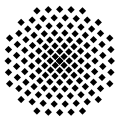
University of Stuttgart - Institute for Modelling Hydraulic and Environmental Systems
Department of Stochastic Simulation and Safety Research for Hydrosystems
Prof. Dr.-Ing. Wolfgang Nowak

Automated Calibration for Numerical Models of Riverflow

Master of Science : Water Resources Engineering and Management WAREM

Betsaida Fernández

Examiners: Prof. Dr.-Ing. Wolfgang Nowak
Dr.-Ing. habil. Sergey Oladyshkin
Supervisor: Dr.-Ing. Rebekka Kopmann
Working period: May 09th, 2016 - Nov 09th, 2016



University of Stuttgart - Institute for Modelling Hydraulic and Environmental Systems
Department of Stochastic Simulation and Safety Research for Hydrosystems
Prof. Dr.-Ing. Wolfgang Nowak

Pfaffenwaldring 5a, 70569 Stuttgart
http://www.iws-ls3.uni-stuttgart.de

Master's Thesis

Automated Calibration for Numerical Models of Riverflow

Betsaida Lo-Amni Fernández Difo
Matriculation Number: 2925222

Examiners: Prof. Dr.-Ing. Wolfgang Nowak
Dr.-Ing. habil. Sergey Oladyshkin
Supervisor: Dr.-Ing. Rebekka Kopmann

Stuttgart, 09 November 2016

Acknowledgements

The present master thesis is the document to support my candidature for the academic degree Master of Science (M.Sc.), in which is detailedly explained my research and findings on the topic *Automated Calibration for Numerical Models of Riverflow*. Mentioned study was developed in Department of Stochastic, Simulation and Safety Research for Hydrosystems (LS^3) at University Stuttgart in cooperation with The Federal Waterways Engineering and Research Institute (BAW).

I would like to thank Prof.Dr.-Ing. habil. Sergey Oladyshkin and Dr.-Ing Rebekka Kopmann who have supervised the academical and professional development of investigation from two different points of views. First, the implementation of mathematical algorithms and second, the understanding of river modeling. Further thank goes to Leopold Stadler for all programming help. Also, I kindly appreciate my family and personal friends; especially to Luz Difo for her enthusiastic support, constant motivation, and constant encouragement, and to Matthias Ansel for the constructive comments that improved the presentation of these findings.

Stuttgart, November 09th, 2016

Betsaida Fernández

Author Declaration

I declare that I have developed and written the enclosed thesis entirely by myself and that I have not used sources or means without declaration in the text. Any thoughts from others or verbatim quotations are clearly marked. The thesis was not utilized in the same or in a similar version to achieve an academic grading or is being published elsewhere. The enclosed electronic copy is identical to the printed versions.

November 09th, 2016

Signature

Abstract

Calibration of the numerical model is fundamental since the beginning of all systems modeling, to approximate the parameters that can mimic the system behavior. Thus, an assessment of different deterministic and stochastic optimizations methods is undertaken to compare their robustness, computational feasibility, and global search capacity. Besides the uncertainty of most suitable methods is analyzed. These optimization methods minimize the objective function that comprises synthetic measurements and evaluated data; synthetic measurements replace observed data set to warranty an obvious unique parameter solution. The input data for the objective function derivate from an hydro-morphological dynamics numerical model that represents an 180-degree bend channel.

As a result, the parameter landscapes show highly ill-posedness in the mathematical problem. The minimization of the objective function by different candidate optimization methods indicates a failure in some of the gradient-based methods as Newton Conjugated and BFGS. Others indicate partial convergence, such as Nelder-mead, Polak und Ribièri, L-BFGS-B, Truncated Newton Conjugated, and Trust-Region Newton Conjugated Gradient. Some others indicate parameter solutions that are out of the physical limits, such as Levenberg-Marquardt and LeastSquareRoot. Furthermore, it exists a significant computational demand for genetic optimization methods, such as Differential Evolution and Basin-Hopping, and for Brute Force method. Deterministic Sequential Least Square Programming and scholastic Bayes Inference theory methods present the optimal optimization results. The application of suitable methods with observed data in the objective function only converges after adding weight to some important measurements points.

keywords: Automated calibration, stochastic optimization methods, deterministic optimization methods.

Contents

List of Figures

List of Tables

1 Introduction

"No man ever steps in the same river twice, for it is not the same river and he is not the same man"

Heraclitus

Automated Calibration of Riverflow Models is still a developing topic; its necessity rises a contemporary investigation to design an accurate, precise and reliable programming routine at low computational cost based on mathematical optimization methods; involving a quality compromise among the typical method features, for instance, the computational feasibility and the minimization capacity.

The present chapter contains a summarized description of the river modeling and the model calibration; it includes the background information with the most relevant scientific articles related to the deterministic, the genetic and the stochastic optimization methods. Furthermore, it introduces the restrictions of the past investigations as challenges ahead, following by a proposal to overcome them as current master's thesis goals. Finally, it closes with a brief explanation of the present study by chapters.

1.1 River Modeling

River systems are employed in the anthropogenic activities since ancient times. Continuously collected knowledge of their individual behavior have developed the *River Engineering* field. The benefices and services provided by rivers through human interventions are justified by the economic and social development of humankind, namely, water supply, irrigation, navigation, hydropower, and channelization. Those activities can receive negative influences, for instances, non-controlled flood events or sediment accumulation.

In general, the primary objective is to offer a sustainable water utilization at least ecological damage, applicable investment and affordable maintenance costs. In this sense, the necessity of river dynamics investigation is conceived to predict natural river system behavior as simply as it is possible.

Mathematical models can represent river dynamics features by computing the governing equations, and constraints functions, and by applying the boundaries conditions with the assistance of informatics tools. The simulation returns the variables as output. Variables represent the characteristics of the river system, such as water depth, bed morphological evolution, water velocities in x and y-direction, and others. Classification of mathematical river models can be according to dimensionality (1D, 2D & 3D) and the level use of governing equations (theoretical, conceptual and empirical). Additionally, in agreement with its linearity in mass conservation, momentum and energy; space and time discretization; and how the variables change, leading to either stochastic model in the case of random variation or deterministic models in case of calculated variation. From that place, it is possible to constitute an approach to the river system with proper simplification and assumptions.

River dynamics incorporate the flow, sediment transport, and morphological processes; those are highly complex and relatively unexplored. They are modernly applied on the computational simulations for large-scale models, although this requires a considerable amount of computer capacity, as explained in Wu (2007). The reliability of those abstracted models depends on the quality of the following factors: the physical processes description by mathematical expressions, the accuracy of space and time discretization, the efficient discretization from differential to simple algebraic equations, and the correct programming routine implementations.

The integrated suite of solvers used in the present study Open-Telemac-Mascaret version 7.1; applied to the field of free-surface flow. It came out originally at the Laboratoire National d'Hydraulique et Environment (LNHE) in France, and it is an open source since 2010. Some of its many simulation modules can be applied to hydro-morphological dynamics.

Telemac2D is the computational hydrodynamic module; it describes the bed and boundary frictions, the turbulence, the waves propagation and the others phenomena. It solves Saint-Venant equations, composed by mass continuity and momentum in both x-y directions, vertically averaged. It uses either FEM or FVM spatial discretization scheme. The

different laws of friction are optional, such as Chézy, Strickler, Manning, and Nikuradse. Turbulence viscosity can be either a constant value or calculated by a turbulence model, such as the Elder, the k-ϵ, or the Smagorinsky models. This module is coupled with the sediment transport and bed evolution module SISYPHE, as is clearly explained in Galland et al. (1991).

SISYPHE is the morphological module that calculates bed evolution with the Exner equation. It calculates the equations for the sediment properties, the total transport rate, and the corrections factors. It can apply different formulas for total sediment transport rate, such as Meyer-Peter and Müller or Engelund-Hansen. Also, it corrects the magnitude and the direction of total sediment transport that can be modified due to slope effect and secondary currents. In general, it also is possible to separate bed and suspended load rates, adding other variables.

1.2 Model Calibration

Calibration is a mathematical inverse problem because it tries to identify the initial and local parameters that describe a system from a given measurement data set of particular variables. Hydrodynamic and morphodynamic parameters are critical in River Engineering because not all of them can be measured directly; therefore, they have to be determined indirectly through calibration by using either observed water depth, flows or sediment evolution, as explained by Anderson et al. (2010).

Traditionally, the calibration of hydrodynamic models has been performed manually. The procedure requires an expert's subjectivity, and the time-consuming trial and error technique. Thus, it leads to the necessity of developing an automatic model calibration routine based on mathematical optimization methods and algorithms, as explained in Dung et al. (2011). Lots of research has been conducted regarding Automatic Calibration models for hydrological catchment; yet only little about hydrodynamic, hydro-morphological and hydrogeological systems.

The calibration process minimizes the *objective function* (OF) by adjusting of the parameter sets; describing the river system behavior and taking into account the physical value ranges of the parameter's values for the flow condition and sediment transport rates. The most commonly used *OF* is the mean-squared-error between simulated and observed data, as exposed in Wasantha Lal (1995).

In theory, infinite multiple combinations of spatially and temporally distributed parameter values might approach to similar solutions; in this case, mesh zonification by values of parameters can reduce under-determination for large and fine grids with parameter information at each point. Nevertheless, in informatics, only a few solution sets stand after the computation and even less can be obtained due to the higher dimensionality of the mathematical problem compared with information data.

The author in Cunge (2003) explains that automatic calibration of hydrodynamic models may have failures; in this work, it is implemented a scientific approach to a modeled domain to some objective functions by varying roughness coefficients. The failures usually are the non-physical values of the roughness coefficients, and the resulting incorrect velocity values. Another issue is the physical variables of the models, namely, water depths, required for calibration, they are measurable only at exact locations over a limited duration, while they are supposed to represent a large space domain over an extended period. The representativity of measurement variables which depend on space and time is unable to function optimally. Following Cunge (2003), who suggests that the quality, the accuracy, and the functionality of the output information of a model depends on the acquisition, the interpretation, and the processing of input data into physically reasonable information.

The author of Dung et al. (2011) clarifies that the calibration algorithms have to evaluate many simulations, from hundreds to thousands depending on the strategy, spatial and temporal distribution of measurement data, and certain parameters. A simulation run of a hydrological model usually takes some minutes; whereas, a simulation run of a hydrodynamic model typically require hours or even days. Consequently, work on automatic calibration of hydrodynamic models is rare, especially in large-scale models. From a computational point of view, an automatic calibration of a large-scale model is nowadays feasible due to the always growing computational power and parallelization techniques.

Different approaches of optimization methods are already implemented for calibration of systems, as following is mentioned a few investigations that cover hydrodynamic, hydrological and groundwater system models calibration through deterministic, genetic and stochastic optimization methods.

Deterministic optimization methods. *A Deterministic Algorithm* computes mathematical functions, always finding the unique solution for a given input parameter. They are considered practical and so far well-studied. Most commonly applied on hydromorphological model calibrations are the Newton methods and its derivations. As for following is presented four different pieces of research.

The research done by Wasantha Lal (1995) uses the *Gauss-Newton* (GN) and other deterministic optimization methods in comparison with the results of the *Singular Value Decomposition*(SYD). Therefore, it investigates the roughness parameters in a one-dimensional unsteady river model; its strategy is grouping the parameters on space which behaves similar, so that the number of unknowns gets reduced, being less than the equations and avoiding underestimated problems, leading to solving a linear inverse problem. It also indicates that the combination of parameters within a group corresponds to a relationship between the river channel geometry and the data measuring location. After five iterations and error at the hundred-thousandth place, it demonstrates the high computational performance and the fastest rate of convergence that GN method owns. As discussed by Willis and Yeh (1987), a solution is possible to be found, only when the optimization problem presents more or an equal number of evaluation points than of parameters. For least-squares, fitting problems, the *Gauss-Newton* algorithm is highly capable for lightly non-linear problems, but it fails for fully non-linear problems, as disclosed in Nowak and Cirpka (2004).

In Liu et al. (2005), the calibration with the Parameter estimator (PEST) of the Geographic Information System (GIS)-based hydrological model applies a robust *Gauss-Marquardt-Levenberg* algorithm, combining the inverse Hessian method and the steep descent method. It brings a fast convergence of the objective function with a meaningful physical solution.

Genetic algorithms (GA). The *Genetic Algorithm* is a heuristic iterative global search method which attempts to find the best parameters combination in a given parameter space. By analogy with genetic biology, it utilizes the Darwinian evolution approach and his individual's survival in a natural environment. It creates a population of candidates solutions, eliminating those that do not fulfill an acceptance test and does not pass a random comparison test. Suitable for minimization of single and multi-objectives functions, the advantages of *GA*s over conventional parameter optimization techniques are appropriate for the ill-posed problems and the highly non-linear parameter domains due to their

adaptiveness. Good *GA*s performance requires the choice of a high crossover probability, a low mutation probability, and a moderate population size, as explained in Cheng et al. (2002).

For multi-objective functions, the design of a global search method is for locating the global optimum and not being trapped in local optima; as mentioned in Madsen (2000). The description of a few of the global searcher methods used for the Multi-objective are in the articles: Multi-objective shuffled complex evolution (SCE) by Duan et al. (1993) and Multi-objective genetic algorithms by Madsen (2000).

The *Shuffled Complex Evolution Algorithm (SCE)* combines probabilistic and deterministic approaches, and it evaluates the objective functions at randomly spaced points in the feasible parameter domain in the direction of global minimum, namely complex shuffling. Finally, it includes a competitive evolution. The level of difficulty in solving a global optimization problem depends on the number of parameters and the constitution of objective function The shuffled complex evolution (SCE) method implies to be robust, effective, and efficient for a broad class of problems, as mentioned in Duan et al. (1993).

Stochastic methods. The stochastic optimization methods generate the random input parameters, to include uncertainties and avoiding sensibility to model errors. It is considered well suitable for highly ill-posses, non-convexed and non-linear problems.

Stochastic methods as *Bayesian Inference Theory* also performs for solving inverse problems; for instance, its variation called Bayesian forecasting system implemented by Krzysztofowicz (1999), it separates the uncertainty into input which is the primary source and to hydrological which is less likely. Calculation of both should be in a separate way. It refines the Prior PDF by not consideration of the hydrological uncertainty.

Clearly, it explained in Oladyshkin et al. (2013) that efficiency and power of Bayesian updating strongly depend on the accuracy of prior information. It is considered arbitrary data to cover all human and tool errors that carry large uncertainties to shape the parameters distribution, from a simple approach at a low computational cost overcomes all disadvantages by Bootstrap filter as conditional PDF.

Recognition of an inverse problem is in Gupta et al. (1998). It explains that errors or uncertainties come from a model as just an approximation of nature. Also, it can not consider as only stochastic variables to fit a threshold curve. Even though, it is possible to

simplify a problem in a single objective function, in reality; it is multi-objective functional, this approaches the model must approach to the real system, not only to measurements.

The optimization methods not only can minimize a single *objective function*, but also there is the possibility of several objective functions. The multi-Objective Method (MOM), also still underdeveloped, is defined as the minimization or maximization of several objective functions (OF). The single *OF* rarely fulfills all features of the river system that are obtained in the observation data, as is implied in Madsen (2000). In hydrodynamics, the *OF* can be implemented either as temporal and spatial or as mean, maximum and minimum for each variable; most commonly used is the water depth. Normalization of the *OF*s values is necessary when the variable are different types, so they performance on the same scale. The solution for multi-objective functions will consist of a Pareto-optimal set of parameter values, which are best from a multi-objective point of view. Among the *OF*s exists a trade-off, which provides the best solution of an *OF* when it is the worst of another *OF* so that the best option is the combination on both, taking into account a compromise.

The most significant limitations for MOM are the low-quality data, the considerable computational time for large scale hydrodynamic models; the use of parallelization techniques brings a better performance. Dung et al. (2011). On the other hand, the quality of the MOM method relies on a good agreement between averaged simulated and observed data in space and time as well as a good agreement between simulated and observed data of the peaks and lows, as seen in Madsen (2000).

1.3 Challenges of Automated Calibration

In the past decades, it arises enormous advances in automation of river model calibration; however, it is a real challenge the reduction of data requirements or the increasing data quality. Likewise, one of the most desirable improvement is the implementation of an appropriated mathematical optimization method that calibrates a system model under 20 iterations. This method must be robust enough to find the solution regardless of the input data; it should propose a meaningful solution which describes the natural behavior of a river system; and it must be computationally economical for the calculation of large river systems, as detailed below.

Data availability. The measurement data of river systems are expensive and not always available in proper resolution in space and time; furthermore, in the case of existences is not each time trustworthy. River systems measurement data relies on human and tools resources which are sensitive to technical errors, wrong interpretations, and even sabotage; so not only quantity is necessary but also quality. The reduction of high costs of measurement data is the primary interest of investment parties. Variables are river systems characteristics that can be directly and indirectly measured in the field, such as water depth, bed evolution and flow rate. Parameters are values that can be quantified in laboratories such as roughness coefficient, sediment sorting, porosity and others, as explained in Xu (2002).

Robustness. Finding the perfect optimization method for the construction of an automated calibration routine mainly depends on its global search capacity. The method must be foolproof; it does not matter either how wrong the observed input data can be, or how far away the initial parameters are at the first guess. An ideal method has to find an approximated global solution of the parameters that fulfill not only mathematically but also those with physical meaning within certain parameter's ranges. To achieve the approximation mentioned above, knowledge of the river system is required to judge the method's given solution as either potentially accurate and consistent with nature or as unlikely results.

By implementing the methods several times with different initial parameter sets on a highly ill-posed problem, it is possible to quantify the method's uncertainty, meaning the probability of finding either the global solution or any locals.

Physically meaningful. Some methods have the potential to be highly precise at finding a local minimum solution in an Ill-posed problem; however, this kind of solution is only mathematical significant. The complexity of nature demonstrates that different multiple parameters sets can reduce a threshold between curves of observed and simulated data, leading to an apparent solution. In nature, hydrodynamic and hydro-morphological parameters lie within limited ranges; for instance, the characteristic sediment diameter can neither be negative nor take too large values. Therefore, a suitable method must find a solution with values that stay within physical limits, according to nature and particular river system.

Computational feasibility. Nowadays techniques of parallelization, availability of high computer capacity and cluster systems have increased the attractiveness of methods that are computationally expensive. For instances, the ones that require first and second derivatives calculations, proper of Newtons derived methods, or stochastic and genetic methods; the previous two reject simulation of the model, meaning that some computations lose resources. While it is true that modern times bring access to better computational technology, the ambition to work with larger river systems has also increased. On theses grounds, a compromise between accuracy and computational cost is an essential requirement for selecting a suitable method.

1.4 Goals to Overcome Challenges

The overall purposes of the present master's thesis are to identify the most suitable mathematical methods for the implementation of an automatic calibration routine and to clarify their applicability on the hydro-morphological numerical models based on Telemac2D -Sisyphe. The description of specific goals follows below.

Goal .1 To propose a robust and computationally cheap optimization method that provides a proper, accurate solution by taking into account physical limit bounds and constraints functions that involve the relationship between parameters; namely, meaningful calibration and correctness not only a small trade-off between simulated and measurement data. Computational cheapness means that the methods must achieve a solution in less than 20 iterations, at the lowest usage of first and order derivatives functions.

Goal .2 To assess the parameters mathematical landscapes, from which it is possible to determine if the problem is either well or ill-posed. In the case of an ill-posed problem, to identify the multi-local minimum and to infer the sensibility of parameters set with the relationship of each other. The method must have enough robustness; despite a wrong non-identified measurement data is provided. The global blind searcher must count with a low probability of being in non-global minimum.

Goal .3 To design a programming routine in Python 2.7 with the best-selected methods after their assessments, to automatically calibrate any numerical model, digitalized by Telemac2D-Sisyphe, at the same time, reducing the professional expertise intervention of manual calibration.

1.5 Outline

The structure of present thesis denominated Automated Calibration for Numerical Model of River flow counts with acknowledgments, thesis' declaration and abstract; also, lists of content, figures and tables, the five main chapters, annexes, and the bibliography.

Chapter I contains the master's thesis *introduction* and *literature review* that briefly describes the main characteristics of river modeling and the model calibration background through citations taken from research articles. As well, explanation of restrictions and challenges of automatic calibration procedures and how the present study plans to overcome them as goals definition. Lastly, the current outline of the thesis.

Chapter II describes the first part of the *methodology*, expressly the main features of hydro-morphological dynamic models. Also, it explains the physical and numerical model. First mentioned details the geometry, hydrograph, and the measurement data; and the latter part entails the description of governing equations, computational modules setup within the initial and boundaries conditions, parameters to be used in the calibration process and the numerical solution proposed by the Telemac-2d-Sisyphe.

Chapter III presents the other part of the *methodology*. It contains the mathematical definitions and explanations of the candidate optimization methods and algorithms, such as gradient and free-gradient based methods, genetic-based algorithms and stochastic method, specifically the Bayesian Inference theory. Also, it introduces the objective function to be minimized.

Chapter IV discloses the hydro-morphological model results. It incorporates the definition of *Euclidean Norm* as the objective function and its components, the ill-posed problem evaluation, and the parameter behavior analysis of the utilized computational model. Also, it includes the assessment of deterministic methods with their performances, convergences and discrepancies, as well as uncertainty investigation of a few selected evaluated methods, and the possibilities of different objective functions. Moreover, it covers the assessment of the stochastic *Bayesian Inference* algorithm. Finally, it examines the calibration of the physical Yen and Lee (1995) model with real measurement data by the highest graded deterministic methods and the Bayesian method.

Chapter V summaries of all preview chapters, remarks the conclusions and it presents the discussions of the present study and follows the suggestions for future research work.

2 The Hydro-morphodynamic Model

The numerical model used in our investigation is the run number four of digitalized physical model proposed by Yen and Lee (1995). This model initially refers to unsteady flow conditions with five different implementations.

The present chapter contains the main features of the physical and the numerical applications. The former relates to geometrics, implemented hydrograph and obtained measurement data. The latter entails the discretization in space and time, which solves the governing equations at each point of the mesh at every time step. Simulations of the numerical model system operate through the computational modules TELEMAC2D for hydrodynamic and SISYPHE for sediment transport and bed evolution module.

2.1 Physical Model: *Bend Channel*

In 1995, Chin-lien Yen and Kwan Tun Lee, working in Hydraulic Research Laboratory of National Taiwan University in Taipei, constructed an 180° channel bend physical model for investigation of bed topography and sediment sorting in unsteady flow. Thereof, geometry channel, selected hydrograph of run 4 and collected measurement data are following described under the denomination of *the Bend Channel Model*.

Geometry. The experimental model consists of a one-meter width flume channel with upstream and downstream straight sections of 11.50 m long each. Both connected by an 180° bend of 4.00 m radius, a constant slope in flow direction equal to 0.002 (m/m) and a characteristic sediment diameter d_{ch} of 1 mm. The upstream straight reach is attached to a stilling basin and the downstream to a controlled weir followed by a settling tank, as seen in figure 2.1.

