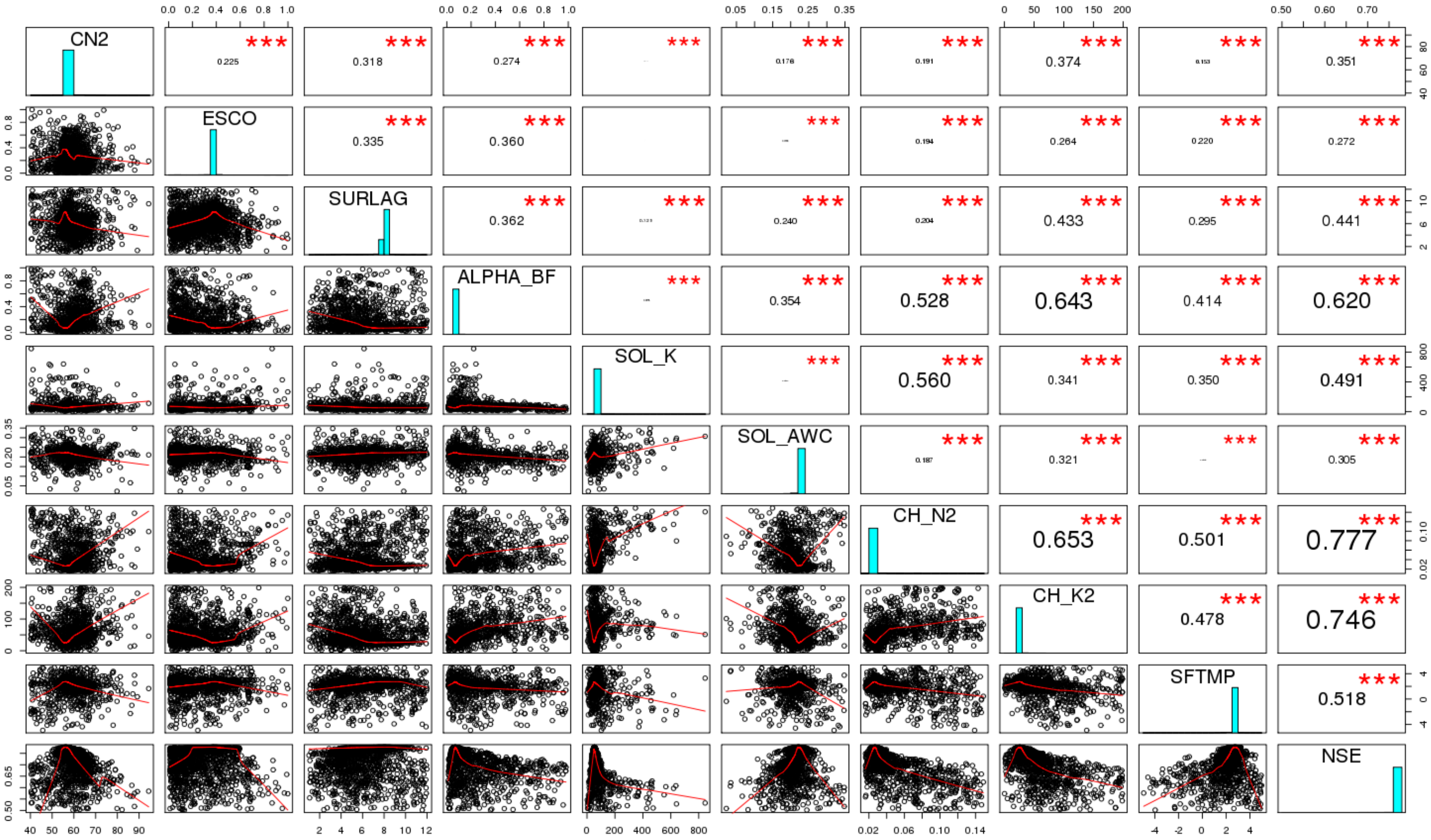

Figure 18: Matrix summarizing the cross-correlation, histograms and statistical significance of the correlation for the (sensitive) parameters listed in Table 1.

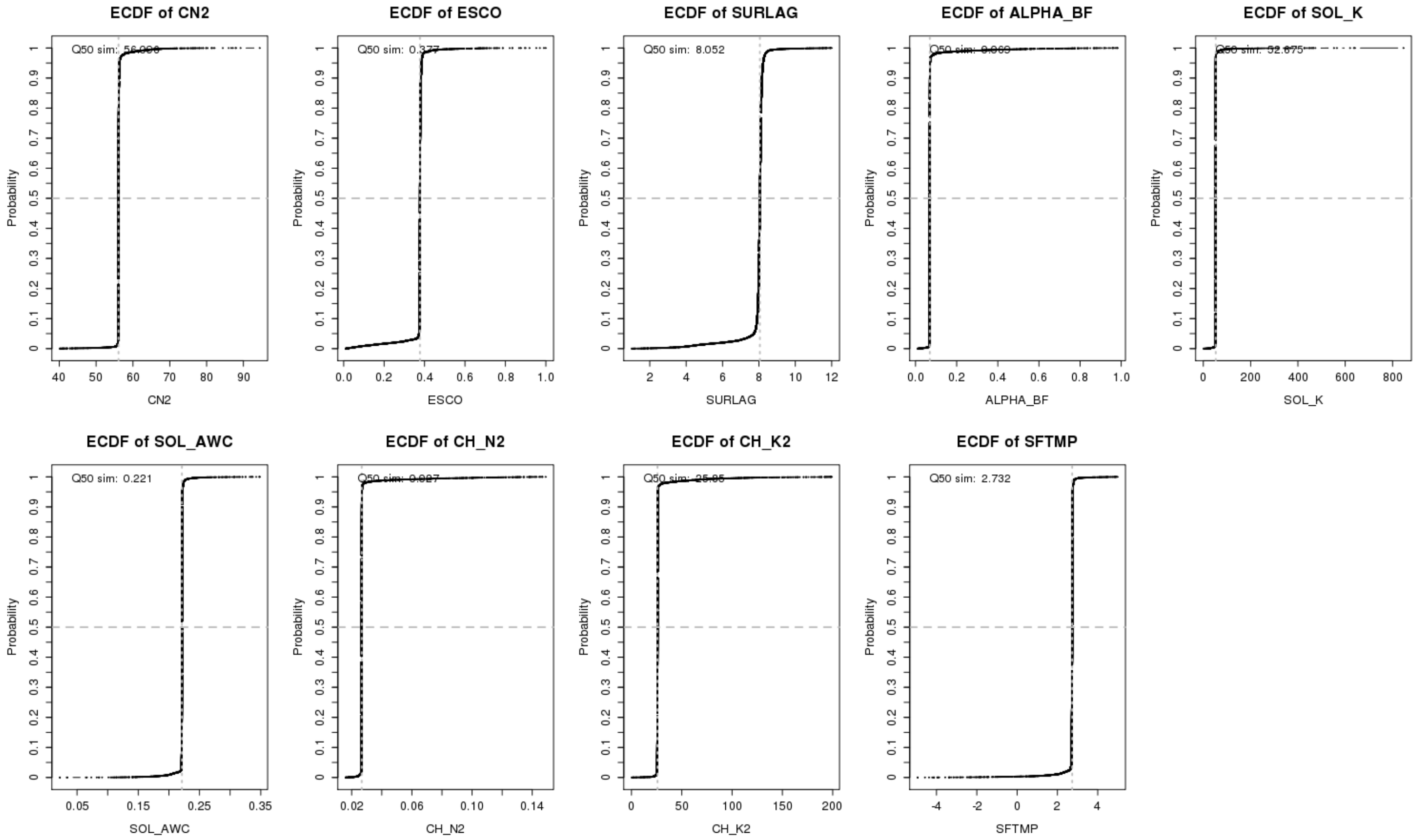

Figure 19: Empirical Cumulative Distribution Functions (ECDFs) for the (sensitive) parameters listed in Table 1. Vertical and horizontal grey dashed-line indicate the location of a user-defined percentile.

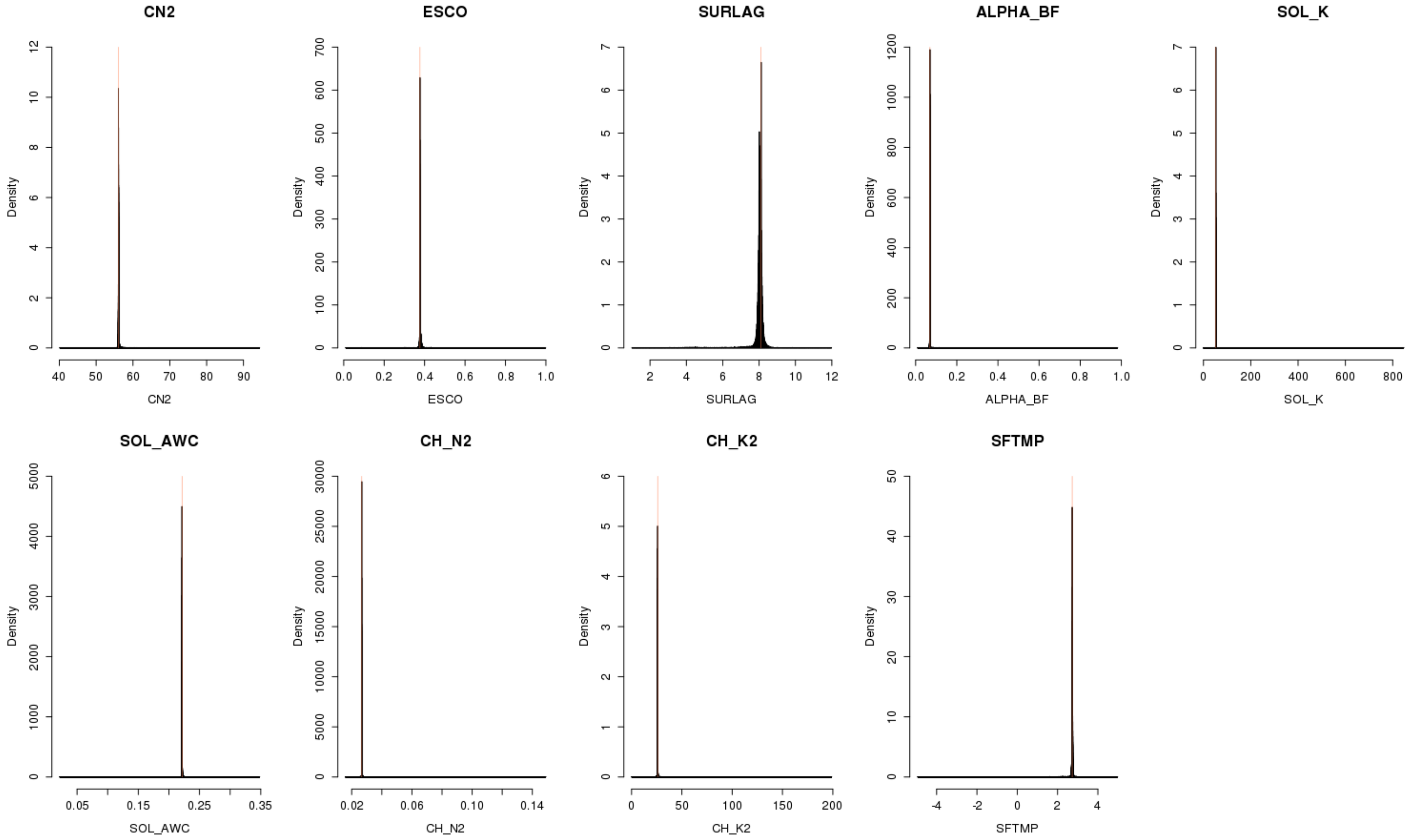

Figure 20: Histograms for the (sensitive) parameters listed in Table 1. Vertical red lines show the location of the highest goodness-of-fit value (Nash-Sutcliffe Efficiency).

