Seventh Framework Programme

Call FP7-ICT-2009-6

Project: 247708 - SUDPLAN

Project full title:

Sustainable Urban Development Planner for Climate Change Adaptation

Deliverable D4.3.3

WP4 Common Services: Hydrological Downscaling Service V3

Due date of deliverable: 31/08/2012

Actual submission date: 06/09/2012

Document Control Page

Title	WP4 Common Services: Hydrological Downscaling Service V3	
Creator	SMHI	
Editor	Lena Strömbäck	
Description	Description of the V2 prototype for Hydrological Downscaling Service. The description focuses on the main functionality and underlying technical implementations to be as a service in other modules. This document does not include the description of the Hydrology Common Service User Interface (SMS).	
Publisher	SUDPLAN Consortium	
Contributors	SMHI	
Type	Text	
Format	application/msword	
Language	EN-GB	
Creation date	08-10-2012	
Version number	1.0	
Version date	06-09-2012	
Last modified by	Lena Strömbäck	
Rights	Copyright “SUDPLAN Consortium”. During the drafting process, access is generally limited to the SUDPLAN Partners.	
Audience	☐ internal ☒ public ☐ restricted, access granted to: EU Commission	
Review status	☐ Draft ☒ WP Manager accepted ☐ PMC quality controlled ☒ Co-ordinator accepted	Where applicable: ☐ Accepted by the PMC as public document
Action requested	☐ to be revised by Partners involved in the preparation of the deliverable ☐ to be revised by Technical reviewer ☐ for approval of the WP Manager ☐ for approval of the Quality Manager ☐ for approval of the Project Co-ordinator ☐ for approval of the PMC	
Requested deadline	31-08-2012	

Revision History

Version	Date	Modified by	Comments
0.1	08-10-2012	Lena Strömbäck	First draft based on D4.3.2
0.2	08-28-2012	Lena Strömbäck	First version for review
1.0	05-09-2012	Lena Strömbäck	Revision based on review
1.0	06-09-2012	Lars Gidhagen	Co-ordinator approval

Table of Contents

1 Management summary

SUDPLAN Common Services represents a common functionality to be used by the four SUDPLAN pilot cities. The functionality is general and can be used in all European cities. The Common Services (CS) offers urban downscaling services for intense rainfall, hydrological variables (including flooding, droughts and water availability) and air quality; accessible on the web through the Scenario Management System (SMS).

The following downscaling functionality is implemented in Common Services:

- Intense rainfall: urban downscaling – generation of short-term precipitation data (time series or IDF curves) for urban hydrological climate change impact assessment
- Hydrological conditions: urban downscaling – local calibration of hydrological variables and generation of future runoff scenarios — COVERED IN THIS REPORT
- Air quality: urban downscaling – generation of local future air quality scenarios taking into account local emissions

This report documents the hydrological downscaling services from an end user point of view. A more detailed IT documentation can be found in the OGC and Back-back end appendices of D4.1.3 Concerted Approach V3. The present document describes how the hydrological downscaling model system fits and works inside the Common Service environment. An important development has been the automatic calibration routine, which exemplify an automation of model execution to optimise parameter settings.

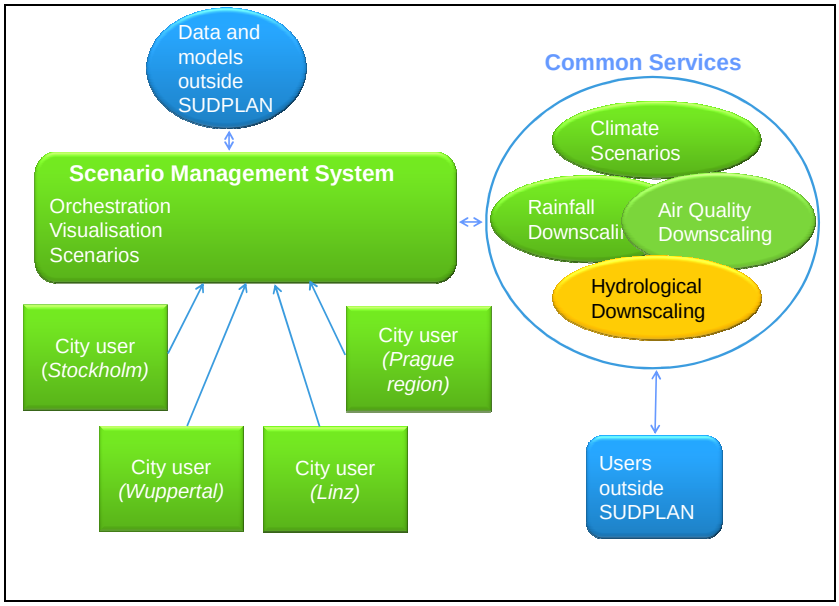
The use of the SUDPLAN platform to manage hydrological downscaling will be demonstrated in WP4 by external end users from the Swedish Water Authority.

This is a public document available at http://sudplan.eu/Results

2 Hydrology: Local downscaling

Common Services (CS) provides the climate and environmental information to end user through the Scenario Management System (SMS), which is a model control, visualisation and integration workbench for all SUDPLAN components (Figure 1). A standardized communication will assure an easy setup connection to the Common Services, also for other software which require climate services of this type.

Figure 1 Overview of SUDPLAN components. The communication between Common Services and the Scenario Management System uses standardized services (OGC).



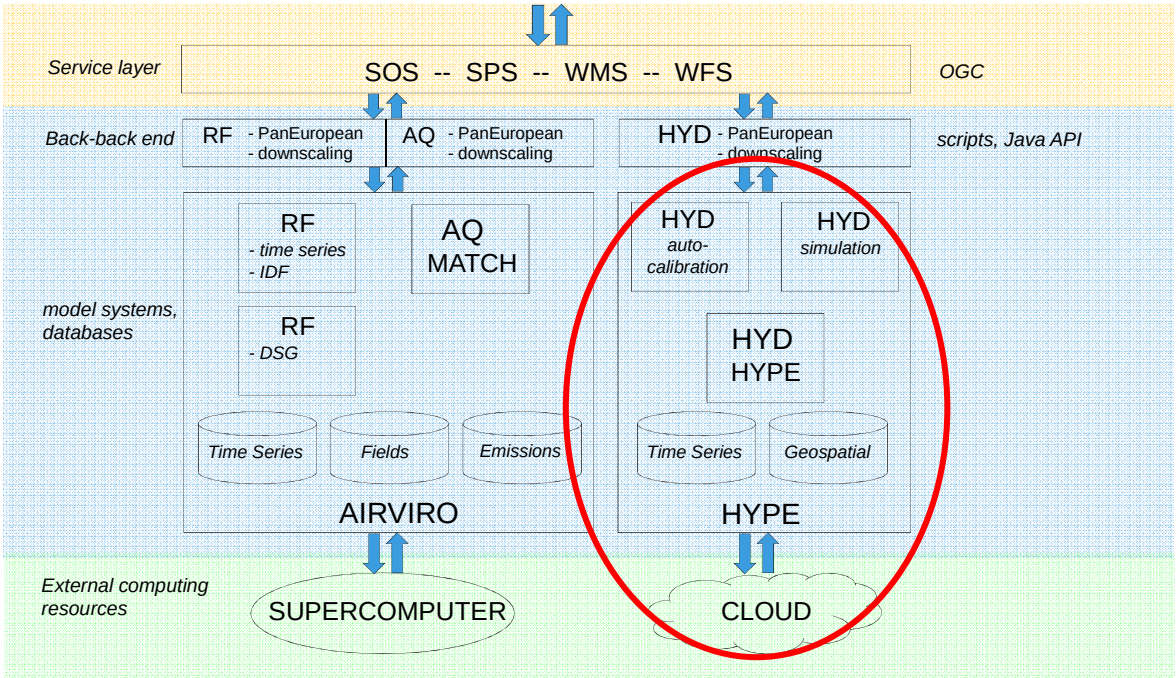


Figure 2 Technical solution of Common Services, with hydrological downscaling components marked.

The technical structure of Common Services is illustrated in Figure 2. The rainfall and air quality downscaling, together with the corresponding Pan-European climate and environmental

information, have been implemented in an existing software, the Airviro system. Input and output data is either point wise time series or gridded time series. The hydrological data is based on point wise time series and irregular polygon data representing watersheds, managed through the existing HYPE model system. Therefore the back-back end solutions are also split in two parts. The service layer does however streamline the communication to all Common Services, so that external end user applications will only have to follow the OGC standards of four services SOS, SPS, WMS and WFS in order to establish communication.

The hydrological downscaling components (marked in Figure 2) form an extension of the HYPE model system. This document describes how the downscaling is performed in SUDPLAN. Technical descriptions concerning the hydrological model HYPE and other components of the hydrological part of Common Services environment is found in D4.1.3 Common Services concerted approach V3. The D4.1.3 document includes appendices documenting the OGC communication and also an appendix which documents the technical solution of the back-back end, i.e. how to communicate with the Common Services, select model, model domain and parameters, input data (emissions, forcing, boundary conditions), proceed to execution and then to access output. The description given in the present document focuses more the user functionality, leaving the specific implementation and specification details to the D4.1.3 document.

The SUDPLAN platform to manage hydrological downscaling will be evaluated by representatives from the Swedish Water Authorities. These authorities, in total five representing different Swedish regions, were formed in response to the EU water directive to provide a comprehensive management of water related issues in Sweden. These users will, as part of WP4, be trained to operate the SMS and Common Services. They will, as external end-users, validate the SUDPLAN functionality of running and visualize hydrological simulations for future years, based on climate scenarios.

The procedure of hydrological downscaling starts by creating a sub-model of the Pan-European hydrological model setup of HYPE, E-HYPE, for the river or stream running through the specific city of interest and the catchment upstream of the city. From the SMS, the user can run an automatic calibration of the model to optimize model parameterization for this catchment. The user can also opt to add their own measured discharge data, if available, to improve this calibration. Once the new, local hydrological model is calibrated, it can be used to run selected climate scenarios.

Note, that unlike the other SUDPLAN downscaling services, the spatial resolution of the output remains the same at European and local scales. It is only the quality of the downscaled results that is improved through calibration with local data. The following sections gives an overview of the main concepts in SUDPLAN hydrological downscaling and climate scenario projections

2.1 Use case: Hydrological downscaling and calibration

The first aim of the service is to provide access to results of future hydrological scenarios on a Pan-European scale, calculated with climate scenarios as input. The second aim of the service is to provide an easy to use downscaling service where the end user can improve and adapt climate scenario data to local measurements. An important aim in defining the service has been to minimize the required user input. Two types of user specifications are needed to launch a model

simulation, the geographic area of interest and the selected climate scenario, with a third option given to add user's local data.

Within the SUDPLAN Czech regional pilot definition plan (D8.1.2) two use cases have been defined that describe the main functionality of the hydrological downscaling service. These use cases is the basis for the downscaling functionality. This section gives a description of the basic use cases, to give an understanding of the problem before describing how the functionality was realised in the SUDPLAN SMS. The realisation of these use cases in the SMS is described in section 5 of this document.

2.1.1 Use case UC-832: Auto calibration of CS hydrological model

This use case provides automatic calibration of the CS hydrological model for one point in the map. The goal with the service is to provide improved simulation results for upstream watersheds in the service. The user needs to be logged in to the CS and has the option of providing a time-series of discharge data from a selected gauging station or using an existing station. The results of the service are new parameter sets that can be used for subsequent runs of the hydrological services and an improved hindcast of discharge at the site using the new parameter set. The use case defines the following steps for the user to perform the service:

1. See on a map the Q stations uploaded in SUDPLAN and upload new observed time series to be used for calibration.
2. Choose point of interest by clicking on it on the map. This should be the location of the chosen gauging station or the point for which new gauging station data is to be uploaded.
3. Select the time series to be used for calibration.
4. Start automated calibration
5. Simulation progress visualisation
6. Retrieve and visualise time-series of simulated and observed variables

2.1.2 Use Case UC-833: Execute CS hydrological model

In this use case the user wants to execute the CS hydrological model to simulate future climate scenarios for the area upstream of the selected point of interest (POI). Auto calibration (previous use case) should have been performed for the same upstream area before this step. The goal of the service is to provide an analysis of the effects of future climate on hydrological conditions, based on a CS climate scenario. The hydrological model uses precipitation and temperature (P, T) as input data from the climate projections as well as the newly calibrated parameter set. The outputs of the use case are as time series of some hydrological variables for the sub-basins upstream of the point of interest. The use case defines the following steps for the user to perform:

1. Select POI (should be the same catchment as for the calibration, otherwise the results will be the same as in the Pan-European results)
2. Select climate scenario (note that the service uses corrected P (precipitation) and T (temperature) from the climate model)
3. Select simulation period (default is the same as input climate scenario)
4. Start model run

5. Simulation progress visualisation
6. Get back daily values time series of values for input P, T and model output variables
7. Export of visualised time series

2.2 Main principles for hydrological downscaling

Figure 5 shows the main principles for the hydrological downscaling and the relation to the user interaction performed from the SMS. The green parts of the figure are the main functions of the hydrological downscaling CS, whereas the blue parts show the interface to the SMS.