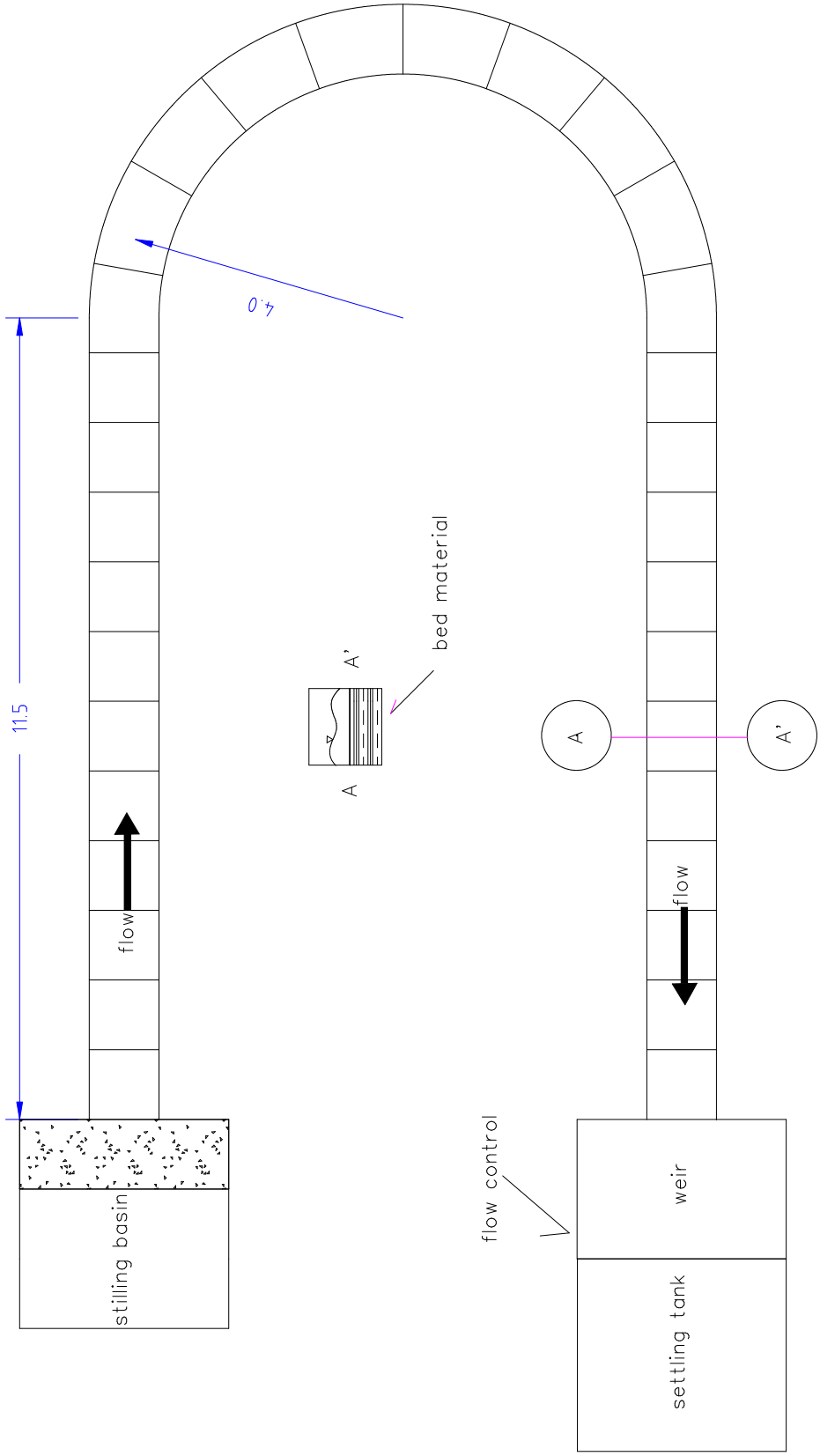


Figure 2.1: Bend channel geometry

Hydrograph. A triangular-shaped inflow hydrograph consists of the same initial and ending condition, namely a base flow of $Q_{baseflow} = 0.02\ m^3$/s and a peak of Q_{peak}=0.053 m^3/ s. Also, a constant water depth at the outflow is controlled by a weir of $h_{baseflow} = 0.0544\ m$ and h_{peak}=0.103 m, in a period of 300 minutes (5 hours), as seen in figure 2.2.

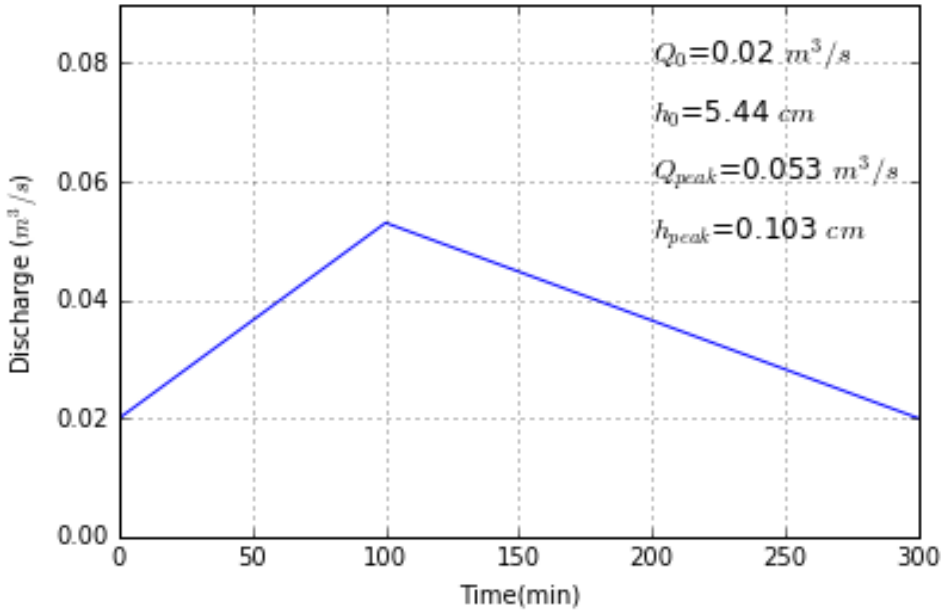
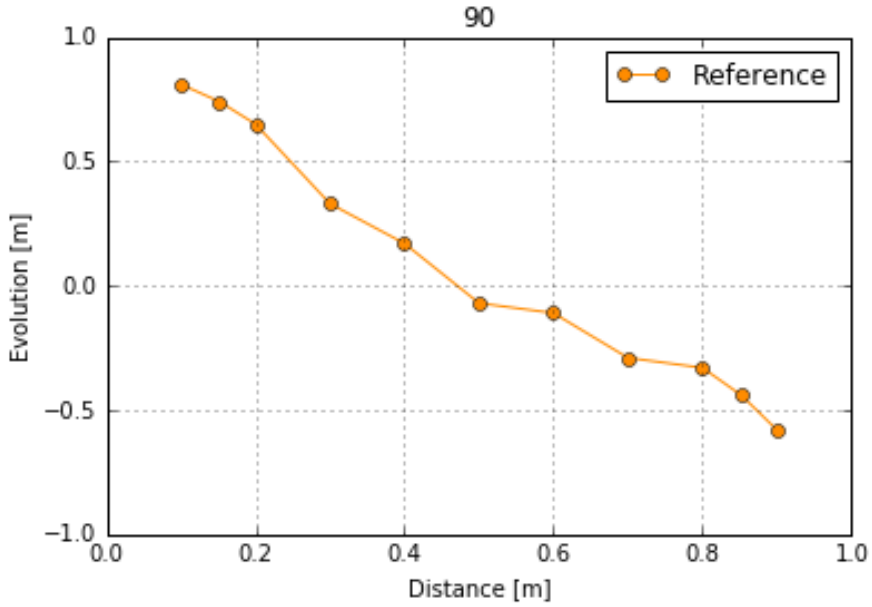
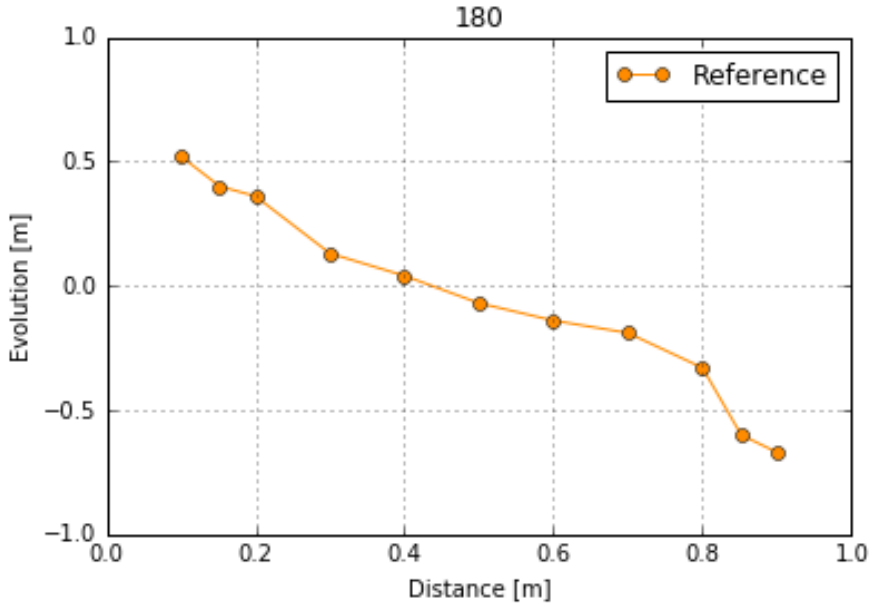


Figure 2.2: Hydrograph

Measurements data. The results of the physical experiment are the measurements of bed topographic evolution. These describe the growth and decay of bed sediment depth at cross sections 90 and 180 degrees of the bend channel after 300 minutes of time; normalized with the initial water depth $h_0 = 0.054m$. As seen in figure 2.3, the evolution profiles admit non-smooth slopes which are proper of nature systems, tools, and human errors. The typical cross-section in channels present a high sediment deposition in the inner bank, and erosion in the outer bank. Specifically, it appears at 90° a higher slope of the bed evolution and more deposition of sediment in the inner bank, in comparison to the 180° cross section that clearly shows a flatter slope and deeper scour depth in the outer bank, as pointed out in Yen and Lee (1995).



(a) Profile at 90 grades°



(b) Profile at 180 grades°

Figure 2.3: Measurement profiles
Source: Federal Waterways Engineering Research Institute, 2016

The description of the main features of the experimental model advises the system understanding, to incorporate it into an abstract mathematical optimization problem with the digitalized features of the hydrodynamic and hydro-morphological system.

2.2 Numerical Model

The numerical model of the *Bend Channel* incorporates an unstructured, triangular mesh with 1050 elements and 709 nodes to be solved by Finite Element Method (FEM). The modules TELEMAC2D-SISYPHE solve the governing equations of Saint Venant and Exner plus others expressions for hydro and- morphological dynamics. From that place, the experimental, and non-directly measurable parameters are chosen for calibration purposes. Moreover, the technical computational set-up of modules and expected numerical solution outputs with their specifications are subsequently summarized.

Governing Equations. The computational hydrodynamic module of TELEMAC2D solves the Saint Venant 2D or shallow water equations; additionally, it calculates the viscosity turbulence by Elder model and the Nikuradse friction law. The Saint Venant consist of the continuity and momentum equations:

$$\frac{\partial(h)}{\partial t}+\frac{\partial(hu)}{\partial x}+\frac{\partial(hv)}{\partial y}=0, \tag{2.1}$$

$$\frac{\partial(u)}{\partial t}+\vec{\mathbf{v}}\cdot\nabla u+g\frac{\partial Z}{\partial x}=\frac{1}{h}\nabla\cdot(h\nu_T\nabla u), \tag{2.2}$$

$$\frac{\partial(v)}{\partial t}+\vec{\mathbf{v}}\cdot\nabla v+g\frac{\partial Z}{\partial y}=\frac{1}{h}\nabla\cdot(h\nu_T\nabla v), \tag{2.3}$$

where u and v are velocities in x- and y- directions, g is the gravity acceleration constant, t is time, h is the water depth, $\vec{\mathbf{v}}$ is the velocity vector, Z is the free surface, ∇ is the divergence of a vector, and ν_T is the sum between mean and turbulent kinematic viscosity.

The Elder Model for Saint-Venant equations offers the possibility of specifying different turbulent viscosities k according to the flow direction, along the current: $k_L=\alpha_L\cdot U*h$, and across the current: $k_T=\alpha_T\cdot U*h$, where the dimensionless dispersion coefficients α_L and α_T are equal to 6 and 0.6 respectively, and $U=\sqrt{\tau/\rho}=\sqrt{\nu(\partial u/\partial z)}$ is the bed shear velocity, τ is the shear stress.

Furthermore, TELEMAC2D solves the quadratic bottom and boundary friction coefficient C_d by different formulations; in this case, Nikuradse friction law is used:

$$C_d = 2\left[\frac{\kappa}{\log(\frac{12h}{k_s})}\right]^2, \tag{2.4}$$

where the Kármán constant κ is equal to 0.4 and k_s is the Nikuradse roughness coefficient. The bottom and boundary roughness k_s admit a constrained proportionality α to the Characteristic sediment diameter d_{ch} of the grains which befall between 1 and 5, defined as $K_s \approx \alpha \cdot d_{50}$ with physical meaning.

On the other side, the computational morphodynamic module of SISYPHE solves the Exner equation for the bed morphological evolution:

$$(1-n)\frac{\partial Z_f}{\partial t} + \nabla \cdot Q_b = 0, \tag{2.5}$$

where n is the dimensionless parameter of *non-cohesive bed porosity*, Z_f is the bottom elevation and Q_b is the gradient of bed load. Moreover, SISYPHE calculates the sediments transport rates and slope bed effect at each grid point; those are dependable on the flow and sediment characteristics.

The total sediment transport rate Φ_T is based on the energy concept by Engelund-Hansen, instead of threshold conditions between critical and boundary shear stress, in this formulation case is recommendable the use of fine sediments with characteristic sediment diameter d_{ch} within the range ($0.2 \text{ mm} < d_{50} < 4 \text{ mm}$), equation can be seen below:

$$\Phi_t = \frac{0.1}{C_d}\hat{\theta}^{5/2}, \tag{2.6}$$

$$\theta' = \frac{\mu \cdot \tau_0}{(\rho_s - \rho)g d_{50}}, \tag{2.7}$$

where the total bed shear stress $\hat{\theta}$ is the function of the dimensionless skin shear stress θ', so that $\hat{\theta} = f(\theta')$. The dimensionless skin shear stress θ' is a function of parameter μ, the correction factor denominated as the *ratio between skin friction and mean diameter*, and τ_0, the local skin friction.

Additionally, the adjustment of the sediment transport rate Φ_t due to the slope bed effect components is executed by correction factors. The transversal inclination of the channel bed appears after mechanical sediment deposition and bed erosion along the cross section, moving the sediments in downslope direction by gravity force, mainly perpendicular to the flow direction. Physically, the slope bed effect influences the magnitude and direction of the sediment transport rate. The correction of sediment-transport-rate magnitude is

included by a multiplication factor on the solid sediment load Q_{b0}, demonstrated in the method of Flokstra and Koch (1981) and presented as:

$$Q_b = Q_{b0}\left(1 - \beta\frac{\partial Z_f}{\partial s}\right), \tag{2.8}$$

where s is the coordinate in the direction of the flow and β is an empirical factor.

The adjustment of sediment-transport-rate direction due to the slope bed effect is taken into account by the formula:

$$\tan\alpha = \tan\delta - T\frac{\partial Z_f}{\partial x}, \tag{2.9}$$

$$T = \frac{1}{\beta_2\sqrt{\theta}}, \tag{2.10}$$

where α is the direction of sediment transport, δ is the direction of bottom stress, both to respect to the flow direction, and x is the coordinate along the perpendicular axis to the flow. T is Deviation coefficient that is calculated based on Talmon et al. (1995). T depends on the Shields parameter θ and the *parameter of deviation* β_2, see equation 2.10.

Due to the channel curvature, centrifugal inertial forces appear, producing the phenomena called secondary currents that separate the sediments transport from the main flow, to take into consideration the radial effect, as explained in Engelund (1970), following formula must be solved:

$$\tan\delta = \frac{7h}{-\rho\alpha'\frac{U^2}{g\frac{\partial Z_s}{\partial y}}}, \tag{2.11}$$

where δ is the bed load transport angular deviation of the main flow, ρ is the water density, U is the velocity resultant, Z_s is the free surface elevation, y is the transversal direction, and α' is the *secondary currents alpha coefficient* that lays between 0.75 and 1.0, according to Tassi and Villaret (2014).

More information about the previous equations, which are implemented on the computational modules TELEMAC2D-SISYPHE, can be found in Hervouet (2007) and Tassi and Villaret (2014).

Numerical solution: Telemac2D-Sisyphe Setup. The spatial discretization scheme utilized corresponds to the *Finite Element Method*. The most relevant features for the functionality of hydrodynamic and hydro-morphological modules appear in the input files. The main files represent the control panel of calculation, denominated *Steering files(.cas)*,

created in a text editor. Previous involve the module configuration of computation including physical and numerical parameter as the list of keywords and their assigned values; conjointly, supporting file names, located in the same directory. For the present investigation, Telemac2d.cas and Sisyphe.cas are the required steering files. The system is internally programmed as subroutines, listed on a *Fortran file (.f)*. The model's mesh and silhouette characteristics information are incorporated in the *Geometry file (.sel)* as a binary format; there, the number of mesh points, elements, nodes per element and coordinates of all the nodes and connectivity table are available.

Three supporting files detail the **initial and boundary conditions** at each point of the mesh boundary. The *Boundary conditions file (.cli)* contains the open boundaries, such as water depth, flow rate, velocity, or friction coefficient, the investigated model has 366 boundaries points. The *Liquid boundaries file (.qsl)* contains the open boundaries in dependency of time, such as hydrograph. The *Previous computation file (.restart)* provides the initial state of the computation optionally.

The resulting variable information after computation at each time step and mesh point are contained in *Result files (.res)* in binary formats, such as velocities in x and y directions, water depth, free surface, bottom, evolution, flow rate, friction, and others for each module.

The possible variable outputs offered by Telemac-2D-Sisyphe, for instance, water depth and bed evolution over the model domains are presented on 2.4.

Parameters. The overall specification of hydrodynamic and hydro-morphological characteristics for the whole domain of the *Yen and Lee Model* are contained into keywords on steering files as the parameter's name and their values. Working along with programming tools is possible to update their values for calibration process iteratively. Specifically, for present investigation the selected parameters used for model optimization purposes are following as listed: Bottom and boundary roughness K_s parameters, denominated by TELEMAC2D as *friction coefficient* and *lateral roughness coefficient of boundaries*.

Also, the most important hydro-morphological parameters are selected. They are directly bounded to hydrodynamics, one more than other due to sensitivity. In SISYPHE, the keywords are *Characteristic sediment diameters* d_{ch}, *secondary currents alpha coefficient* α', *ratio between skin friction and mean diameter* μ, *non-cohesive bed porosity* n, parameters for direction deviation β_2 and magnitude of deviation β.

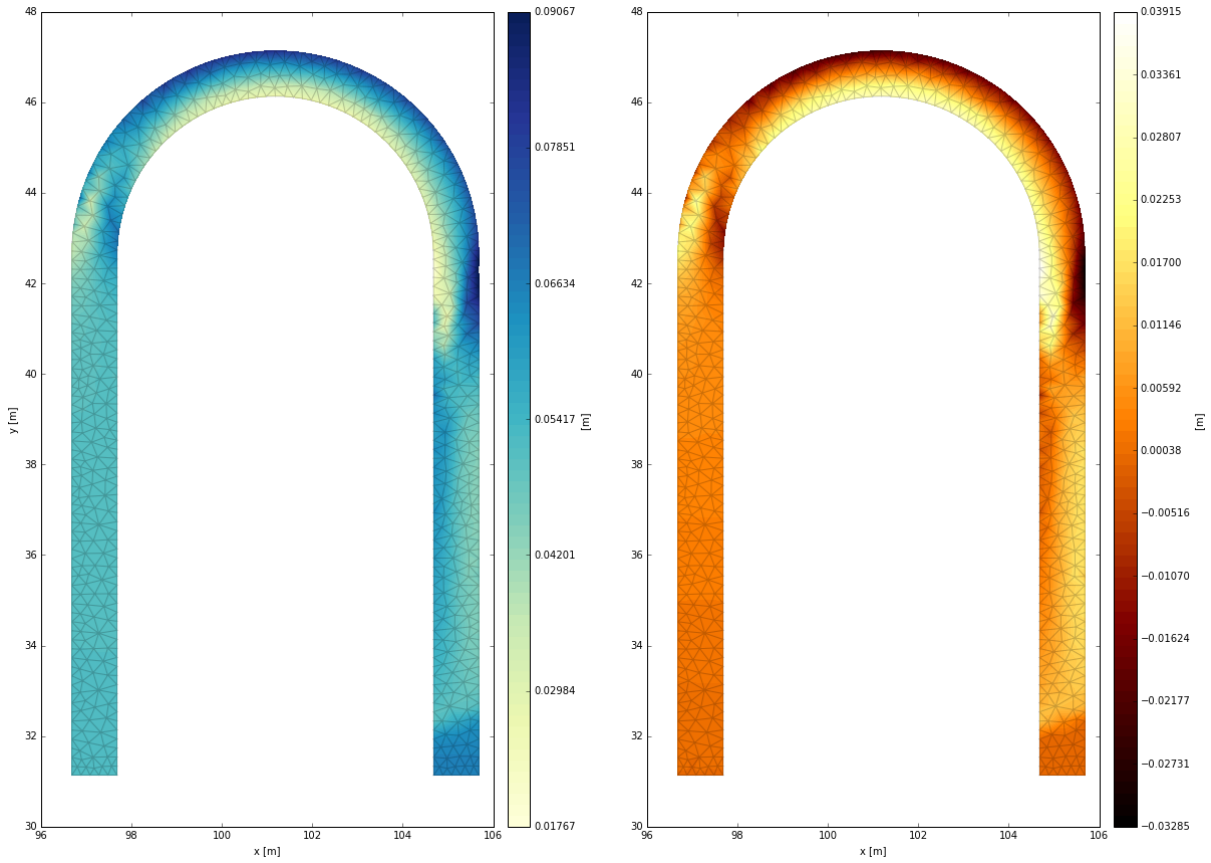


Figure 2.4: Water Depth and Bed Evolution of the Bend Channel

As mentioned, different hydrodynamic and hydro-morphological variables calculated by Telemac2D-Sysiphe can be the results or output data after simulation with certain input parameters. For instances, those variables can be the velocity in x -and y-direction, the water depth, the free surface, the scalar flow-rate, the scalar velocity, the bottom friction and the bed morphological evolution. Any variables, which values are at every time step, and every mesh point, can be utilized for the calibration process by inserting them into the objective function that must be minimized by the candidate optimization methods. The main criteria to choose the variable is the type of measurement data and the step time that was collected.

3 Optimization Algorithms and Methods

The mathematical optimization methods and algorithms, which are selected as the potential candidates to be investigated, are *Nelder-Mead simplex*, *Newton Conjugate Gradient*, *Levenberg-Marquardt*, *Sequential Least Square Programming* and others. According to each method, they can solve multivariate functions, either constrained or unconstrained, also bounded or unbounded. In our investigation candidates are implemented through the SciPy optimization package, version 17, for the programming language *Python 2.7 and 3.5*. The purpose of the candidates'assessment is to integrate the best suitable methods into a calibration routine for hydro-morphological models.

The present chapter introduces the objective function; then it explains the analytical background of conceivable optimization methods and algorithms under the following categories: gradient and free-gradient, genetic and stochastic based methods.

3.1 Objective Function

The contemplated *objective function* for the classical inverse problem is the absolute distance between the observed and the simulated data. Both data are spatially and temporally continuous; however, only the last time step simulation is considered for the objective function. The objective function's definition is an weighted *Euclidean Norm*; the (L_2-norm) is the minimization's goal of optimization methods, see equation:

$$\min \quad \phi(\mathbf{x^*}) = \left(\sum_{i=1}^{n} w_i \cdot |f_{x_i} - y_{obs}|^2 \right)^{1/2}, \tag{3.1}$$

where:

$\phi(x^*)$ is the objective function, appropriately weighted according to credibility of observed data; with a total n misfits between the outputs variables and the measurement points at selected cross sections,

$f_x - f_{obs}$ is the residual function, the misfit between measurement and simulated variable data.

x^* is the optimal solution after minimization.

i stands for the position of simulated variable point. n is the total variables points,

w_i is the weight vector, which takes either same dimension as the simulated and observed data for the different importance of measurement points, or scalar value for equally weighted data.

$f(x_i)$ the simulated function is used for the sediment evolution for the optimization method assessments and water depth for objective functions evaluation,

y_{obs} the observed data of sediment evolution at 90 and 180 degrees, or its replacement of generated synthetic evolutions and/or water depths data with known input parameters at any cross section of the bending channel.

Also recommended is the minimization of different norms or combination of them with relatively weighted adjustment, according to the importance of each observation point.

Sun and Sun (2015) supports larger weight assignments for the observations that are more accurate, that represent the system's state in a large spatial and/or temporal scale, and that are sensitive to relevant model parameters.

3.2 Gradient-Based Algorithms

The gradient-based methods, either Newton or the conjugated gradient-based, as explained in Nocedal and Wright (2006), in turn rely on either the line search or the trust region strategies. The former strategy computes a search direction p_k and then chooses a line search value α_k; both are included in the iteration process. The objective functions that are to be minimized can be either in linear or in quadratic form. The equations can be seen below:

$$\min_p \quad m_k^L(x_k + p) = f_k + p_k^T \nabla f_k, \tag{3.2}$$

$$\min_{p} \quad m_k^Q(x_k+p) = f_k + \nabla f_k^T p_k + \frac{1}{2} p_k^T B_k p_k, \tag{3.3}$$

$$x_{k+1} = x_k + \alpha_k p_k, \tag{3.4}$$

where m_k^L is the linear model function, m_k^Q is the quadratic model function, $f_k$is the objective function to be p is the line search direction. Also, α is the line search magnitude, B is a Hessian matrix, k is the iteration number, and T stands for transposed of a matrix.