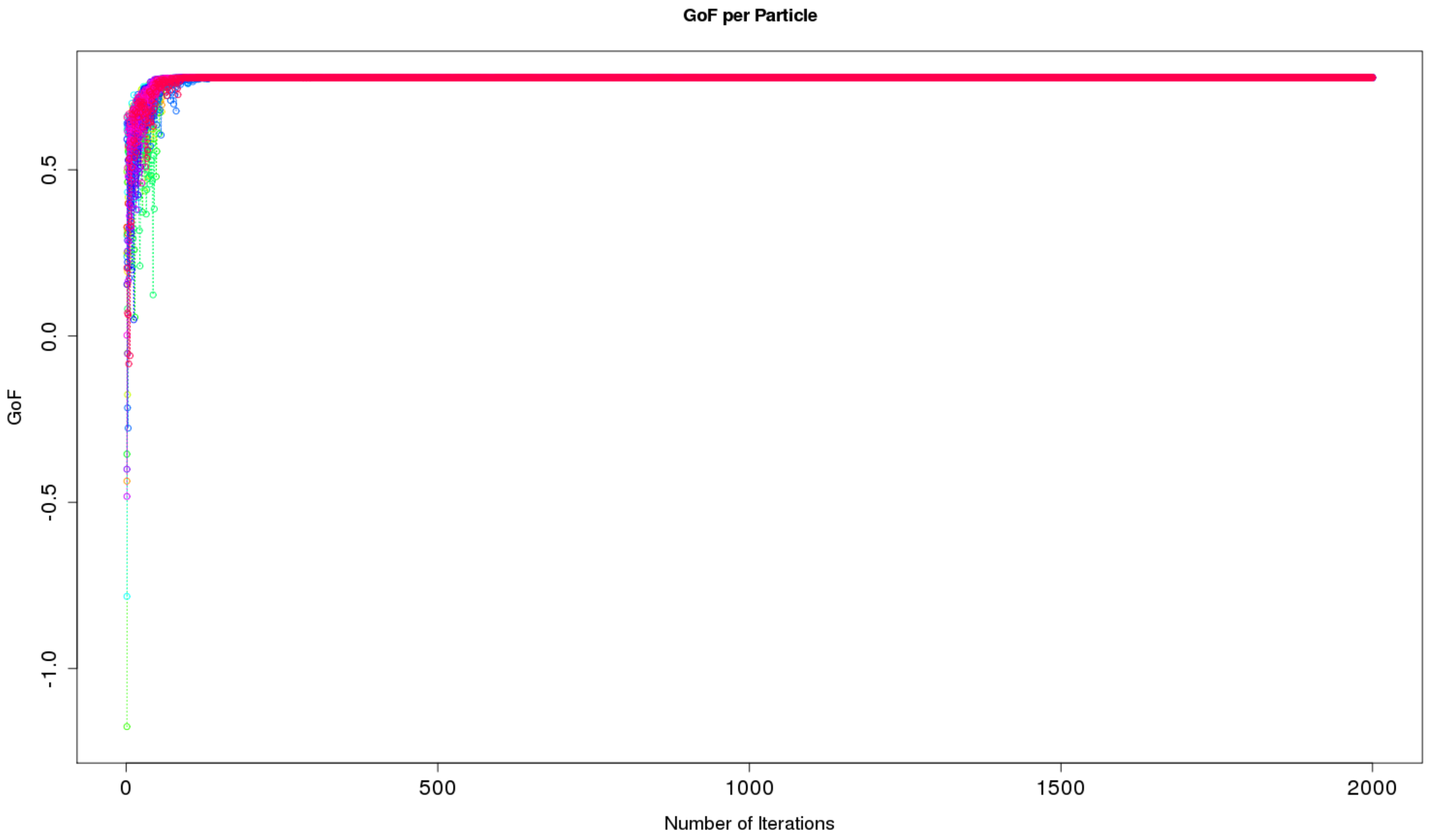

Figure 21: Nash-Sutcliffe Efficiency for all 20 particles for 2000 iterations.

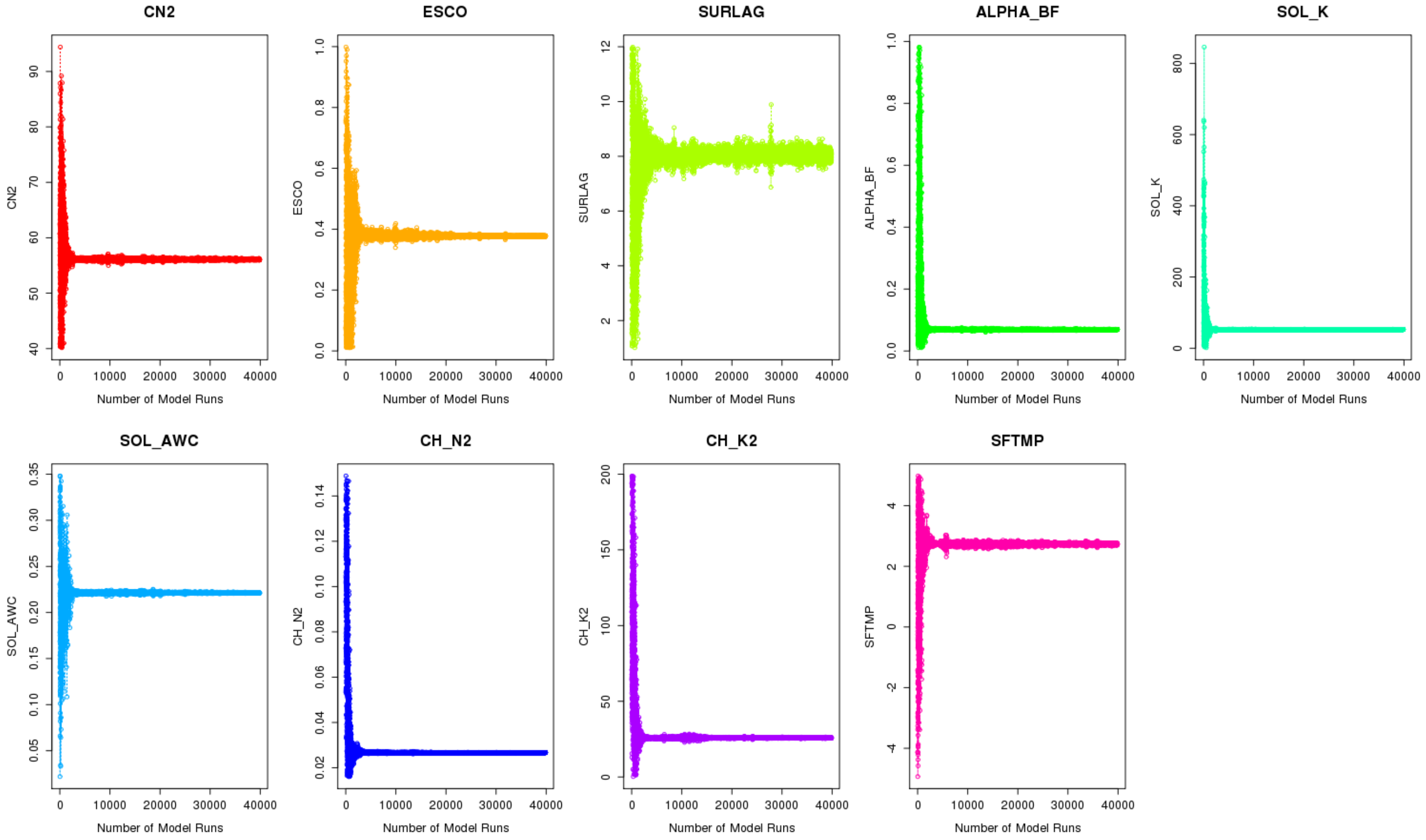

Figure 22: Convergence of the different (sensitive) parameters as a function of the iteration number.

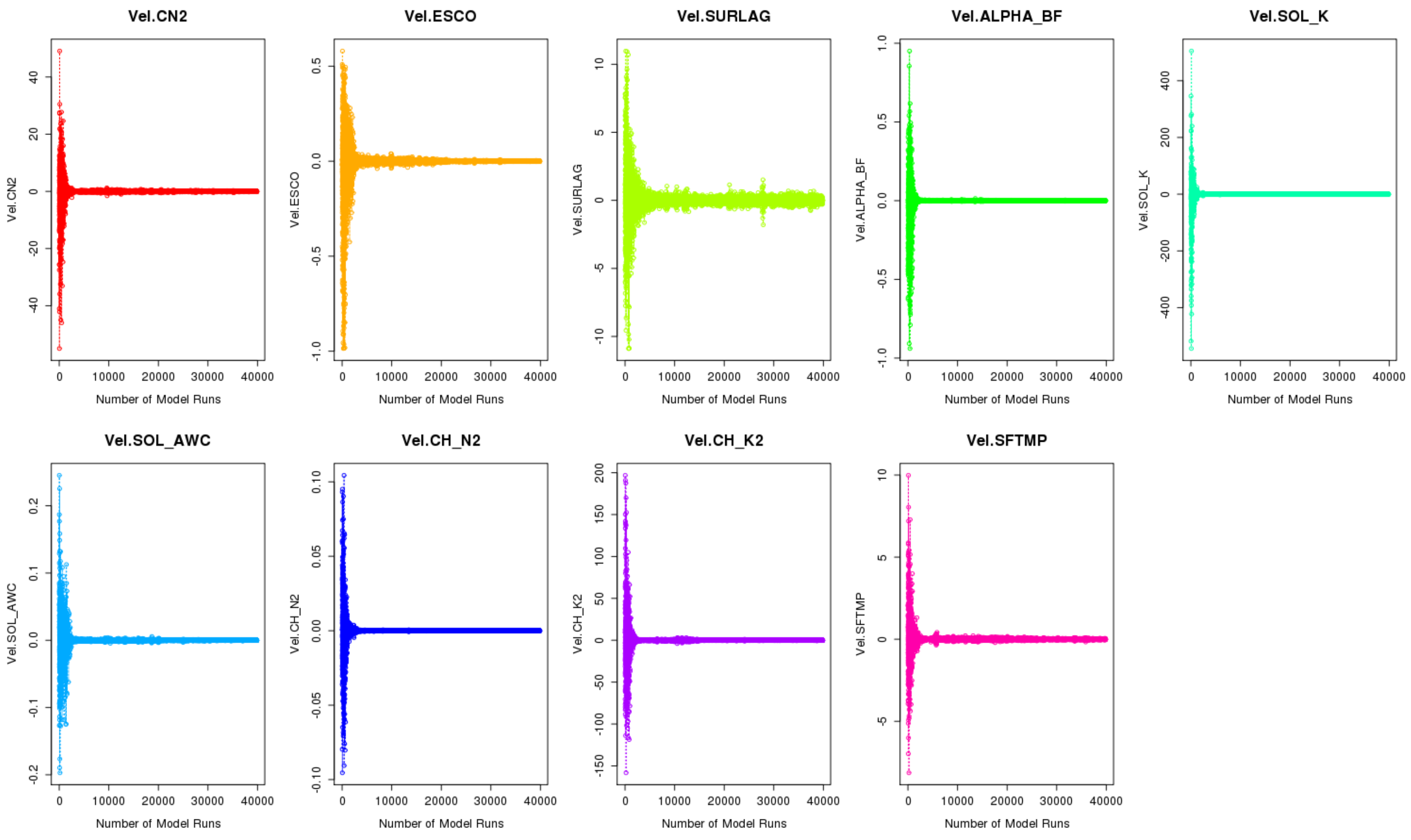

Figure 23: Convergence of the velocities for each (sensitive) parameter as a function of the iteration number.

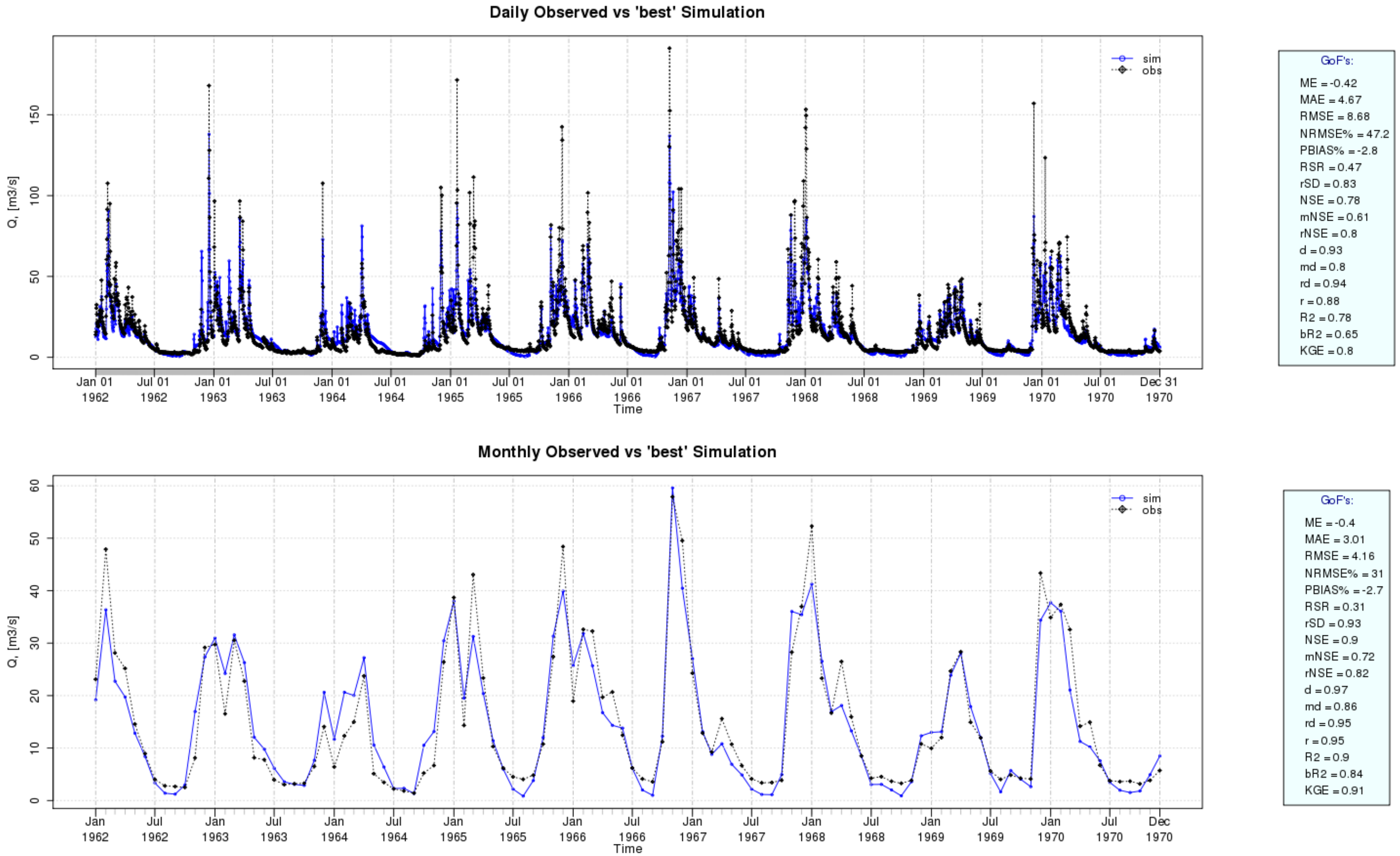

Figure 24: Simulated versus Observed discharges at the station Q071 (see Figure 13). Upper panels show daily calibrated time series and summary box. Lower panels show monthly (aggregated) time series and summary box.