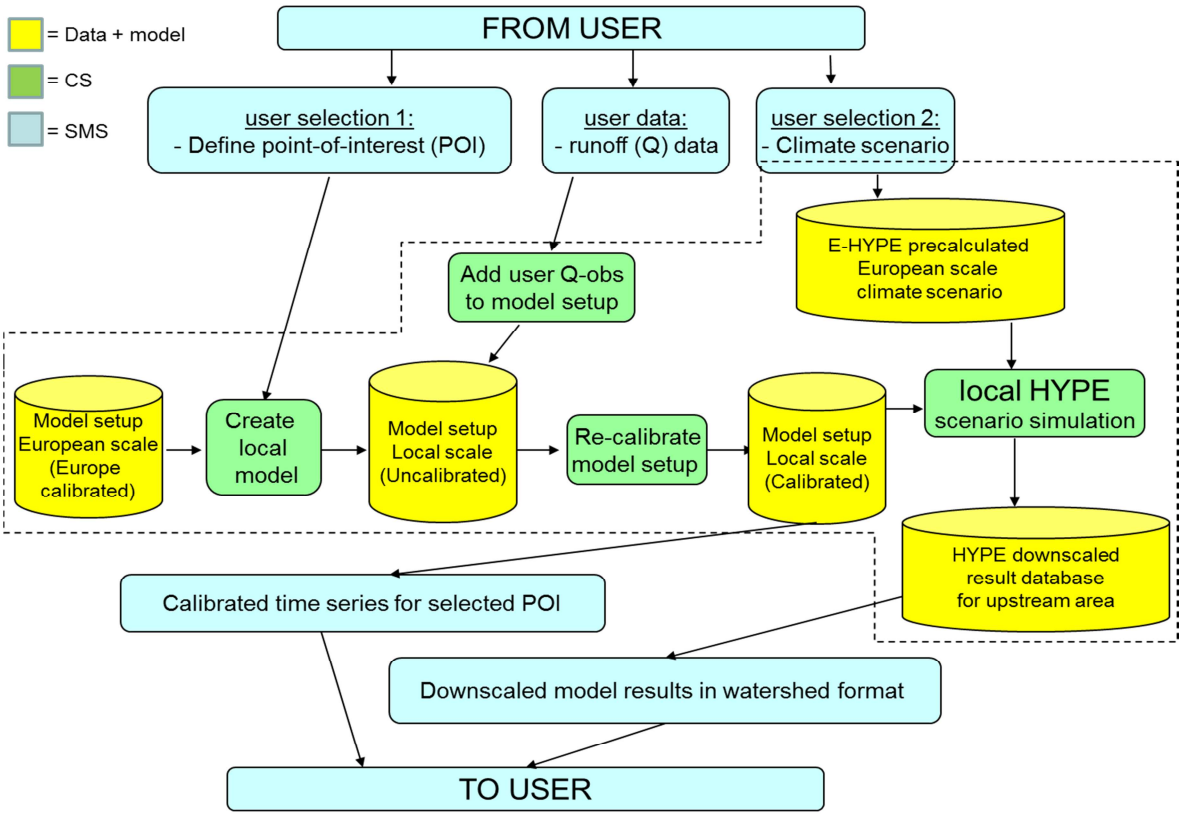


Figure 5: Components and data flows between SMS and CS for hydrological downscaling.

The model execution includes the following steps:

1. Creation of the local sub-model from the Pan-European model
2. Recalibration of the local sub-model (using historical precipitation and temperature inputs)
3. Running of local sub-model to simulate future hydrological scenarios using local sub-model (using climate scenario precipitation and temperature inputs)

The technical solution of the CS includes each of the individual processes and (the green parts) in Figure 5, as well as models, data and automation of the steps linking each of these processes (yellow in the figure). The implementation of these modules and how they connect to the SMS environment is further described in Section 4 and 5 of this document

Before and after each step described in the use case, the user will be able to visualise the modelled results and, where relevant, compare these to measured results via the SMS. The climate scenario precipitation and temperature will be stored. The local model setup including any new Q data (observed discharge) added by the user and new parameters resulting from auto calibration will be stored by the CS, time-series outputs from the scenario runs will be temporary, and statistics calculated as a result of the scenario runs will be stored and catalogued by the SMS.

In the course of the project it was necessary to extend HYPE functionalities and develop routines for sub-model creation and auto calibration to support the use cases described.

2.3 Outputs

For the generation of Pan-European E-HYPE scenario runs, a post-processing is made to yield a number of statistical variables of interest to the user. Table 1 lists these outputs including a short description, valid for the PE visualisation.

Table 1 E-HYPE output data for spatial visualisation and time series export on the Pan-European scale

Raw model output (daily/monthly/annual/10-year)	
Mean Q	River discharge
Mean specific runoff	Local runoff
Mean relative soil moisture	The soil moisture deficit in the root zone (in mm), in relation to the field capacity
Groundwater	Relative groundwater level
DBS[1]-corrected temperature	Temperature after preparation for hydrological simulation by DBS scaling
DBS-corrected precipitation	Precipitation after preparation for hydrological simulation by DBS scaling
Statistical output (only available as 30-year mean)	
Cout_highflow-T10	The discharge magnitude that on average occurs once every 10 years.
Cout_highflow-T50	The discharge magnitude that on average occurs once every 50 years.
Mean High Flow (MHQ)	The mean annual high water discharge in a river/stream.
Mean Low Flow (MLQ)	The mean annual low water discharge in a river/stream
Hydrological drought, number of days	The number of days (per year) with hydrological drought. This is defined as the number of days when the flow is higher than the 10th percentile of the flow in the reference period.
Hydrological drought, intensity	The intensity of hydrological drought, i.e. how much below a threshold value for the reference period the flow is for the dry days.
Agricultural drought, number of days	The number of days with agricultural drought. Similar to the hydrological drought, but based on days over the 90th percentile of the relative soil moisture compared to the reference period.
Agricultural drought,	The intensity of agricultural drought, i.e. how much above a threshold

[1] Distribution Based Scaling

intensity	value for the reference period the relative soil moisture is for the dry days.
Snow days	The number of days where the snow depth is more than 1 cm.
Snow max	Maximum snow depth

The output of the hydrological downscaling contains the model output and is summarised in Table 2.

Table 2 Time series output from Common Services hydrological downscaling model HYPE

Output (daily values)	
Mean Q	River discharge
Mean specific runoff	Local runoff
Mean relative soil moisture	The soil moisture deficit in the root zone (in mm), in relation to the field capacity
Groundwater	Relative groundwater level
DBS[2]-corrected temperature	Temperature after preparation for hydrological simulation by DBS scaling
DBS-corrected precipitation	Precipitation after preparation for hydrological simulation by DBS scaling

[2] Distribution Based Scaling

3 Hydrological downscaling using HYPE

The Hydrological Predictions for the Environment (HYPE) model is a dynamic, semi-distributed and process-based model based on well-known hydrological and nutrient transport concepts [2]. It can be used for both small and large scale assessments of water resources and water quality. In the model, landscape is divided into classes according to soil type, vegetation and altitude. The soil representation is stratified and can be divided in up to three layers, each with individual characteristics and hence calculations. The model simulates water flows, transport and turnover of nitrogen, phosphorus and inert trace substances. These substances follow the same pathways as water in the model: surface runoff, macro pore flow, tile drainage and groundwater outflow from the individual soil layers. Rivers and lakes are described separately with routines for turnover of nutrients in each environment. Model coefficients are global, or related to specific characteristics of Hydrological Response Units (HRU), i.e. combinations of soil type and land-use. Internal model components are checked using corresponding observations from different sites [3]. A schematic overview of the model and its main components is given in figure 4.

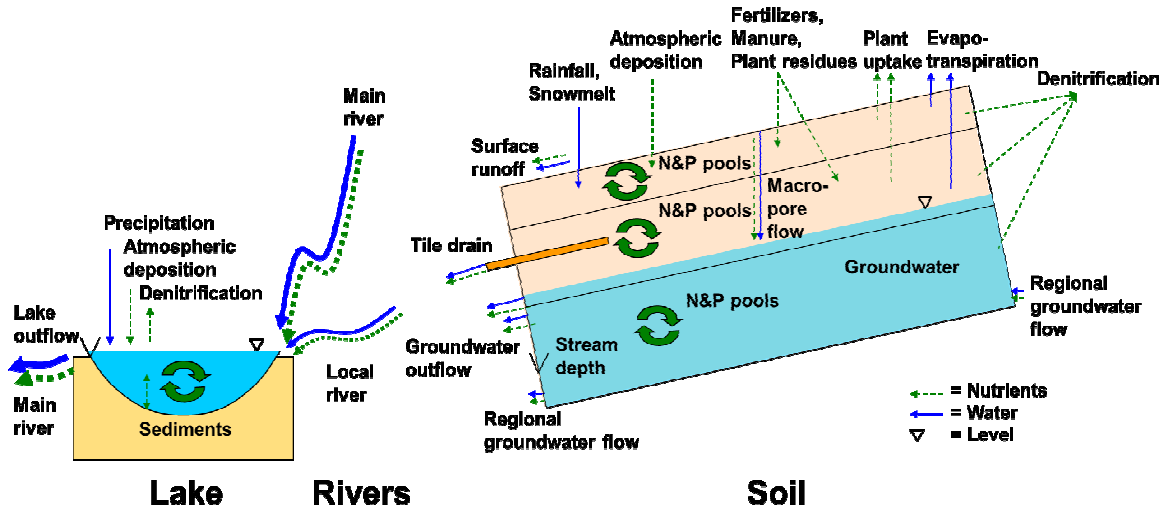


Figure 4 The main components of the Hype hydrological model.

So far, the HYPE model has been applied pan-European for water (E-HYPE1.0), while nutrients are only modelled for Sweden (S-HYPE) and the Baltic Sea basin (Balt-HYPE). The HYPE model code is structured so that the model can be easily applied with high spatial resolution over large model domains, which is also facilitated by linking coefficients to physical characteristics and the multi-basin calibration procedure [4]. The model is not calibrated site specifically, but simultaneously for groups of parameters referring to different parts of the model structure. This makes the model robust for large regions although specific sites may not be simulated as well as if local calibration is applied.

In this section we will describe how SUDPLAN uses E-Hype to perform hydrological downscaling, including selection of relevant geographical area, automatic calibration and details about the geographical and input parameters setup used in the project.

3.1 E-Hype 2.0

A first version of a Pan-European hydrological model (E-HYPE) with high resolution is operational in the SMHI production environment. The model has been initiated and funded by GMES and several EU FP7 projects (e.g. Geoland2, MyOcean, OPERR, Ecosupport). The large-scale application of the hydrological model is possible due to the ready availability of regional and global databases which are handled in a specially designed system for automatic generation of model input data, WHIST [5]. The current version (E-HYPE2.0) has been set up, calibrated, and validated against observed time-series. Table 3 gives an overview of input data for E-Hype and E-Hype 2.0. E-Hype 2.0 gives the basic data setup for the Pan-European data simulations as well as for the local downscaling simulations available in SUDPLAN.

Table 3. E-HYPE 2.0 model applications and input data

	E-HYPE2.0
Areal extent	8.8 million km2
Median Sub-basin Resolution	214 km2
No. Sub-basins	35447
Topography/routing	Hydrosheds [8] and Hydro 1K (for latitude > 60 deg) [7]
Forcing Data	ERA-INTERIM
Landcover	CORINE [10] and Globcover 2000 (for areas not covered by CORINE) [11]
Urban Area	Euroland SoilSealing 2009 [12]
Lake area and spatial distribution	GLWD (Global Lake and Wetland database) [13]
Lake and reservoir information	GLWD (Global Lake and Wetland database) [13] ERMOBST [14] FLAKE-Global [15] International Water Power & Dam Construction yearbook 2011 [16] ILEC World Lake database[17] LEGOS [17] Swedish Water archive (SMHI) [6]
Irrigated area	European IrrigationMap [18] GIAM [19]
Soil types	European Soils Database [20] and DSMW (Digital Soil Map of the World) [22]
Discharge measurements	GRDC [23], EWA [25], BHDC [24]
Evaporation measurements	COSMO[27] and FLUXDATA[28]

3.2 Creation of local sub-model

The first step in selecting the local sub-model is to extract a relevant area and based on this create a local version of E-Hype 2.0 for this area. An executable program, “SelectAro” was developed to create this local sub-model from the Pan-European model. Input to this program is the sub-basin identification number at the outlet of the local sub-model, i.e. the downstream point-of-interest. This number is extracted when the user clicks on the sub-basin in the map. All sub-basins upstream of the selected area(s) are automatically selected by the executable program to create a complete sub-model which can be run by HYPE.

Based on the selected sub-basin and the derived upstream sub-basins, SelectAro reads the indata files for E-Hype 2.0, selects data relevant for the new model to be created and generates new

input data files for the local model. This new model can be used for predictions for the selected area, but results can be improved by calibrating the model with local observed data.

3.3 Automated calibration

For SUDPLAN an automatic calibration routine has been implemented and added to the HYPE model source code. The SUDPLAN system also provides testing and definition of inputs to the automatic calibration routine so that initiation of this routine is also automated. The auto calibration builds on well-established numerical methods and the implemented package provides a number of different algorithms for the expert user. In this section we present an overview of these functionalities and methods. We also give some results for validating the algorithms. A more detailed validation of the methods is presented in D 8.2.2 Czech Pilot V2 [29].

3.3.1 Provided methods for automatic calibration

The implemented auto calibration package provides a set of methods for automatic calibration of E-Hype with historical discharge data for a local area. All implemented methods are adaptations of well-established algorithms within numerical optimization. In this section we give a brief overview of the selected functionality a more detailed description of the theory behind the functions is given in appendix A.

The auto calibration package offers two classes of calibration methods, sampling methods and directional methods. The sampling method offered by the package is the **Progressive Monte Carlo method**. This method provides a simple way to gather knowledge of an objective function by sampling of realizations of this function under variation of its argument values. The Monte Carlo method relies on random sampling; parameter sets are generated without any form of organization, and all parameter values are varied simultaneously. This method offers the advantage of being applicable to parameter spaces of any dimension. The Monte Carlo method used in the auto calibration incorporates a refinement, consisting of a progressive, stage-wise reduction of the parameter space around promising parameter sets. The concept is based on the radius of the original parameter space that is defined as half the distance between the original parameter space boundaries. From the computational point of view, sampling methods present the advantage that the amount of function evaluations is prescribed by the user, which allows for estimating rather accurately the computational time required to perform the task.

The package offers two different classes of directive methods Quasi Newton and Brent. **Quasi-Newton** methods are a class of optimization methods based on the first order Newton expansion of the gradient of the objective function. After determining the gradient the method performs a line search in the calculated direction. A wide range of methods satisfying these conditions have been developed in the framework of optimization. Three of those where of interest for our auto calibration routine: Steepest descent, DFP[3] and BFGS[4].

In addition, the package implements a second directive method the **Brent method**. In the Brent method, determination of the step direction is trivial since the method permutes through all

[3] Davidson-Fletcher-Powell formula
[4] Broyden-Fletcher-Goldfarb-Shanno formula

dimensions of the parameter space, reducing the problem to that of successive line searches. Given a starting parameter set, $\vec{p}_0$, the algorithm first proceeds to a line search along the dimension of the first component, from boundary to boundary, but keeping the value of all other parameter set components constant in the process. The line search is performed so as to find a first component value that improves the objective function in the sense of optimization. Once a better value has been found, the parameter sets first component is updated. The algorithm then proceeds similarly to improve the other parameters in the set component. Thus, the method performs a series of successive updates of all parameter set components, one after the other, in a fixed order.