It must be assured that $p_k = -B_k^{-1}\nabla f_k$ is in decent direction and B is a symmetric non-singular matrix by choosing a line-search α_k that reduces the objective function without spending additional computational time. To accomplish the previous statement, the fulfillment of certain conditions is necessary, either the Wolfe conditions:

1) The Armijo rule: $f(x_k+\alpha p_k) \leq f(x_k) + c_1\alpha\nabla f_k^T p_k$, for decrease of objective function, and 2) The curvature condition: $p_k^T \nabla f(x_k+\alpha_k p_k) \geq c_2 p_k^T \nabla f(x_k)$ for selection of strongest negative slope,

or The Strong Wolfe conditions:

1) The Armijo rule and 2) the modified curvature $|\nabla f(x_k+\alpha_k p_k)^T p_k| \leq c_2 |\nabla f_k^T p_k|$,

where constants c_1 and c_2 depends on the method, always under the conditions $0 < c_1 < c_2 < 1$. In practice, $c_1 = 10^{-4}$ and $c_2 = 0.9$ for the Newton and the Quasi Newton methods, and $c_2 = 0.1$ for nonlinear conjugate gradient.

For the trust region strategy, the search radius plays around the current iteration, within the choosing step length and direction for the next function evaluation, defined by the solution of the quadratic form equation 3.3.

Where $\| p \| \leq \Delta_k$, the norm of line search direction is smaller or equal than the trust-region radius. If the new step length fails the function evaluation, it follows a radio reduction.

In general, the *Gradient-based methods* and the *Conjugated gradient methods* require the Jacobian matrix and/or the Hessian matrix, as explained in Nocedal and Wright (2006); otherwise, an approximation of the first and the costly computational of second derivative, in both cases leading to a higher number of function calls for a single iteration. On the other hand, Jones et al. (2001–) explains that the inversion of provided gradient or Hessian functions become unstable for highly dimensional problems, bringing singularities. In this case, *Quasi-Netwon Methods* approximate the inverse Hessian matrix directly, forcing to

be positive defined, is suitable for until 250 unknowns; such as BFGS (Broyden-Fletcher-Goldfarb-Shanno algorithm) and its bounded version L-BFGS-B. Varoquaux et al. (2015). The following sections describe the gradient-based methods for later implementation.

3.2.1 Levenberg-Marquardt Method (LM)

The combination of Levenberg (1944) and Marquardt (1963), uses the line search strategy and B as an approximated Hessian matrix. The LM method proposes a smart self-adaptive optimization algorithm for the numerical solution of least-squared problems by continuously iterative updating estimation of the non-linear parameters until finding a local and in the case of fortunateness the global minimum. This method applies the *Gauss-Newton* and the *Steepest Descent* Methods. The LM method is a modification of the former by incorporating the curvature information of the objective function parameter space, property of the latter. The LM method amplifies the approximated Hessian diagonal with a damping factor λ; thus moving further in the smaller gradient direction to get into the error valley. As explained in Moré (1978), the updating iterations are given by expression:

$$x_{k+1} = x_k - \left[J_{x_k}^T \cdot J_{x_k} + \lambda \cdot Diag(J_{x_k}^T \cdot J_{x_k})\right]^{-1} \cdot J_{x_k}^T \cdot [f_{x_k} - y_{obs}], \tag{3.5}$$

where x_k is the parameter vector of k iteration, J_{xk} is the Jacobian matrix, which entries are first derivative of parameters x at k iteration, λ is the damping factor, $Diag$ is the diagonal matrix, $f_{x_k} - y_{obs}$ is the residual function, and T stands for a transposed matrix.

The approximated the Hessian matrix B is the dot product between the transposed Jacobian and the Jacobian matrix, the mathematical approach that arises from the *Gauss-Newton Method*. In this way, the LM method requires only the first derivative, avoiding the highly computational cost of the second derivative. It keeps the fast convergence of the Hessian properties, but it reduces the robustness of the *Steepest Gradient*, making the LM method more sensitive to the initial parameters. In addition to robustness, the main feature of this implementation is the proper use of implicitly scaled variables.

The LM method auto-adjusts according to the residual function evaluation at each iteration. In a case of a larger residual in comparison to the previous one, the LM method rejects the iteration and returns the previous parameter set, then when λ is increased, performs as the gradient descent method, being highly robust and slow convergent. On

the contrary, if the iteration is accepted, then the λ gets a reduction, expecting a *Gauss-Newton* behavior, becoming more sensitive and fast convergent. In which the quadratic error that assumes a linear approximation of the function is only valid near to a minimum. This way, it is possible to suppress oscillations and not to overshoot, as explained in Nowak and Cirpka (2004).

The LM method does not necessary find the global minimum. As described by Seong et al. (2015) in the PEST employment, its successful search depends on a good first guess parameter set, specified by the user. Meanwhile, the LM method accepts the first local minimum as a good value.

In the optimization library of Python exists two different ways of LM method's implementation. One way searches for the root of a vector function, in this case, can be the residual function between the observed data and the simulated data. The other way wraps a FORTRAN's library subroutine, denominated Minipack, to solve the least square of non-Linear equations. The disadvantages of both LM's method implementation is the lack of bounds to keep the parameter solution within the physical meaning. They require more observations than parameters, do not work on underdetermined problems. They are highly adaptable according to the chosen factor λ and scaling of the diagonal matrix, approaching to the *Gauss Newton* approximation method when λ is 0.1, and to the *Descent Gradient* when λ approximates to 100.

3.2.2 Polak and Ribière Method (CG)

The *Polak and Ribière* method is a nonlinear conjugate gradient algorithm that minimizes the objective function $\phi(x)$ with the line search strategy and the Wolfe conditions to test it. Given an initial parameter set x_0, that is used to initiate the residual function $f_0 - y_{obs}$ and its gradient ∇f_0, whose first value is assigned to the line search direction (p_0). It computes iteratively $x_{k+1} = x_k + \alpha_k p_k$ by introducing the β in the line search direction term. The Polak and Ribière Algorithm is solved as follows:

- The line search is defined by $\alpha_{k+1} = arg \quad minf(x_k + \alpha_k p_k)$
- The line search direction is defined by the sum of negative gradient and an upgraded Polak and Ribieri beta plus, $p_{k+1} = -\nabla f_k + \beta_k^{PR+} p_k$
- The parameter beta for Polak and Ribière $\beta_{k+1}^{PR} = \dfrac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\| \nabla f_k \|^2}$

- Due to *Beta* addition, the Wolfe conditions do not guarantee p_k decedent direction, so the following must be chosen $\beta_{k+1}^{PR+} = max\{\beta_{k+1}^{PR}, 0\}$

In the optimization library of Python, the Polak and Ribiere (CG) method requires a callable Jacobian function and the order of the norm, in this case, *Euclidean* is the scalar value 2.

3.2.3 Newton Conjugate-Gradient (NCG) or truncated Newton method (TNC)

The Newton Conjugate Gradient, also known as the Truncated Newton Method, is an inexact Newton method that solves indirectly the exact Newton System equations $\nabla^2 f_x \bar{p}^N = -\nabla f_x$ by computing an approximated line search direction $p_k \approx p_k^N$ as the *Conjugate Gradient method* and the line search length α_k that satisfies the Wolfe conditions, see expressions:

$$p_{k+1} = -r_{k+1} + \alpha_k \beta_{k+1} p_k, \tag{3.6}$$

$$r_k = \nabla^2 f_k p_k + \nabla f_k, \tag{3.7}$$

$$\beta_k = \frac{r_{k+1}^T \cdot \nabla^2 f_k \cdot p_k}{p_k^T \cdot \nabla^2 f_k \cdot p_k}, \tag{3.8}$$

where r_k is the residual between exact and inexact newton solution, α is the line search length, p is the line search direction, f_k is the objective function, and β is the scalar value that ensures p_{k+1} and p_k are conjugated with $\nabla^2 f_k$, so that $p_k^T \cdot \nabla^2 f_k \cdot p_{k+1} = 0$.

As termination criteria, the exact approximates to inexact solution after relative size of residual is small enough, so forcing the sequence $\eta_k = min(0.5, \sqrt{\| \nabla f_k \|})$ is introduced to determine $\| r_k \| = \eta_k \| \nabla f_k \|$. The method does not require the knowledge of full Hessian matrix $H_k = \nabla^2 f_x$, instead the Hessian-vector product $H_k \cdot p = \nabla^2 f_x \cdot p$ by the second approximation of the Taylor's theorem, presented on expression:

$$\nabla^2 f_k p \approx \frac{\nabla f(x_k + hp) - \nabla f(x_k)}{h}, \tag{3.9}$$

where p is the search direction vector and h is the delta parameter step as small can be changes visible, for every iteration the curvature test $p_j^T H_k p_j \leq 0$ is fulfilled, ensuring that the search direction is decedent, as explained in Nocedal and Wright (2006).

In the optimization library of Python, the Newton Conjugate-Gradient method (NCG) and the truncated Newton method (TNC) are separated optimization options. The TNC

method differs from the NCG method as it wraps a C implementation and allows each parameter to pass upper and lower bounds. For both methods, the requirement of full Hessian is optional. Otherwise, it is used the Hessian-Vector product. Unfortunately, the methods become unsuitable for singular Hessian matrix which appears in ill-problems, although the TNC method performs better than NCG in this situation probably due to its bounds requirements, as explained in Jones et al. (2001–).

3.2.4 Truth-Region Newton Conjugate Gradient Method (Truth-ncg)

The *trust-region Newton method* consists of constructing a model function m_k over a radius of investigation Δ of the current parameter set iteration x_k with the information from the objective function $\phi(x)$. By solving the Newton equations $\nabla^2 f_x \bar{p}^N = -\nabla f_x$ with the search direction p and with an step in the trust region, a candidate x that produces enough decrement on $\phi(x)$ becomes x_{k+1} for next iteration. In the contrary case, the radial region is shrunk, repeatedly trying the candidate evaluation until finding a reduction on $\phi(x)$. The model function m_k, subjected to $\| p \| \leq \Delta_k$, is minimized; it has usually a quadratic form on expression 3.10. Subsequently, it calculates the ratio between actual and predicted reduction as 3.11 in order to choose the trust-radius at each iteration.

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T H_k p \tag{3.10}$$

$$\rho_k = \frac{f(x_k) - f(x_k + p)}{m_k(0) - m_k(p_k)} \tag{3.11}$$

where m_k is the model function, f_k is the objective function at k iteration, H is the Hessian matrix, p is the line search direction, and ρ is the ratio between actual reduction and predicted reduction by the model.

If $\rho \leq 0$, the new function evaluation is greater than the previous one, then the solution is rejected $x_{k+1} = x_k$ and the radius is shrunk for next iteration $\Delta_{k+1} = \frac{1}{4} \| p \|$. If $\rho \approx 1$, there is an agreement between the model function m_k and the objective function f_k, then the expansion of the radius for the next iteration is proposed $\Delta_{k+1} = 2\Delta_k$ and $x_{k+1} = x_k + p$. If $\rho > 0$ and < 1, then radius for next iteration remains the same $\Delta_{k+1} = \Delta_k$ and $x_{k+1} = x_k + p$.

In the optimization library of Python, the computation of the expensive full Hessian matrix is mandatory required, the first radio Δ and the maximal $max\Delta$ are optional. Choosing a fixed radius for a high dimensional problem, where every parameter has a different scale, is not proper.

3.2.5 Quasi-Newton Methods

Broyden (1970), Fletcher (1970), Goldfarb (1970), Shanno (1970), propose a Quasi-Newton method, usually referred as (BFGS) method, which is an iterative method for solving unconstrained nonlinear optimization problems. It is based on the secant method, which finds the root of the first derivative for multidimensional problems. The costly inverse Hessian matrix H_k^{-1} replacement for a positive defined M_k is modified gradually during the iteration process. It approximates the parameters step size by the use of either Armijo or Wolfe line search, assuring a positive curvature. In case a positive curvature is not possible, Dai (2013) says that a non-convex multidimensional problem converges when the step size is the first local minimizer of the line search. The expressions of iteration and the inverse positive Hessian matrix, are presented below:

$$x^{k+1} = x^k - \alpha_k \left[H_k\right]^{-1} \cdot J_{x^k}^T, \tag{3.12}$$

$$M_{k+1} = M_k - \frac{\delta_k \gamma_k^T M_k + M_k \gamma_k \delta_k^T}{\delta_k^T \gamma_k} + \left(1 + \frac{\gamma_k^T M_k \gamma_k}{\delta_k^T \gamma_k}\right) \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k}, \tag{3.13}$$

where the relative displacement of the parameter vector expressed as $\delta^k = x^{k+1} - x^k$, and the relative displacement of the gradient indicated as $\gamma^k = g_x^{k+1} - g_x^k$, are part of the inverse Hessian matrix iteration.

The L-BFGS is a modification presented by Liu and Nocedal (1989), where the L stands for "Limited-Memory" that works well for big problems where the number of parameters n is larger of given information m; it is based on the BFGS method but requires fewer operations per iteration and less memory. Instead of storing the dense n x n Hessian approximation matrix, it maintains a history of the past m updates of the position parameter and the gradients of the objective function.

In L-BFGS-B, B stands for bounds. It includes two new steps, a gradient projection, and a subspace minimization. The first step reduces the problem dimensionality by determining the function evaluation of parameter values that are within their bounds, reducing the initial dimensionality of the problem, the number of iterations and running time, so an active set is defined as expression 3.14. After minimization of quadratic form 3.3, $x*$ becomes x^c, also known as *Cauchy point*.

$$\mathcal{A}(x^*) = x_i^* = l_i \quad or \quad x_i = u_i \tag{3.14}$$

The second step restricts the solution under lower and upper bounds $[l_i < x_i < u_i]$ box. It iteratively solves similar to the L-BFGS with a stepsize limit. The method uses the

unconstrained line search direction $\hat{p}$, evaluates the Wolfe conditions for line search length and delivers a solution in the constrained box $x_{k+1} = x^c + \alpha_k \hat{p}_k$, as explained by Henao (2014).

In the Python optimization library, the BFGS and the L-BFGS-B implementations are separated, both request the Jacobian matrix as a callable function for best performance, differing on the bounds' limits for the parameters.

3.2.6 Sequential Least SQuare Programming Algorithm (SLSQP)

The SLSQP method is a highly efficient optimization algorithm, based on the Sequential Quadratic Programming (SQP); it is designed to minimize multivariate nonlinear bounded and constrained *objective function*, written as a software package by Kraft et al. (1988) in Fortran 77 and implemented in the Python optimization library.

The SLSQP optimizer replaces the quadratic programming subproblem (QP) by a linear least squares subproblem (LSEI) as proposed by Schittkowski (1982) to determine the search direction. An adapted Lawson and Hanson (1974) solver for NNLS (non-negative least squares) is implemented.

$$\textbf{(QP):} \quad \min_p \quad \phi(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^2 \nabla^2_{xx} \mathcal{L}_k p + \mu m_k(p), \tag{3.15}$$

$$\textbf{(LSEI):} \quad \min_p \quad \phi(p) = \| \, (D_k)^{\frac{1}{2}} (L_k)^T p + D_k)^{-\frac{1}{2}} (L_k)^{-1} \nabla f(x_k) \, \|, \tag{3.16}$$

$$\mathcal{L}_k(x_k, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x), \tag{3.17}$$

where $\phi(p)$ is the objective function, $f(x_k)$ is the residual function and its gradient is $\nabla f(x_k)$, $c(x_k)$ is the constrain function, $\nabla^2_{xx} \mathcal{L}_k$ is the estimate Hessian of Lagrangian function, $D_k = diag(d_1, d_2, ..., d_n)$ is the diagonal Hessian matrix, $L_k = L(l_1, l_2, ..., l_n)$ is the lower triangular matrix of the Hessian, μ is the penalty parameter, and $(\lambda_1, \lambda_2, ..., \lambda_m)^T$ is a multiplier vector approximation; iterating at x_k parameters.

The subproblems are subjected to the following linearized equality and inequality constrain functions below:

$$\left\{ \begin{array}{l} \nabla c_i(x_k)^T p + c_i(x_k) = 0, \; i \in \mathcal{E}, \\ \nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \; i \in \mathcal{I}, \end{array} \right\},$$

within lower and upper bounds on parameters $[L_i < x_i < U_i]$. Where $\mathcal{E}$ stands for equations, and $\mathcal{I}$ stands for inequations.

The SLSQP method implements the Han (1977) and Powell (1978) approach, first adds penalty L_1 test as a merit function for constrains in the line search, defined as $\mu \cdot m_k(p)$, with μ being the penalty parameter 3.18. Second, it uses the Quasi-Newton method to replace the Hessian matrix of Lagrangian $\nabla_x x^2 \mathcal{L}(x_k, \lambda_k)$ by updating the approximated Hessian matrix H_k^{BFGS} on the equation:

$$m_k(p) = \sum_{i \in \mathcal{E}} |c_i(x_k) + \nabla c_i(x_k)^T p| + \sum_{i \in \mathcal{I}} |c_i(x_k) + \nabla c_i(x_k)^T p|^-, \tag{3.18}$$

$$H_{k+1} = H_k - \frac{H_k \delta_k \delta^T H_k}{\delta_k^T H_k \delta_k} + \frac{\gamma_k \gamma^T}{\gamma_k^T \delta_k}, \tag{3.19}$$

where the model function is m_k, the Hessian matrix is H_k, the relative displacement of the parameter vector is expressed as $\delta^k = x^{k+1} - x^k$, and the relative displacement of the gradient indicated as $\gamma^k = \nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_k)$.

In the optimization library of Python, the arguments of the SLQSP method include the constraints function, the bound intervals for each parameter and the callable Jacobian matrix function; all are optional but recommendable for a better optimization method's performance.

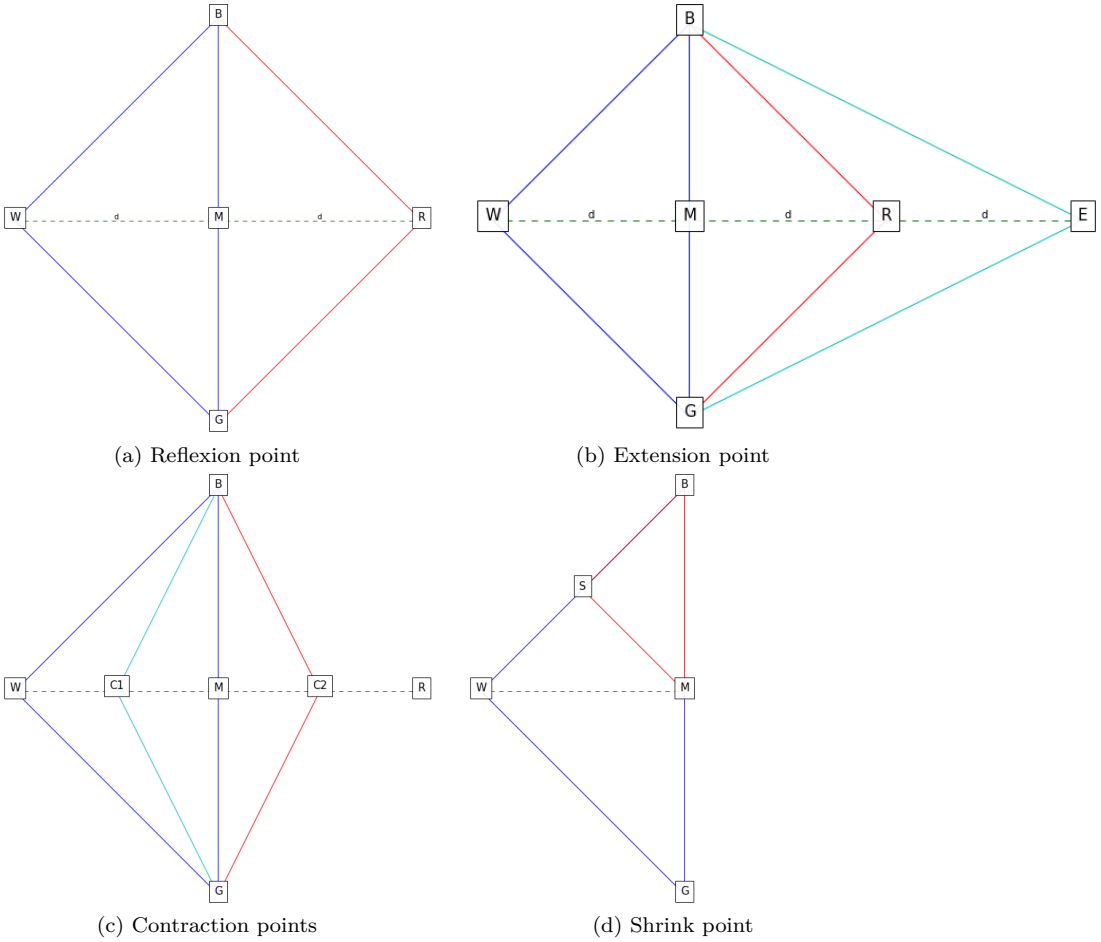
3.3 Gradient-Free Algorithms

The gradient-free methods require a low level of mathematical expression, practically just geometrics and arithmetics; a simple analysis is their main features. As following two candidate optimization methods are selected, namely Nelder-Mead which is based on triangulation, and Bruce force method which builds on a grid generation of evaluated objective function.

3.3.1 Nelder-Mead Method

The simplex method *Nelder-Mead* finds a local minimum without the gradient calculations. It evaluates the *objective function* with one given initial parameter vector and two other generated parameter vectors in different directions. The three evaluated parameter

vectors form a triangle, where a ranking qualification is assigned (best B, good G and worse W) for each vertex, according to the objective function aim. The parameter sets are saved as: $B = (x_1^1, ..., x_n^1)$, $G = (x_1^2, ..., x_n^2)$, $W = (x_1^3, ..., x_n^3)$. The midpoint between the best vertex and the good vertex is calculated by $M_k = \frac{B+G}{2}$, the distance between M and W is named the reflexion distance d.



(a) Reflexion point

(b) Extension point

(c) Contraction points

(d) Shrink point

Figure 3.1: Strategies of the Nelder-Mead method

Henceforth, the advanced triangle is formed from the past best vertex, the good vertex and a new vertex, which is obtained by projecting in a mirror way the previous worse vertex. The reflected vertex R is located by $R = M + (M - W) = 2M - W$. In case, R does not result in a new worse vertex; the expansion vertex is calculated by E, $E = R + (R - M) = 2R - M$. Contrary, if neither E nor R vertexes perform a better objective function evaluation than W, then the two contracted evaluation triangles are measured. Previous is done from the midpoint with a half reflexion distance in the direction of R and the other in the direction of W. Even more, if none of the constrained new vertexes

reduce the objective function lower than W, then a smaller triangle from BMS, being S in between worse and best situation, is constructed.

After the obtainment of a new vertex that replaces W, the triangulation process continues with the ranking qualification process for every new triangle and the search of a new third vertex. The method searches until the achievement of the lowest function evaluation value and then assesses the third vertexes sequentially, updating the worse vertex with better ones. Otherwise, other searching techniques of a new third vertex are implemented, as explained Nelder and Mead (1965).

Non-control of the evaluated suggested parameter vectors by the Simplex method, which is included in the optimization library of Python, generates unsatisfactory unphysical parameters sets, producing computational conflicts in TELEMAC2D. The *Nelder-Mead* Method performs highly recommendable in the proximity of local or global minimum since the computational costly first and second derivative can be evaded.

3.3.2 Brute Force Method

The *Brute force* method is a global searcher that does not require high mathematical formulation; *Brute Force* computes the *objective function* value at each point of a multi-dimensional grid. Grid generation is in the range of provided parameter bounds' limits. The chosen number of evaluation points at each interval constitutes the space structure which can be fine or coarse. It does not require an initial parameter set; the final result is the lowest function evaluation of the multidimensional grid.

In the optimization library of Python, *Brute Force* method permutes over the grid points in the parameter space, with non-repetition of parameters and it considers the parameter's order, as it is presented:

$$P(n, N_s) = \frac{n!}{(n - N_s)!}, \tag{3.20}$$

where n is the number of parameters, N_s is the number of grid point along the axes; therefore, $P(n, N_s)$ is the number of function evaluation. After the potential global minimum is found, the package offers the possibility of refinement by choosing any other type of method in the optimization library.