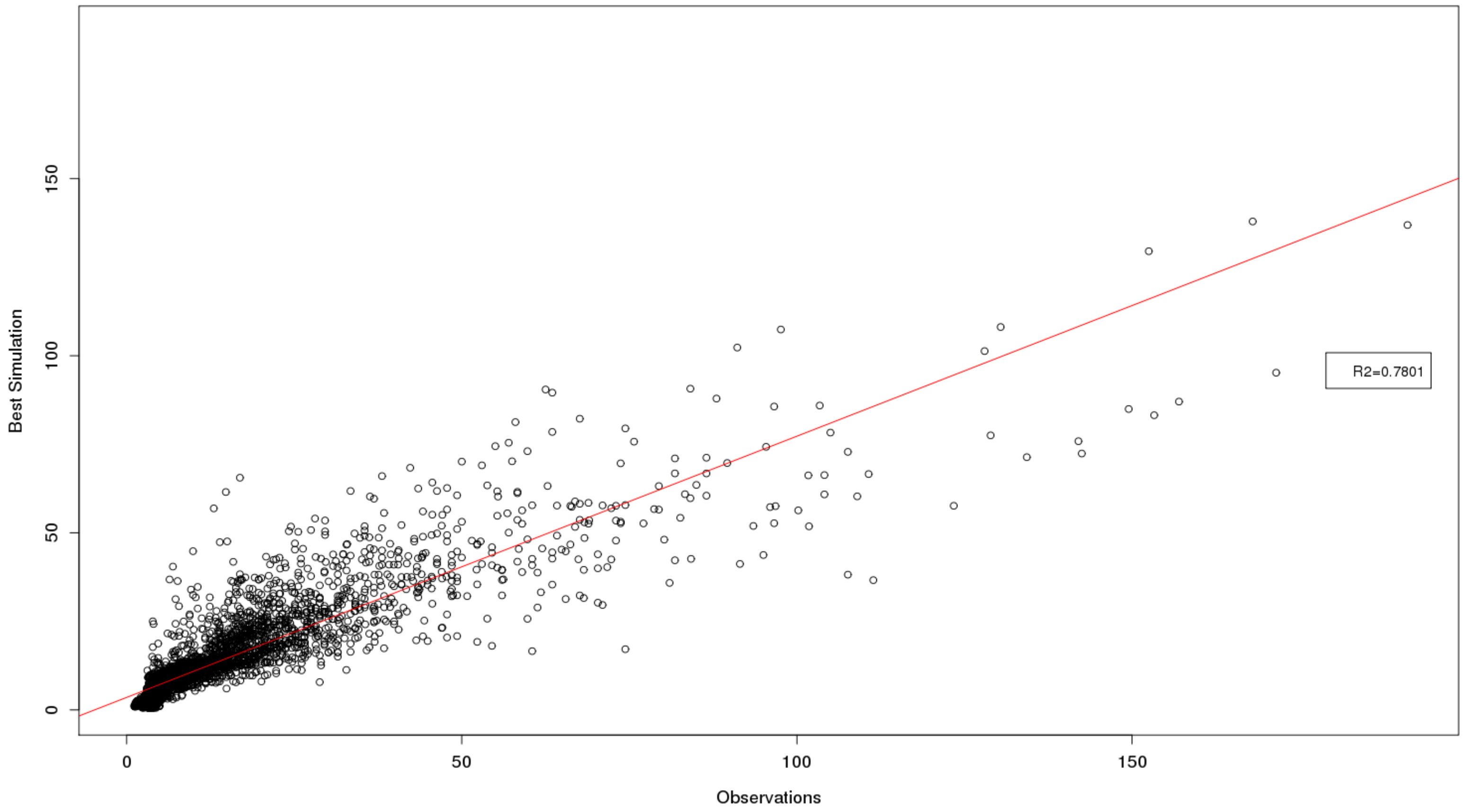

Figure 25: Scatter plot of best simulated versus observed daily discharges at the station Q071 (see Figure 13).

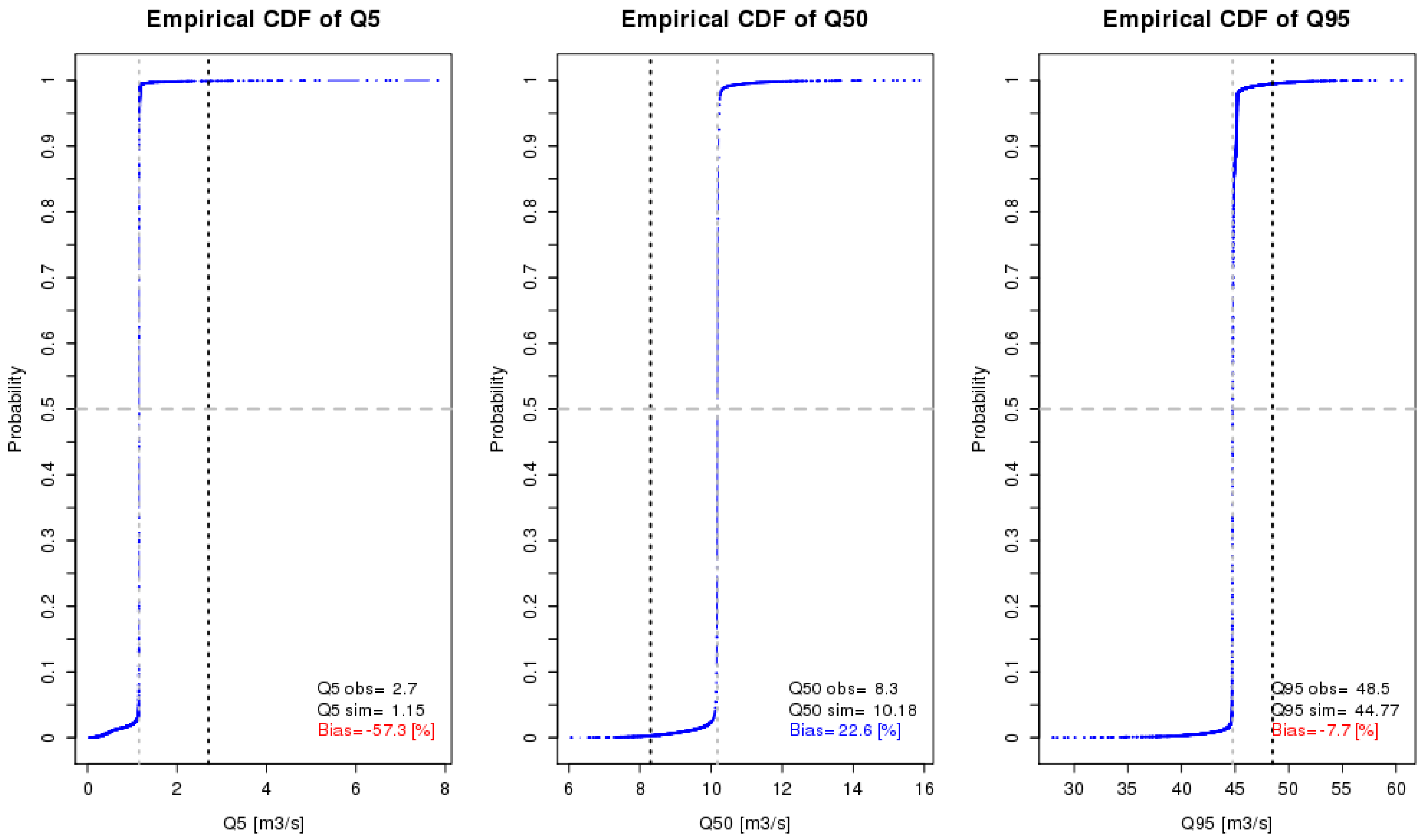

Figure 26: ECDFs for user-defined quantiles (5, 95 and 95). Vertical dashed-line represents the observed quantile.

5 Calibration of a Groundwater Flow Model Using *hydroPSO*

5.1 Groundwater System and Conceptualization

The model to be calibrated is termed “M2” and is similar to the model implemented in *Rojas et al.* (2010). The groundwater system is described by an unconfined aquifer which receives lateral groundwater recharges at the eastern boundary and over all domain through deep fissures in basement rocks. The system exchanges groundwater with contiguous aquifers at the western and southern boundaries, whereas at the northern boundary it shows a groundwater divide. Main outputs of groundwater from the system corresponds to high transpiration rates from forested areas and high evaporation rates from “playas”. The groundwater system is assumed to be under steady-state conditions. An schematic representation of the system is shown in Figure 27.

This groundwater system is conceptualized as a one-layer aquifer, with the northern and southern interaction with external forcings expressed as constant-head boundaries using the BAS6 package of MF2005. Recharge mechanisms are expressed by lateral constant flux and spatially-distributed recharge rates using the WEL and RCH packages of MF2005, respectively. We consider 22 hydraulic conductivity zones obtained from *Rojas and Dassargues* (2007) implemented in the LPF package, one transpiration zone defined as negative rates using the RCH package and one evaporation zone implemented in the EVT package. In total, 30 parameters are considered for the calibration of model M2. The definition of parameters as well as their feasible ranges to implement *hydroPSO* are shown in Table 2. In addition, we consider 42 observation wells (evenly) distributed in the modelled domain for calibration purposes.

Table 2: Parameters used to calibrate model M2 with the *hydroPSO* package.

Parameter		Range	
		Min	Max
Recharge (Lateral fluxes) [m^3 d^{-1}]	RECH	0	345600
Recharge from basement rocks [m^3 d^{-1}]	RECH_BAS	0	172800
Transpiration forested areas [m^3 d^{-1}]	TRANSP	0	172800
Discharge to eastern aquifers [m^3 d^{-1}]	NORIA	0	86400
Evaporation rate [m d^{-1}]	EVTR	0	0.01
Extinction depth [m]	EXTD	0	20
Elevation constant head north [m]	CH_N	1075	1120
Elevation constant head south [m]	CH_S	875	920
Hydraulic conductivity for 22 zones [m d^{-1}]	*K*	0	100

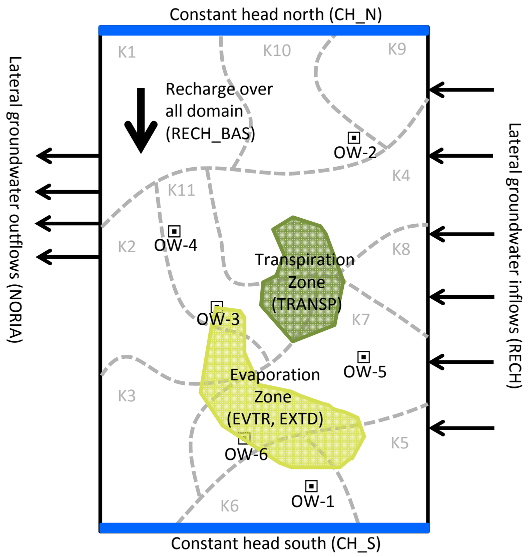
Constant head north (CH_N)
K1
K10
K9
Recharge over all domain (RECH_BAS)
OW-2
K4
K11
OW-4
K2
Transpiration Zone (TRANSP)
K8
OW-3
K7
Evaporation Zone (EVTR, EXTD)
OW-5
K3
OW-6
K5
OW-1
K6
Constant head south (CH_S)
Lateral groundwater outflows (NORIA)
Lateral groundwater inflows (RECH)

Figure 27: Groundwater system conceptualization for model "M2". Note that M2 is set up with 22 hydraulic conductivity zones and 42 observation wells.

5.2 Interfacing *hydroPSO* and MF2005 and ZB

The interaction between *hydroPSO* and M2 (MF2005/ZB) is depicted in Figure 28. *run_me.bat* is a simple batch script executing programmes `preproc`, `mf2005` and `zonebud_hydroPSO` sequentially. `preproc` is a simple read/write FORTRAN code reading the *pre_param.txt* file for M2, which is an ASCII file containing updated parameter values produced by *hydroPSO* after each iteration. `preproc` is problem-specific and copies the (updated) parameter values in the files M2.BA6, M2.EVT, M2.LPF, M2.RCH and M2.WEL, which are the files containing the parameters for M2. Therefore, for a different groundwater model possibly having different conductivity zones, geometry, boundary conditions, etc., it will be necessary to adapt `preproc` to that particular problem calibration. Special care must be taken in reading parameters as reported in the *pre_param.txt* file, whose order is defined by the user through the file *ParamRanges.txt*. Subsequently, `mf2005` executes MF2005 for the "Name File", M2.NAM. MF2005 produces a summary and the main results of the groundwater flow model (heads and global flow components) in files M2.LST, M2.BUD and M2.HED. Finally, `zonebud_hydroPSO` calculates the balance for each specific "budget" zone defined in the M2.ZON file and saves the results into the M2.BAL file. `zonebud_hydroPSO` is a modified version of the original ZB code where keyboard input is skipped by using pre-defined (hard-coded) default options.