The Brent method does not make use of the knowledge of the objective function gathered from previous iteration. In addition, the steepest descent method does not make use of the knowledge of the objective function gathered from previous iterations. Therefore, the DFP method was the first quasi-Newton method tested for the auto calibration routine. As all quasi-Newton methods it provides a faster progression than the Brent method. This is due to optimizing all parameters together, instead of looping through one parameter at a time. However, the curvature estimate by the DFP formula was found to be poor, leading to an accumulation of unnecessary small suggested steps, $\vec{d}_k$. A more robust update formula was therefore required, and attention was turned toward the nowadays widely accepted BFGS method. Of the quasi-Newton methods, the **BFGS** update was found to be the most robust and fastest method. It has the best ability to estimate the objective surface curvature, what results in the most efficient next step suggestions, and overall the best progression for a given amount of iterations.

3.3.2 Line search algorithm

All directional methods above are dependent on determining a step direction, $\vec{d}_k$, and the step length, λ_k., for each iteration. Provided a step direction, $\vec{d}_k$, the other ingredient of directional methods is the selection of a step length, λ_k. This is achieved by means of a so-called line search algorithm. The line-search algorithm used in the calibration package is a mixture of interval segmenting and quadratic fitting.

The interval segmentation part of the algorithm is based on the golden ratio, $\varphi = \frac{1+\sqrt{5}}{2} = 1.618034$, the only positive solution to the equation $1 + 1/\varphi = \varphi$. Typically, a test point is taken at $1 - 1/\varphi = 0.381966$ within the interval, measured from the lower interval boundary. The objective function is evaluated there, and the result is compared with earlier evaluations of the objective function at the interval boundaries. If the situation is favorable, a new test point is obtained by quadratic fitting of the three test points with a parabola, and localization of the parabola minimum. Otherwise the interval is further segmented, in accordance with the golden ratio proportionality principle. At any rate, the most unfavorable boundary point is always left out. Proceeding this way, the interval is gradually reduced, until its size triggers one or another condition for interruption.

All line search algorithms, incl. the one just described, suffer the same problem: if the objective function features a local minima, it cannot be guaranteed that the algorithm will converge toward the global minima. In the case at hand, this is clearly a consequence of the assumption that the objective function can be fitted with a quadratic function, by means of a parabola, which in itself already implies the assumption that the function knows only one minimum over the interval of interest. As this is generally not the case, convergence toward the global minimum cannot be

fully ensured. However, this algorithm was chosen because of its robustness and its capacity to shrink the original interval size relatively quickly.

3.3.3 Interrupters

An important issue is to determine how to terminate the calibration. In the 'autocal' package, natural ways of termination for the optimization methods above are implemented, either to stop the line search algorithm, or to exit the optimization work all together. Here, we discuss those termination methods, called interrupters.

3.3.3.1 Interrupters for sampling methods

Sampling methods do not terminate until the number of function calls specified by the user is reached.

- For the scan functionality, the amount of function calls is given by the product of the amount of grid points, in both dimensions.
- For the progressive Monte Carlo method, the total number of function calls is given by $N_s \cdot N_c \cdot N_e$, where N_s is the amount of successive stages, N_c is the amount of centers per stage, and N_e is the amount of run per center; the first stage features $N_e \cdot N_c$ function calls by default, even if there is only one center (the mean of the user-specified parameter space boundaries).

3.3.3.2 Interrupters for directional methods

For directional methods there are several termination criteria of interest:

- the maximum total calibration time, specified in hours.
- a maximum amount of iterations allowed. For the Brent method, one iteration corresponds to a full loop of line searches through all parameter space dimensions, incl. the (optional) diagonal step. For the quasi-Newton methods, one iteration corresponds to the gradient determination, the inverse Hessian determination, as well as the subsequent line search in the suggested direction.
- a precision for function value at optimum can be given, as well as a certain amount of tolerated iterations. If the function value does not vary more than the specified precision within the specified amount of last iterations, the algorithm will stop as it is considered to no longer progress sufficiently.
- for each parameter value considered, a determination precision must be specified. Further, a certain amount of tolerated iterations can be specified. If ALL parameter values do not vary more than the specified precision within the specified amount of last iterations, the method is terminated because it is considered that the optimal parameter values are determined well enough.

The quasi-Newton family of methods can further be interrupted by yet another criterion:

- a lower tolerance for the gradient norm can be specified, and the algorithm stops as soon as the gradient norm of the objective function is found lower than the specified tolerance.

3.3.3.3 Interrupters for the line search algorithm

Termination of the line search algorithm can be triggered in two ways:

- the algorithm stops as soon as the interval length is shorter than a user-specified tolerance
- termination is triggered if, for ALL parameters, the interval length corresponds to parameter values smaller than the specified precisions

3.3.4 Evaluations and conclusions

In the performed evaluations the goal was to investigate the optimal performance of each of the methods. Therefore we have run the algorithms using a high number of steps trying to find the optimal parameter setup for the local model. Table 4 shows the stopping conditions used for each algorithm. For the Monte Carlo method, we only used a stopping criteria based on number of iterations, while we for the other methods also looked at other criteria, such as, change in parameter values and improvement in result criteria.

Table 4: Stop conditions used in evaluating the different algorithms

Interruptor	**Monte Carlo**	**Brent**	**Quasi-Newton**
Max amount of iterations	Featured	Featured	Featured
Max amount of time	-	Featured	Featured
Criteria changed less than specified tolerance over specified amount of iterations	-	Featured	Featured
ALL parameter values changed less than specified tolerance over specified amount of iterations	-	Featured	Featured
Gradient norm smaller than specified tolerance	-	-	Featured
Perpendicularity of gradient and step	-	-	Featured

The evaluations were performed based on the E-HYPE parameterization. The driving data based on Pan-European information were replaced with local data for the Prague area. The results were evaluated against observed data from 2 gauging stations. This evaluation shows (Table 5) that the Monte Carlo methods gives the best performance for the auto-calibration (mean R^2 0.45), however, the drawback of this method is that it requires most time for the computation. Another interesting method is BFGS which also shows an almost as good result (mean R^2 0.29) using less than half the computation time and number of model runs compared to Monte Carlo.

Table 5: Result of the evaluations

	Calibration Method	**Mean R^2**	**Model runs**	**Time**
	E-HYPE default (none)	-0.29	1	10s
1	Monte Carlo	**0.45**	30 000	3 days 19h
2a	BFGS	**0.29**	13 534	1 day 11h
2b	DFP	-0.06	3 183	8h
3	Brent	0.26	23 958	2 days 16h

Finally we wanted to investigate whether it was possible to reduce the computation time by changing the stop conditions forcing the calibration methods to perform a lower number of iterations. Figure 5 shows a plot over the improvement expressed as mean R^2 for each iteration of three of the methods. This graph shows that all the methods give a high improvement for the first iterations, while subsequent runs give less of an improvement. This is promising as it suggests that it should be possible to achieve a much faster auto calibration by performing a lower number of iterations and using the mean R^2 as the stopping criteria. However, as shown by

the DFP method, the increase in performance can at some points be stabilized at one level, and then suddenly increase after several iterations which means that that R^2 alone as a stopping criteria may not be sufficient. We will further investigate these issues to improve the computing time of the auto calibration method.

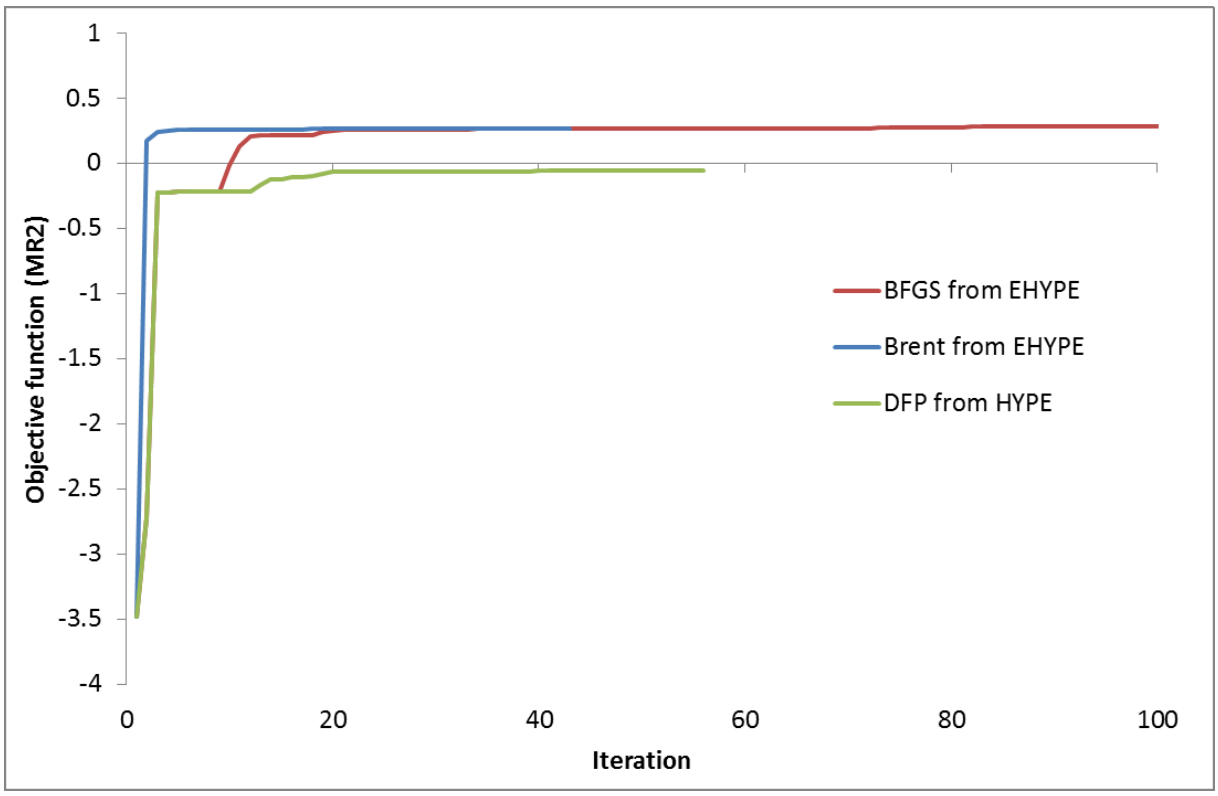


Figure 5 improvement of the auto calibration result per iteration.

3.4 Local HYPE Scenario Simulation

The goal of the hydrological downscaling service is to provide an analysis of the effects of future climate on local hydrological conditions, based on a CS climate scenario. The local model created and calibrated based on the local area and local data can be run via the SMS to calculate these predictions. The hydrological model uses precipitation and temperature (P, T) from the climate projections as well as the newly calibrated parameter set as input data. Note that the precipitation and temperature data from the climate projections have been bias corrected using the method of Yang *et al.* [30], Distribution Based Scaling (DBS). The outputs of the scenario can be the same as for the pan-European applications, i.e. spatial maps of hydrological states as well as time-series of some hydrological variables for the sub-basins upstream of the point of interest. For the implemented SUDPLAN prototype we demonstrate the functionality with a selection of these variables.

4 Technical description of the service

This section gives an overview of the implemented API (Application Programming Interface) for providing the hydrological downscaling functionality to other applications. A more detailed description of the API is given in Appendix A of D4.1.3. The implementation covers the Back-back-end of the hydrology part of the common services, i.e. the web-services to run Hype-simulations and store the results of these model runs.

4.1 General architecture

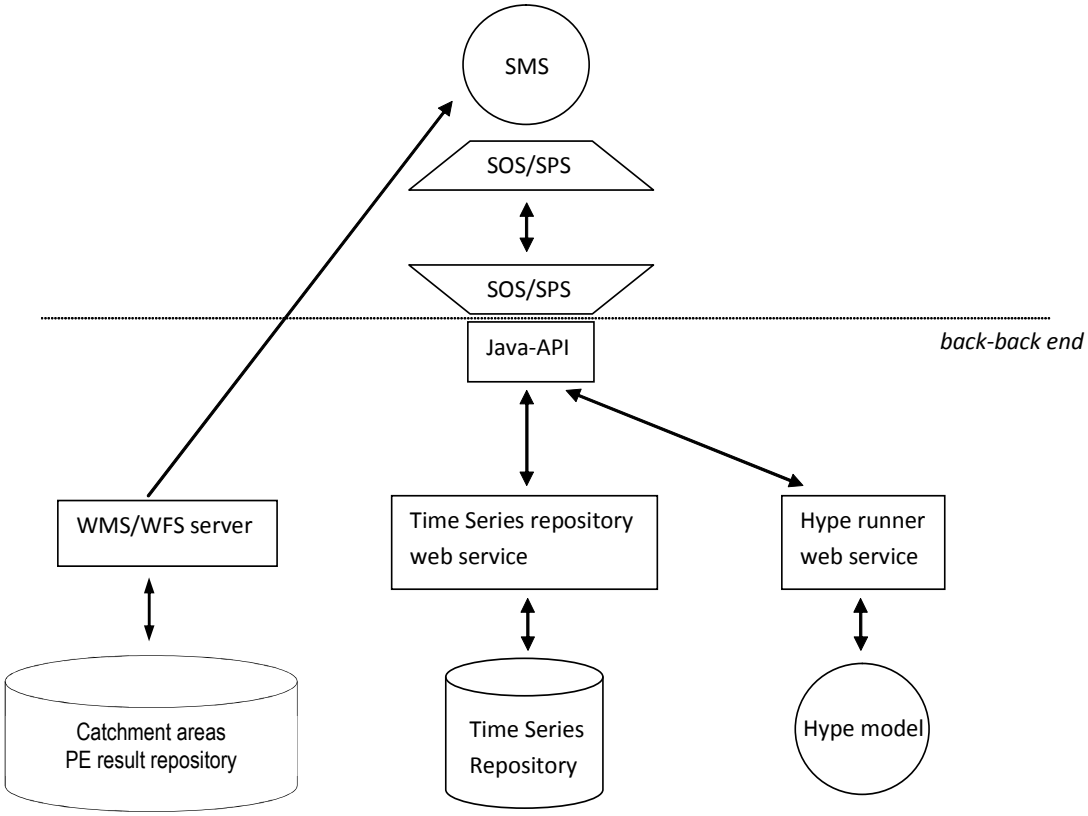


Figure 6: General architecture for the hydrological downscaling CS service.