3.4 Genetic Based Algorithms

It is likely that the algorithm finds a local instead of the global minimum in a very ill-posed problem. Hence it is appropriate to use first a robust global minimum searcher such as *Basin Hopping* or *Differential Evolution*, both included in the optimization library of Python.

3.4.1 Basin Hopping Algorithm

The *Basin Hopping BH* method, also named *Monte Carlo Minimization MCM* is a genetic algorithm with some stochastic features. The BH method attempts to find the global minimum by exploration "Hopping" and to achieve a local minimum by "exploitation." The BH method solves the smooth scalar objective functions with dependence on the highly multi-parameter. The BH algorithm was initially is described by Wales and Doye (1997) as a searcher for the lowest-energy structure of an atomic-and-macromolecular cluster in chemical physics; even though, it is proved to be valid in other science fields. The BH method is particularly useful for a highly ill-posed problem which separation of the local minima has large barriers. The Basin Hopping's algorithm is explained on figure 3.2.

Given an initial parameter set, the BH method proposes a new parameter set after adding some random perturbation - replacing the gradient direction steps in deterministic methods - to either the initial or the previous set, $x_{k+1} = x_k + perturbation$. The Metropolis criterion of the standard Monte Carlo is used as the acceptance test, in case the new function evaluation performs better than the previous $f(x_{k+1}) < f(x_k)$, a new generated random perturbation is added to the current parameter set. Conversely, the hopping probability is calculated $P^{BH} = e^{-\Delta f/T}$, to be compared with a generated uniform random number r on the opened interval [0.0, 1.0). If $P^{BH} \geq r$, then the new parameter set is accepted, then it adds a new perturbation to it. Contrary, a new perturbation is added to the old parameter set. In any case, it repeats the process until either the convergence or the achievement of iterations' limit. The temperature parameter T is assigned by the user, directly proportional to an averaged separation distance among local minima. If $T \approx \infty$ non-rejection is allowed enhancing the global search ability. Dangerously, it stays long calling the same parameter set when there is a non-proper addition of the perturbation, according to the parameter characteristics.

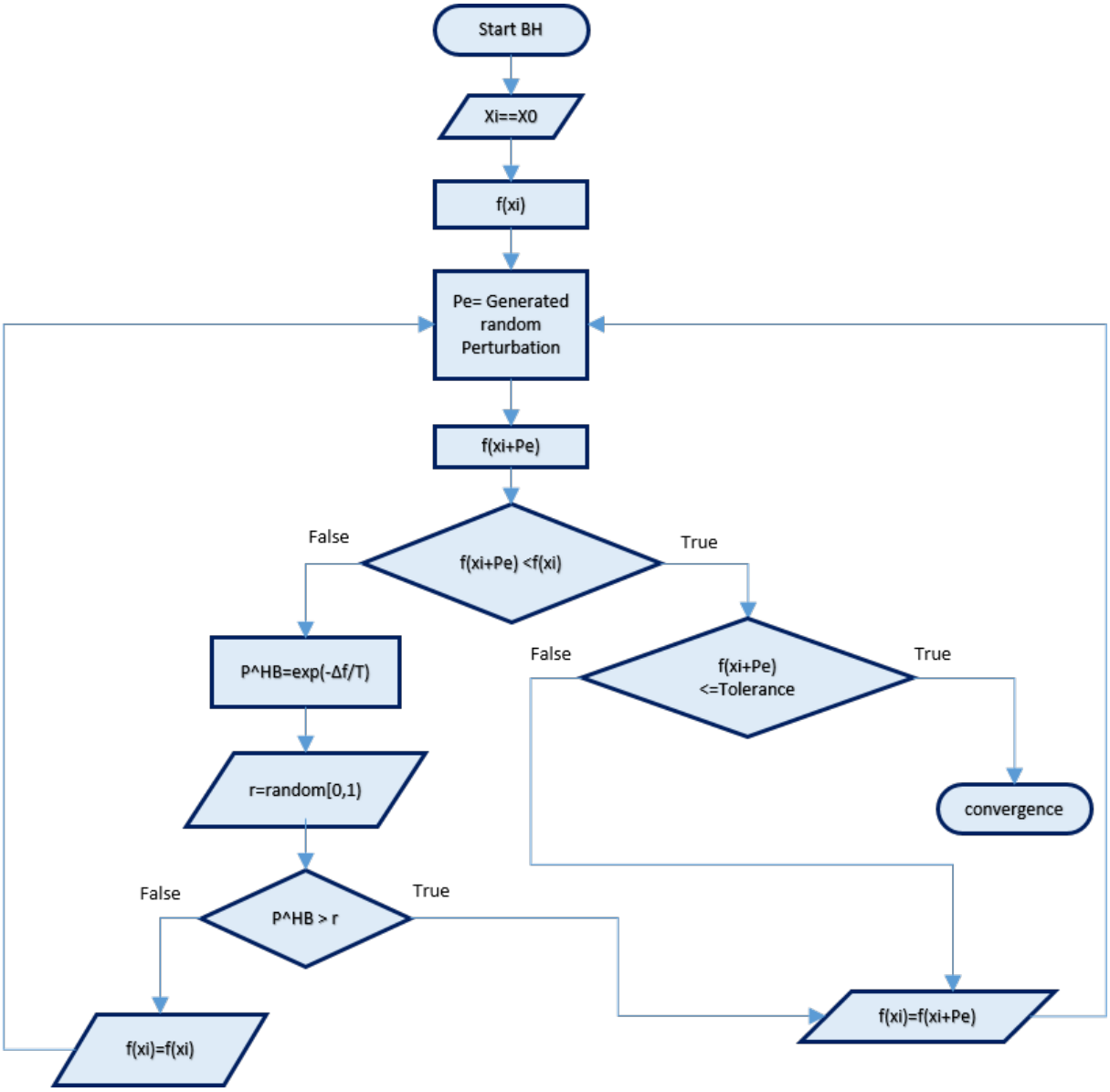


Figure 3.2: Flowchart of the Basin Hopping method

In Python optimization library, the arguments such as *acceptance test* can insert the interval bounds as selection criteria, and *take step* can choose the random distribution type of the perturbation. Both are used to improve and customize the algorithm performance. If the acceptance test is not fulfilled even at last iteration, the refinement of local begins with the BFGS method and with the initial parameter sets by default, which in case, are not specified since improvement is mandatory. Any other of the optimization method for refinement can be utilized, if it is previously selected.

3.4.2 Differential Evolution (DE)

Anther genetic algorithm disclosed by Storn and Price (1997) is the *Differential Evolution DE* with an evolutionary strategy, highly recommended for the large, noisy, nonsmooth, non-linear, non-continuous and non-stationary problems. It randomly searches into the large bounded parameters population per generation, storing candidates or agents to replace them after evaluation by better ones, following the steps of initialization, mutation, recombination, and selection.

The *Initialization* is when uniform random values on bound intervals are assigned to initial parameter vectors, as many population size is required, first generation; namely candidates with the form of $x_{i,G} = [x_{j,i,G}, ..., x_{D,i,G}]$, where D[j] is the index of parameter, N[i] is the index of population and G is the generation index.

The *Mutation* expands the search spaces, for a given population of parameter vectors, three different parameter vectors are randomly selected from the population, which receive the index of population (r_1, r_2, r_3). It is added the weighted difference of two of the vectors to the third. This is denominated donor vector and it is defined as $u_{i,G+1} = x_{r_1,G} - F(x_{r_2,G} - x_{r_3,G})$, the mutation weighted factor F lies on [0,2].

The *Recombination* incorporates the evaluation test for effectiveness. This step provides a uniform random number r on [0,1] which is compared with the CR (cross over probability) or recombination factor, initially determined by the user. If the former r is smaller than the latter CR, then donor vector $u_{j,i,G+1}$ becomes a trial vector $v_{j,i,G+1}$; otherwise, the parameter vector $x_{j,i,G}$ recovers its position in the population of the next generation.

The *Selection*, creates a new generation thereafter the successful solution from the last generation, the trial vector $v_{j,i,G+1}$ is evaluated in the optimization function $f_{op}(v_{j,i,G+1})$, if the output value is smaller than the previous one $f_{op}(x_{j,i,G})$, then this becomes in a parameter vector for the next generation.

Remarkably, by increasing the mutation factor F rises the radius of search, and increasing crossover probability CR allows a larger number of mutant or donor vectors to progress to the next generation. Unfortunately, DE requires scores of function evaluations.

In the optimization library of Python, the differential evolution method belongs to the global searcher package, arguments such as *popsize* stand for the size of the population. Besides, other different strategies for the mutation step are available options. The mutation factor can be not only an integer but also randomly selected within a given interval.

Another customization is to choose the initial parameters sets population by either Latin-HyperCube or Random options. Former tries to cover as best the range of given bounds interval, and latter accepts the initial population randomly. There is the possibility of refinement after the potential found global minimum, electing a minimization method with the option *polish*, its default is L-BSGS-B method.

3.5 Stochastic Method: *Bayesian Inference Theory*

The stochastically based Bayesian inference method is a classical inverse solver that performs from given information to infer consequent information, according to Bayes' Theorem, see equation 3.21. The Bayesian method refers to the information or the lack of it as the probability distribution functions (PDF). The imperfectly known generated information is named *Prior probability distribution function* (Prior PDF) from hypothetical parameters values x. The data of real measurement y_{obs} constitutes the conditional PDF as likelihood function. The prior and the conditional PDF are combined to reduce uncertainty in the estimated parameters solutions that are employed to construct the *Posterior probability function* (Posterior PDF), normalized by marginal PDF or model evidence. With a large amount of data, the Posterior PDF becomes similar to likelihood; as can be seen on Bayesian theorem:

$$P(x|y_{obs}) = \frac{P(x) \cdot P(y_{obs}|x)}{P(y_{obs})}, \tag{3.21}$$

where $P(x)$ is the prior probability function, $P(y_{obs}|x)$ is the conditional probability likelihood function, $P(x|y_{obs})$ is the posterior probability function, $P(y_{obs})$ is the marginal probability distribution function for normalization.

Prior PDF. The prior information, which contains the uncertainties, is constructed by a significant amount of simulation with randomly generated input parameters within physical ranges.

Conditional probability likelihood function. Bootstrap filtering (BF) is an updating likelihood function that can be used as conditional PDF in the Bayes theorem. Meanwhile, there is a significant number of previous models simulations; it is well known as a direct

and straightforward implementation based on Monte-Carlo, as explained in Oladyshkin et al. (2013), the Bootstrap filtering expression composes the conditional PDF:

$$P(y_{obs}|x) = e^{(-0.5 \cdot L_1 \cdot E^{-1} \cdot L_1^T)}, \tag{3.22}$$

where E^{-1} is the inverse diagonal matrix that is filled with designated acceptance error ϵ. The $L(x)$ is the computed error, which can be a threshold between observed and computed data, also denominated the $L_1 - norm$. Then, $P(y_{obs}|x)$ is considered as a weight vector. After the weight normalization, the values higher than zero and lower than one stand for partial fulfillment of acceptance error; conversely, values below zero do not fulfill the criteria.

Posterior PDF. The uniformly randomized generated number r is compared with the weight or the conditional PDF value $P(y_{obs}|x)$. If the latter is larger than the former, the conditional PDF value evolutes to the posterior PDF value. The mentioned filter proceeds until evaluation of whole conditional PDF vector, and completion of Posterior PDF vector.

4 Hydro-morphological Model Calibration

An automated calibration routine, which is applied on the hydro-morphological numerical model of the *Bend Channel*, minimizes the *Objective Function* (OF) with different optimizations methods and algorithms to analyze and compare their performances and their efficiencies. They depend on gradient, non-gradient, genetic and stochastic approaches. Continuously, the calibration process computes the implementations of the hydrodynamic module TELEMAC 2D, coupled with the sediment transport and bed evolution module SISYPHE as many times as each method requires. The simulated variables along with either the real or synthetic measurements are used to construct the *Objective Function* (OF).

The present chapter entails the definition of *Euclidean norm*, the analysis of the parameter's landscapes in order to understand the model; moreover, the assessment of the deterministic and the stochastic optimization methods; firstly, with an inserted synthetic data in OF, and secondly, with real data in OF. Also other possibilities of objective functions are investigated.

4.1 Euclidean Norm as Objective Function

In our study, a customized Euclidean Norm $L_2 - norm$ is used as the *Objective Function* to be minimized by the optimization methods through the python optimization library. The norm is defined in equation:

$$\phi(\mathbf{x^*}) = 0.50\sqrt{\sum_{i=1}^{11}(f_x^{90} - y_{synth}^{90})^2} + 0.50\sqrt{\sum_{i=1}^{11}(f_x^{180} - y_{synth}^{180})^2}, \tag{4.1}$$

where f_x^{90} plus f_x^{180} are simulated data and y_{synth}^{90} plus y_{synth}^{180} are the synthetic data that replaces the real measurement data, computed at two different cross section of bend channel Yen and Lee (1995) model, one at 90° and other at 180° with eleven points each, identical to the real measurement data. The chosen output variable to perform in the objective function is the *Bed morphological evolution* due to the existence of real measurement data of the same type. Therefore, a later comparison between objective function with real measurement data and with synthetic data is possible.

Synthetic measurements data. In a classical inverse problem, the parameters, which produce the real measurement data and approximately mimic the system behavior, are calculated. However, there are three reasons to replace the real measurement data by the synthetic measurement data, also denominated *The twins-Experiment*. First, in a highly ill-posed problem, such as the experimental model example presented in Yen and Lee (1995), the real measurement data may not objectively judge the effectivity of the algorithm performance. Thus, different optimization algorithms find their local minima. Second, the real measurement data can carry human and tool errors. Third, the possible additional numerical behavior of the numerical model can be incorporated; which is not included in the real model, leading to an unachievable optimization.

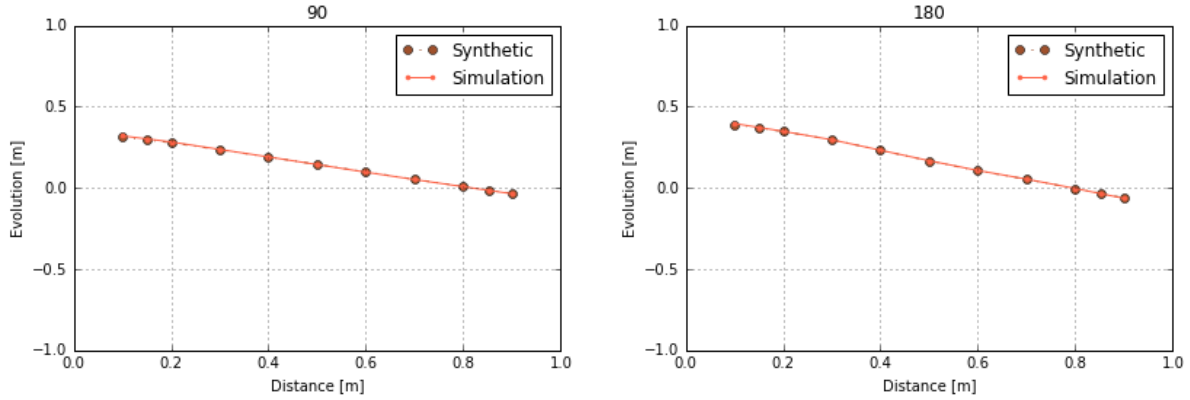
Therefore, to improve the acceptance of reliability on the computational model program and to analyze the optimization methods from global minimum searcher ability point of view, the generation of synthetic measurement data within a physically bounded parameter set is necessary to replace the real measurement data in the *objective function*. This synthetic measurement data is considered as the fixed term in the objective function.

The synthetic objective function can be done by simulating the output data from a chosen parameter set, denominated the synthetic parameters or the global solution, as can be seen in the synthetic values in table 4.1; this creates a unique and existing solution. Knowing in advance the global solution is possible to compare the methods not only with the minimization of the *objective function* capacity but also with the discrepancies between the proposed parameter solution and the synthetic parameter set.

Table 4.1: Hydro-and- morphological parameters for calibration

ID	Name	Synthetic	Initial	Low Bound	Upper Bound	Unit
P_0	Sediment diameters	0.0008	0.0010	0.0003	0.0015	m
P_1	Friction coefficient	0.0012	0.0020	0.0003	0.005	m
P_2	Roughness coefficient of boundaries	0.0008	0.0010	0.0003	0.001	m
P_3	Ratio between skin friction and mean diameter	3.50	3.00	1	5	-
P_4	Non cohesive bed porosity	0.40	0.35	0.25	0.5	-
P_5	Parameter for deviation	0.80	2.00	0.2	2.5	-
P_6	Beta	1.00	1.30	1	2	-
P_7	Secondary currents alpha coefficient	0.85	1.00	0.75	1	-

The synthetic measurements can be any of the TELEMAC2D-SYSIPHE output variables, obtained at each desired position after a triangulation between the nodes in the mesh, specifically a fixed *Bed Evolution* at 90° and another at 180°. Soft profile slopes are shown on the figure 4.1. Conversely, as it is presented by the real measured data in figure 2.3.



(a) Synthetic of evolution at 90 °

(b) Synthetic evolution at 180 °

Figure 4.1: Synthetic Bed Morphological Evolution profiles

Simulated variable data. In the same way, as the generation of the fixed synthetic measurement data, the simulated variable data are interactively computed with the updating parameter sets proposed by the optimization methods. Both, the synthetic and the simulated measurement data are used to calculate the *objective function*. The first simulated data to be utilized is the data which is generated by the initial parameter set as shown in table 4.1. The delivered simulated data set must be of the same variable type, located at the same spatial and temporal position as the synthetic measurement data set, as mentioned: the Bed Morphological Evolution at 90 ° and 180 ° after 18,000 seconds.

4.2 Ill-posed Problem Assessment: *Numerical Parameter Behavior*

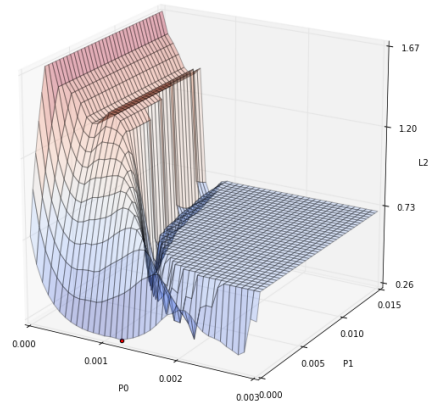
The mathematical 3D landscapes based on the multiple residual function evaluations are calculated with an alternated pair of parameters grid within a reasonable physical interval while the other parameters are constant values. This is done to observe the *Bend Channel* model problem convexity type and detect the parameter combinations problems. The *Euclidean Norm* also is the objective function used to creates the mathematical landscapes:

$$\| f_e \| = (\sum_{i=1}^{n} x_i^2)^{1/2}, \tag{4.2}$$

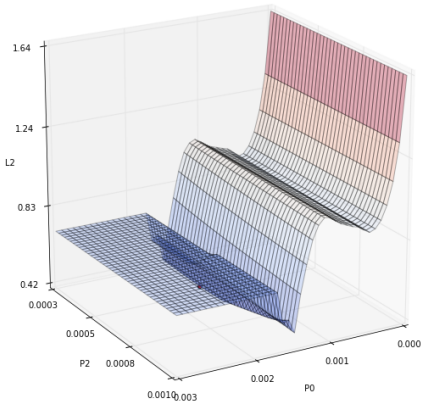
where x_e is the different between a fixed synthetic and simulated measurement data at each grid point, and f_e is the least-squared-error function or the *Euclidean Norm* value.

The grids contain 40 x 40 points with the $L_2 - norm$ values; the lowest value of each landscape is represented as a red point. The *Brute Force* method permute the grid values. From eight different parameters, 28 landscape combinations of pair parameters are possible. The characteristic sediment diameter d_{ch} is the most sensitive parameter, its behavior in conjunction with each other parameters is shown in figure 4.2. Additionally, the other parameter combinations are also presented in figures 4.3, 4.5, 4.6, 4.7, 4.8 and 4.9.

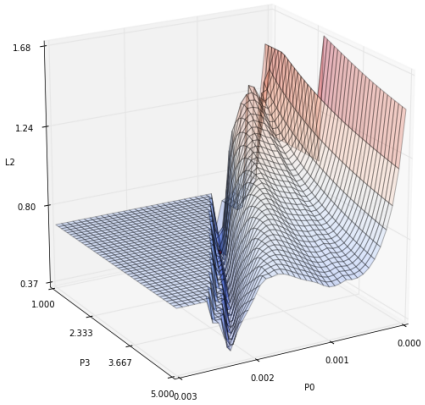
Characteristic sediment diameter (d_{ch})(P_0). This parameter presents two curvy channels in combination with the Bed friction coefficient (P_1). Two straight channels are parallel with the Ratio between Skin Friction and Mean diameter (P_3). Also, the convexity with the Parameter for Deviation (P_5) shows a light local. Two parallel linear channels are in only one direction with the Boundary friction coefficient (P_2). The Non-cohesive bed porosity (P_4) and the Beta (P_6) do not change for any variation of the Characteristic sediment diameter P_0. Plateaus are presented in all combinations. Furthermore, in conjunction with the Secondary currents alpha coefficient (P_7) exists one soft slope linear channel and another irregular stronger slope. See figure 4.2.



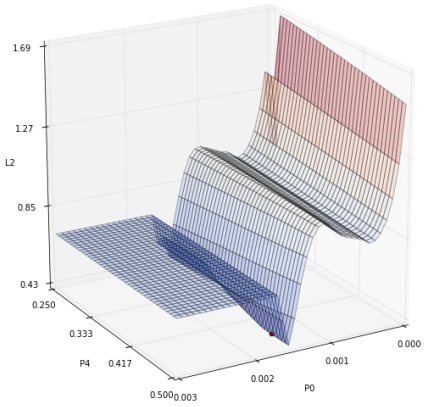
(a) Characteristic sediment diameter (P_0) and Friction Coefficient (P_1)



(b) Characteristic sediment diameter (P_0) and Roughness Friction Boundaries (P_2)



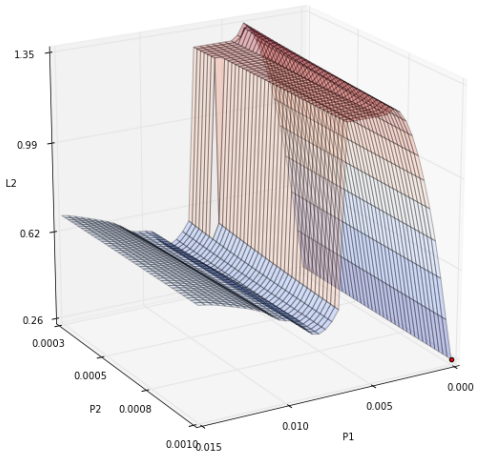
(c) Sediment Diameter and Radio between Skin Friction (P_3) and Characteristic sediment diameter (P_0)



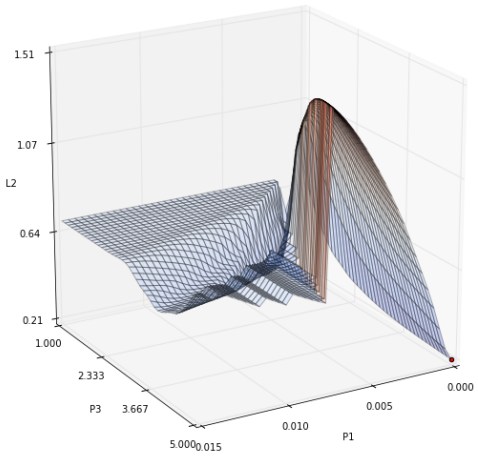
(d) Characteristic sediment diameter (P_0) and Non cohesive bed porosity (P_4)

Figure 4.2: Mathematical landscapes of the Characteristic Sediment Diameter (P_0)

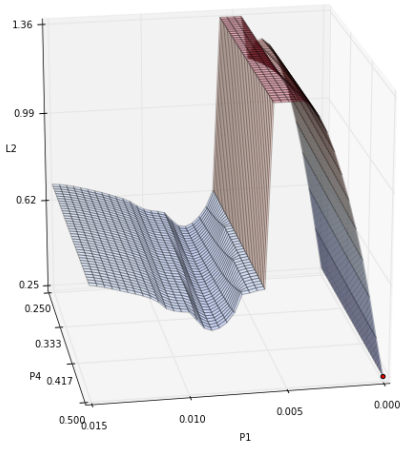
Friction coefficient (K_s)(P_1). This friction on the bed parameter presents a light linear proportionality with the Non-cohesive bed porosity (P_4) and the Beta (P_6), meaning a almost non-change on the evaluated function over different Bed friction coefficient (P_1). A convexity is visible in combination with the Parameter for Deviation (P_5). Also, a mild slope channel in both directions is presented with the Ratio between Skin Friction and Mean diameter (P_3) and peak irregularities with the Secondary currents alpha coefficient (P_7). See figure 4.3.