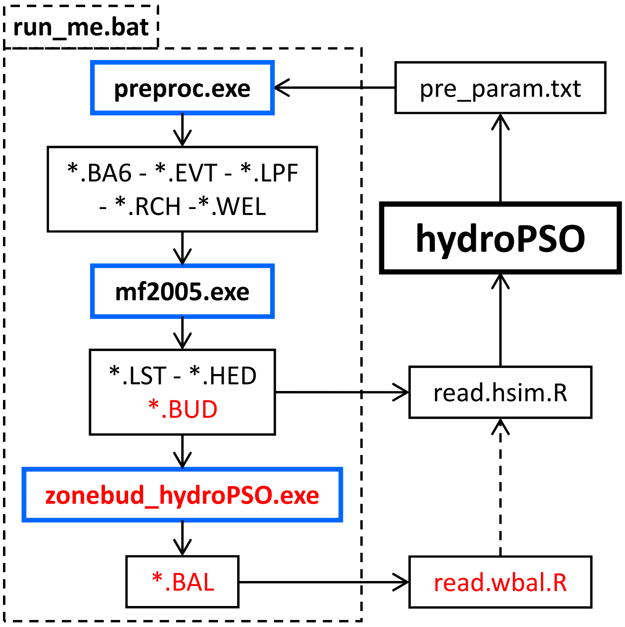

Figure 28: Interaction of *hydroPSO* with MF2005/ZB (M2) and main I/O wrapper functions defined.

5.3 Definition of *ParamFiles.txt* and *ParamRanges.txt* files

Two basic pieces of information are required to interface M2 and *hydroPSO*. First, the names and location of the (multiple) parameters to be calibrated, together with specific location and decimal positions. Second, feasible and physically meaningful parameter ranges. This information is entered in two text files, *ParamFiles.txt* and *ParamRanges.txt*, which must be stored in the subdirectory `./PSO.in`. Note that care must be taken in numbering and naming the parameters as they require to be consistent in both files.

ParamFiles.txt

```
ParameterNmbr ParameterName Filename      Row.Number Col.Start Col.End DecimalPlaces
1             HK_1          pre_param.txt 2          1         20      3
2             HK_2          pre_param.txt 3          1         20      3
3             HK_3          pre_param.txt 4          1         20      3
4             HK_4          pre_param.txt 5          1         20      3
5             HK_5          pre_param.txt 6          1         20      3
6             HK_6          pre_param.txt 7          1         20      3
7             HK_7          pre_param.txt 8          1         20      3
8             HK_8          pre_param.txt 9          1         20      3
9             HK_9          pre_param.txt 10         1         20      3
10            HK_10         pre_param.txt 11         1         20      3
11            HK_11         pre_param.txt 12         1         20      3
12            HK_12         pre_param.txt 13         1         20      3
13            HK_13         pre_param.txt 14         1         20      3
14            HK_14         pre_param.txt 15         1         20      3
```

```
15            HK_15         pre_param.txt 16          1         20      3
16            HK_16         pre_param.txt 17          1         20      3
17            HK_17         pre_param.txt 18          1         20      3
18            HK_18         pre_param.txt 19          1         20      3
19            HK_19         pre_param.txt 20          1         20      3
20            HK_20         pre_param.txt 21          1         20      3
21            HK_21         pre_param.txt 22          1         20      3
22            HK_22         pre_param.txt 23          1         20      3
23            R_TRANSP      pre_param.txt 24          1         20      4
24            R_BSMNT       pre_param.txt 25          1         20      4
25            R_CHN         pre_param.txt 26          1         20      4
26            R_CHS         pre_param.txt 27          1         20      4
27            R_EVTR        pre_param.txt 28          1         20      8
28            R_EXTD        pre_param.txt 29          1         20      4
29            R_RECH        pre_param.txt 30          1         20      4
30            R_NOR         pre_param.txt 31          1         20      4
```

ParamRanges.txt

```
ParameterNmbr ParameterName MinValue MaxValue
1             HK_1          0        100
2             HK_2          0        100
3             HK_3          0        100
4             HK_4          0        100
5             HK_5          0        100
6             HK_6          0        100
7             HK_7          0        100
8             HK_8          0        100
9             HK_9          0        100
10            HK_10         0        100
11            HK_11         0        100
12            HK_12         0        100
13            HK_13         0        100
14            HK_14         0        100
15            HK_15         0        100
16            HK_16         0        100
17            HK_17         0        100
18            HK_18         0        100
19            HK_19         0        100
20            HK_20         0        100
21            HK_21         0        100
22            HK_22         0        100
23            R_TRANSP      0        172800
24            R_BSMNT       0        172800
25            R_CHN         1075     1120
26            R_CHS         875      920
27            R_EVTR        0        0.01
28            R_EXTD        0        20
29            R_RECH        0        345600
30            R_NOR         0        86400
```

5.4 Basic I/O Wrapper Functions

For a successful calibration a number of observations are required for comparison against the simulated equivalents and, as such, they constitute the basis for the calculation of the objective function. For MF2005, we can define observations for a series of flow processes, e.g. specified-head flows, drains, rivers, groundwater heads, streams, etc., see *Hill et al.* (2000). For model

M2, however, we employ 42 groundwater head observations contained in the M2.HOB file, which are read through a simple R function (*read.hobs.R*). As explained later, *read.hobs.R* is called only once and is not necessarily required. We include it to illustrate how additional observations for other flow components can be read, stored in an R object and used for calculating the objective function.

read.hobs.R

```
read.hobs <- function(fname="M2.HOB",ncol=9,skip=2) {
  x <- read.table(file=fname,skip=skip,header=FALSE)
  x <- x[,ncol]
  return(x) }
```

The main I/O function to interface *hydroPSO* and M2 (MF2005/ZB) is *read.hsim.R*. This function reads simulated groundwater heads, global groundwater balance components and residuals, and calculates a customized goodness-of-fit measure as objective function. For this case, we read both simulated heads and calculated residuals from the M2.LST file. Alternatively, we could read simulated groundwater heads from the M2.HED file (for specific well locations), however, this does not allow us to read in one-step the global groundwater balance components available in the M2.LST file. By directly reading the residuals as reported in the M2.LST, we avoid having to read the groundwater head observations contained in the M2.HOB file.

At the same time, *read.hsim.R* calculates a Gaussian likelihood measure (L) using,

$$L = (2\pi)^{-N/2} \left|C\right|^{-1/2} \exp\left(-\frac{1}{2}\left(h_{sim} - h_{obs}\right)^{T} C^{-1}\left(h_{sim} - h_{obs}\right)\right) \qquad (10)$$

where C is the covariance matrix of the observed system variables, N is the number of observations, and the likelihood for the corresponding *hydroPSO* iteration (i.e. parameter set) is obtained using a product inference function (see, e.g., *Rojas et al.*, 2008). Note that alternative formulations can be implemented as objective functions, however, the current version of *hydroPSO* works only with single-objective functions.

Finally, *read.hsim.R* writes the calculated likelihood, which is processed by *hydroPSO* to assess the quality of the particles' positions, and the global groundwater balance components to simple ASCII files, `lik_gauss.txt` and `WBAL.txt`, respectively.

read.hsim.R

```
read.hsim <- function(fname="M2.LST",nobs=42) {
sim <- rep(NA,nobs)
lik <- NA
out <- rep(NA,7)
L <- 0
x <- readLines(fname)
stg <- " HEAD AND DRAWDOWN OBSERVATIONS"
n <- which(x==stg)
L <- length(n)

if (L > 0) {
  suppressWarnings(tmp <- read.table(file=fname,skip=n+3,header=FALSE,nrows=nobs,colClasses
  =c("NULL","NULL","numeric","numeric"),fill=TRUE,na.strings="OMITTED"))
  sim <- as.numeric(tmp[,1])
  res1 <- as.numeric(tmp[,2])
  na.index <- which(is.na(sim))

  if ((length(sim)==nobs) & (length(na.index)==0)) {
    # Gaussian likelihood (stdev = 10 same for MCMC analysis Rojas et al. 2010)
    gauss1 <- 2*pi
    gauss2 <- 10*sqrt(gauss1)
    gauss3 <- 1/gauss2
    gauss4 <- 2*(10^2)
    res2 <- res1^2
    lik1 <- gauss3*exp(-(res2/gauss4))
    lik2 <- prod(lik1)
    lik <- lik2^(1/nobs) #likelihood using "product" inference function (Rojas et al. 2010)

          system2("zonbud_hydroPSO.exe")
  stg <- "  VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP  1 IN STRESS PERIOD   1"
    n <- which(x==stg)
    L <- length(n)
    if (L > 0) {
      suppressWarnings(tmp <- read.table(file=fname,skip=n+11,header=FALSE,nrows=9,fill=TRU
      E,stringsAsFactors=FALSE))
      rech <-as.numeric(tmp[1,3])
      evap <-as.numeric(tmp[8,3])
      transp <-as.numeric(tmp[9,3])
      out1 <- c(rech,evap,transp )
      names(out1) <- c("rech","evap","transp")
      out2 <- read.wbal("M2.BAL")
      out <- c(out1, out2)
    }
  } else {
      sim <- rep(NA,nobs)
      lik <- NA
      out <- rep(NA,7)
    }
}
# Adding the results of the water balance to "WBAL.txt"
wb.Text.file <- file("WBAL.txt","a")
writeLines(as.character(out),wb.Text.file,sep=" ")
writeLines("",wb.Text.file)
close(wb.Text.file)
write(lik,"lik_gauss.txt")
return(sim) }
```

In principle, using *read.hsim.R* and correctly defining the *ParamFiles.txt* file, should suffice for interfacing *hydroPSO* and M2 (MF2005/ZB). However, to consider alternative "goodness-of-fit" measures and/or observations might require using *read.hobs.R*.

In addition, for this tutorial we develop an R script to read the results from the ZB program. *read.wbal.R* reads the results for 4 “budget” zones defined in the M2.ZON file, and saved in the M2.BAL file by `zonebud_hydroPSO`. These flows are recharge due to deep fissures in basement rocks (`rechdeep`), outflows at the aquifer’s southern cross-section (`cgordo`), (point) southernmost incoming lateral flux (`chaca`), and outflows to western aquifers (`noria`). These results are stored in an R object (`out`), which is saved in the WBAL.txt file by the *read.hsim.R* script.

read.wbal.R

```
read.wbal <- function(fname="M2.BAL") {
if (length(readLines(fname)) > 34 ) {
  suppressWarnings(rechdeep <- as.numeric(read.table(file=fname,skip=34,header=FALSE,nrows=
  1,colClasses=c("NULL","NULL","numeric"),fill=TRUE)))
  suppressWarnings(cgordo <- as.numeric(read.table(file=fname,skip=42,header=FALSE,nrows=1,
  colClasses=c("NULL","NULL","NULL","NULL","NULL","numeric"),fill=TRUE)))
  noria <- NA
  chaca <- NA

  x <- readLines(fname)
  stg <- "     Flow Budget for Zone  5 at Time Step  1 of Stress Period  1"
  n <- which(x==stg)
  L <- length(n)
  if (L > 0) noria <- read.table(file=fname,skip=n+18,header=FALSE,nrows=1,colClasses=c("NU
  LL","NULL","numeric"),fill=TRUE)

  stg <- "     Flow Budget for Zone 16 at Time Step  1 of Stress Period  1"
  n <- which(x==stg)
  L <- length(n)
  if (L > 0) chaca <- read.table(file=fname,skip=n+8,header=FALSE,nrows=1,colClasses=c("NUL
  L","NULL","numeric"),fill=TRUE)
  out <- c(rechdeep,noria,cgordo,chaca)
} else out <- rep(NA,4)
names(out) <- c("rechdeep","noria","cgordo","chaca")
return(out) }
```

5.5 Implementation Details and Results of the Calibration

1. Once `ParamFiles.txt` and `ParamRanges.txt` files have been created, they need to be stored in the `./PSO.in` directory within the folder containing the main MF2005 model files, which for this tutorial corresponds to `./MF2005`.

2. Several auxiliary files (described below) are included in `./MF2005`:

   Auxiliary Files in MF2005

   ```
   hydroPSO-MF2005.R -> Main R script to run hydroPSO
   lik_gauss.txt -> ASCII file containing the likelihood value
   pre_param.txt -> ASCII file containing the updated parameters
   ```

```
TOY_LPF_M2.txt -> Template for the LPF package used by preproc
TOY_RCH_M2.txt -> Template for the RCH package used by preproc
```

3. The setup for the problem as well as all the options implemented to calibrate the groundwater flow model are defined in the `hydroPSO-MF2005.R` script. By default all results from *hydroPSO* are saved into the `PSO.out` folder, however, this can be redefined by using the `drty.out` argument.

hydroPSO-MF2005.R

```
###################################################################################
## Example to interface MODFLOW2005 and ZONE BUDGET with hydroPSO. This script     #
## allows hydroPSO to take control over the execution of MODFLOW2005 and ZONEBUDGET#
## through the definition of a batch file (run_me.bat) and a series of simple I/O R#
## scripts                                                                         #
##                                                                                 #
## Part of the hydroPSO R package                                                  #
## http://www.rforge.net/hydroPSO/ http://cran.r-project.org/web/packages/hydroPSO #
## Copyright 2011-2012 Mauricio Zambrano-Bigiarini & Rodrigo Rojas                 #
## Distributed under GPL 2 or later                                                #
##                                                                                 #
## Created by Mauricio Zambrano-Bigiarini and Rodrigo Rojas. 26-Oct-2011           #
## Last saved: 13-Feb-2012                                                         #
###################################################################################

###Loading required libraries
library(hydroPSO)
library(hydroTSM)
library(hydroGOF)

###Definition of working directory: input, output and model files paths
model.drty <- "~/MF2005"
setwd(model.drty )

###Customized I/O functions (R scripts) to interface MF2005 with hydroPSO
source("read.hobs.R")
source("read.hsim.R")
source("read.wbal.R")
source("read.lik.R")

###Goodness-of-fit, either customized or pre-defined from hydroGOF
gof.FUN <- "read.lik"
gof.FUN.args <- list()

###Getting the OBSERVATIONS (not strictly necessary for this example)
obs.fname <- "M2.HOB"
obs.fname <- paste(file.path(model.drty),"/",obs.fname,sep="")
obs <- read.hobs(fname=obs.fname)

###MAIN model function
model.FUN.args=list(
   model.drty=model.drty,
   param.files=paste(model.drty,"/PSO.in/ParamFiles.txt",sep=""),
   exe.fname="run_me.bat",
   ###Function for reading the simulated equivalents
   out.FUN="read.hsim",
   out.FUN.args=list(
      fname="M2.LST",
      nobs=42),
  ###Function assessing the simulated equivalents against the observations
  gof.FUN=gof.FUN,
  gof.FUN.args=gof.FUN.args,
  obs=obs
) ###END model.FUN.args

###MAIN PSO ALGORITHM
###For hydroPSO fine-tuning parameters, see Zambrano-Bigiarini and Rojas,2012
set.seed(1111)
hydroPSO(
   fn="hydromod",
   model.FUN="hydromod",
   model.FUN.args=model.FUN.args,
   method="pso",
   control=list(
      MinMax="max",
      npart=70,
      maxit=3000,
      reltol=1e-10,
      use.IW=TRUE,IW.type="linear",IW.w=1/(2*log(2)),IW.exp=1,
      use.TVc1=TRUE,TVc1.type="non-linear",TVc1.rng=c(1.28,1.05),TVc1.exp=1.5,
      drty.out="PSO.70p3000i.rand_TVc1_TVlambda.out",
      REPORT=50
   ) ###END control options
) ###END MAIN hydroPSO ALGORITHM
```