Conceptually the back-back end consist of three separated parts, the hype simulation functionality, a repository for storing and accessing time series data and finally a WMS/WFS server. Therefore, the implementation of the back-back end of the common services consists of the following four parts:

- A web-service to run Hype simulations. This web service access the E-hype model and provides functionality for running E-hype 2.0 simulations, as well as access to downscaling, area selection and auto calibration of E-hype for a selected area.
- A web-service to store time-series, in particular this includes methods for storing and retrieving results from E-Hype simulations.
- A Java API to access those web services, which provides the interface to the back-end functionality.
- A WMS/WFS server for access to geospatial data, to service map layers and geographic information on basins that can be used for example to visualize upstream basins.

4.2 Overview of the API

In this section we give a brief description of the main functionality of the provided Java API. More details about the functionality are given in Appendix 2 of the D4.1.2 Concerted Approach V2.

4.2.1 Time Series results repository

Time-series data can be stored (*storeTimeSeries*) in and retrieved from (*getTimeSeries, getTimeSeriesMonthly, getTimeSeriesDaily. getTimeSeries10Years*) the "Time-series Repository". The result when retrieving a time-series is simply a list of date-value pairs. The repository can also be queried to retrieve metadata about time-series (*describeTimeSeries*). This operation returns information such as unit, resolution and which variable the time-series represents. Finally there is a possibility to get all available time-series identifiers or all available time series identifiers for a specific basin (*listTimeSeries*).

4.2.2 Hype model runner

The functionality provided by the Hype Model runner can be divided into three different groups;

1. preparing a simulation by setting up parameters for a downscaled simulation,
2. setting up an automatic calibration and finally
3. running the simulation scenario.

Setting up a simulation scenario involves setting a number of parameters. First, there is a need to create a work area at the "Hype Model runner" (*createSimulation*). A handle to this work area that can be used to set properties and perform calibration on the simulation before execution is returned. This involves selecting simulation scenario (*setDefaultScenario*), the geographical point of interest for the simulation (*setPointofInterest*) and selecting the time frame (*setSimulationTime*).

When properties are set, a simulation can be started (*runSimulation*). As a result a handle to the running simulation process is returned. This handle can be used to poll the status of the running process (*getExecutionStatus*).

When the execution status indicates that the execution is complete the result can be stored in the "Time-series repository" (*storeSimulationResult*). When storing the simulation result a unique identifier is connected to the time-series for later retrieval.

The procedure for auto calibration for an area is very similar to the above process. Also for the calibration process a work area has to be created for the process (*createCalibrationSimulation*). It is possible to reuse results from a previous calibration to perform further calibrations of the model (*useCalibrationFrom*). The relevant sub-area for the simulation is selected (*setPointofInterest, createSubmodel*). The observation data from the calibration is fetched from the repository (*getTimeSeries*) and merged to the given model data (*mergeObservations*).

4.3 API usage for the given use cases

In this section we show how the API is used for the above use cases.

4.3.1 UC-832 Auto calibration of CS hydrological model

This section describes necessary steps used in this use case.

- Upload observed time-series: *storeTimeSeries*
- Prepare a calibration
  Calibrate Model: *createCalibrationSimulation*
  Retrieve a previously stored time-series: *getTimeSeries*
  Add time-series to model: *mergeObservations*
  Create sub-model: *createSubmodel*
  Wait until it is done: *getExecutionStatus*, until status is DONE.
- Start the calibration simulation: *runSimulation*
  Wait until it is done: *getExecutionStatus*, until status is DONE.
- Simulation progress visualization: *getResultfileStatus*
- Handle simulation result
  Store simulation result in result repository: *storeSimulationResult*
- Retrieve simulation result: *getTimeSeries*

4.3.2 UC-833 Execute CS hydrological model

This section describes necessary steps used in this use case.

- See what Scenarios exists: *listScenarios*
- Prepare model run
  Create Simulation: *createSimulation*
  Select simulation period: *setSimulationTime*
  Select POI: *setPointOfInterest*
- Start model run: *runSimulation*
  Wait until it is done: *getExecutionStatus*, until status is DONE.
  Simulation progress visualization: *getResultfileStatus*
- Store result in result repository: *storeSimulationResult*
- Retrieve simulation result: *getTimeSeries*

5 Realisation in the SMS client

In this section we will describe how the hydrology local calibration and simulation scenarios have been integrated with the SMS client.

The first steps in the auto calibration use case, lets the user select his point of interest, and assign time series to this area. In figure 7 and 8 we show how this is done in the SMS client. Figure 7 shows the selected catchment area. In the catchment area, a pre-loaded gauging station with data is shown and the user can select this for calibration in later steps. Figure 8 shows the same area, with all upstream basins visible for the user.

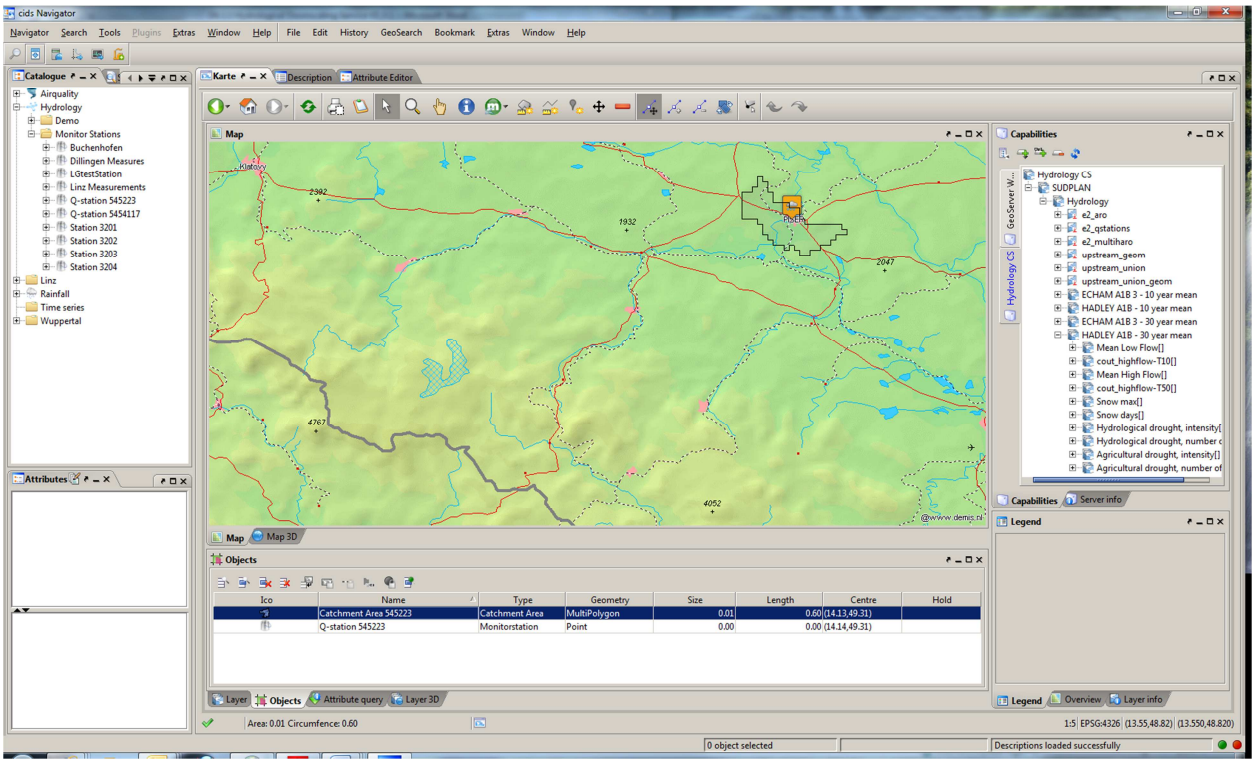


Figure 7: One sub-basing selected with available calibration data.

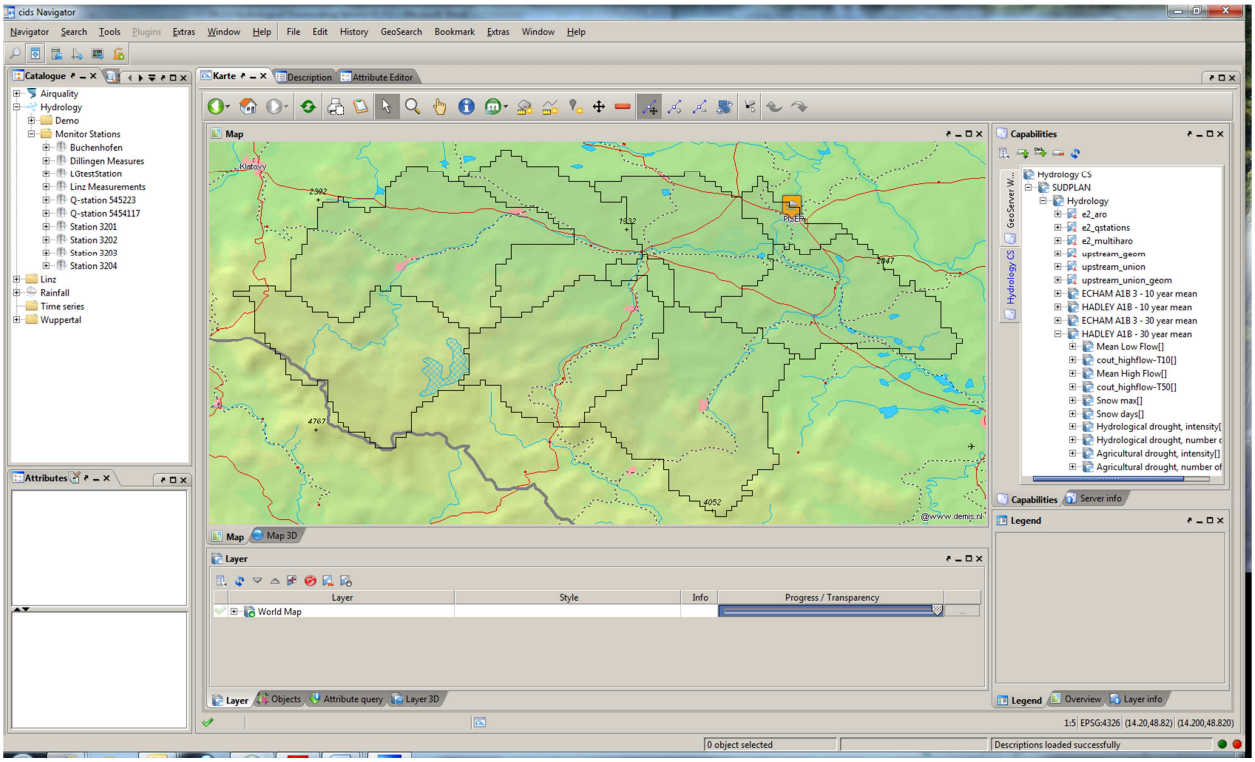


Figure 8: The upstream areas of the selected area highlighted.

The user can now create a local model by right clicking on the selected area. This result in a dialogue where the name of the new local model can be specified (figure 9 left). The created local model shows up in the left pane of the SUDPLAN client (figure 9 right).

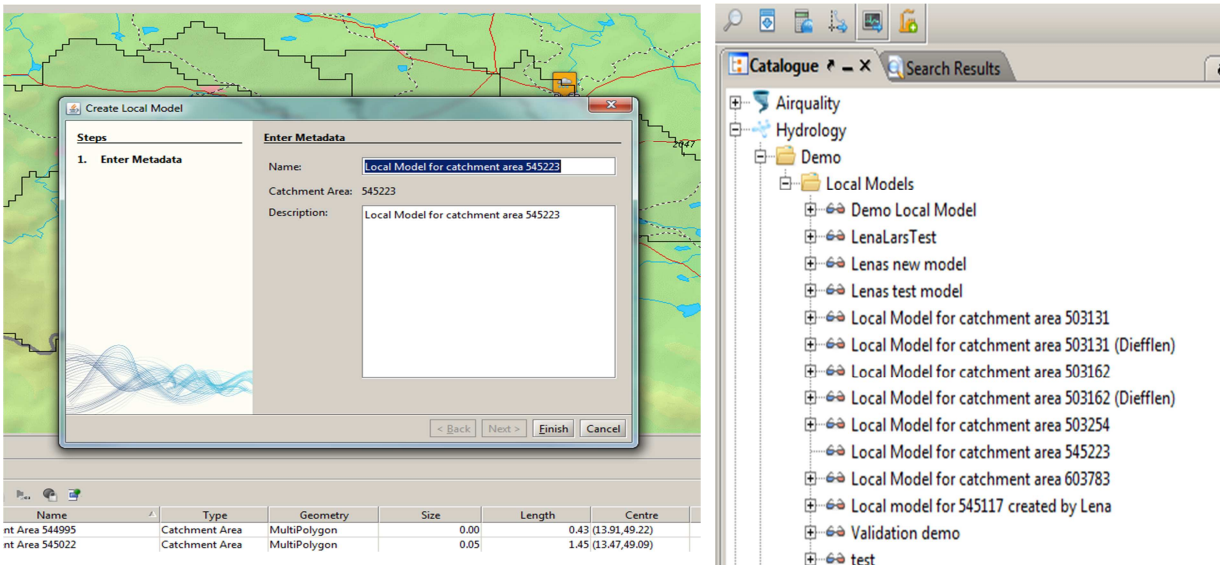


Figure 8: Creation of a local model in the SMS client.

After the creation of a local model, the user can assign time series for calibration and then start the calibration. The SUDPLAN system provides testing and definition of inputs to the calibration

routine so that initiation of this routine is automated to simplify for the end user. For the prototype we use the Brent method for Calibration. However, for future developments a selection of methods with different properties can be offered. The resulting calibrated model shows up in the left pane of the SMS client (figure 9). The user can select this result and compare the measured data with the calibrated time series (figure 10).