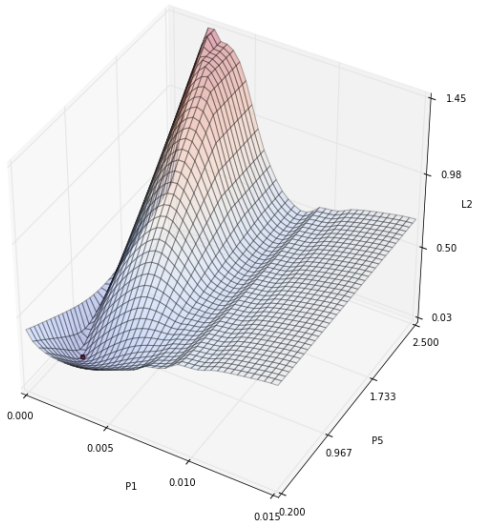
(a) Friction Coefficient (P_1) and Roughness Friction of Boundaries (P_2)



(b) Friction Coefficient (P_1) and Radio between Skin Friction and Mean Diameter (P_3)



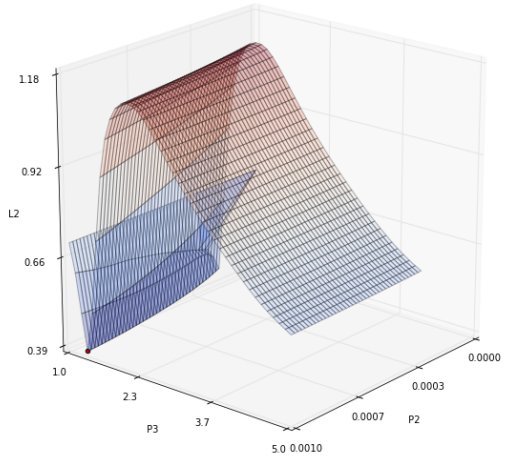
(c) Friction Coefficient (P_1) and Non cohesive bed porosity (P_4)



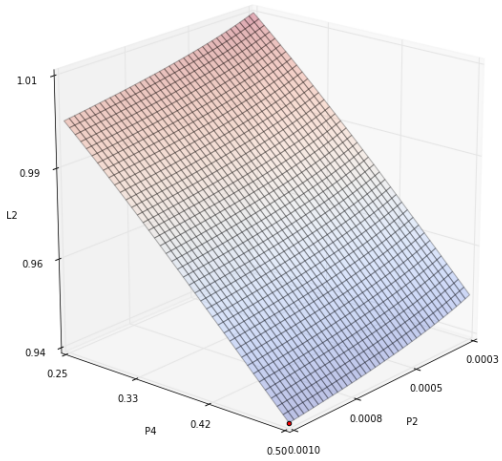
(d) Friction Coefficient (P_1) and Parameter for Deviation (P_5)

Figure 4.3: Mathematical landscapes of the Bed Friction Coefficient (P_1)

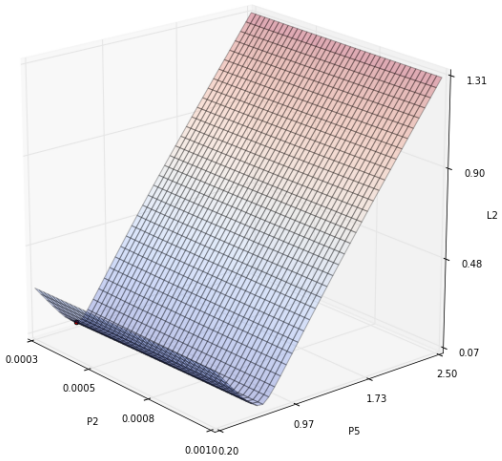
Lateral Roughness Friction of Boundaries (K_s)(P_2). This friction on lateral boundaries has little proportionality with the other parameters. Meanwhile, the other parameters vary in different ways, the lateral roughness friction of boundaries P_2 maintains a soft linear change.



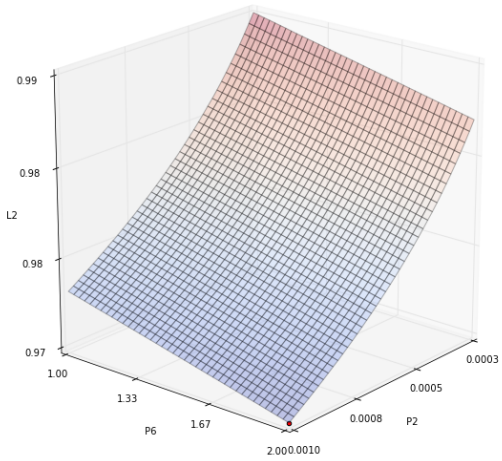
(a) Roughness Friction of Boundaries (P_2) and Sediment Diameter and Radio between Skin Friction (P_3)



(b) Roughness Friction of Boundaries (P_2) and Non cohesive bed porosity(P_4)



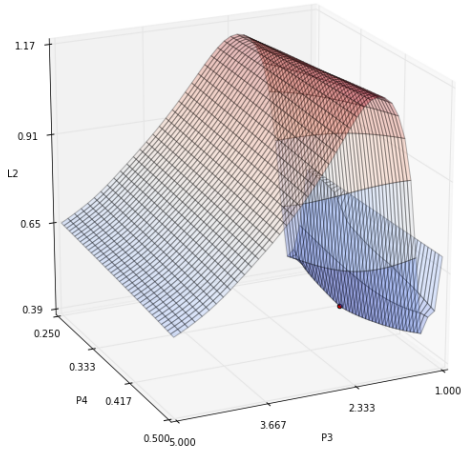
(c) Roughness Friction of Boundaries (P_2) and Parameter for Deviation (P_5)



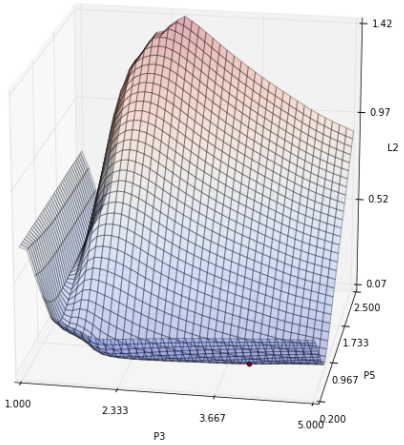
(d) Roughness Friction of Boundaries (P_2) and Beta (P_6)

Figure 4.4: Mathematical landscapes of the Roughness Friction of Boundaries (P_2)

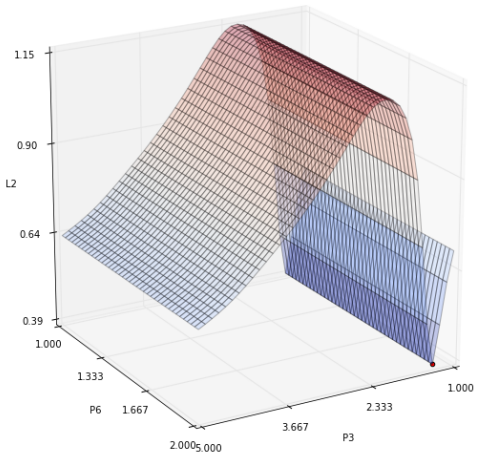
Ratio between Skin Friction and Mean Diameter (μ)(P_3). It presents light proportionality with the Non-cohesive bed porosity (P_4) and the Secondary currents alpha coefficient (P_7). Therefore, a channel in a single direction with a large maximum is visible. Also, almost non-proportionality with Beta (P_6) is shown, meaning there is a non-variation of function by changing this parameter, so there is a narrow channel in a single direction. Convexity only exists with the Parameter of Deviation (P_5). See figure 4.5.



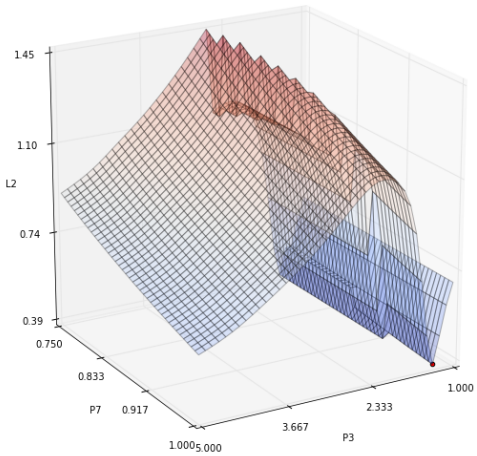
(a) Radio between Skin Friction and Mean Diameter (P_3) and Non-cohesive bed porosity (P_4)



(b) Radio between Skin Friction and Mean Diameter (P_3) and Parameter for Deviation (P_5)



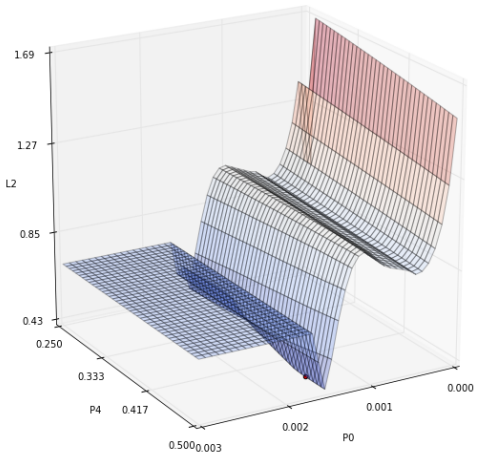
(c) Radio between Skin Friction and Mean Diameter (P_3) and Beta (P_6)



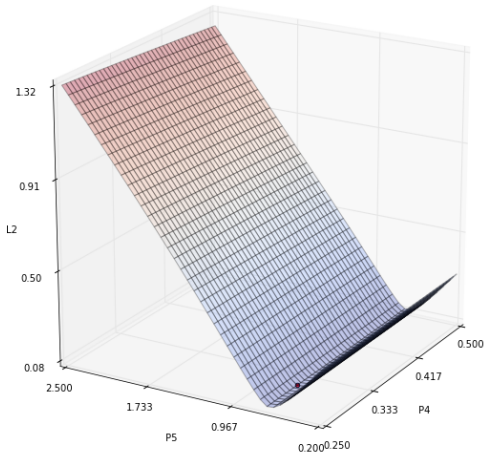
(d) Radio between Skin Friction and Mean Diameter (P_3) and Secondary currents alpha coefficient (P_7)

Figure 4.5: Mathematical landscapes of the Ratio between Skin Friction and Mean Diameter (P_3)

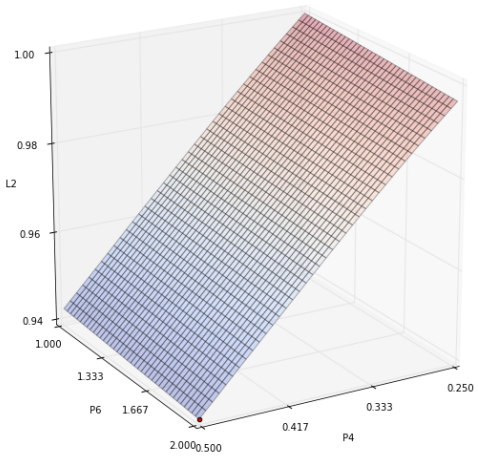
Non-cohesive bed porosity (n)(P_4). This parameter presents a linear functionality in combination with the Parameter for Deviation (P_5), with the characteristic sediment diameter (P_0), with the Bed friction coefficient (P_1), and with the Ratio between Skin Friction and Mean diameter (P_3). Therefore, the derivative are possible in a single direction. Also, some combinations present a parallel maximum to the minimum. The combinations with the Beta (P_6) and the Secondary currents alpha coefficient (P_7) show linear proportionality and non-convexities, as a result the first derivative is a constant. See figure 4.6.



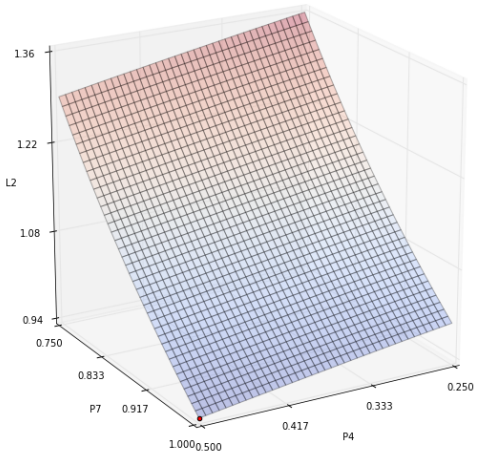
(a) Characteristic sediment diameter (P_0) and Non-cohesive bed porosity (P_4)



(b) Non-cohesive bed porosity (P_4) and Parameter for Deviation (P_5)



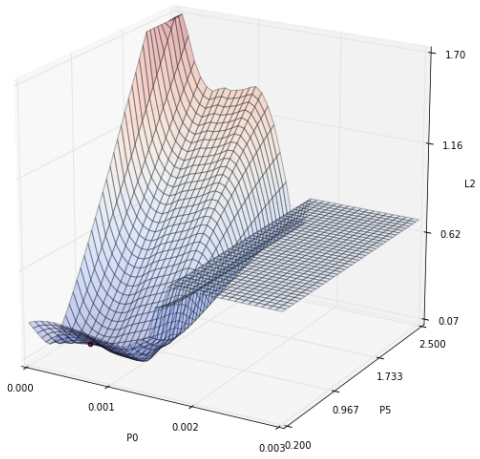
(c) Non-cohesive bed porosity (P_4) and Beta (P_6)



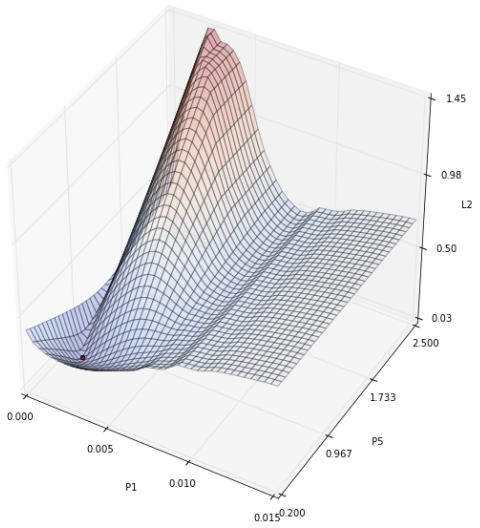
(d) Non-cohesive bed porosity (P_4) and Secondary currents alpha coefficient (P_7)

Figure 4.6: Mathematical landscapes of the Non cohesive bed porosity (P_4)

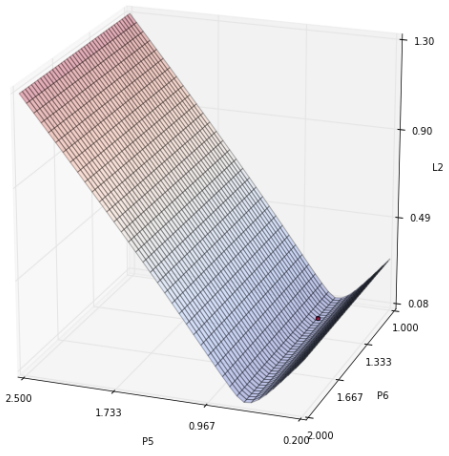
Parameter for Deviation (β_2)(P_5). This parameter in combination with the Non-cohesive bed porosity (P_4), the Beta (P_6) and the Secondary currents alpha coefficient (P_7), it presents a high linear channel. Therefore, only the first derivatives are possible, meaning non-significant-changes in the Euclidean Norm for the companion parameter of the Parameter for Deviation (P_5). On the other hand, β_2 shares convexities with the Characteristic sediment diameter (P_0), the Bed friction coefficient (P_1) and the Ratio between Skin Friction and Mean diameter (P_3), thus, the second derivative is possible, and a local solution is visible. See figure 4.7.



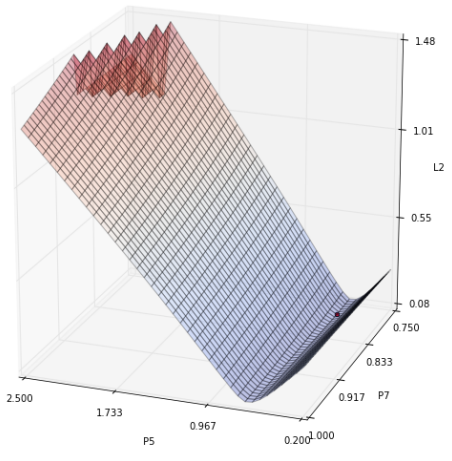
(a) Characteristic sediment diameter (P_0) and Parameter for Deviation (P_5)



(b) Bed friction coefficient (P_1) and Parameter for Deviation (P_5)



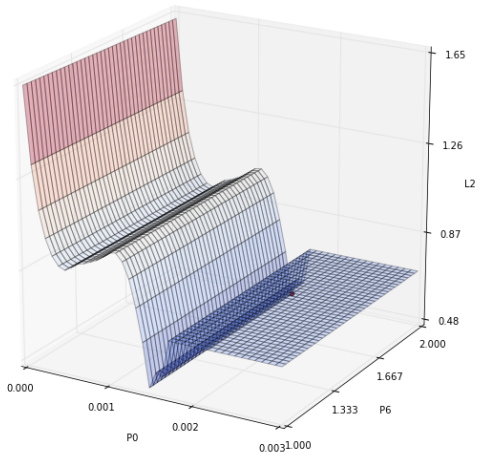
(c) Parameter for Deviation (P_5) and Beta (P_6)



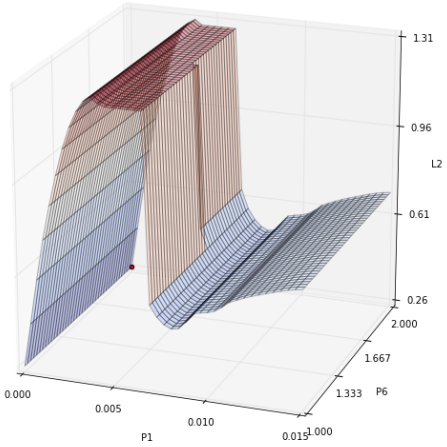
(d) Parameter for Deviation (P_5) and Secondary currents alpha coefficient (P_7)

Figure 4.7: Mathematical landscapes of the Parameter for Deviation (P_5)

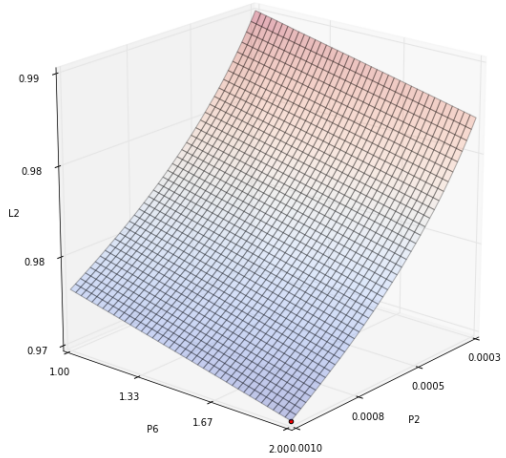
Beta (β)(P_6). The slope effect corrector for magnitude of the sediment transport rate presents some linear decedent planes, meaning that derivative is a scalar value, in combination with the Non-cohesive bed porosity (P_4) and the Secondary currents alpha coefficient (P_7), meaning that change on the Beta does not support significantly to the optimization. The combinations with the Characteristic sediment diameter (P_0), the Bed friction coefficient (P_1), the Ratio between Skin Friction and Mean diameter (P_3), and the Parameter for Deviation (P_5) show linear channels where it is also visible that the same solution is kept along the Beta axis. See figure 4.8.



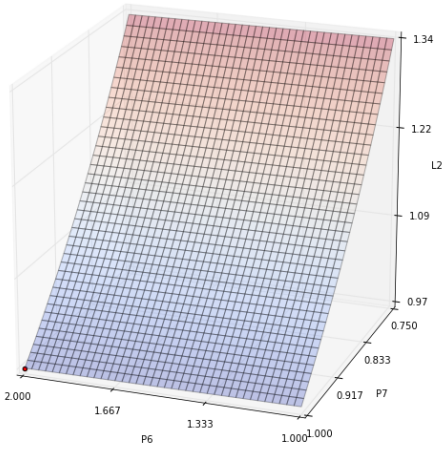
(a) Characteristic sediment diameter (P_0) and Beta (P_6)



(b) Bed friction coefficient (P_1) and Beta (P_6)



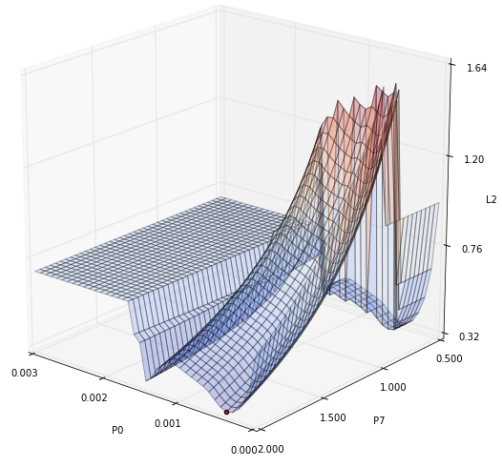
(c) Roughness Friction of Boundaries (P_2) and Beta (P_6)



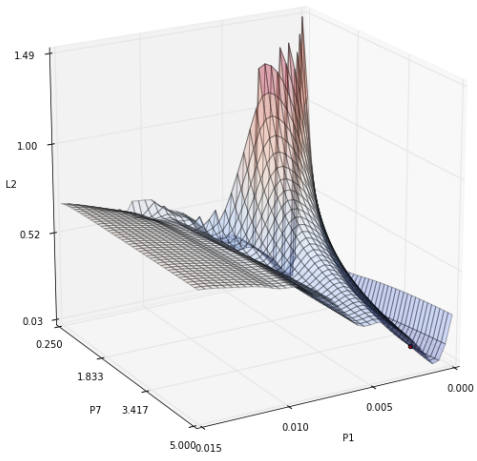
(d) Beta (P_6) and Secondary currents alpha coefficient (P_7)

Figure 4.8: Mathematical landscapes of the Beta (P_6)

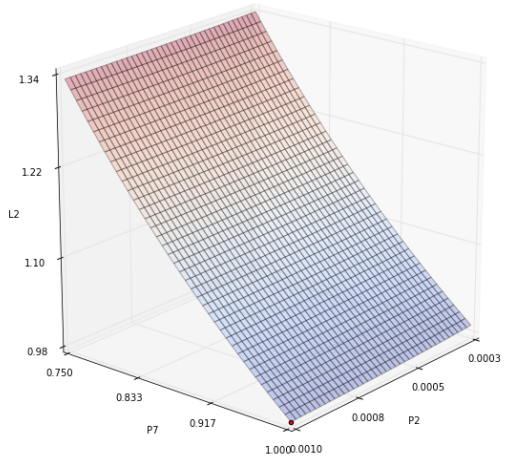
Secondary currents alpha coefficient (α')(P_7). It presents irregular channelization with the Characteristic sediment diameter (P_0), the Bed friction coefficient (P_1), and the Ratio between Skin Friction and Mean diameter (P_3). In combinations with the Non-cohesive bed porosity (P_4) and the Beta (P_6) is presented a lineal decent plane which the first derivative is scalar value. Also, it presents a perfect linear channel with some irregularities combined with the Parameter for Deviation (P_5). See figure 4.9.



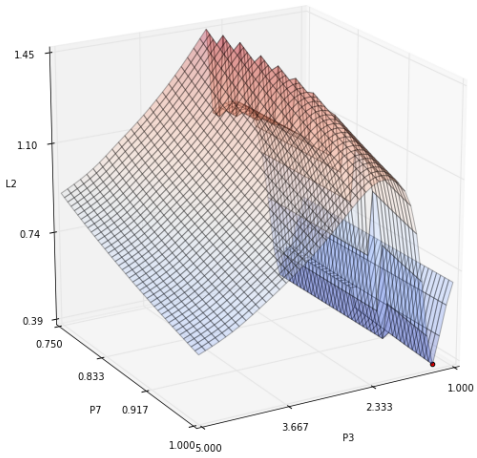
(a) Characteristic sediment diameter (P_0) and Secondary currents alpha coefficient (P_7)



(b) Bed friction coefficient (P_1) and Secondary currents alpha coefficient (P_7)



(c) Roughness Friction Boundaries (P_2) and Secondary currents alpha coefficient (P_7)



(d) Radio between Skin Friction (P_3) and Secondary currents alpha coefficient (P_7)

Figure 4.9: Mathematical landscapes of the Secondary currents alpha coefficient (P_7)

In general, the combinations of the mathematical landscapes of the hydro-morphological parameters present characteristics, such as the non-convexity shape, the either linear or

curvy channelization, the non-continuous downhill slopes due to maximum, the irregularities, the inclined planes, and the plateaus. It is indicated that the numerical model of the *Bend Channel* is an ill-posed problem.