In the `hydroPSO-MF2005.R` script we use `read.hobs.R` to read groundwater head observations from M2.HOB, `read.hsim.R` to read simulated equivalents and calculate a likelihood measure, `read.wbal.R` to read groundwater balance components defined in the `M2.ZON` file, and `read.lik.R` to read the one-line ASCII file containing the likelihood measure. When a model code other than pre-defined test functions coded in *hydroPSO* is used, the `fn` argument **must** take the value `hydromod`. The latter will indicate *hydroPSO* to expect for external files containing the model executable(s), input(s) and output files. We use the PSO algorithm (`method="pso"`) to maximize (`MinMax="max"`) the likelihood calculated by `read.hsim.R`. For that purpose, we employ a swarm of 70 particles (`npart=70`) and a maximum number of iterations equals to 3000 (`maxit=3000`). A constant (`IW.type="linear"`) inertia weight (`use.IW=TRUE`) value of $1/(2 * log(2))$ (`IW.w`), a non-linear time-variant cognition coefficient (c_1) between [1.28,1.05] with exponent (`TVc1.exp=1.5`), and a linear (`TVlambda.type="linear"`) time-variant lambda factor (velocity clamping factor) (`use.TVlambda=TRUE`) between [1.0,0.5] are used for fine-tuning the algorithm. As model arguments we define `run_me.bat` as the main code running the (sequence of) model(s) together with the main model outputs (`out.FUN="read.hsim"`) and the likelihood of that particular parameter set (`gof.FUN="read.lik"`).Finally, we save results in folder `PSO.70p3000i.rand_TVc1_TVlambda.out`.

4. Here, we illustrate the use of individual *hydroPSO* "reading" and "plot-

ting" functions to obtain customized graphs. First, we set up the corresponding directory with the results of the calibration:

```
> setwd("~/MF2005/PSO.70p3000i.rand_Tvc1_TVlambda.out")
```

5. Plotting the evolution of the global optimum:

```
> read_convergence(do.png=TRUE)
```

Figure 29 shows the evolution of the Gaussian likelihood (customized goodness-of-fit measure) as a function of the number of iterations. In general, we see that there is an initial exploratory phase of ca. 500 iterations where significant improvements in the Gbest are found. After iteration nr. 1000, Gbest stabilizes and the NSR seems to reduce oscillation converging toward an attraction zone.

6. Plotting the evolution of the 30 parameters. First, we read the file `Particles.txt` and then we call function `plot_ParamsPerIter` for plotting:

```
> parts <- read_params(file="Particles.txt",param.cols=
+                        4:33,plot=FALSE)
> plot_ParamsPerIter(parts[["params"]])
```

Figure 30 shows the result from the previous command lines. In this figure we see the evolution of the parameters of M2 as a function of the number of model evaluations (i.e. 70 particles × 3000 iterations). We see that several parameters show a relatively insensitive behaviour (e.g. HK_6, HK_10-15, HK_21, R_CHN, R_TRANSP, R_EVTR, R_EXTD), whereas others show a clear zone of attraction. These results are in full agreement with the findings by *Rojas et al.* (2010).

7. From the previous item it is clear that not all parameters are sensitive. So, here we plot sensitive parameters only using the options of the `plot_results()` *hydroPSO* function:

```
> plot_results(drty.out=getwd(),MinMax="max",do.png=TRUE,
+               param.names=c("HK_2","R_CHS","R_NOR","HK_19",
+               "HK_16","R_RECH","R_BSMNT","R_CHN","R_TRANSP",
+               "R_EVTR","R_EXTD","HK_1"))
```

This command line will produce several figures summarizing the results of the calibration for M2 (see, e.g., Section 4.4.2). Figure 31 shows dotty-plots for the subset of sensitive parameters identified in Figure 30. Here, we see clear zones of attraction for some parameters (e.g. HK_2, R_CHS, and R_RECH), whereas others show slightly sensitive (non-symmetric) likelihood response surfaces. Figure 32, in turn, shows (projected) 2D dotty plots among parameters highlighting non-linear interactions.

Figure 33 shows the ECDs for the sensitive parameters of M2, where the most likely parameter value is highlighted. Finally, Figure 34 shows a good correspondence between the best simulated and the observed groundwater heads.

8. At the same time, it is possible to analyse the results defining a "behavioural" threshold for the full set of simulated parameters. Here, we select all simulations with a Gaussian Likelihood greater than 3.8×10^{-2}:

```
> plot_results(drty.out=getwd(),MinMax="max",do.png=TRUE,
+              param.names=c("HK_2","R_CHS","R_NOR","HK_19",
+              "HK_16","R_RECH","R_BSMNT","R_CHN","R_TRANSP",
+              "R_EVTR","R_EXTD","HK_1"),beh.thr=3.8e-2)
```

Figures 35 and 36 show the full ECDFs for the simulated groundwater heads at 42 observation wells, highlighting the groundwater head observations and the percentage bias for the quantile 50. Figure 37 show the resulting plots for the (projected) 2D interaction among parameters, where a clearer picture of the non-linear interactions for the zone of attraction can be seen for several parameters (e.g. R_RECH vs. R_NOR and R_RECH vs. HK_2). At the same time, Figure 38 shows the resulting ECDFs for a behavioural threshold of 3.8×10^{-2}.

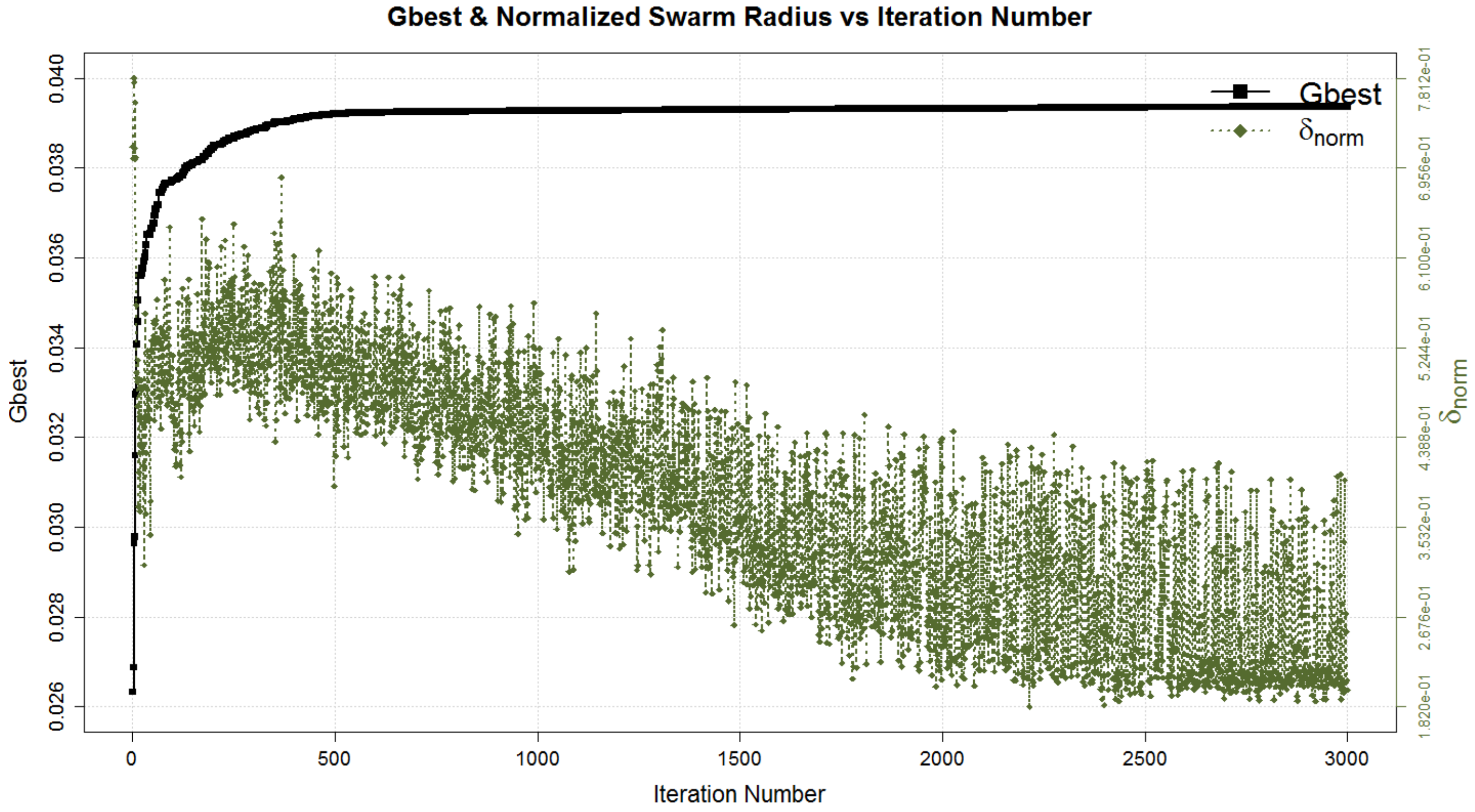

Figure 29: Evolution of the global best (Gbest) and the Normalized Swarm Radius (NSR) versus iteration number for the calibration of the Ega headwater catchment.

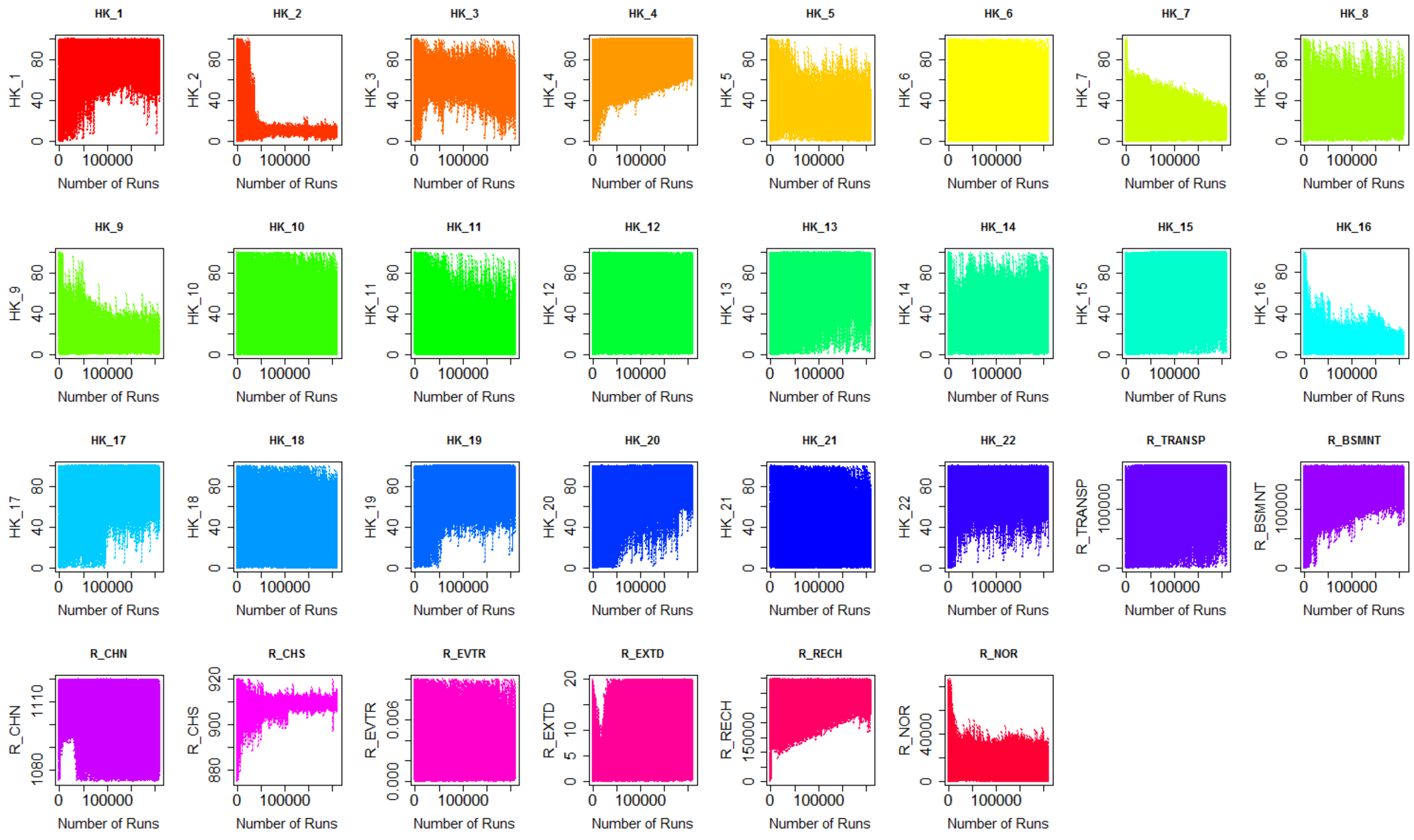

Figure 30: Convergence of the different MF2005 parameters as a function of the iteration number.