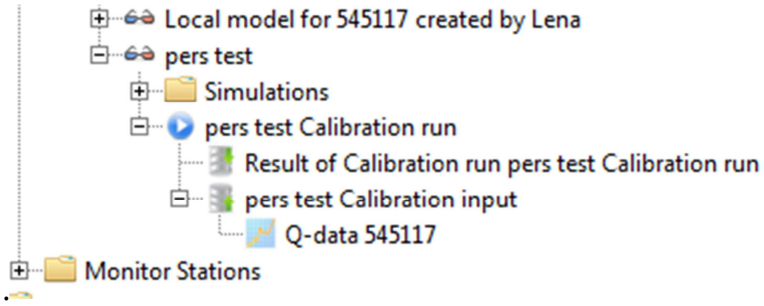


Figure 9: Resulting calibrated model in the left pane of the SMS client.

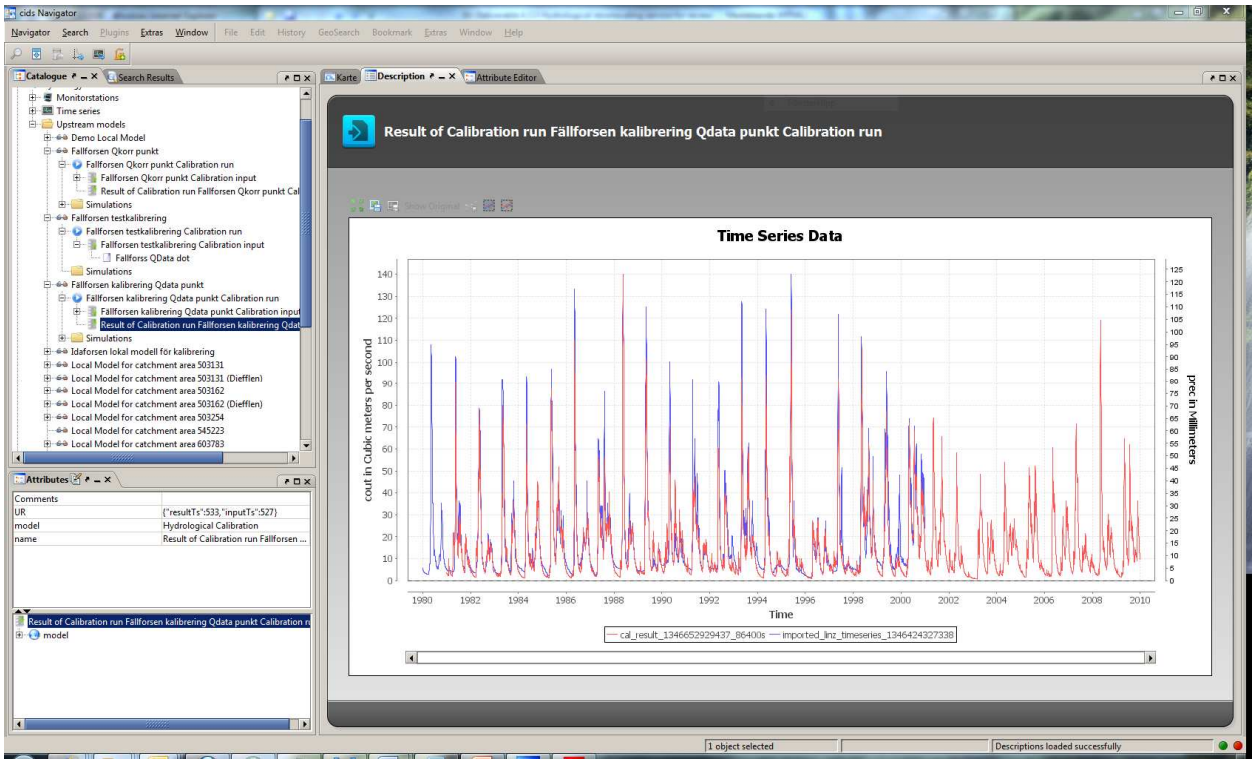


Figure 10: Measured and calibrated data for the point of interest.

Running a simulation is performed in a similar way. The simulation is started by selecting a calibrated model, the desired climate scenario, the time range and possibly meta-data about the calibration (Figure 11) and after this the simulation will be started. A progress bar indicates the running process (Figure 12).

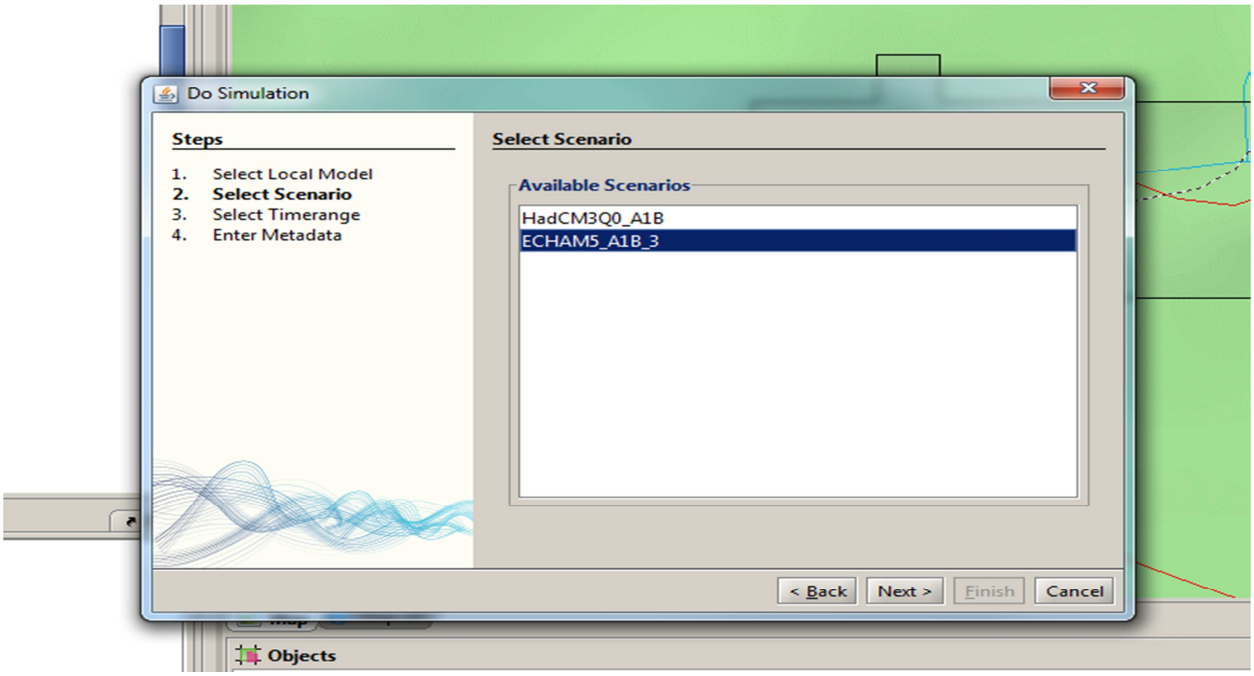


Figure 11: Selecting scenario for a simulation.

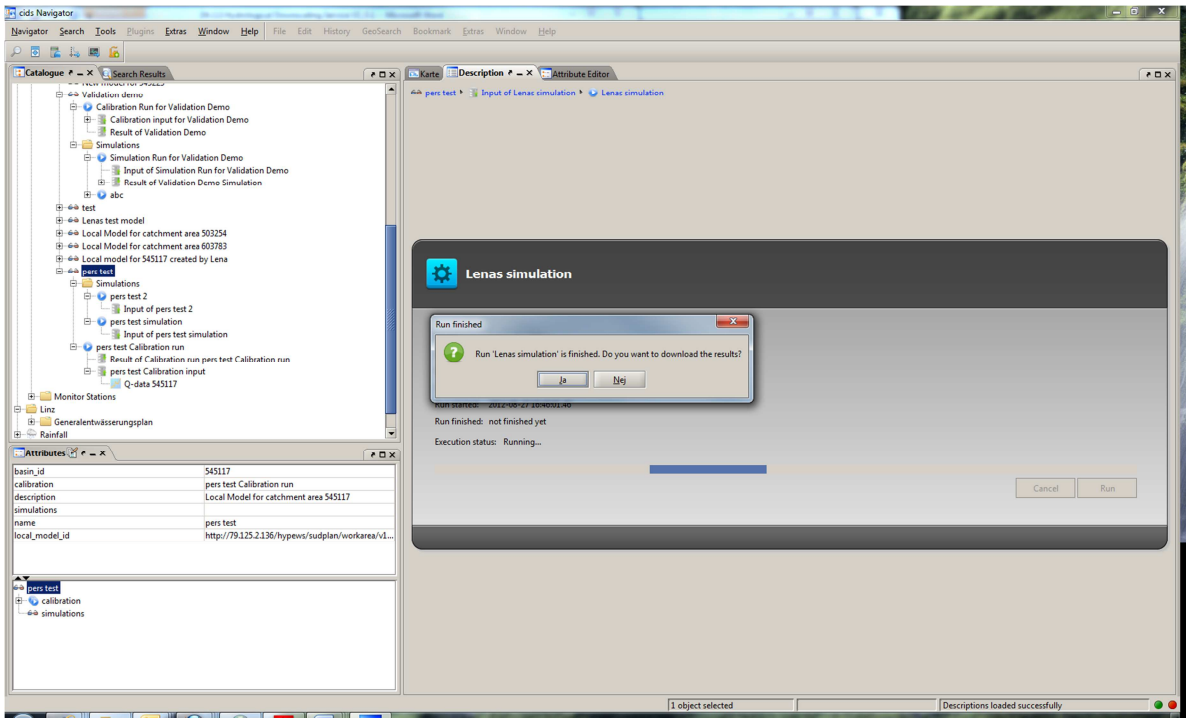


Figure 12: Progress indicator during simulation.

6 Conclusion

This report describes the hydrological downscaling services and focuses on the back-back end implementation as well on the necessary extensions in the context of E-Hype. The implementation provides functionality for creating a sub-model of the Pan-European hydrological model, E-HYPE, for the river or stream running through the specific city of interest and the catchment upstream of the city. In addition the automatic calibration package for optimizing model parameterization based on measured discharge data is supported. The new local hydrological model can be used to run chosen climate scenarios. We also provide an API that encapsulates and exposes the back-back end functionality.

Moreover, the API is integrated into the Hydrological Common Services with the SUDPLAN platform, the SMS and its user interface. We will further evaluate the application with end users regarding the expected functionality, ease of use and more detailed specifications of calibration routines and model output.

References

1. D8.1.2 Czech Pilot Definition Plan V2.doc
2. Lindström, G., Pers, C. Rosberg, J., Strömqvist, J., Arheimer, B.: Development and test of the HYPE (Hydrological Predictions for the Environment) model - A water quality model for different spatial scales", Hydrol. Research 41 (3-4): 295-319 (2010)
3. Arheimer, B., Dahné, J., Lindström, G. Marklund, L., Strömqvist, J.: Multi-variable evaluation of an integrated model system covering Sweden (S-HYPE). IAHS Publ. 345 (2011)
4. Donnelly, C., Dahné, J., Lindström, G., Rosberg, J., Strömqvist, J., Pers, C., Yang, W., Arheimer, B.: An evaluation of multi-basin hydrological modelling for predictions in ungauged basins. IAHS publ. 333:112-120 (2009)
5. Strömqvist, J., Dahné, J., Donnelly, C., Lindström, G., Rosberg, J., Pers, C., Yang, W., Arheimer, B.: Using recently developed global data sets for hydrological predictions. IAHS publ. 333:121-127 (2009)
6. Swedish water archive (SVAR) http://svarwebb.smhi.se/
7. USGS (US Geological Survey): Hydro1k Elevation Derivative Database. http://edc.usgs.gov/products/elevation/gtopo30/hydro/index.html
8. Lehner, B., Verdin, K., Jarvis, A.: New global hydrography derived from spaceborne elevation data. Eos, Trans., AGU 89(10), 93–94 (2008)
9. Johansson, B. 2002. Estimation of areal precipitation for hydrological modelling in Sweden. PhD Thesis, Earth Sciences Centre, Dept Phys. Geog., Göteborg University. Sweden.
10. EIONET,2003.Central Data Repository. http://cdr.eionet.eu.int. [Date accessed: 14 Feb 2009]
11. GLC2000; http://bioval.jrc.ec.europa.eu/products/glc2000/data_access.php
12. Euroland Soil Sealing 2009: HR Layer Imperviousness 2009 - Pan european - P-EL-03. Produced by: GeoVille, Planetek, Infoterra
13. GLWD (Global Lakes and Wetlands database) Lehner, B. and P. Döll (2004): Development and validation of a global database of lakes, reservoirs and wetlands. Journal of Hydrology 296/1-4: 1-22
14. EEA (register called ERMObst) European Environmental Agency Waterbase (EEA, 2009) EEA (2009) European Environmental Agency's Waterbase. http://www.eea.europa.eu/data-and-maps/data/waterbase-lakes-6. Accessed 15 January 2010.
15. Kirillin, G., et al. 2011, FLake-Global: Online lake model with worldwide coverage, *Env. Modell. Soft.*, in press, doi:10.1016/j.envsoft.2010.12.004
16. (http://edition.pagesuite-professional.co.uk/launch.aspx?referral=other&pnum=225&refresh=cZ14S05wdF09&EID=e09e710c-eda7-4a00-8395-19b66f2dea5d&skip=&p=225
17. (http://wldb.ilec.or.jp/LakeDB2/) (ILEC= International Lake Environment Committee Foundation)
18. LEGOS (Laboratoire d'Etudes en Géophysique et Océanographie Spatiales) http://www.legos.obs-mip.fr/en/

19. European Irrigation Map: Wriedt, G., Van der Velde, M., Aloe, A., Bouraoui, F., 2009. A European irrigation map for spatially distributed agricultural modelling. Agricultural Water Management 96, 771-789, doi:10.1016/j.agwat.2008.10.012
20. IWMI Global Irrigated Area Map (GIAM) http://www.iwmigiam.org/info/main/index.asp
21. Soil database, Quaternay deposits 1:1M, SGU, Swedish Geological Survey. http://www.sgu.se/sgu/eng/produkter-tjanster/kartvisare-jord_e.html
22. JRC: http://eusoils.jrc.ec.europa.eu/ESDB_Archive/ESDB/index.htm European Soils Database
23. http://www.fao.org/nr/land/soils/digital-soil-map-of-the-world/en/
24. Global Runoff Data Centre: River Discharge Data, Koblenz, Federal Institute of Hydrology (BfG) (2009b)
25. BHDC: Baltex Hydrological Database. http://www.smhi.se/sgn0102/bhdc/
26. Global Runoff Data Centre: European Water Archiver (EWA) / Global Runoff Data Centre. Koblenz, Federal Institute of Hydrology (BfG)
27. COSMO: http://www.cosmo-model.org/srnwp/content/default.htm
28. FLUXDATA: source: www.fluxdata.org
29. D4.1.1 Common Services concerted approach V1.pdf
30. Yang, W., Andréasson, J., Graham, L.P., Olsson, J., Rosberg, J., and F. Wetterhall (2010) Improved use of RCM simulations in hydrological climate change impact studies, Hydrol. Res., 41, 211-229.