The non-full convexity, presented in the most landscapes, implies that the second derivative used may deliver a error; since there is a derivable in only one direction. Previous brings singularity in the inversion of any matrix, showing multiple local solutions.

The total-one-direction-channelizations indicate the almost non-proportionality of the pair parameters; thus, the change of one parameter almost does not affect the other. The curvy-channelizations suggest that multiple solutions can be proposed by different combinations of the same pair parameters.

The non-continue downhill due to a maximum in the way of the global minimum indicates the existence of multiple local minima. It can be a challenge for gradient-based optimization methods with the line search strategy.

The irregularities are proper of either the numerical model or the combination of chosen equations. Some combinations of the hydro-morphological parameters can not be performed mathematically .

The inclined planes between the pair or parameters indicate their linear directed and inverse proportionality.

Unfortunately, the plateaus stand for numerical conflicts in TELEMAC2D when there are different combinations of parameters, even though, the parameters's values are within the bounded physical intervals, the combination is not runnable; continuously dropping the previous correct result, which is saved in the memory, as a new result and repeating until a runnable physical parameter set is calculated. Last, it is dangerous for methods based on gradients, since the derivative on the plateau surface is zero; therefore, the optimization methods *misconceives the detection of a local minimum.*

4.3 Assessment of Deterministic Algorithms

In our study, the evaluation of the optimization methods comprises four criteria, the computational feasibility, the minimization of the objective function capacity, the achievement of low discrepancies with the global solution, and the functional physical constraints. The computational criterion is only the first step to identify the most suitable methods. After finding the most robust, computational feasible and acceptable global minima searchers; the most appropriate optimization methods' probability of certainties through statistics interpretations are analyzed. Finally, the minimization results can be influenced by the different weighted measurements points; therefore, several combinations of the weights and other variable types are introduced in a modified objective function.

4.3.1 Optimization Methods' performance

The main characteristics of the different optimization methods are compared relatively, such as the computational performance, the robustness, the velocity of convergence, the accuracy, and the global minimum search ability. The calibration of the hydro-morphological *Bend Channel* model is implemented through the provided optimization methods and algorithms included in the optimization library of Python. All the optimization methods begin with the same conditions, such as the initial parameter set, see 4.1, with the initial $L_2 - norm$ of 0.976, the tolerance of (0.001) and twenty iterations. For the actual ill-problem, the mentioned tolerance is hard to reach; thus, the optimization methods that can reduce L_2 to 90 %, are considered converged.

Some gradient-based methods requirements are a Jacobian and a Hessian matrix as callable functions, which are programmed according to each parameter necessity. In the way, that delta parameter is a fraction multiple of six of bounds range $\Delta X_i = (U_b - L_b)/6 * a$, being $a = 8$ for the most sensitive parameter and $a = 1$ for the less. The bounds intervals are considered within a realistic physical meaning in nature, only they are used for methods that demand them. Another condition is the constraint functions, essential for the parameters that keep a relationship in between, such as the Characteristic sediment diameter (P_0) and the Bed friction coefficient (P_1), where $P_0 \leq P_1 \leq 5P_0$.

Hereafter, the method's performances are mainly linked to the mentioned particular numerical model, meaning other results of the methods can perform with another type of the numerical model.

Nelder-Mead. The simplex method, which does not require the use of the derivative, can be easily overestimated. It is observed that after some triangle iterations, the method can be just trapped, since it does not find a better solution than the actual triangle function evaluations, leading to an unnecessary computation misuse. A non-bounded and a non-constrained algorithm likely carries a non-meaningful proposed solution. However, after a long computation time, the objective function's reduction is 88 percent, compared with the initial objective function value. The constraint conditions present fulfillments, and the total discrepancy percent lies on 25 percent, at 20 feasible computational iterations.

Polak and Ribièri. This gradient-based method sounds more attractive than previous. Unfortunately, for ill-posed problems, a decent line searcher goes directly to the first local minimum, depending on the initial parameter set. For every iteration, some non-physical parameter sets can be proposed (negative or out of range) due to the non-bounded conditions. Non-robust enough with a global parameter discrepancy of 57% but relative fast compared with the others, it minimizes an 85% of the L_2 at 16 iterations under this specific conditions.

L-BFGS-B. Also, a gradient-based method, which uses the Jacobian matrix, promises more due to the incorporation of bounds for each parameter solution and a maximum number of the line search. It notably reduces the L_2 to 78% from the initial value at six iterations. Nevertheless, it is also sensitive to a local minimum, being not robust enough, and non-constrained functional due to the non-physical relationship between the parameters. The global parameter discrepancy rises to 31.51%.

BFGS. As well, the non-constrained, non-bounded and gradient-based BFGS method fails with the robustness, the minimization capacity, and the performance. Again, it proposes non-physical parameters set after each iteration. Distinctly, it is not considered suitable the weak reduction of 28 % of L_2, due to the total collapse of the model calibration and the negative proposed Characteristic sediment diameter (P_0). As a result of the misconceiving detection of a local minimum, after the line search arrives on a plateau in the landscape, this method proposes a wrong solution at one iteration.

Newton-CG. It is a conjugate gradient-based method, where the Jacobian and the Hessian matrix are optional; in the case of their usage, they must be callable functions,

because of the specific requirements of each delta parameters. A non-bounded implementation produces a non-meaningful solution. Also, the misconception of a local minimum detection, when the line searcher finds a plateau, reduces the possible potential of this optimization method for this ill-posed problem. So that, in the present specific case, it stops at two iterations, reducing a moderate amount of 21.97% of the L_2.

TNC. The truncated conjugated Newton, in mathematical optimization terms, is synonymous of the Newton-CG method, in the optimization library of Python, differs only of the bounds' limits, since the TNC method includes boundaries intervals for each of the parameters, being more persuasive. It reduces a significant 83.42% of the norm at 20 iterations with a discrepancy of 28%; however, since it is also a gradient based method, it is highly sensible to search for a local minimum, meaning a relative low robustness.

Trust-ncg. The gradient-based method based on the trust-region searcher requires mandatory the Hessian matrix as a callable function, the radius of the search must be provided; it is non-meaningful to share the same radius of search when the parameters have different dimensionality. It minimizes the L_2 to 75.76% at 20 iterations with a 51% of discrepancy from the global minimum. Finally, it is non-robust enough and high computational expensive due to the second derivative evaluations of every parameter per iteration.

SLSQP. It is a highly recommendable, bounded, constrained and gradient-based method implementation for ill-posed problems. Tremendous robust and its satisfying performance reduces a 95% of L_2 at five iterations. The discrepancy of the global minimum gets a 19 %, the lowest of all evaluated methods, see table 4.3. To rely on the robustness capacity of the indicated method, the uncertainty evaluation is developed in chapter 4.3.3. It brings the possibility to introduce the multiple layers of sediments, which are fractionally related, by constrained functions. So far, only the constraint function between d_{ch} and K_s is used, so that latter one is equal or five times larger than former.

Least-Squares. This method is based on the Levenberg-Marquardt approach; also it is gradient-based. Even though it is a non-bounded method, it highly reduces the L_2 with apparent physical meaningful values. After the first iterations, the method proposes too large or negative parameter values, out of meaning. Afterward, it smartly finds, even

without bounds interval specifications, suitable parameter values that describe a local minimum with a L_2 reduction of 97.45% of the initial norm at 13 iterations but with significant discrepancies to the global minimum, the highest of all evaluated methods, see table 4.3. The beta parameter value is around 30 times the global one and out of physical bounds. Also, the constraints function does not fulfill that the Bed friction coefficient (P_1) must be larger or equal than the Characteristic sediment diameter (P_0).

The chosen damping factor recommended by the optimization library is 10. Also, it exists the possibility of weighted diagonal according to the parameter requirements, but it is not used.

Root Levenberg-Marquardt. The result is almost identical to the previously described implementation *Least-Squares* in all the positive and the negatives aspects, only it improves the iteration number to 10, leading with nearly identical solution parameters and performances.

Basin-Hopping. The genetic global minimum searcher is a non-gradient based and non-directly bounded. It is non-constrained, but an acceptance test is designed to reject solution out of bounds and limit intervals. The step sizes are customized according to the parameter sensitivity, adding to a uniformly random number the previous step size from the fractioned range of the bounds. The step size is equal to $Stepsize+ = uniform \quad random(a*L_i, a*U_i)$, where $a = 0.25$ for the Characteristic sediment diameter parameter, the most sensitive, and $a = 0.50$ for all others. The temperature factor of 0.1 is used, as well, for its simplicity "Nelder-Mead" is implemented as the local searcher method for refinement. The highest minimization capacity, with a reduction of the L_2 to 99.70%, good robustness, second best place of discrepancies with the global minimum with 27% at 20 iterations, but large-rejected-function calls that make computational expensive. It is mandatory for the method to combine itself with a local minimum searcher method.

Differential Evolution. The genetic algorithm is designed as a global searcher, non-gradient based and bounded method. The chosen strategy for mutation is "best1in" as described in 3.4.2. The mutation weighted factor is uniformly randomized taken from the interval (0.5, 1.5). The population for every parameter is 10. The recombination factor is 0.7. The initial parameter set is chosen with the "Latin hypercube" strategy that

tries to cover more space as possible in the limited conditions, differing from the other methods by the different initial parameter set. The possibility of local refinement with any other methods, which is included in the optimization library, offers a pair optimizers, global and then local; however, it is not used, since it is not mandatory. This method minimizes the *objective function* to a high value of 92% at 20 iterations with a significant amount of rejected function calls at expenses of the computational cost. Nevertheless, the discrepancies of the global searcher arise an 83%. The calculations show that the most difficult parameter, to calibrate by this method, is the Bed friction coefficient (P_1) and the Boundary friction coefficient (P_2), adding the high weight to the discrepancies, contrary to another parameter that sounds more global calibrated. In any case, it achieves the constraint requirement.

Brute Force. It is a global searcher method and computationally expensive, forming an 8-dimensional calculation grid of three points per axis with the 336 permutations as iterations of the parameter values within the bounds ranges. It does not require the initial parameter set, due to the complete parameter landscape evaluation to find the global minimum. Irrelevantly, it calculates same parameters set with a different order to formed a dense matrix, since the order matters. High computational expensive without economizing resources presents a reduction of L_2 to 85% with a discrepancy of the global minimum of 57%, and a non-achievement of constraints demand. A possible refinement is not implemented for mentioned model.

In summary, as the computational performance condition, a maximal of 20 iterations is the first criterion for the optimization method's evaluation. Not all optimization methods converge under this conditions. Indeed, each optimization method differs with the number of the objective functions calls and evaluations per iteration. For instance, the gradient-based methods require the call of the 8 (size of parameters vector) objective functions to form the Jacobian matrix for each iteration, and the genetic methods require several objective functions calls that may be not accepted by their evaluation test. The table 4.2 presents the computational efficiency comparison among the optimization methods.

The table 4.2 presents that the number of iterations is neither proportional to objective functions and the Jacobian calls nor the evaluations of them. On the other hand, the time for a single call of the objective function depends on the utilized compiler; thus, one simulation in TELEMAC2D-SISYPHE lasts between 10 and 30 seconds.

Table 4.2: Computational performance of the optimization methods

Method	Func. Calls	Jac.calls	Func.Eval.	Jac.Eval.	Iteration	Time (s)
Nelder Mead	34	-	20	-	20	975
Polak and Ribièri	161	16	28	16	16	1811
L-BFGS-B	120	12	12	12	9	3797
BFGS	139	13	13	13	1	1139
Newton CG	96	9	15	2	2	2731
TNC	210	21	21	20	20	4877
SLSQP	65	20	20	5	5	1816
Trust-ncg	2591	10	21	10	20	59513
Basin Hopping	579	-	20	-	20	16560
LeastQ	136	13	136	13	13	3700
Root LM	109	10	109	10	10	2544
Differential Evolution	664	-	20	-	20	14527
Brute force	6562	-	6562	-	336	176659

From the performance point of view, the optimization methods that achieve the chosen acceptable L_2 of 0.1 under 20 iterations are the SLSQP (5), the Root LM (10), and the LeastQ (13) methods. Even though the Basin Hopping and the Differential Evolution methods fulfill the tolerance characteristics, the objective functions calls are much larger than the evaluated objective functions, reducing their attractiveness due to their computational time requirements.

4.3.2 Optimization Methods' validations

As mentioned, the same initial parameter set on each method are applied. However, there are two exceptions, the first is the *Brute force* method and the second is the *Differential Evolution*, both of them adopt particular initial parameter sets. The remaining three criteria for the optimization method's evaluation to discuss are the minimization capacity of the objective function, the global search abilities and the fulfillment of the constraint functions.

The minimization capacity of the objective function by each optimization method is presented in the table 4.4; also, the proposed parameter solutions, the minimized objective function values and the reductions in percent of L_2 are listed. A strong convergence can be perceived when the minimized objective function $\phi(x^*)$ approaches to zero or in this case a chosen acceptance of 0.1, under 20 iterations, as seen in table 4.4. The Basin

Hopping (99.79%), the Root LM (97.47%), the LeastQ (97.45%), the SLSQP (93.03%), and the Differential Evolution (92.92%) are the most suitable optimization methods from local minimum searcher point of view, due to their converge fulfillment. It is important to repeat that in a highly ill-posed problem, any reached local minima also approaches to zero and the misconception of solution for the gradient-based methods when searcher lands on any landscape plateaus. For instance, the BFGS and the Newton-CG methods stop at one and two iterations after finding a misconceived local minimum (plateaus landscape where all derivative are zero).

The global search ability of the method is a criterion that can be evaluated only with the existence of synthetic measurement data, so it is conceivable to determine which method fulfills the global solution. In nature, it is not possible to detect these discrepancies. The suggested parameter solutions differ per method, clarifying the existence of the multiple local minima; this statement is only applicable to the converging optimization methods. Therefore, not only a fast converge, neither a great minimization of the objective function are sole criteria of the optimal method to calibrate a hydro-morphological numerical model in a physical meaning, but it also makes the reduction of discrepancies.

The table 4.3 presents the relative error between the proposed solutions and the global synthetic solution. The SLSQP (19%), the Nelder-Mead (25%), and the Basin Hopping (27.25%) methods achieve the lowest discrepancy values in percent, obtained by the normalized sum of the parameters.

Table 4.3: Relative error with the global solution by optimization methods

Method	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	%
Nelder Mead	0.26	0.63	0.33	0.01	0.02	0.10	0.37	0.29	25.00
Polak and Ribièri	1.06	0.75	0.25	0.43	0.38	0.69	1.00	0.08	57.90
L-BFGS-B	0.88	0.40	0.25	0.00	0.25	0.26	0.31	0.18	31.51
BFGS	1.86	0.57	0.31	0.14	0.13	1.50	0.30	0.23	62.94
Newton CG	0.85	0.49	0.28	0.14	0.13	1.50	0.30	0.18	48.26
TNC	0.23	0.75	0.25	0.02	0.12	0.43	0.30	0.18	28.51
SLSQP	0.24	0.36	0.19	0.02	0.09	0.18	0.31	0.15	19.19
Trust-ncg	0.55	0.91	0.46	0.14	0.12	1.50	0.30	0.18	51.96
Basin Hopping	0.05	0.07	0.71	0.10	0.24	0.02	0.97	0.03	27.25
LeastQ	0.60	0.26	2.49	0.83	0.63	2.33	31.88	1.70	509.11
Root LM	0.59	0.26	2.49	0.83	0.63	2.33	31.88	1.71	508.91
Differential Evolution	0.05	4.48	0.45	0.17	0.25	0.39	0.79	0.12	83.73
Brute force	1.06	0.75	0.25	0.43	0.38	0.69	1.00	0.03	57.29

Table 4.4: Proposed solution parameter set by the optimization methods

Method	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	Norm	Reduction
Nelder Mead	1.00E-03	1.96E-03	1.07E-03	3.53	0.41	0.88	1.37	1.09	0.1180	87.91
Polak and Ribièri	1.65E-03	3.00E-04	1.00E-03	5.00	0.25	1.35	2.00	0.92	0.1429	85.36
L-BFGS-B	1.50E-03	7.20E-04	1.00E-03	3.49	0.50	0.59	1.31	1.00	0.1419	85.47
BFGS	-6.90E-04	5.20E-04	1.05E-03	3.00	0.35	2.00	1.30	1.04	0.6986	28.43
Newton CG	1.20E-04	6.10E-04	1.02E-03	3.00	0.35	2.00	1.30	1.00	0.7618	21.97
TNC	9.86E-04	3.00E-04	1.00E-03	3.58	0.35	1.14	1.30	1.00	0.1619	83.42
SLSQP	9.90E-04	1.63E-03	9.50E-04	3.58	0.43	0.66	1.31	0.98	0.0675	93.09
Trust-ncg	1.24E-03	1.07E-04	1.16E-03	3.00	0.35	2.00	1.30	1.00	0.2363	75.79
Basin Hopping	7.60E-04	1.12E-03	1.37E-03	3.16	0.50	0.81	1.97	0.87	0.0020	99.79
LeastQ	1.28E-03	8.90E-04	2.79E-03	6.40	0.15	2.67	32.88	2.30	0.0249	97.45
Root LM	1.27E-03	8.90E-04	2.79E-03	6.39	0.15	2.66	32.88	2.30	0.0247	97.47
Differential Evolution	7.60E-04	6.58E-03	4.40E-04	2.91	0.50	1.11	1.79	0.95	0.0691	92.92
Brute force	1.65E-03	3.00E-04	1.00E-03	5.00	0.25	1.35	2.00	0.88	0.1410	85.56

The last criterion to evaluate the optimization methods is the physical functional constraint of $d_{ch} < K_s < 5 * d_{ch}$ or $P_0 < P_1 < 5P_0$; in this case, the friction of bed is only due to the grain and not to the forms. The constraint results of each method are visible in table 4.5, this is valid only for the converging method. The SLSQP, the Basin-Hopping, and the Nelder-Mead methods manage mentioned basis. The other methods propose parameter values out of this physical constraint.

Table 4.5: Constraints function achievements by optimization methods

Method	P_1/P_0
Nelder Mead	1.95
Polak and Ribièri	0.18
L-BFGS-B	0.48
BFGS	-0.75
Newton CG	5.08
TNC	0.30
SLSQP	1.65
Trust-ncg	0.09
Basin Hopping	1.47
LeastQ	0.70
Root LM	0.70
Differential Evolution	8.66
Brute force	0.18

The evaluation of the optimization methods and algorithms, taking into account the four criteria, consists of the assignment of a punctuation according to the fulfillment of each criterion, as shown in table 4.6. The given punctuation are 30, 20 or 10, and they are accumulative.

- C1) The criterion for minimization assigns the highest scores for the lowest obtained norm, meaning a score of 30 points for a norm lower than 0.05, a score of 20 points for a norm lower than 0.05, and a score of 30 points for a norm lower than 0.05. As seen in table 4.4.
- C2) The criterion for the computational performance assigns the highest scores for the lowest number of iterations; thus, a score of 30 points for less than five iterations, a score of 20 points for less than ten iterations, and a score of 10 points for less than or equal twenty iterations. In case the reduction of the norm is less than 30%,

then this criterion takes zero value, such as it does the BFGS and the Newton-CG methods. As seen in table 4.2.

- C3) The criterion for the achievement of the low parameter discrepancy assigns the highest scores for the lowest total discrepancy percent; meaning a score of 30 points for less than 10%, a score of 20 points for less than 20%, and a score of 10 points for less than 30%. As seen in table 4.3.
- C4) The criterion of the constraint function assigns a score of 10 points to the solution within constrained conditions and a score of zero points for the ones outside, this is assigned if there is non-misconception of a local minimum. Only three methods fulfill this criterion, namely the Nelder-mead, the SLSQP, and the Basin-Hopping methods. As seen in table 4.5.

The highest possible punctuation is 100 and the lowest is 0, according to the evaluation, the optimization methods that obtain the highest scores are the SLSQP with 80 points, the Basin-Hopping with 60 points, and the root LM with 50 points. The optimization methods that fail due to local-minimum misconception receive automatically zero points in all criteria.

Table 4.6: Evaluation of the optimization methods

Method	C1	C2	C3	C4	Sum
Nelder Mead	10	10	10	10	40
Polak und Ribièri	10	10	0	0	20
L-BFGS-B	10	20	0	0	30
BFGS	0	0	0	0	0
Newton CG	0	0	0	0	0
TNC	0	10	10	0	20
SLSQP	20	30	20	10	80
Trust-ncg	0	10	0	0	10
Basin Hopping	30	10	10	10	60
LeastQ	30	10	0	0	40
Root LM	30	20	0	0	50
Differential Evolution	20	10	0	0	30
Brute force	10	0	0	0	10
Punctuation					
30	≤ 0.05	≤ 5	≤ 10	-	
20	≤ 0.1	≤ 10	≤ 20	-	
10	≤ 0.15	≤ 20	≤ 30	≤ 5	& ≥ 1

4.3.3 Uncertainty Analysis of the Optimization Methods

To truly investigate the robustness, the uncertainty analysis of the most suitable optimizations methods brings the probability to find the global solution under 20 iterations. The several optimizations of the objective function differ in the uniform randomly generated initial parameter sets. Each proposed solution is part of the statistic data set.

The statistic data is the relative error between the proposed and the global (synthetic) parameter which is represented in the Box Diagram per optimization method; namely, the Nelder-Mead, the Polak and Ribièri, the L-BFGS-B, the SLSQP, the Levenberg Marquardt and the Differential evolution methods. After that, the statistics expressions describe the behavior of the Probability density function (PDF) or distributed solutions. In such a case, the relative error is calculated by equation:

$$error = \frac{P_{solution} - P_{synth}}{P_{synth}}, \tag{4.3}$$

where $P_{solution}$ is the parameter solution proposed by the optimization methods which uncertainty is analyzed, P_{synth} is the synthetic parameters or global solution.

The Box Plot Diagrams present in quartiles the relative error as the statistic data, separating the extreme values. After that, some important statistic features of the proposed solutions are identified, such as the mean, the median, the standard deviation, the skewness and the kurtosis values.

The mean value shows the general misfit with the global solution, also taking into account the large and non-meaningful proposals. The zero value stands for the whole data set is concentrated around the global solution; on the contrary, other values stand for either the method's proposals are around other local or the solutions are highly distributed.

The median value shows the position of the center value, the influence of extreme values are same weighted as the other values; the zero's value stands for equally distributed proposals in both directions, either the extreme or close values. Any other value represents a significant concentration of proposals in one single direction.

The standard deviation shows the range of distributed solutions; small values mean that the proposals are similar, so the method finds the same solution in different optimizations, which means robustness. Contrary, large standard deviation values mean different found a local minimum.

The skewness shows the direction of distributed extreme values, the value of zero stands for reasonably concentrated solutions, similar to the Gauss distribution. Contrary, the large values represent extreme solutions much bigger (positive skewness) or much smaller (negative skewness) than the global solution.

The kurtosis shows weather the solutions are either spread or peaked. The spread solutions (negative kurtosis) stands for multiple local minima, and the peaked (positive kurtosis) stands for a single local minimum. It is important to mention that it could happen a high kurtosis value which data is not concentrated in the mean equal zero, this means that the method proposes intensively the same local minimum solution, ignoring the global.

Uncertainty analysis of Nelder-Mead method. The robustness of the most simple optimization method, *Nelder-Mead*, is considered for analysis, since its failure of converging might be a consequence of the numerical model problem type. Therefore, it is investigated the possibilities of improvements. The figure 4.10 shows the Box Diagram for 500 optimizations solutions by Nelder-Mead, and the table 4.7 shows the statistic concepts. The l_2 achieves mostly a non-desirable tolerance and a non-convergence under 20 iterations, only 29 percent present a norm lower than 0.1. Nevertheless, it is only the Bed friction coefficient (P_1), roughness coefficient parameter, which performs a large relative error for most of the optimization results. Previous is proper of the negative kurtosis, which means a flatten probability distribution function (PDF). Also, the mean value bigger than the median deliveries a positive lightly skewness, indicating that for mentioned optimization method, it is not easy to propose a desirable solution for this parameter (P_1).