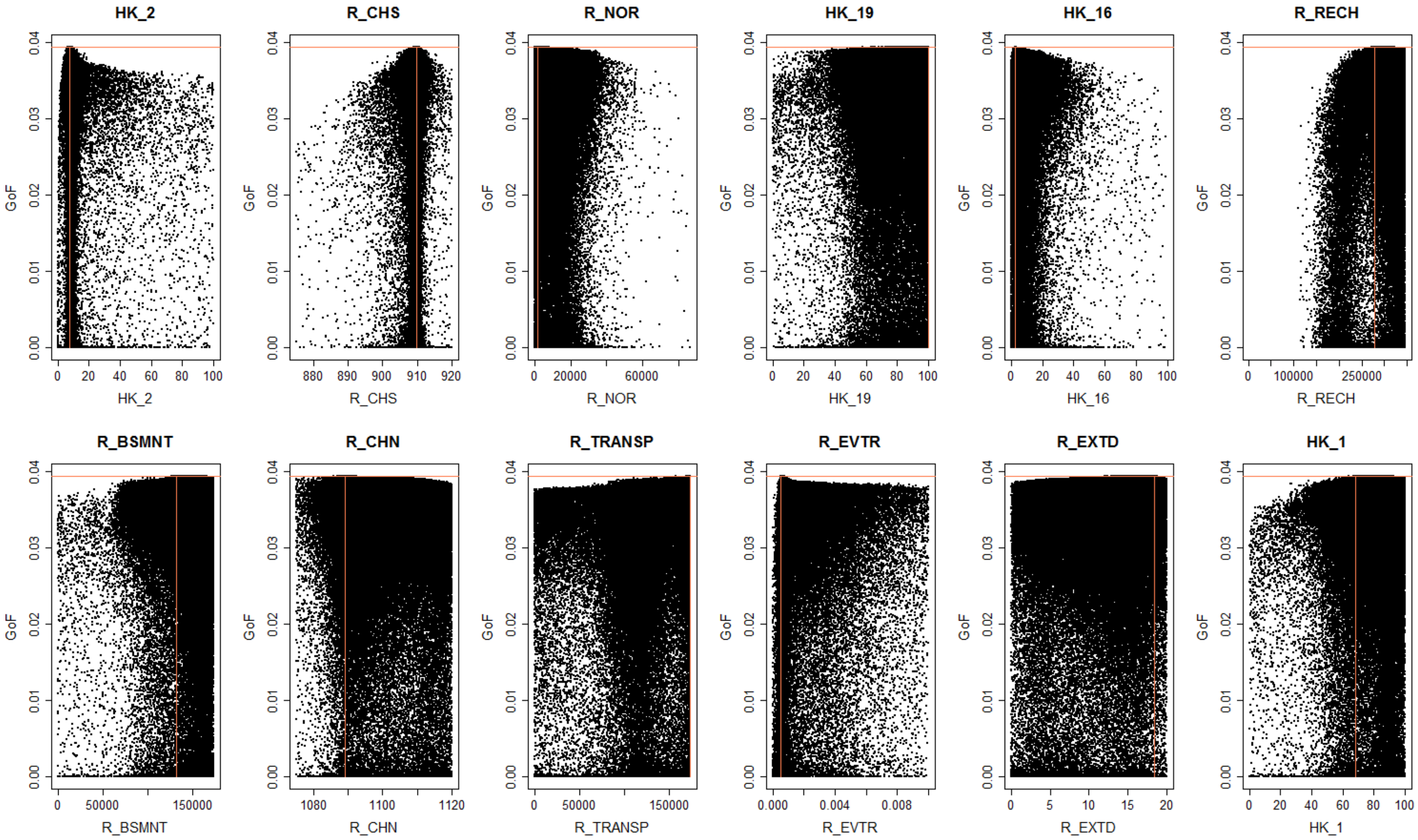

Figure 31: Dotty-plots for the sensitive parameters obtained from Figure 30 for M2. Vertical and horizontal coloured lines show the location of the highest goodness-of-fit value (Gaussian Likelihood).

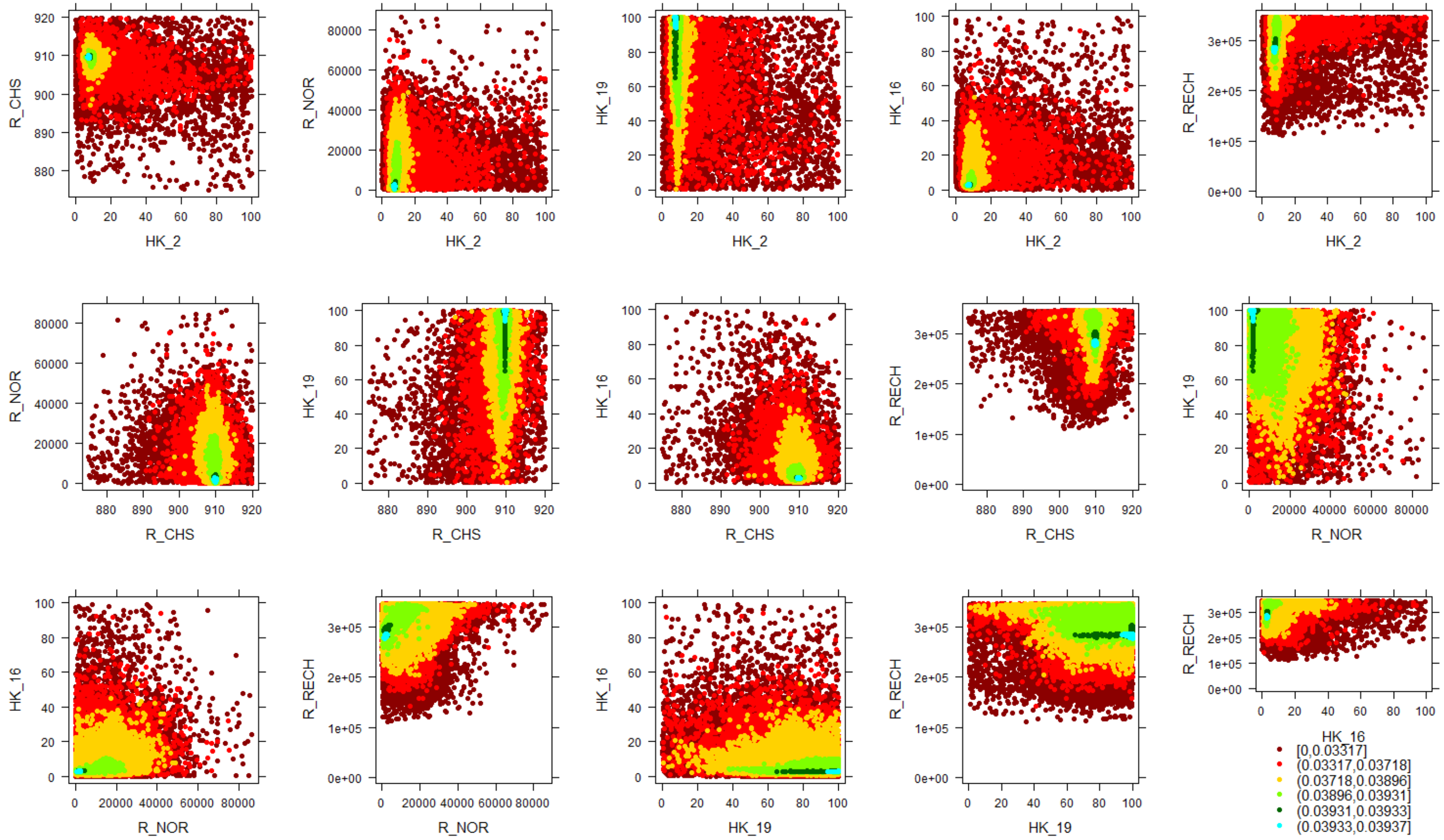

Figure 32: 2-dimensional projected dotty-plots highlighting the interaction among the first seven sensitive parameters obtained from Figure 30 for M2.

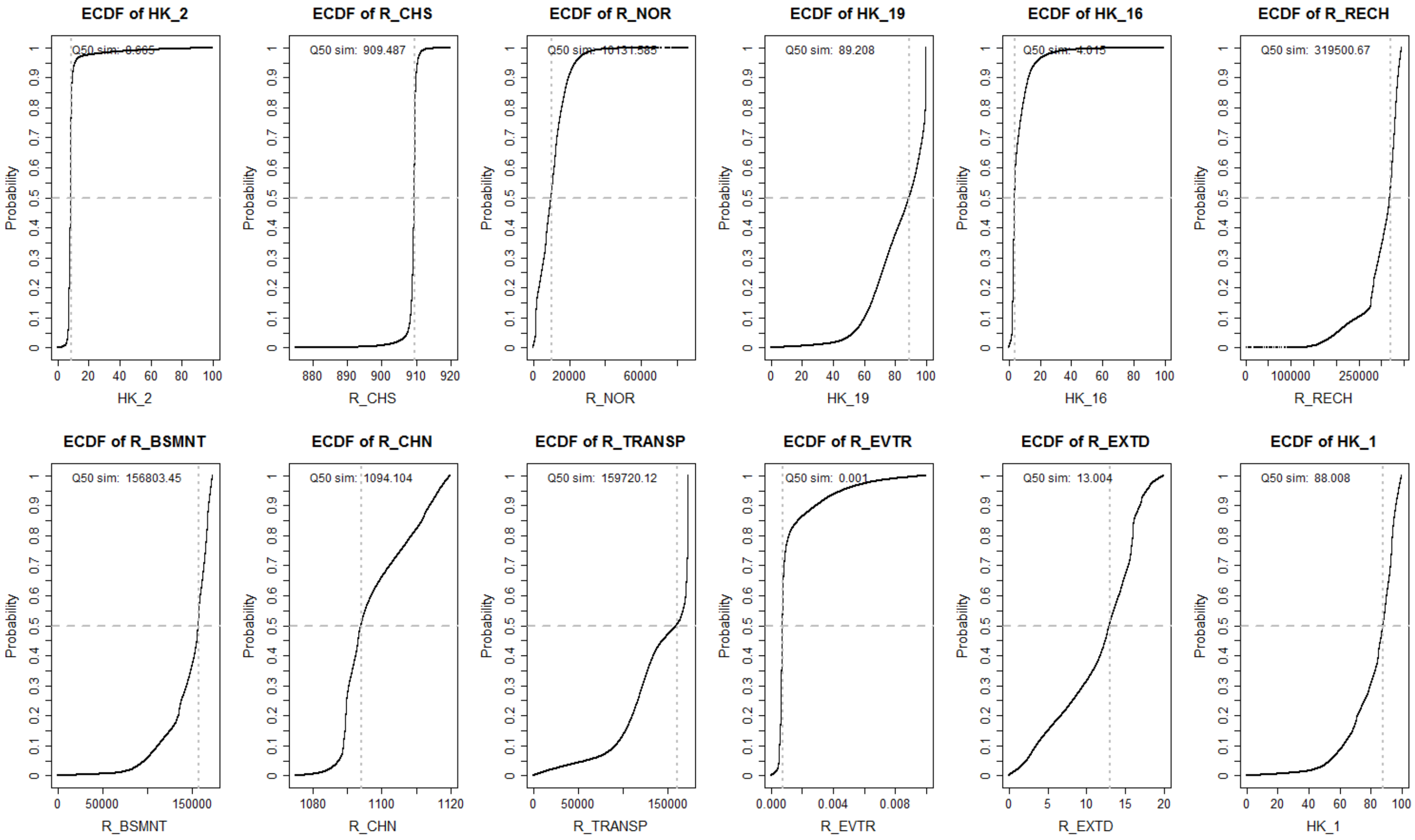

Figure 33: Empirical Cumulative Distribution Functions (ECDFs) for the sensitive parameters obtained from Figure 30 for M2. Vertical and horizontal grey dashed-line indicate the location of a user-defined percentile.