7 Glossary

2D	Two-dimensional, typically a field that varies in east-west and north-south direction. The field may also vary in time –this is typical for e.g. air pollution and population density. The former varies from one hour to another while the latter maybe varies from one year to another.
3D	Three-dimensional, typically a field that varies in east-west and north-south direction as well as vertically. The field may also vary in time.
4D	Four-dimensional. Most often 3D field that explicitly also varies in time. It could also be when a certain 3D parameter (e.g. a particular air pollutant) also varies according to another 3D parameter (e.g. temperature). It will then be possible to study the variation of the first 3D parameter as a function of space (x,y,z) and the second parameter.
Airviro	Air quality management system consisting of databases, dispersion models and utilities to facilitate data collection, emission inventories etc, see http://www.Airviro.smhi.se/
Climate scenario	*Climate scenarios* means the resulting climate evolution over time, as simulated by global (GCMs) and regional (RCMs) climate models. Climate scenarios are products of certain emission scenarios that reflect different economic growth and emission mitigation agreements.
Common Services	*Common Services* is the climate downscaling services for rainfall, river flooding and air quality, developed in the SUDPLAN project and accessed through the SUDPLAN platform (Scenario Management System)
Common Services server	*Common Services* models will be executed at a SMHI server, accessible through OGC communication.
Emission scenario	These are of three types, of which the first one is behind the climate scenarios used in all SUDPLAN Common Services. The two remaining emission scenario types are only relevant for air quality downscaling.

IPCC emission scenarios	*IPCC emission scenarios* are estimates of future global greenhouse gas concentrations based on assumptions about global development (economic growth, technical development, mitigation agreements, etc). During the first two years of the SUDPLAN projects, the climates scenarios based on SRES (Special Report on Emission Scenarios) A1B scenario from the 4th assessment have been used. The SRES emission scenarios do not include emissions of the pollutants of interest for air quality. If available the climate scenarios based on the 5th assessment RCP (Representative Concentration Pathways) emissions scenarios will also be used within the SUDPLAN project. They include emissions of air pollutants.
European tracer gas emissions (air pollutants)	*European tracer gas emissions (air pollutants)* thus may or may not be included in IPCC emission scenarios. For creating Pan-European air quality fields under climate scenarios driven by the SRES A1B emission scenario, SUDPLAN uses tracer gas emissions from the more recent RCP emission scenarios. This inconsistency will be solved when climate scenarios based on RCP emission scenarios are available.
Local emission scenarios	*Local emission scenarios* (to the atmosphere) are those of a particular European city. These will to a large extent influence future air quality in the city, but have little influence on global climate, nor do they influence air pollution concentrations in incoming long-range transported air. SUDPLAN will typically need gridded emissions with 1x1 km or finer spatial resolution as input to its urban air quality downscaling model.
Hind cast	A simulation of a historical period. Often done to compare model simulations with data which is available during that period.
Hot spot	Point (or small area) which is very different from its surroundings. In the present context, most often high concentrations of air pollutants, or extreme meteorological conditions.

Information product	Raw data, such as the results of mathematical modelling, and the analysis thereof, will often need to be packaged in such a way as to be accessible to the various stakeholders of an analysis. The medium can be one of a wide variety, such as print, photo, video, slides, or web pages. The term *information product* refers to such an entity.
Mockup	A model of a design used for demonstrating the functionality of a system.
Model	A *model* is a simplified representation of a system, usually intended to facilitate analysis of the system through manipulation of the model. In the SUDPLAN context the term can be used to refer to mathematical models of processes or spatial models of geographical entities.
PM_{10}	'PM10' shall mean particulate matter which passes through a size-selective inlet as defined in the reference method for the sampling and measurement of PM10, EN 12341, with a 50 % efficiency cut-off at 10 μm aerodynamic diameter;
$PM_{2.5}$	'PM2,5' shall mean particulate matter which passes through a size-selective inlet as defined in the reference method for the sampling and measurement of PM2,5, EN 14907, with a 50 % efficiency cut-off at 2,5 μm aerodynamic diameter;
Profile	Within SUDPLAN a *profile* is a set of configuration parameters which are associated with an individual or group, and which are remembered in order to facilitate repeated use of the system.
Regional downscaling	A climate scenario may be downscaled to a higher spatial resolution, typically 25-50 km, by a Regional Climate Model (RCM). The regional downscaling in SUDPLAN will be performed by SMHI's RCM (RCA, see below) and will generate climate scenarios at 44 or 22 km resolution.
Report	A *report* is a particular type of information product which is usually static and might integrate still images, static data representations, mathematical expressions, and narrative to communicate an analytical result to others.

Scenario	A *scenario* is a set of parameters, variables and other conditions which represent a hypothetical situation, and which can be analysed through the use of models in order to produce hypothetical outcomes. In SUDPLAN a scenario is an individual model simulation outcome to be used in urban planning. The model simulation may or may not include Common Services downscaling (with specific input) and may or may not include a local model simulation (with specific input and parameters).
Scenario Management System	*Scenario Management System* is synonymous with SUDPLAN platform
Scenario Management System Framework	The *Scenario Management System Framework* is the main Building Block of the Scenario Management System. It provides the Scenario Management System core functionalities and integration support for the other Building Blocks.
Scenario Management System Building Block	Scenario Management System Framework is composed of three distinct *Building Blocks*: The Scenario Management System Framework, the Model as a Service Building Block and the Advanced Visualisation Building Block.
Street canyon	Volume between high buildings in cities. Due to poor circulation (and high emissions) prone to poor air quality. Street canyons have unexpected circulation patterns, thus dedicated models are needed to study air pollution here.
SUDPLAN application	A *SUDPLAN application* is a decision support system crafted by using the SUDPLAN platform and integrating models, data, sensors, and other services to meet the requirements of the particular application.
SUDPLAN platform	The *SUDPLAN platform* is an ensemble of software components which support the development of SUDPLAN applications.
SUDPLAN system	*SUDPLAN system* is synonymous with SUDPLAN application

Urban downscaling	This refers to further downscaling of the regional climate scenarios for Europe to the urban scale within SUDPLAN. This will be possible for a) *rainfall/precipitation* where the temporal resolution will be 30 minutes or less. The spatial resolution will be that of a precipitation gauge, i.e. representative for a point rather than a certain area. b) *hydrological variables (river runoff, soil moisture etc*) where the temporal resolution is daily and the spatial resolution linked to catchment areas which presently count approximately 35000 and with average size 240 km^2. *c) air quality (PM, NO2/NOx, SO2, O3, CO).* The temporal resolution will be hourly for gridded output fields and the spatial resolution typically 1x1 kilometres.
User	The term *user* refers to people who have a more or less direct involvement with a system. Primary users are directly and frequently involved, while secondary users may interact with the system only occasionally or through an intermediary. Tertiary users may not interact with the system but have a direct interest in the performance of the system.
Web-based	Computer applications are said to be *web-based* if they rely on or take advantage of data and/or services which are accessible via the World Wide Web using the Internet.

8 Abbreviations and Acronyms

Acronym	Description
A1B	Emission scenario used for global climate modelling in IPCCs Fourth Assessment Report (AR4)
Airviro	Air quality management system to facilitate data collection, emission inventories etc, see http://www.airviro.smhi.se/
CS	Common Services
AVDB	Airviro Time Series database (used for storage in Common Services)
AR4, AR5	Fourth and Fifth Assessment Report of IPCC
AQ	Air Quality
C API	Application Programming Interface written in C
CMIP5	Coupled Model Intercomparison Project, phase 5 (coordinated model exercise in support to AR5)
CS	Common Services (SUDPLAN functionality)
CTM	Chemistry Transport Model
CTREE	FairCom CTREE database (Index database, core of AVDB)
DBS	Distribution-Based Scaling, a method to bias-correct (i.e. remove systematic errors in) the temperature and precipitation of the RCM output
DoW	SUDPLAN Description of Work
DSS	Decision Support Systems
ECHAM5	GCM developed at Max Planck Institute for Meteorology, DE
ECMWF	The European Centre for Medium-Range Weather Forecasts (also co-ordinating FP7-SPACE project MACC)
EDB	Airviro Emission database
EEA	European Economic Association
E-HYPE	HYdrological Predictions for the Environment (European set-up), hydrological rainfall-runoff model developed and used by SMHI
EM&S	Environmental Modelling and Software
ESA	European Space Agency
ESDI	European Spatial Data Infrastructure
EU	European Union
GCM	Global Climate Model or, equivalently, General Circulation Model. Physically based computer model that simulates the global climate on a 200-300 km resolution. Can be used both to reproduce historical climate and estimate future climate, e.g. in response to changes in greenhouse gas concentrations.
GHG	GreenHouse Gases
GTE	Georeferenced Time-series Editor
GIS	Geographic Information System
HadCM3	GCM developed at Met Office Hadley Centre, UK

HIRLAM	HIgh Resolution Limited Area Model, numerical weather prediction model developed and used operationally by SMHI
ICT	Information and Communication Technologies
ID	Identifier
IDF-curve	Intensity Duration Frequency-curve, a curve (or a table of values) showing the rainfall intensity associated with a certain duration (i.e. time period) and frequency (i.e. probability, generally expressed as a return period). Calculated from short-term rainfall observations and widely used in design of urban drainage systems.
iEMSs	International Environmental Modelling & Software Society
IFIP	International Federation for Information Processing
IPCC	The Intergovernmental Panel on Climate Change, the leading body for the assessment of climate change
IPR	Intellectual Property Rights
ISAM	Indexed Sequential Access Method, a method for indexing data for fast retrieval
ISO	International Standardization Organisation
ISESS	International Symposium on Environmental Software Systems
IST	Information Society Technology
MATCH	Multiple-scale Atmospheric Transport and Chemistry modelling system, a CTM developed and used by SMHI.
MODSIM	International Congress on Modelling and Simulation
OASIS	Organization for the Advancement of Structured Information Standards Open Advanced System for Disaster and Emergency Management (FP6 project)
OGC	Open Geospatial Consortium
O&M	Observation and Measurements
ORCHESTRA	Open Architecture and Spatial Data Infrastructure in Europe (FP6 IST-511678)
OSGeo	Open Source Geospatial Foundation
OSIRIS	Open architecture for Smart and Interoperable networks in Risk management based on In-situ Sensors (FP6 IST-33799)
PMC	Project Management Committee
RC	Rossby Centre, climate research unit at SMHI
RCA	Rossby Centre Atmospheric model, RCM developed by SMHI and used in SUDPLAN
RCM	Regional Climate Model, commonly used to increase the spatial resolution of climate scenarios to 25-50 km in a specific region.
RCP4.5	Radiative Concentration Pathways: A set of four emission scenarios to be used for the AR5 simulations. The scenarios are named according to their radiative forcing at 2100, e.g. 4.5 W/m^2.
RNB	Airviro Field database
SANY	Sensors Anywhere (FP6 IST-033654)
SDI	Spatial Data Infrastructure

SISE	Single Information Space in Europe for the Environment
SISE	Single Information Space in Europe for the Environment
SMHI	Swedish Meteorological and Hydrological Institute
SMS	Scenario Management System
SOA	Service Oriented Architecture
SOS	Sensor Observation Service
SPS	Sensor Planning Service
SWE	Sensor Web Enablement
SUDPLAN	Sustainable Urban Development PLANner for climate change adaptation
SWE	Sensor Web Enablement
Tbd	To be determined
UWEDAT	AIT environmental data management and monitoring system
WCC	World Computer Congress
WCS	Web Coverage Service
WFS	Web Feature Service
WP	Work Package
WPS	Web Processing Service
WMS	Web Map Service

Appendix A: Auto calibration: methods and functions

In this appendix we give a more detailed description of the theory of the methods used in the auto calibration package. All functionalities described in this appendix are discussed and illustrated in terms of seeking a minimum, and illustrated over the 2-parameter test function :

F1: $$f(x,y) = 3\,(1-x)^2 \cdot e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)\cdot e^{-x^2-y^2} - \frac{1}{3}\,e^{-(x+1)^2-y^2}$$

The global minimum of this function lies at $x = 0.228879$, $y = -1.626176$, which is therefore the target parameter set for all optimization routines described and illustrated here.

Throughout the discussion of optimization methods, we use mathematical In what follows, parameter sets are considered as vectors, and noted with vector notation, *i.e.* $\vec{p}$. In this spirit, a particular component of $\vec{p}$ therefore refers to an individual parameter value. The vector size, denoted N, corresponds to the amount of parameters involved in the optimization problem, and is referred to as the *dimensions* of the problem. Finally, curly brackets refer to ensembles of several parameter sets, *i.e.* $\{\vec{p}\}$.

1. Sampling methods

A simple way to gather knowledge of the objective function is to sample evaluations of this function under variation of its arguments values. Sampling can be performed by either an organized (scan mode) or a random (Monte Carlo mode) variation of the parameter set. Provided that the amount of sampling points is sufficient to catch the functions complexity, sampling methods can give a rough estimation of the optimum. Further, they are found useful to provide a starting point for directional optimization algorithms.

From the computational point of view, sampling methods offer the advantage that the amount of function evaluations is defined by the user, which in principle allows for estimating rather accurately the computational time required to perform the task.

1.1 2-parameter organized scan

This is a functionality incorporated in the optimization package that allows for systematically varying two parameter values, with constant interval between values. Though limited to two parameter values, this method not only provides a quick way to roughly determine the objective function value optimum within a given parameter space, but also returns a matrix of objective function evaluations sampled over the parameter space grid, that can be used for visualization.

Figure 5 illustrates the scan functionality for the test objective function (F1), for a 100x100 equidistant sampling over the argument intervals $x \in [-3.0, 3.0]$, $y \in [-2.5, 2.5]$. A simple analysis of the function evaluations matrix provides a rough estimate of the global minimum of the objective function over that interval.