All parameters that receive a negative kurtosis, excluding the Parameter for Deviation (P_5) and the Secondary currents alpha coefficient (P_7), present a flat and spread PDF for their solutions. The Ratio between Skin Friction and Mean diameter (P_3) approaches to the standard distribution due to its approximately zero value in the skewness. The largest skewness values for the Parameter for Deviation (P_5) brings a significant number of extreme proposed values. In conclusion, the easiest parameter to be calibrated by Nelder-Mead is the Secondary currents alpha coefficient (P_7); it is visible with the lowest standard deviation value, indicating the short range of its proposals.

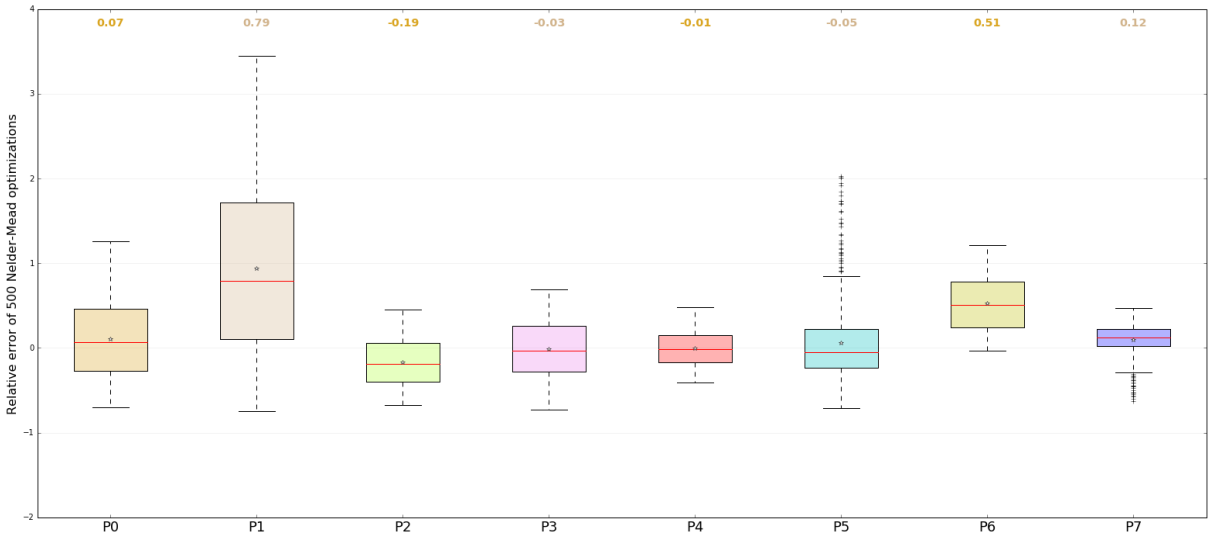


Figure 4.10: Box Graph for Nelder-Mead method

Table 4.7: Nelder-Mead uncertainty statistics of the relative error

Stats Nelder	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.103	0.936	-0.172	-0.014	-0.010	0.059	0.528	0.098
Median	0.068	0.788	-0.187	-0.029	-0.013	-0.050	0.507	0.125
Std.deviation	0.437	1.020	0.267	0.344	0.200	0.490	0.311	0.193
Skew	0.249	0.382	0.104	0.000	0.044	1.669	0.160	-1.175
Kurt	-1.013	-0.804	-1.136	-0.924	-0.924	3.420	-1.114	1.861

Uncertainty of Polak und Ribièri (CG). The 100 optimizations with Polak and Ribièri method brings a 23% of success with tolerance under 0.1. The Box Diagram, in figure 4.11, presents the large values for the standard deviation, the skewness, and the kurtosis in three parameters. They are the characteristic sediment diameter (P_0), the Bed friction coefficient (P_1) and the Roughness Friction of Boundaries(P_2), meaning a lot of extreme proposal values, but still concentrated solutions due to the peaked PDF; this demonstrates the existence of multiple local minima. The other parameters present a mean and a median value around zero, a small standard deviation and skewness, and a negative kurtosis, which means the solutions are concentrated in a short range of values and a few extreme values but spread PDF over the short range. See table 4.8.

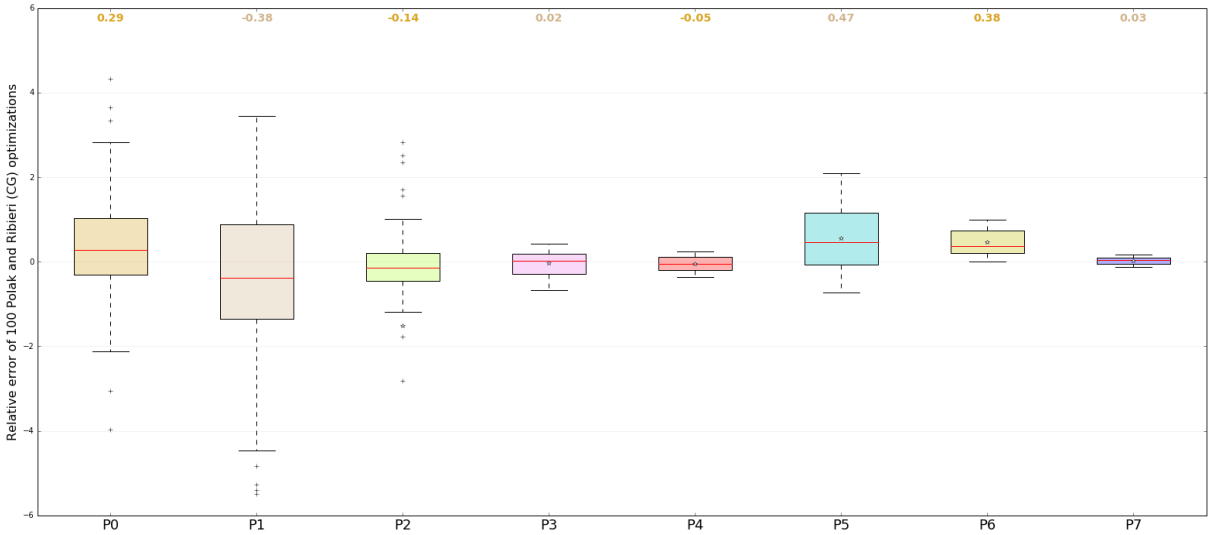


Figure 4.11: Box Graph for Polak and Ribièri method

Table 4.8: Polak und Ribièri (CG) uncertainty statistics of the relative error

stats CG	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	140.063	186.652	-1.516	-0.039	-0.058	0.547	0.454	0.029
Median	0.286	-0.381	-0.143	0.019	-0.046	0.469	0.375	0.032
Std.deviation	2529.251	1782.618	94.598	0.284	0.176	0.785	0.293	0.087
Skew	4.141	8.124	-5.872	-0.374	-0.040	0.335	0.283	0.058
Kurt	52.699	71.582	61.499	-1.005	-1.188	-0.971	-1.295	-1.161

Uncertainty analysis of L-BFGS-B method. By implementing 100 optimizations with the L-BFGS-B method and the randomized initial parameter sets is possible to visualize the robustness of this approach. Only the 37 percent achieves the norm below the value of 0.1, showing that this method is sensitive with the first guess of parameters. The skew values are only positive for the Bed friction coefficient (P_1) and the Parameter for Deviation (P_5) which means extreme values in the positiveness tails. The significant positive kurtosis presented in the Characteristic sediment diameter (P_0) and the Parameter for Deviation (P_5) shows that solutions are dense in a peaked PDF. The extended range of the PDF is submitted for the Characteristic sediment diameter (P_0), the Bed friction coefficient (P_1) and the Parameter for Deviation (P_5); contrary, only the Secondary currents alpha coefficient (P_7) contains a short range of possible solutions. See table 4.9.

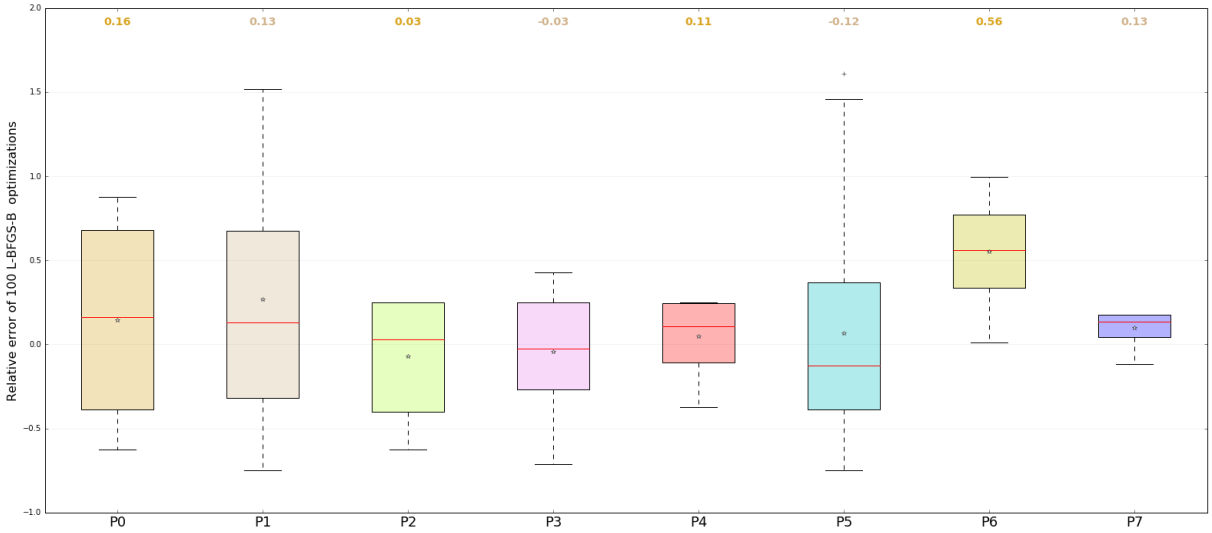


Figure 4.12: Box Graph for L-BFGS-B method

Table 4.9: L-BFGS-B uncertainty statistics of the relative error

stats. LBFGSB	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.142	0.268	-0.073	-0.045	0.046	0.067	0.553	0.096
Median	0.161	0.128	0.031	-0.026	0.109	-0.124	0.562	0.135
Std.deviation	0.546	0.876	0.333	0.313	0.205	0.591	0.275	0.087
Skew	-0.106	1.483	-0.442	-0.129	-0.676	1.103	-0.277	-0.917
Kurt	-1.401	2.630	-1.437	-1.103	-0.910	0.489	-1.094	-0.393

Uncertainty analysis of SLSQP method. The best qualified deterministic method so far, some 900 optimization routines are calculated with the purpose to investigate the robustness. A percent of 64 achieves a lower tolerance than 0.1, indicating its high capacity of a minimum local searcher. The statistics concepts are shown in a Box Diagram in figure 4.13 and in the table 4.10. The parameter Bed friction coefficient (P_1) and the Parameter for Deviation (P_5) present a large positive skewness value, characteristics of extremes, and a significant positive kurtosis, indicating a peaked PDF. The remaining parameter with negative kurtosis have a flat and spread PDF distribution. The Beta (P_6) presents an approximatively symmetrical shape due to the skew value around zero. Furthermore, there are discrepancies in every solution of the Beta (P_6), since global is the exact low limit, indicating non-robustness due to the short range of boundaries. In conclusion, the Secondary currents alpha coefficient (P_7) sounds to be a straightforward parameter for indicated optimization method, possible to see it in the lowest range and standard deviation.

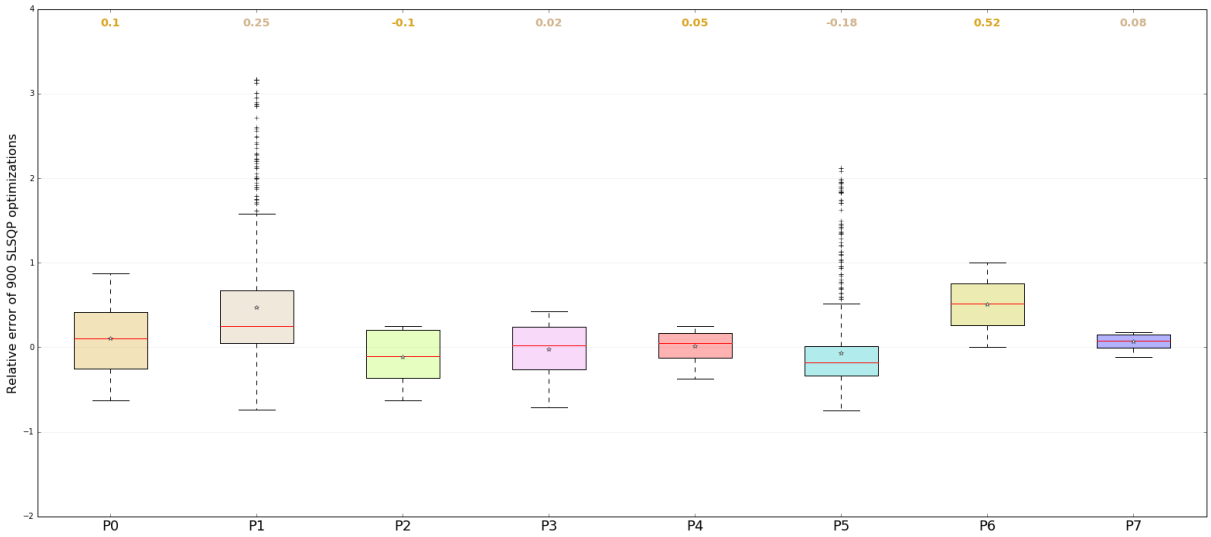


Figure 4.13: Box Graph for SLSQP method

Table 4.10: SLSQP uncertainty statistics of the relative error

Stats SLSQP	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.104	0.466	-0.116	-0.026	0.013	-0.074	0.508	0.064
Median	0.105	0.250	-0.102	0.019	0.049	-0.182	0.519	0.076
Std.deviation	0.449	0.724	0.304	0.315	0.185	0.486	0.290	0.095
Skew	0.079	1.976	-0.282	-0.334	-0.516	2.432	0.005	-0.514
Kurt	-0.980	4.070	-1.246	-0.951	-0.851	6.528	-1.196	-0.908

Uncertainty Analysis of Root Levenberg Marquardt method. The Levenberg-Marquardt method is implemented 100 times in the investigating model, where 33 percent converges under L_2 of 0.1. Entirely non-bounded space of searching, it can propose non-meaningful physical solutions but pleasant numerical ones. The Box diagram in figure 4.14 presents the parameter solutions in quartiles; mainly, the high positive Beta's Kurtosis performs a peaked PDF, also with a significant negative skew value showing that the mean is lower than the median. The relative negative and positive significant skew values in all parameter exposes the extreme solutions. The large positive kurtosis admits intensively peaked PDF. Last, the LM method implies the existence of possible numerical solutions beyond the bounded physical intervals that fulfill the tolerance requirements. It sounds that the Beta (P_6) is the most open to find a solution. For the Parameter for Deviation, (P_5), a huge negative mean value and the largest standard deviation demonstrate a significant amount of extremes values. See table 4.11.

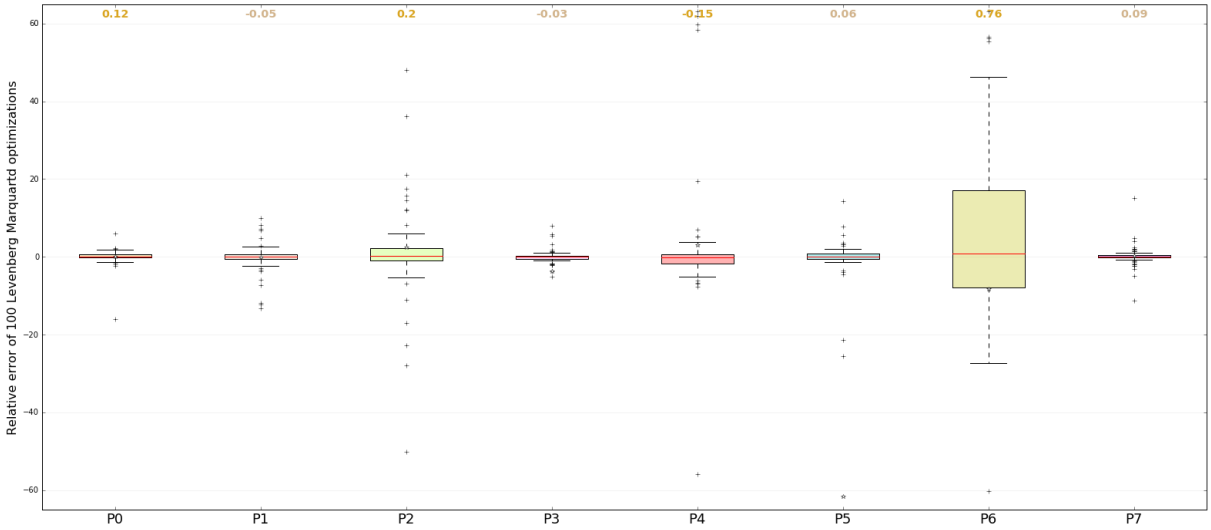


Figure 4.14: Box Graph for Levenberg-Marquardt method

Table 4.11: Levenberg Marquardt uncertainty statistics of the relative error

Stats LM	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.057	-0.197	2.485	-3.779	2.984	-61.653	-8.258	0.152
Median	0.116	-0.050	0.205	-0.027	-0.151	0.060	0.757	0.088
Std.deviation	1.861	3.115	16.995	37.877	28.732	518.771	144.571	2.214
Skew	-6.225	-1.156	5.189	-9.826	1.891	-1.738	-8.534	1.707
Kurt	54.878	6.970	41.713	94.703	14.971	13.037	78.242	26.055

Uncertainty Analysis of Differential Evolution method. This method is highly computational intensive due to a large number of function evaluations, approximated 1681 per optimization; on the other hand, a significant accuracy level of 70 percent achieves the 0.1 tolerance value. It is only implemented 30 global minimum searchers with the optimization library of Python and with uniform random initial parameters sets. The figure 4.15 presents the well-behaved parameter solution and the narrowed range of error between 1 and -0.5. The table 4.12 shows that the skewness values of every parameter are around zero, meaning not extreme values or tails. Also, the six negative values on kurtosis represent spread or flatten PDF distribution, so possible different local solutions are found. In conclusion, although, a fixed number of maximal iterations is set, the rejected evaluated objective functions affects the computational time. The short range in every parameter solution can be seen in the low standard deviation values. See also table 4.12.

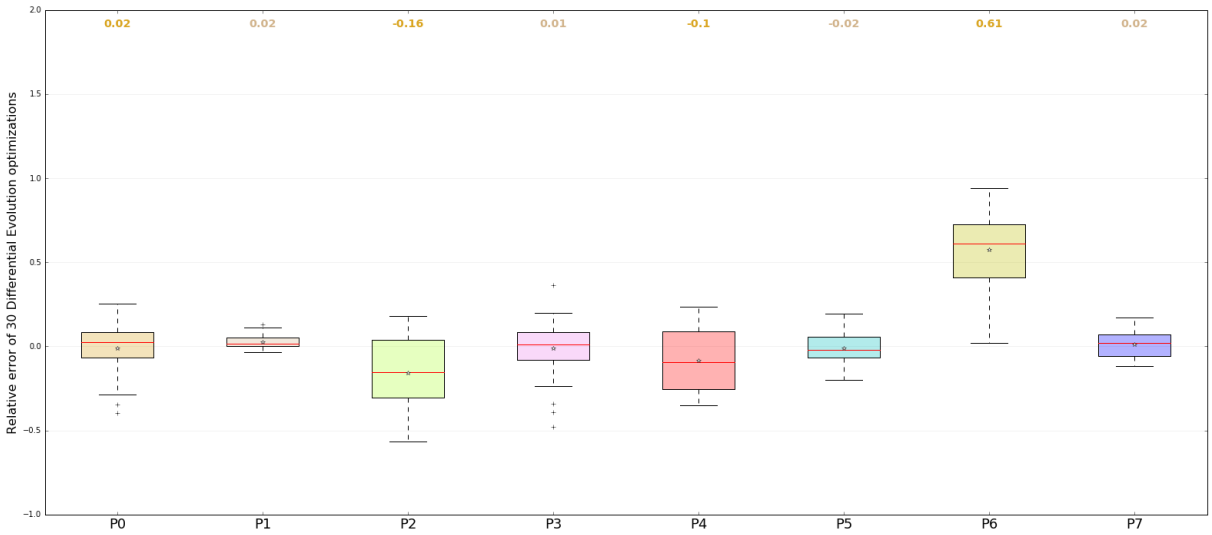


Figure 4.15: Box Graph for Differential Evolution method

Table 4.12: Differential-Evolution uncertainty statistics of the relative error

Stats DiffEvol	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	-0.010	0.026	-0.160	-0.012	-0.087	-0.012	0.572	0.012
Median	0.024	0.017	-0.156	0.010	-0.096	-0.023	0.612	0.020
Std.deviation	0.146	0.042	0.226	0.179	0.191	0.102	0.233	0.084
Skew	-0.976	0.640	-0.158	-0.703	0.216	0.130	-0.578	0.203
Kurt	0.647	-0.184	-1.001	0.586	-1.292	-0.558	-0.284	-1.101

In general, it is clear that each of the optimization methods performs differently under the same circumstances. For the uncertain analysis, only the most suitable methods are chosen, such as the Nelder-Mead, the Polak and Ribièri, the L-BFGS-B, the SLSQP, the Levenberg Marquardt and the Differential Evolution.

The mean value of the relative error per parameter is around zero for the Nelder-Mead, the L-BFGS-B, the SLSQP and the Differential Evolution methods; this means that in average, including the extreme value, the proposed solutions approach to the global. Contrary, the Levenberg Marquardt method insists in other local solution for some parameters.

The median value of the data set is around zero for all analyzed methods; this means that the global solution stays near the center of the statistic data set; so the amount of upper and lower values are similar.

The standard deviation differs from every assessed method, a zero value with tenth parts for each parameter are presented in the L-BFGS-B, the SLSQP, the Differential Evolution methods; this means that all proposals values are nearby, so the same solution is found. The other methods with large standard deviation values present that the local minima are spread in the physical range.

The skewness represents the extreme proposals, including the greater and smaller than global solutions. The highest skew values in all parameter sets are presented in the Levenberg-Marquardt method, the second place; it is for the Polak and Ribièri method. The lowest skew values are performed by the Differential Evolution, around zero and tenth parts. The other have some extreme values in a single parameter.

The kurtosis, which represents the distributions of the proposals, is mainly positive for the Levenberg-Marquardt, meaning peaked solutions, the method finds mostly the same local minima. The other methods present for some parameters negative kurtosis and other positives.

4.3.4 Impacts of Measurement Data in the Objective Function

So far, an equal weighted *Euclidean Norm* between the simulated and the synthetic evolution at 90 and 180° are used as input to the objective function. Other possible combinations can upgrade the actual objective function when measurement data of other variables are given. On the contrary case, a creation of the synthetic measurement data for another variable is feasible to analyze, the performance of different objective functions. Therefore, the output variable of the Bed Evolution and the Water Depth at position 45, 90, 135 and 180 over the bend-channel are added to the overall synthetic data; the plan view is shown in figure 4.16.