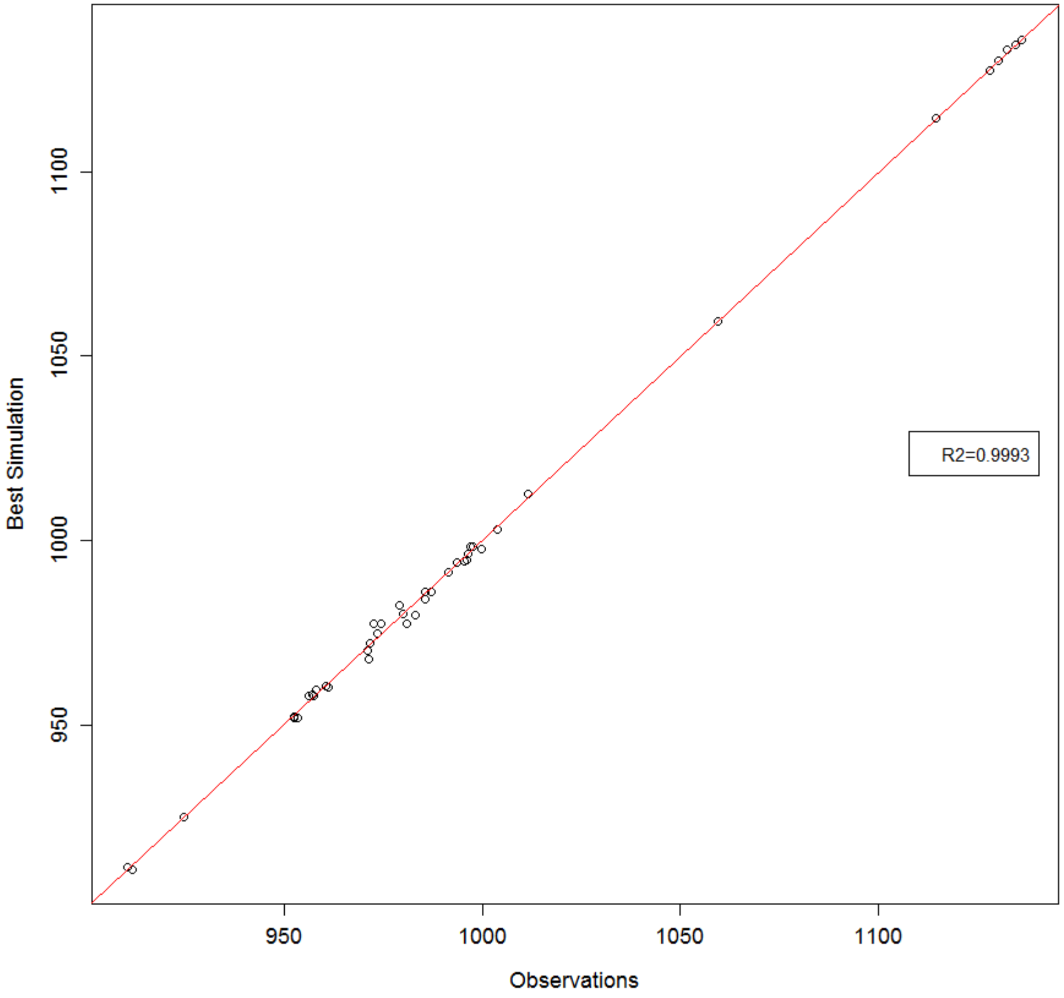

Figure 34: Scatter plot of best simulated versus observed groundwater heads at 42 observation wells for M2.

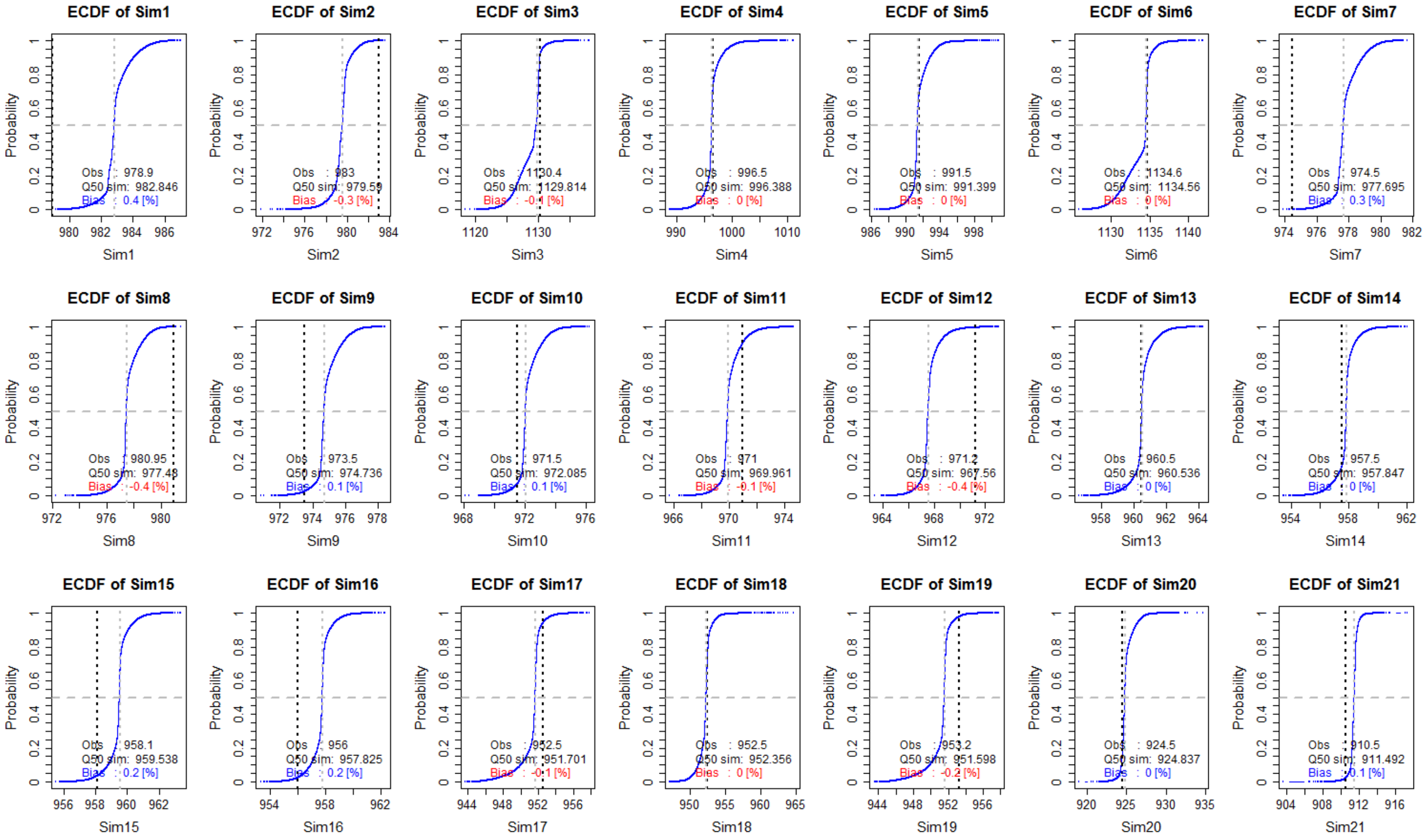

Figure 35: ECDFs for groundwater heads at 42 observation wells for M2. Vertical dashed-line represents the observed value.

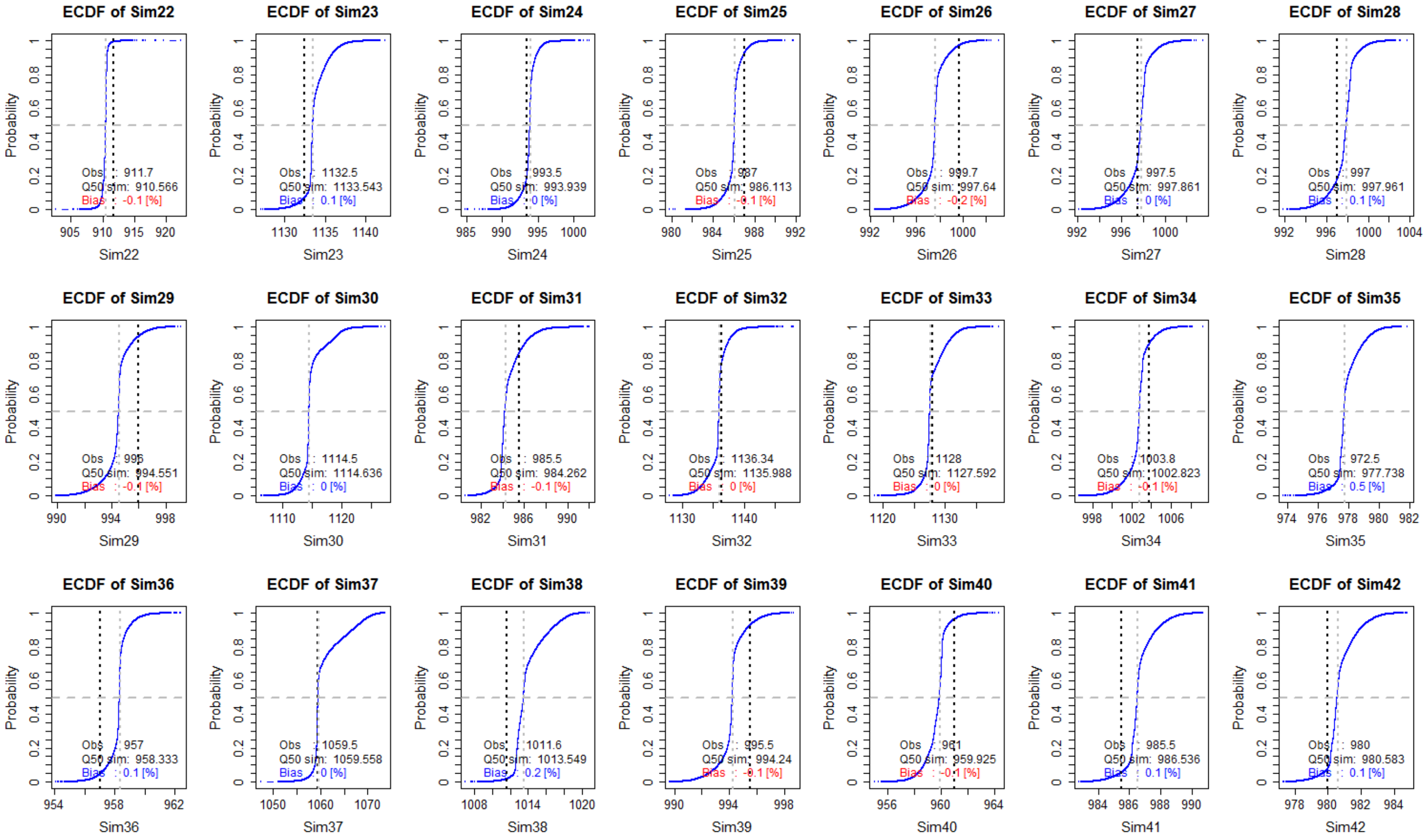

Figure 36: ECDFs for groundwater heads at 42 observation wells for M2. Vertical dashed-line represents the observed value.

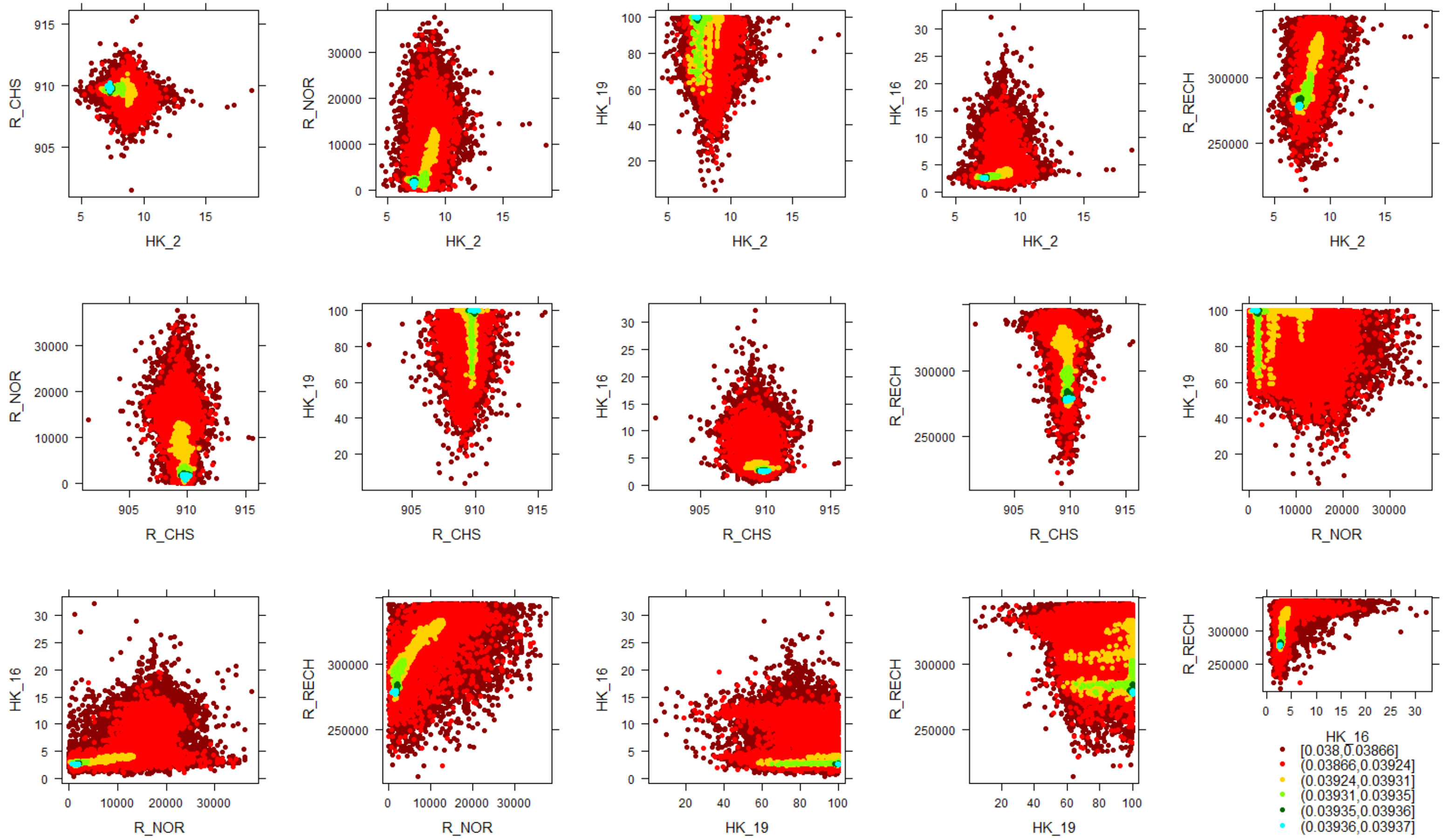

Figure 37: Idem as Figure 32 with a "behavioural" threshold of 3.8×10^{-2}.