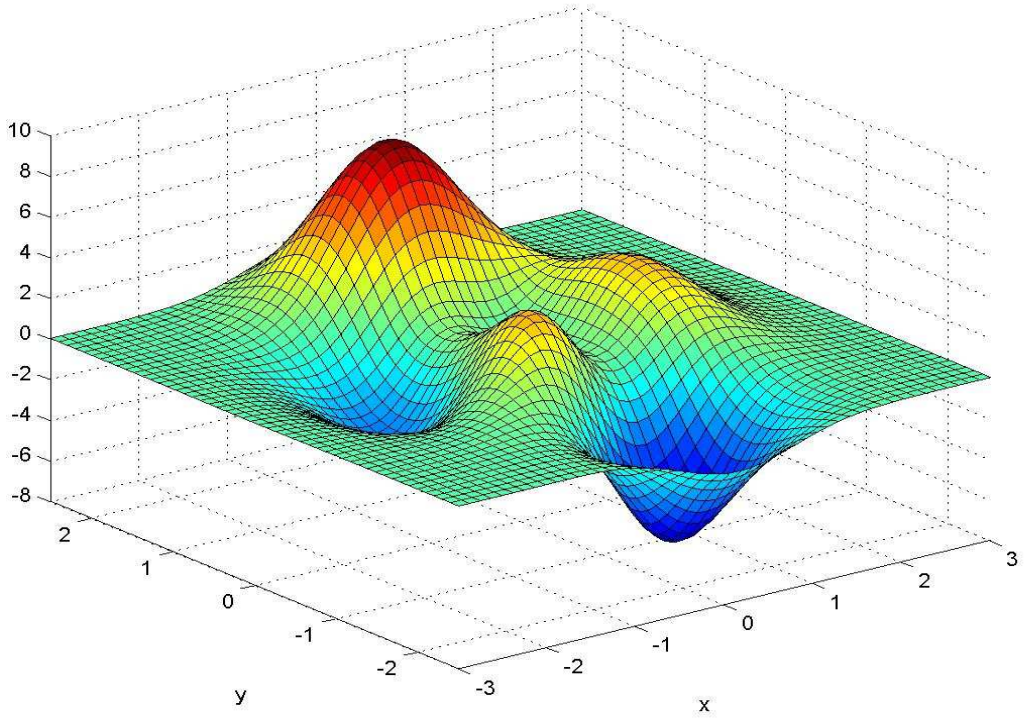


Figure A1: visualization of the test objective function by 2-parameter scan

1.2 Progressive Monte Carlo method

The Monte Carlo method relies on random sampling. Parameter sets are generated without any form of organization, and all parameter values are varied at once. With respect to the organized 2-parameter scan functionality described above, this method offers the advantage of being applicable to parameter spaces of any dimensions. The Monte Carlo method used here incorporates a refinement, consisting in a progressive, stage-wise reduction of the parameter space around promising parameter sets. The concept is based on the radius of the original parameter space, denoted $\vec{r}_0$ here, that is defined as half the distance between the original parameter space boundaries. The algorithm requires the user to specify a few numerical parameters:

the parameter space shrinking coefficient from stage to stage, ξ: $0 < \xi < 1$,
the amount of stages, N_s
the amount of centers (parameter sets corresponding to the most optimal objective function evaluations at previous stage), N_c
the amount of objective function evaluations per center, N_e

The initial stage of the algorithm (referred to as stage 0) corresponds to a usual, simple Monte Carlo sampling phase. $N_e \cdot N_c$ vectors $\vec{\chi}$, whose components are all limited to the interval [-1,1], are randomly generated. Those vectors are subsequently component-wise multiplied with the

initial parameter space radius, $\vec{r}_0$, and an ensemble of parameter sets is then obtained by adding the products to the center of the parameter space, $\vec{p}_0$:

$$\{\vec{p}\} = \vec{p}_0 + \vec{\chi} \cdot \vec{r}_0$$

In this formula the multiplication sign refers to component-wise multiplication. The objective function is evaluated for each parameter set of the ensemble. The results are sorted in accordance to the optimization criteria, and the N_c best parameter sets are retained as parameter space centers for the next stage.

Denoting the stage index with k: $1 \leqslant k \leqslant N_s - 1$, the following algorithm is applied at each of the subsequent stages:

- determine the stage parameter space radius: $\vec{r}_k = \xi^k \cdot \vec{r}_0$
- using each of the N_c best parameter sets retained from the previous stage, $\vec{p}_k$, as a temporary parameter space center, successively generate N_c ensembles of N_e parameter sets each, using the equation
- $\{\vec{p}\} = \vec{p}_k + \vec{\chi} \cdot \vec{r}_k \qquad \forall\ \vec{p}_k$
- check that all parameter sets fall within the original parameter space boundaries, and regenerate those who do not, until they comply with the original boundaries
- evaluate the objective function for all $N_e \cdot N_c$ parameter sets
- sort the results in accordance with the optimization criteria, and retain the N_c best parameter sets as parameter space centers for the next stage

Each stage of the method requires $N_e \cdot N_c$ function calls, and total computational work of the method corresponds to $N_s \cdot N_e \cdot N_c$ function calls.

To illustrate the process we applied the progressive Monte Carlo method to the test objective function (F1). Figure 6 shows the progression of the algorithm over the first 6 stages represented from above by means of elevation lines. In the case at hand, we chose to run 15 evaluations per center ($N_e = 15$), and retain the 6 best parameter sets for stage progression ($N_c = 6$). Further, the radius shrinking factor was taken relatively small, $\xi = 0.6$, ensuring a relatively fast algorithm progression.

Figure A2a) shows the initial stage, where the entire parameter space is sampled; the black dots show the 6 best parameter sets out of the $N_e \cdot N_c$=90 function evaluations performed. Those best parameter sets are used as temporary parameter space centers in the next stage (grey dots in figure A2b), where 15 new parameters sets are generated only within each of the 6 local, shrunk parameter spaces, sketched in grey. The parameter sets corresponding to the 6 minimum sampling values are marked in black. This iterative process is then repeated until the prescribed amount of stages, N_s, is reached.

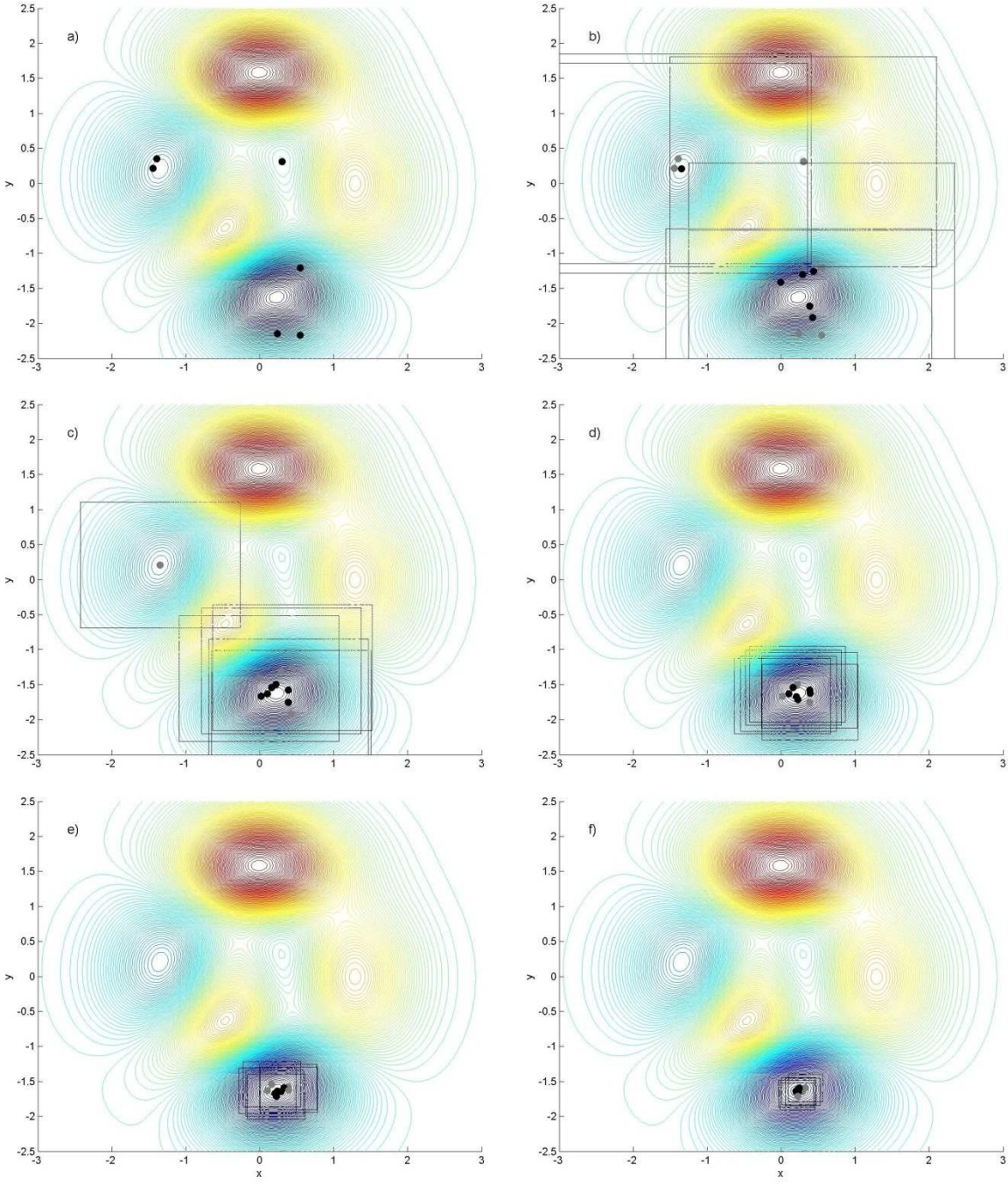


Figure A2: progression of the Monte Carlo method over the 6 first stages

2. Directional methods

The idea behind directional methods is to progress from a given set of model parameters toward a new set, in a way that corresponds to an improvement in the sense of optimization of the objective function. This is achieved by determining a direction of improvement and the optimal step length in that direction for each iteration. Denoting parameter sets as vectors, progression at iteration *k* is given by the equation

$$\vec{p}_{k+1} = \vec{p}_k + \lambda_k \, \vec{d}_k$$

where $\vec{p}_k$ is the starting parameter set at iteration *k*, $\vec{d}_k$ is the step direction and λ_k is the step length. The new parameter set, $\vec{p}_{k+1}$, corresponds to an improvement of the objective function and serves as starting point for the next iteration. Therefore, at each iteration the task of the optimization algorithm is essentially to first determine the step direction, $\vec{d}_k$, and then the optimal step length, λ_k.

Determination of the direction is achieved by means of methods that fall into two categories: the Brent method, described in section 2.1, where the step direction is successively permuted through all dimensions of the parameter space in a prescribed, organized manner; and the quasi-Newton family of methods, described in section 2.2, where the step direction is a function of the objective function gradient evaluated at $\vec{p}_k$. The step length on the other hand is obtained by a unique line-search method.

Before discussing the technicalities behind those different methods, it is necessary to point out that directional methods require an entry point, *i.e.* a starting parameter set, $\vec{p}_0$, from which to progress. This is usually provided by means of a guess, or by means of a parameter set obtained by one of the sampling type of routines described in section 3.2.1. Further, it must be emphasized that the optimization performances depend on the starting set, $\vec{p}_0$. Application of the optimization algorithms to various starting sets will obviously lead to different optimization times, and can even possibly lead to different (local) optima. It is therefore essential to provide the algorithm with an adequate starting set, *i.e.* the best set obtained by Monte Carlo sampling, or a qualified guess based on a priori knowledge of the objective function.

2.1 Brent method

In the Brent method, determining the step direction is trivial since the method permutes through all dimensions of the parameter space, reducing the problem to that of successive line searches. Provided a starting parameter set, $\vec{p}_0$, the algorithm first proceeds to a line search along the dimension of the first component, from boundary to boundary , but keeping the value of all other parameter set components constant in the process. The line search is performed so as to find a first component value that improves the objective function in the sense of optimization. Once a better value has been found, the parameter sets first component is updated. The algorithm proceeds similarly to improve the second parameter set component, maintaining all other components constant, including the corrected first one. This iteration continues for all parameter components. Therefore, the method performs a series of successive updates of all parameter set components, one after the other, in a fixed order.

Once all components have been updated once, a line search is performed along the diagonal joining the current, fully updated parameter set, and the starting parameter set. Since the line search is diagonal, all components are subject to an update. The resulting parameter set, $\vec{p}_1$, is then used as starting parameter set for the next iteration. The same procedure is repeated for several iterations, until one of the conditions for stopping the algorithm is triggered; see section 3.3.4 for a description of implemented interrupters.

The Brent method is illustrated in Figure A3, where it is applied to the test function (F1). For the sake of illustrative clarity, the starting parameter set, drawn as a purple dot in illustration a), was chosen relatively far from the optimum, in particular far from the results given by the Monte Carlo method.