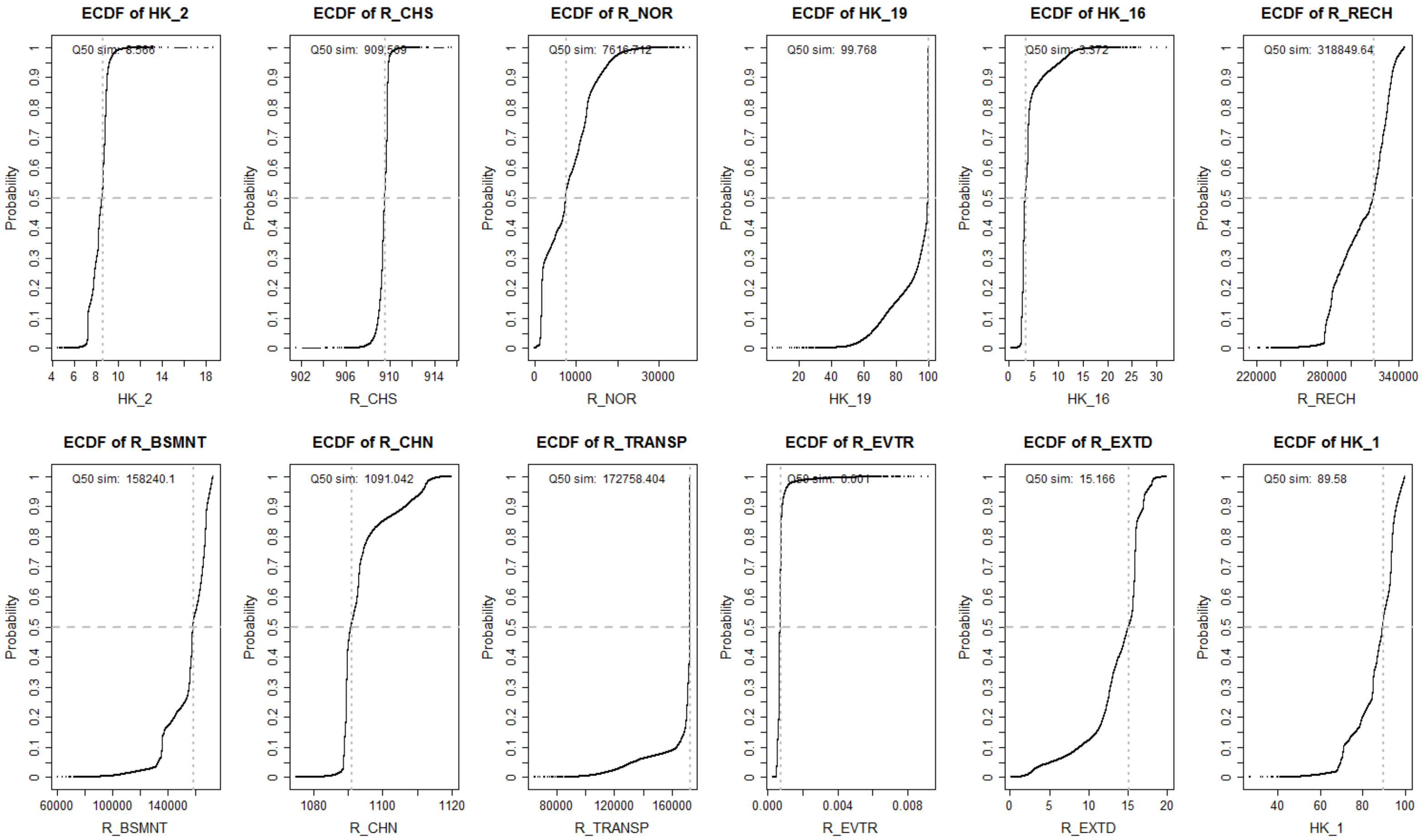

Figure 38: Idem as Figure 33 with a “behavioural” threshold of $3.8{\times}10^{-2}$.

6 Closing Remarks

1. This tutorial aimed at providing a general overview of the capabilities of the *hydroPSO* R package. In particular, we illustrated how to calibrate two model codes commonly used in hydrology-related applications, SWAT-2005 and MODFLOW-2005. Given the flexibility of *hydroPSO*, we believe this calibration/optimisation engine can be applied to a wider range of environmental models requiring some form of parameter estimation.

2. The version of *hydroPSO* used in this tutorial does not take advantage of multi-core machines nor from parallelized platforms. We hope in the next future to add parallel capabilities to *hydroPSO* to benefit from modern architectures and alleviate the computational burden.

3. In *hydroPSO* we consider only single-objective functions for optimisation. Again, we hope in the next future to include multi-objective functionalities to the main algorithm.

4. Finally, investigation on improvements to the canonical PSO algorithm is one of the most dynamic and active research areas in Particle Swarm literature. Therefore, we are particularly interested in constantly updating the *hydroPSO* package according to the most relevant innovations in this research area.

This tutorial was built under:

```
[1] "i386-pc-mingw32/i386 (32-bit)"

[1] "R version 2.14.1 (2011-12-22)"

[1] "hydroPSO 0.1-51-3"
```

References

Arnold, J., and N. Fohrer (2005), SWAT2000: current capabilities and research opportunities in applied watershed modelling, *Hydrological Processes*, *19*(3), 563–572, doi: 10.1002/hyp.5611.

Arnold, J., R. Srinivasan, R. Muttiah, and J. Williams (1998), Large area hydrologic modeling and assessment part I: Model development, *Journal of the American Water Resources Association*, *34*(1), 73–89.

Beven, K. (2006), A manifesto for the equifinality thesis, *Journal of Hydrology*, *320*(1–2), 18–36, doi: 10.1016/j.jhydrol.2005.07.007.

CHE (2000), Los aprovechamientos en la cuenca del ebro: Afección en el régimen hydrológico fluvial -2000-ph-24i [the exploitation in the ebro basin: Alterations to the hydrological regime of the river - 2000-ph-24i], *Tech. rep.*, Confederación Hidrográfica del Ebro, available on http://oph.chebro.es/DOCUMENTACION/Hidrologicos/RegimenHidrologico.htm. [last accessed Oct-2011].

Clerc, M. (2007), Back to random topology, *Tech. rep.*, Particle Swarm Central.

Clerc, M., and J. Kennedy (2002), The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, *6*, 58–73, doi: S10890778X(02)02209-9.

Clerc, M., A. Auger, T. Blackwell, D. Bratton, S. Croussette, R. Eberhart, N. Hansen, J. Keko, H. Kennedy, R. Krohling, W. Langdon, W. Li, V. Miranda, R. Poli, P. Serra, and M. Stickel (2011), Standard PSO 2007 - SPSO2007.

Di Luzio, M., R. Srinivasan, and J. Arnold (2004), A GIS-coupled hydrological model system for the watershed assessment of agricultural nonpoint and point sources of pollution, *Transactions in GIS*, *8*(1), 113–136, doi: 10.1111/j.1467-9671.2004.00170.x.

Eberhart, R., and J. Kennedy (1995), A new optimizer using particle swarm theory, in *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, pp. 39–43, doi: 10.1109/MHS.1995.494215.

Eberhart, R., and Y. Shi (1998), Comparison between genetic algorithms and particle swarm optimization, in *Evolutionary Programming VII*, vol. 1447, edited by V. Porto, N. Saravanan, D. Waagen, and A. Eiben, pp. 611–616, Springer Berlin / Heidelberg, doi: 10.1007/BFb0040812.

Eberhart, R., and Y. Shi (2001), Particle swarm optimization: developments, applications and resources, in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 81–86, doi: 10.1109/CEC.2001.934374.

Evers, G., and M. Ghalia (2009), Regrouping particle swarm optimization: A new global optimization algorithm with improved performance consistency across benchmarks, in *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009*, pp. 3901–3908, doi: 10.1109/ICSMC.2009.5346625.

Gelman, A., J. Carlin, H. Stern, and D. Rubin (2004), *Bayesian data analysis*, second ed., 696 pp., Chapman & Hall/CRC, New York.

Harbaugh, A. (1990), A computer program for calculating subregional water budgets using results from the u.s. geological survey modular three-dimensional ground-water flow model, *Open file report 90-392*, United States Geological Survey, Reston, Virginia, USA.

Harbaugh, A. (2005), MODFLOW–2005, the U.S. Geological Survey modular ground-water model–the Ground–Water Flow Process, *Techniques and methods 6–a16*, United States Geological Survey, Reston, Virginia, USA.

Hill, M., E. Banta, A. Harbaugh, and E. Anderman (2000), MODFLOW–2000. the U.S. Geological Survey modular ground-water model–User guide to the observation, sensitivity, and parameter–estimation processes and three post processing programs, *Open-file report 00-184*, United States Geological Survey, Denver, Colorado, USA.

Holvoet, K., A. van Griensven, P. Seuntjens, and P. Vanrolleghem (2005), Sensitivity analysis for hydrology and pesticide supply towards the river in swat, *Physics and Chemistry of the Earth, Parts A/B/C*, *30*(8–10), 518–526, doi: 10.1016/j.pce.2005.07.006.

Kannan, N., S. White, F. Worrall, and M. Whelan (2007), Sensitivity analysis and identification of the best evapotranspiration and runoff options for hydrological modelling in SWAT-2000, *Journal of Hydrology*, *332*(3-4), 456–466, doi: 10.1016/j.jhydrol.2006.08.001.

Kennedy, J., and R. Eberhart (1995), Particle swarm optimization, in *Proceedings IEEE International Conference on Neural Networks, 1995*, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

Kennedy, J., and R. Mendes (2002), Population structure and particle swarm performance, in *Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02*, pp. 1671–1676, doi: 10.1109/CEC.2002.1004493.

Mendes, R., J. Kennedy, and J. Neves (2004), The Fully Informed Particle swarm: Simpler, maybe better, *IEEE Transactions on Evolutionary Computation*, *8*(3), 204–210, doi: 10.1109/TEVC.2004.826074.

Muleta, M., and J. Nicklow (2005), Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model, *Journal of Hydrology*, *306*(1-4), 127–145, doi: 10.1016/j.jhydrol.2004.09.005.

Nash, J., and J. Sutcliffe (1970), River flow forecasting through conceptual models. Part I–A discussion of principles, *Journal of Hydrology*, *10*(3), 282–290, doi: 10.1016/0022-1694(70)90255-6.

Neitsch, S., J. Arnold, J. Kiniry, R. Srinivasan, and J. Williams (2005), *Soil and Water Assessment Tool Theoretical Documentation Version 2005*, Grassland, Soil and Water Research Laboratory; Agricultural Research Service 808 East Blackland Road; Temple, Texas 76502; Blackland Research & Extension Center; Texas Agricultural Experiment Station 720 East Blackland Road Temple, Texas 76502, USA.

Poli, R., J. Kennedy, and T. Blackwell (2007), Particle swarm optimization, *Swarm Intelligence*, *1*(1), 33–57, doi: 10.1007/s11721-007-0002-0.

R Development Core Team (2011), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Rojas, R., and A. Dassargues (2007), Groundwater flow modelling of the regional aquifer of the Pampa del Tamarugal, northern Chile, *Hydrogeology Journal*, *15*(3), 537–551, doi: 10.1007/s10040-006-0084-6.

Rojas, R., L. Feyen, and A. Dassargues (2008), Conceptual model uncertainty in groundwater modeling: Combining generalized likelihood uncertainty estimation and Bayesian model averaging, *Water Resources Research*, *44*(12), W12,418, doi: 10.1029/2008WR006908.

Rojas, R., O. Batelaan, L. Feyen, and A. Dassargues (2010), Assessment of conceptual model uncertainty for the regional aquifer Pampa del Tamarugal–North Chile, *Hydrology and Earth System Sciences*, *14*(2), 171–192, doi: 10.5194/hess-14-171-2010.

Shi, Y., and R. Eberhart (1999), Empirical study of particle swarm optimization, in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, p. 1945, doi: 10.1109/CEC.1999.785511.

van Griensven, A., T. Meixner, S. Grunwald, T. Bishop, M. Diluzio, and R. Srinivasan (2006), A global sensitivity analysis tool for the parameters of multi-variable catchment models, *Journal of Hydrology*, *324*(1–4), 10–23, doi: 10.1016/j.jhydrol.2005.09.008.

van Liew, M., J. Arnold, and D. Bosch (2005), Problems and potential of autocalibrating a hydrologic model, *Transactions of the ASAE*, *48*(3), 1025–1040.

van Liew, M. W., T. L. Veith, D. D. Bosch, and J. G. Arnold (2007), Suitability of SWAT for the Conservation Effects Assessment Project: Comparison on USDA Agricultural Research Service watersheds, *Journal of Hydrologic Engineering*, *12*(2), 173–189, doi: 10.1061/(ASCE)1084-0699(2007)12:2(173).

Zambrano-Bigiarini, M., and R. Rojas (2012), hydroPSO: A model-independent particle swarm optimization software for calibration of environmental models, *Environmental Modelling & Software*, *submitted.*

Zhao, B. (2006), An improved particle swarm optimization algorithm for global numerical optimization, in *Computational Science - ICCS 2006*, *Lecture Notes in Computer Science*, vol. 3991, edited by V. Alexandrov, G. van Albada, P. Sloot, and J. Dongarra, pp. 657–664, Springer-Verlag Berlin Heidelberg.