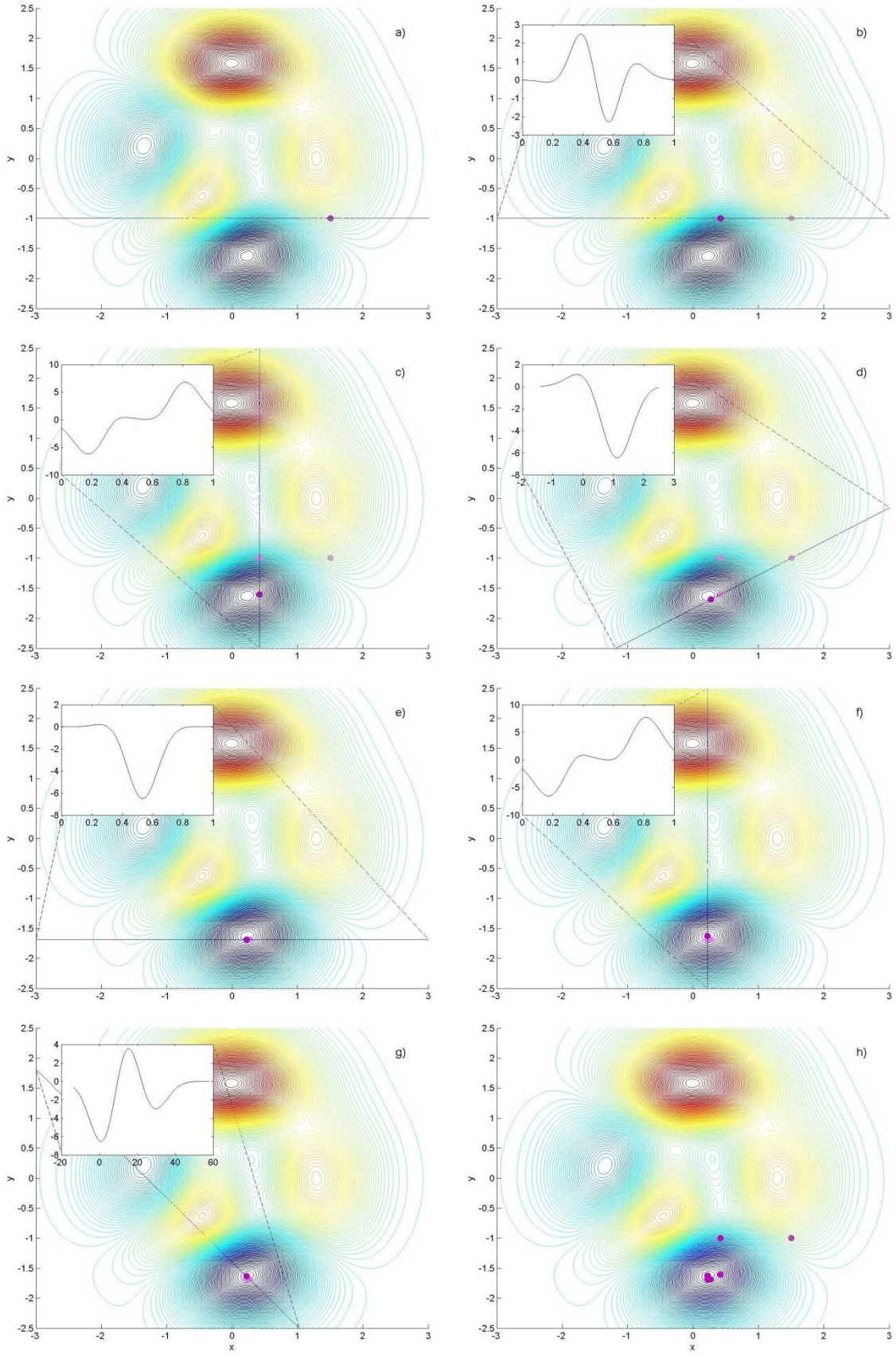


Figure A3: progression of the Brent method with diagonal step

Fig. A3b) shows the line search performed boundary-to-boundary along the x-dimension, and the inset shows the that minimum along that line is at approximately 0.58 times the parameter space width. The resulting updated parameter set, marked with a purple dot, is the starting point of the next line search, performed along the y-dimension (see Fig. A3c). A new, lower minimum is found and the parameter set is updated accordingly.

When all dimensions have then been looped through, a line search along the diagonal joining the current and initial parameter sets is performed, as illustrated in Figure A3d). A parameter set relatively close to the global minimum is found, that serves as starting parameter set for the next iteration, illustrated in figure A3e)-g). The last figure, figure A3h), shows the total path followed by the method until the global minimum was approximated to sufficient accuracy and one of the interrupters for the method was triggered.

2.2 Quasi-Newton methods

Quasi-Newton methods are based on the first order Newton expansion of the gradient of the objective function. Denoting the objective function f, the current parameter set $\vec{p}_k$ and the parameter set corresponding to the optimum $\vec{p}_{opt}$, the Newton series for the gradient gives:

$$\vec{\nabla} f(\vec{p})|_{\vec{p}_{opt}} = \vec{\nabla} f(\vec{p})|_{\vec{p}_k} + (\vec{p}_k - \vec{p}_{opt}) \cdot H|_{\vec{p}_{opt}} + \mathcal{O}(\partial^3 \vec{p}_k)$$

where H is the Hessian matrix (second order partial derivatives of the objective function). Since, by definition, the gradient vanishes at optimum, discarding the higher order terms gives the following formula for the step direction, $\vec{d}_k = \vec{p}_k - \vec{p}_{opt}$:

$$\vec{d}_k = -H^{-1}|_{\vec{p}_{opt}} \cdot \vec{\nabla} f(\vec{p})\big|_{\vec{p}_k}$$

Assuming that higher order derivatives do not contribute to the above expression, $\vec{d}_k$ would point directly at the optimum if both terms in the expression could be exactly determined. This however cannot be achieved in practice, even within the limits of approximations to second order derivatives.

The gradient at the current parameter step, $\vec{\nabla} f(\vec{p})\big|_{\vec{p}_k}$, can be evaluated numerically only, at relatively low computational cost. The method of central differences was found to be most practical, and the option of using a 2-, 4-, 6- or 8-points stencil is implemented in the optimization package; higher order stencils were allowed for in an attempt to average the eventual accumulation of numerical errors in the evaluation of the objective function, *f*. Evaluation of the inverse Hessian, $H^{-1}|_{\vec{p}_{opt}}$, is however a priori impossible since the optimum set, $\vec{p}_{opt}$, is the unknown of the problem. Instead, the inverse Hessian has to be approximated and computed afresh or updated at each iteration. Further, the matrix H has to satisfy a series of requirements, known as the *Wolfe conditions*, for the method to converge toward an optimum. In particular, one of the requirements for $\vec{p}_{opt}$ to correspond to a minimum is that the inverse Hessian must be symmetric and positive definite at each iteration of the algorithm.

A wide range of methods satisfying those conditions have been developed in the framework of optimization. Three of those methods retained our attention and were implemented.

2.2.1 Steepest descent

As its name indicates, the steepest descent method is limited to follow the opposite of the gradient, and the inverse Hessian matrix is always approximated as a unit matrix, without any form of updating:

$$H^{-1}|_{\vec{p}_{opt}} = \mathbb{1}$$

A unit matrix verifies the Wolfe conditions trivially.

2.2.2 DFP (Davidson-Fletcher-Powell formula)

The DFP quasi-Newton method was developed in 1959 by W.C. Davidson. The Davidson-Fletcher-Powell formula provides an update for the Hessian matrix that maintains symmetry and positive definiteness.

Denoting the difference in gradient between the last two steps with

$$\vec{y}_k := \vec{\nabla} f(\vec{p}_k + \vec{d}_k) - \vec{\nabla} f(\vec{p}_k),$$

the DFP update formula for the Hessian update for the next iteration is

$$H_{k+1} = \left(\mathbb{1} - \alpha_k \, \vec{y}_k \, \vec{d}_k^{\,T}\right) H_k \left(\mathbb{1} - \alpha_k \, \vec{d}_k \, \vec{y}_k^{\,T}\right) + \alpha_k \, \vec{y}_k \, \vec{y}_k^{\,T}$$

where $\alpha_k := 1 / \vec{y}_k^{\,T} \, \vec{d}_k$ is a scalar factor common to all terms. For implementing the quasi-Newton algorithms, it is rather the inverse Hessian that is useful. In this case update of the inverse is given by:

$$H_{k+1}^{-1} = H_k^{-1} - \frac{H_k^{-1} \, \vec{y}_k \, \vec{y}_k^{\,T} \, H_k^{-1}}{\vec{y}_k^{\,T} \, H_k^{-1} \, \vec{y}_k} + \alpha_k \, \vec{d}_k \, \vec{d}_k^{\,T}$$

The products in the expression are performed left-right, and they take either the form of dot products between vectors, or regular matrix products. At the very first iteration (k = 0), the inverse Hessian H_1^{-1} is estimated by taking $H_0^{-1} = \mathbb{1}$, a regular unit matrix of dimension equal to the optimization problem parameters amount. This implies that the first step of the DFP quasi-Newton method is completely identical to a steepest descent step.

The DFP method was the first quasi-Newton method tested for the auto calibration routine. As all quasi-Newton methods, in principle it provides a faster progression than the Brent method, because it optimizes all parameters together, instead of looping through one parameter at a time. However, the curvature estimate by the DFP formula was found to be poor, leading to an accumulation of unnecessary small suggested steps, $\vec{d}_k$. A more robust update formula was therefore required, and attention was turned toward the nowadays widely accepted BFGS method.

2.2.3 BFGS (Broyden-Fletcher-Goldfarb-Shanno formula)

The BFGS formula, was suggested in 1970 to supersede the DFP formula and can be seen as the dual of the latter as it simply inverses the roles of $\vec{d}_k$ and $\vec{y}_k$. The formula for the Hessian update is therefore:

$$H_{k+1} = H_k - \frac{H_k \, \vec{d}_k \, \vec{d}_k^{\,T} \, H_k}{\vec{d}_k^{\,T} \, H_k \, \vec{d}_k} + \alpha_k \, \vec{y}_k \, \vec{y}_k^{\,T}$$

or, equivalently, for the computationally more useful inverse Hessian:

$$H_{k+1}^{-1} = \left(1 - \alpha_k \, \vec{d}_k \, \vec{y}_k^{\,T}\right) H_k^{-1} \left(1 - \alpha_k \, \vec{y}_k \, \vec{d}_k^{\,T}\right) + \alpha_k \, \vec{d}_k \, \vec{d}_k^{\,T}$$

where the scalar α_k is defined as above. As with the DFP method, the inverse Hessian is approximated as a unit matrix at the very first iterations, and the progression of the algorithm starts with a pure steepest descent step.

Figure A4 illustrates the progression of the BFGS method, applied to the test function (F1). The starting point $\vec{p}_0$, *i.e.* the dark green dot in Fig. A4a), is the same as for the illustration of the Brent method. The figure features in light green the auxiliary points for derivative computation by central differences. The first BFGS step, a pure steepest descent step, places the next guess outside the parameter boundaries (blue line, suggested next point not featured). The maximum allowed value of λ is determined to be around 0.4, and the line search determines the optimum to be around $\lambda = 0.21$, giving the starting point for the next iteration shown as a dark green dot in Fig. A4b). The algorithm is repeated to find a third point, Fig. A4c)-d). At iteration 3, Fig. A4e)-f), the suggested next point lies well within the parameter space and λ is allowed to be as large as 1.618034, the golden ratio.

The same method is applied until one of the conditions for algorithm termination is triggered. Fig. A4g) shows the global path taken by the BFGS quasi-Newton method. In Fig. A4h), we compare with the path taken by the DFP algorithm, where DFP iteration steps are shown with red dots.

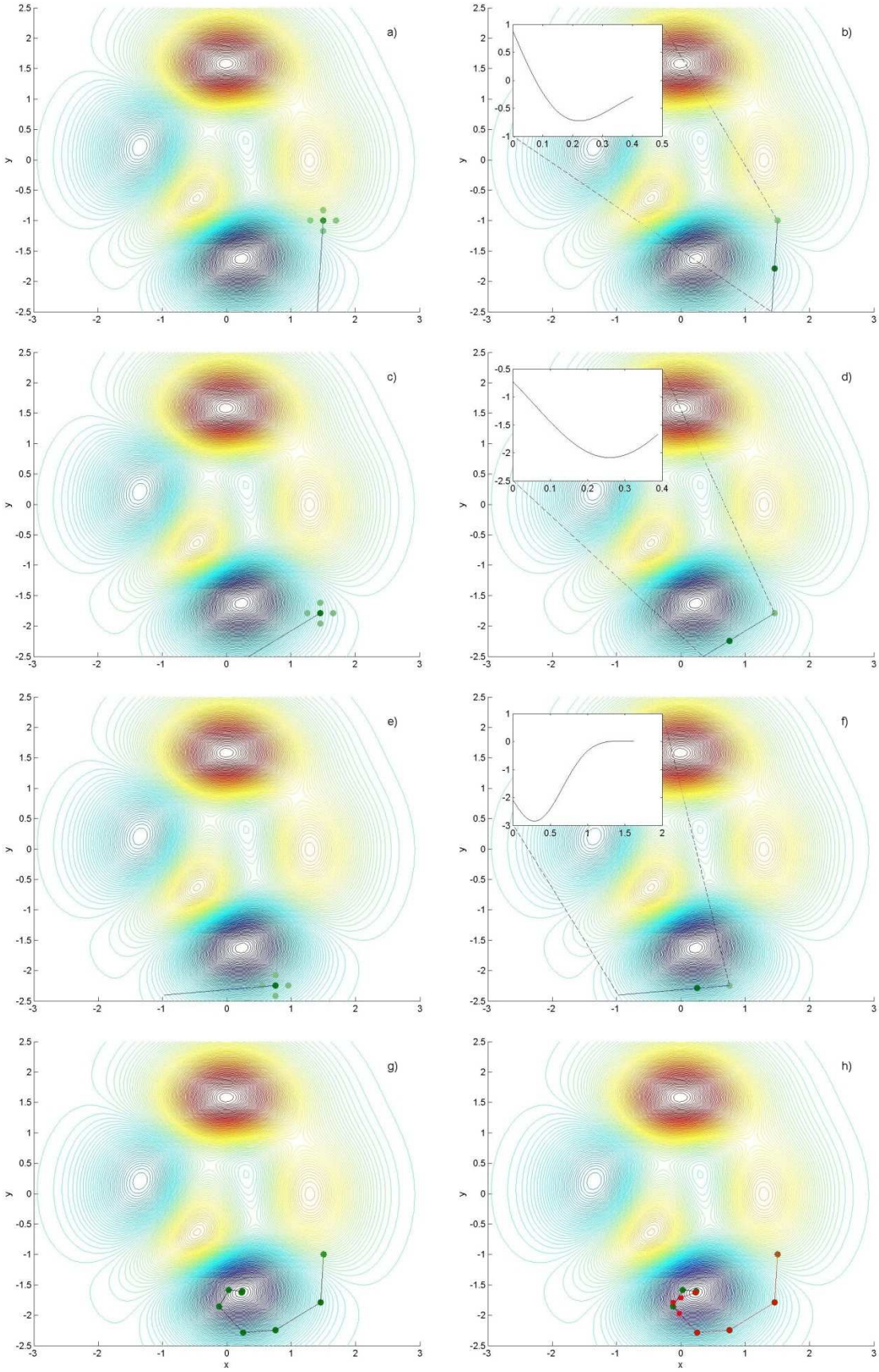


Figure A4: progression of the BFGS quasi-Newton method