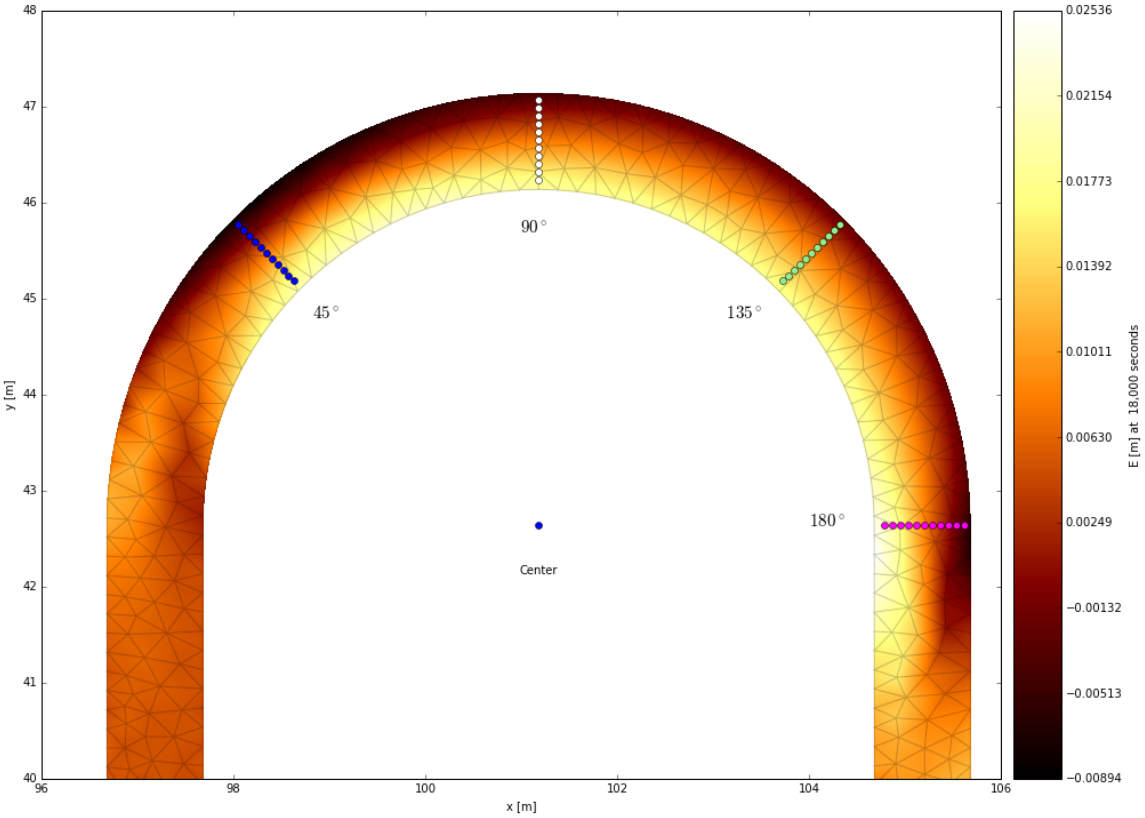
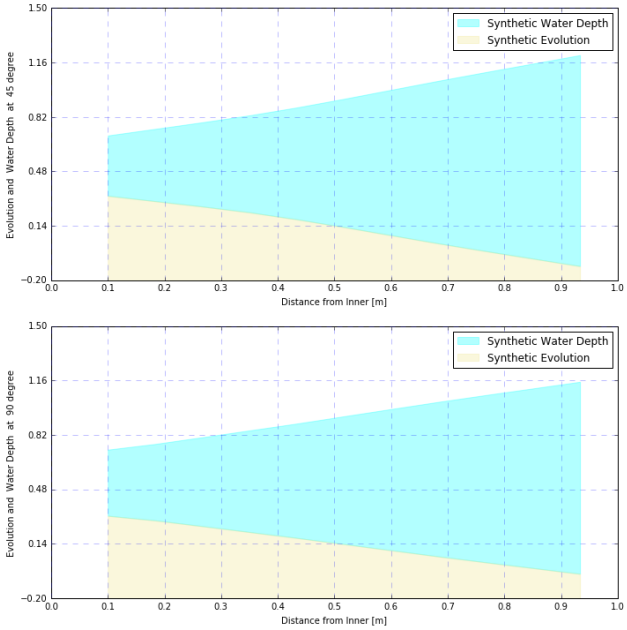


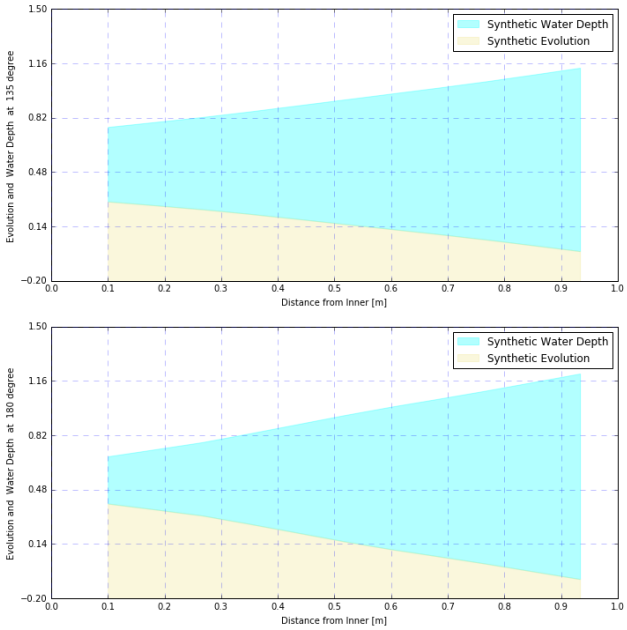
Figure 4.16: Plan view of four different cross sections

Four different sceneries are computed, substituting the previous synthetic data as fixed term in the *objective function*, weight differs from 0.5 to 0.25, summing up the value 1.

1. Bed evolution at same places 90°, 180° to compare with new proposals.
2. Bed evolution 90°, 180°, plus new ones at 45° and 135°.
3. Bed evolution and Water depth at 90°, 180°.
4. Alternate Bed evolution at 90°, 180° and Water depth at at 45° and 135°.



(a) Profiles at 45 and 90°



(b) Profiles at 135 and 180°

Figure 4.17: Synthetic Water Depth and Bed Evolution at four different location

For the setting, the same initial parameter set, a reduced tolerance for the objective function of 1E-6, and the SLSQP method as optimizer are used. First scenery which is utilized as the reference, only with two morphological cross sections, achieves the lowest minimized $norm - L_2$ value, although it requires the full 20 iterations for convergence; also, it obtains the lowest discrepancy parameter value. The second case, duplicating the morphological information variables, convergences after five iterations with much less

precision. The third case and apparently the most unfavorable case, due to largest L_2 value and discrepancy, stops at four iterations with a proposed solution. Last case, two different variable at two other locations converges at five iterations with an acceptable norm and discrepancy percent value. See table 4.13.

In General, all cases accomplish a moderate discrepancy with the global and significantly minimize the $L_2 - norm$ under a required number of iterations, see tables 4.14. The reference case, scenario one with the less given data minimizes at best the objective function at the expense of all iterations. Nevertheless, omitting the reference case, it can be interfered by case four that different variables at different locations cover more the problem. Thus, more measurement data need less iterations due to the reduction of the over-determination. Moreover, weighting data, according to relevance and confidence of the measurement points, increases the possibility of finding a realistic solution.

Table 4.13: Proposed solutions by the SLSQP method for the different sceneries

Scenary	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	Norm
1	9.30E-04	1.18E-03	9.85E-04	3.54504	0.29078	0.83856	1.32811	0.98592	0.00471
2	9.34E-04	1.79E-03	4.92E-04	3.68376	0.43122	0.61531	1.30936	0.99996	0.09391
3	9.46E-04	1.95E-03	1.00E-03	3.65386	0.42350	0.56902	1.31198	0.99499	0.12458
4	1.29E-03	1.29E-03	1.00E-03	3.70898	0.37900	0.74669	1.31043	0.99999	0.05417

Table 4.14: Relative error for the different sceneries

Scenary	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	%
1	0.16	0.01	0.23	0.01	0.27	0.05	0.33	0.16	15.37
2	0.17	0.50	0.38	0.05	0.08	0.23	0.31	0.18	23.69
3	0.18	0.63	0.25	0.04	0.06	0.29	0.31	0.17	24.18
4	0.61	0.07	0.25	0.06	0.05	0.07	0.31	0.18	19.94

Table 4.15: Performance of the different sceneries

Scenary	Func. Calls	Funct.Eva	Jac.calls	Iteration	Time (s)
1	334	145	21	20	10293
2	71	26	5	5	2088
3	52	17	4	4	1610
4	68	23	5	5	2082

4.4 Assessment of Stochastic Algorithm: *Bayesian Inference Theory*

In consideration that the deterministic methods do not always perform a solution that describes properly the hydrodynamic and hydro-morphological system for Yen and Lee (1995); the Bayes' Inference Theorem, which is a stochastic optimization method, is also investigated. Indeed, it is the most computationally expensive, but very robust version of the stochastic methods. Another version can be implemented as well.

The Bayesian inference equation is defined as $P(x) = c * P_0(x) \cdot L(x)$, where $P_0(x)$ is the prior probability distribution function (PDF). It is constructed in a way that also covers the sources of uncertainty, such as the human and the numerical ones. The prior PDF must be built with many simulations, in this case, some 1000 is chosen.

For the present investigation, two different distribution type of Prior PDF is used. First, a group of 1000 uniformly distributed random parameter sets is generated, where the limit values correspond to the physical boundary values of each parameter. Second, other 1000 normally distributed random parameter sets is also generated, where the mean value of each parameter correspond to the initial parameters suggested by the literature, and the standard deviation is a sixth of the range between the low and the upper bounds. After that, the parameter sets are used to simulate the numerical model of *Bend Channel*, to store the bed evolution output variable at 90° and 180°, and to calculate the objective function for each set.

The conditional probability distribution or the likelihood function $L(x)$, which is used to calculate the weighs of prior PDF according to the chosen error value, as explained in Oladyshkin et al. (2013), for independent measurement data $L(x)$ takes the form of:

$$L(x) = e^{(-0.5 \cdot L_1 \cdot E^{-1} \cdot L_1^T)}, \tag{4.4}$$

where L_1 is the first norm or the deviation between the fixed synthetic evolution and the randomly generated evolution, both at 180°; also, the inverse diagonal matrix is E^{-1}, with same dimension of parameters, and its entries are the allowed error ϵ of 0.0009.

The posterior PDF is composed after filtering the normalized weighted Prior PDF, under the condition that the normed weights are greater than a uniformly generated random number between 0 and 1. The statistics concepts of the Prior and Posterior data, such as the mean, the standard deviation, the skewness and the kurtosis values are plotted on every graph for a best analysis and interpretation.

4.4.1 Uniformly randomized parameters for Prior PDF

Prior Probability Distribution Function. It is the generation of 1000 uniformly distributed random parameter sets with unknown values within the physically bounded intervals. The graphs of Prior PDF are shown in figure 4.18. Mostly flat shaped PDF dominates, which is naturally of a negative kurtosis.

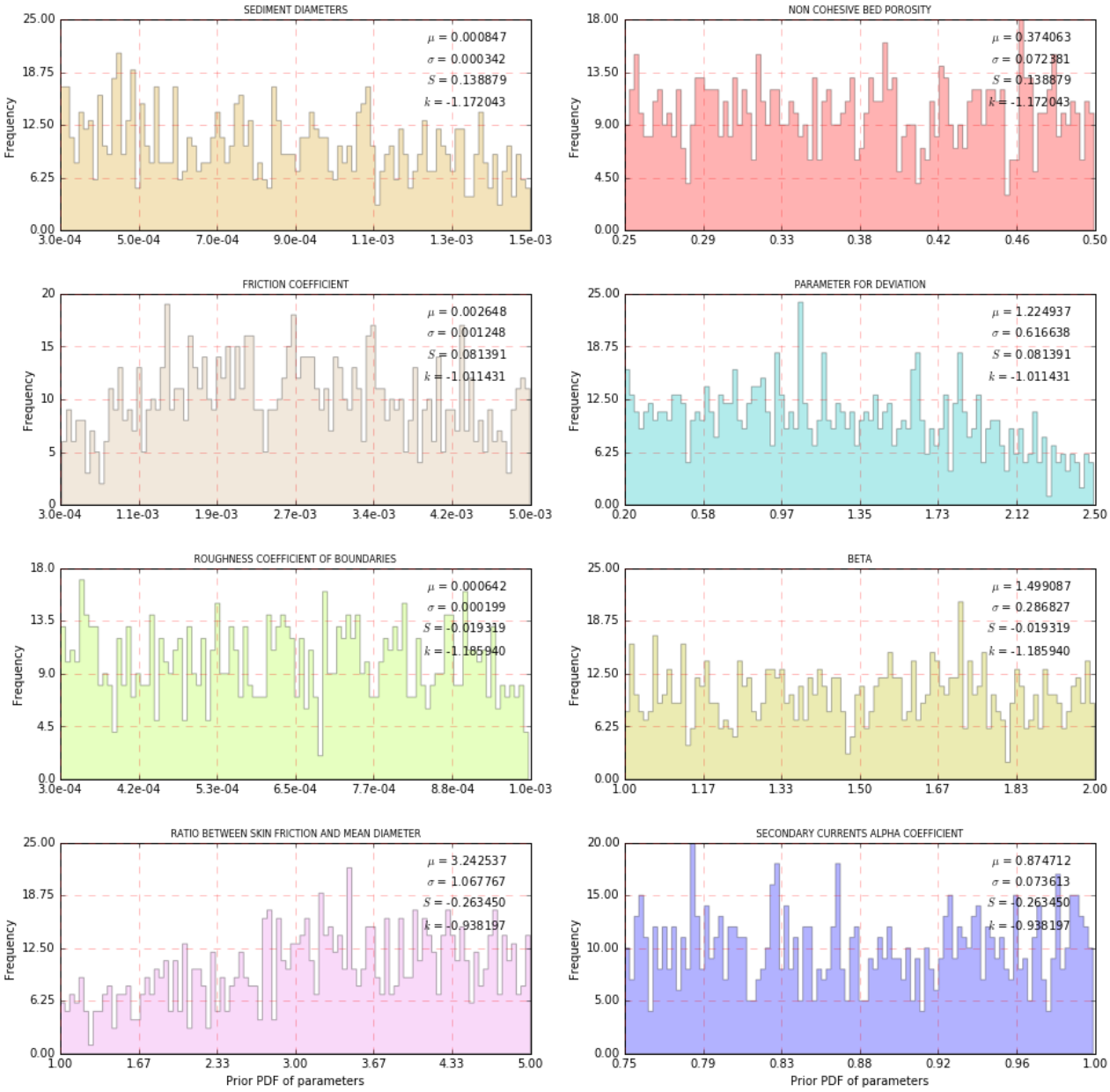


Figure 4.18: The Prior PDF of the uniformly randomized parameters

As it is expected due to the uniformly randomized parameters, the skewness values are around zero, and all kurtosis values are negative.

Table 4.16: Statistics of the Prior PDF for the uniformly randomized parameters

stats Prior	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.0008	0.0026	0.0006	3.2425	0.3741	1.2249	1.4991	0.8747
Median	0.0008	0.0026	0.0006	3.3128	0.0008	0.0026	0.0006	3.3128
Std.deviation	0.0003	0.0012	0.0002	1.0678	0.0724	0.6166	0.2868	0.0736
Skew	0.1389	0.0814	-0.0193	-0.2635	0.1389	0.0814	-0.0193	-0.2635
Kurt	-1.1720	-1.0114	-1.1859	-0.9382	-1.1720	-1.0114	-1.1859	-0.9382

Posterior Probability. From 1000 prior size, a total of five simulations becomes the posterior PDF, see table 4.17. All of them present a relative well minimized L_2 values and a reasonable discrepancy value under 31%, see the table 4.18. Therefore, the high robustness of Bayes' Theorem with uniformly distributed random parameters is demonstrated.

The statistics of the posterior solutions present some extra information. The negative skewness dominates over all parameters, meaning that the global solution values are larger than the proposed solutions. The negative kurtosis stands for the spread and the flat PDFs, demonstrating the existence of multiple local solutions, see table 4.19.

Table 4.17: The Posterior parameters after filtering

Solutions	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	Norm
0	6.34E-04	1.58E-03	5.00E-04	4.887	0.299	0.828	1.528	0.920	0.03751
1	1.22E-03	1.49E-03	6.12E-04	4.129	0.255	0.598	1.607	0.931	0.08678
2	4.47E-04	1.88E-03	6.36E-04	3.433	0.390	0.679	1.733	0.801	0.07151
3	1.16E-03	1.11E-03	5.74E-04	4.165	0.389	0.920	1.357	0.952	0.03931
4	9.93E-04	9.50E-04	6.95E-04	4.428	0.261	0.975	1.775	0.948	0.03238

Table 4.18: Relative error between the Posterior parameters and the global solution after filtering

solution ID	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	%
0	0.21	0.32	0.38	0.40	0.25	0.04	0.53	0.08	27.44
1	0.53	0.25	0.23	0.18	0.36	0.25	0.61	0.09	31.29
2	0.44	0.57	0.21	0.02	0.02	0.15	0.73	0.06	27.51
3	0.45	0.07	0.28	0.19	0.03	0.15	0.36	0.12	20.56
4	0.24	0.21	0.13	0.27	0.35	0.22	0.78	0.11	28.77

Table 4.19: Statistics of the Posterior PDF of parameters

stats post	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.0009	0.0014	0.0006	4.2086	0.3189	0.7999	1.6003	0.9101
Median	0.0010	0.0015	0.0006	4.1654	0.0010	0.0015	0.0006	4.1654
Std.deviation	0.0003	0.0003	0.0001	0.4728	0.0597	0.1424	0.1501	0.0558
Skew	-0.3509	-0.0180	-0.2359	-0.2716	-0.3509	-0.0180	-0.2359	-0.2716
Kurt	-1.5385	-1.3579	-0.8982	-0.6968	-1.5385	-1.3579	-0.8982	-0.6968

The normalized weights of the Prior PDF are displayed in figure 4.19, where the highest is less than 0.3, meaning that a non-single simulation achieves the chosen error of 0.0009 since non-solution has a weight of 1. The gaps between the solutions indicate that moving

just a fraction of the parameter value rises to a non-solution or a maximum. Therefore, the deterministic methods can fail, even though the numerical value is highly approximated to the optimal solution.

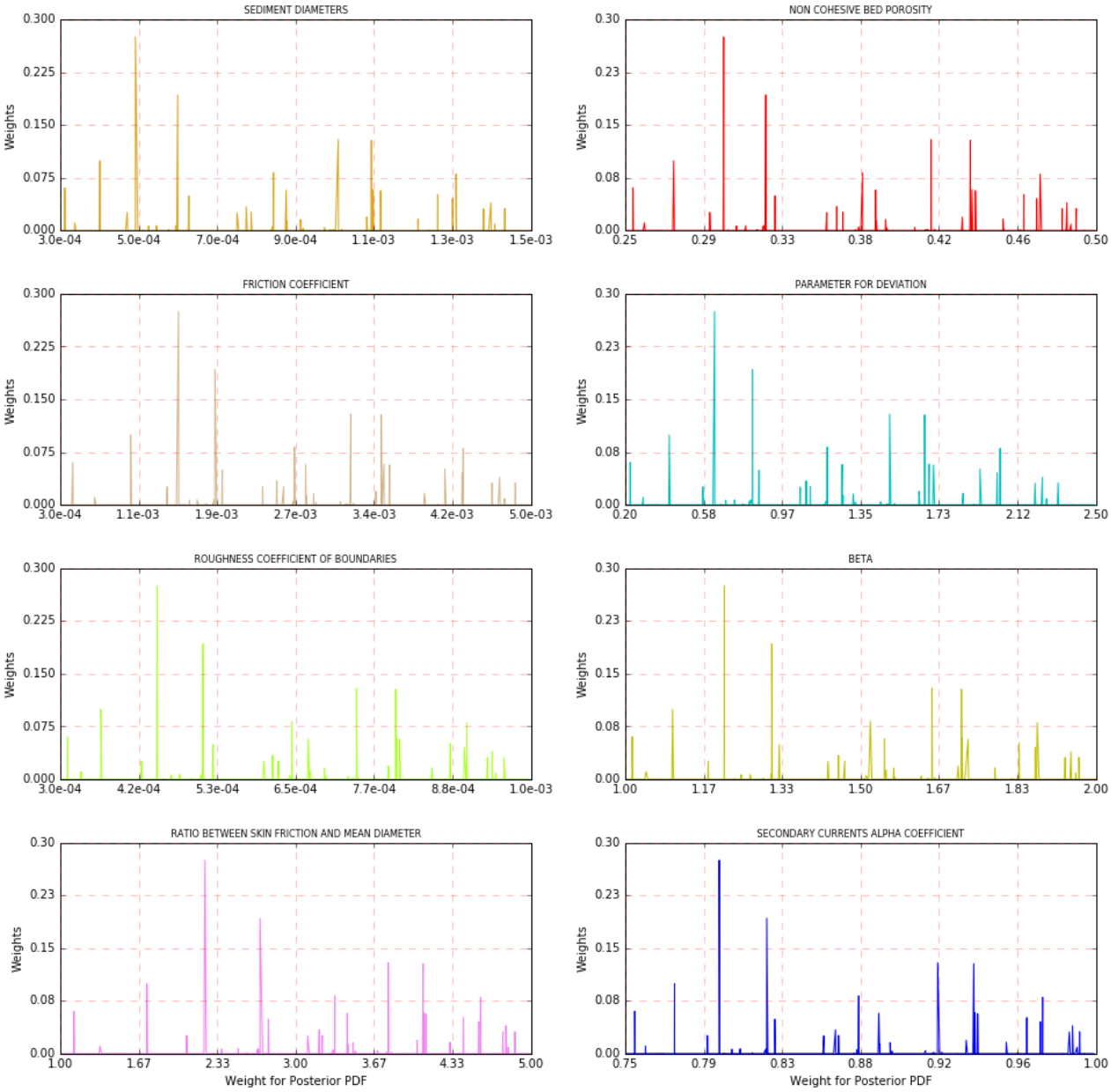
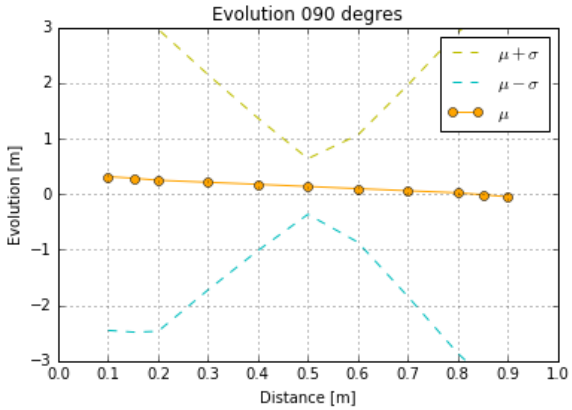


Figure 4.19: Weights of conditional PDF, uniformly distribution

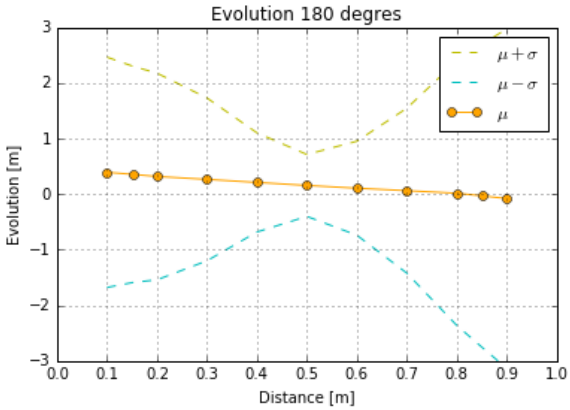
Deviation of bed evolution profiles. From the Bayesian Inference theorem, not only is possible to see the performance of the parameters, but also the output variable behavior. The simulated variables proposed for each parameter set, in this case, the bed morphological evolution at 90° and 180°, can also be used for investigation. The figure 4.20 presents the misfit or the deviation between the synthetic measurement data as mean value, and this (μ) mean or synthetic measurement value plus the standard deviation calculated from all prior and posterior measurement data, as presented:

$$\sigma^2 = \frac{1}{n-1}\sum(x_i - \mu^2), \tag{4.5}$$

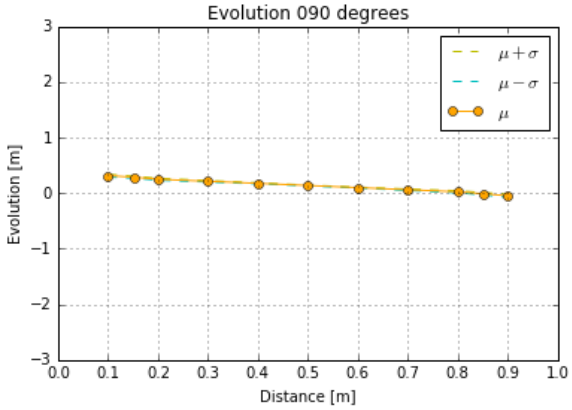
where μ is the value of the synthetic measurement point (bed evolution) and x_i is every value of the prior simulated bed evolution. n stands for the size of the prior, in this case 1000. The standard deviation of prior evolution is much larger than the posterior. It seems that the profiles match with a solution from the minimization of $L_2 - norm$ point of view; yet it is already seen the discrepancies with the global solution.



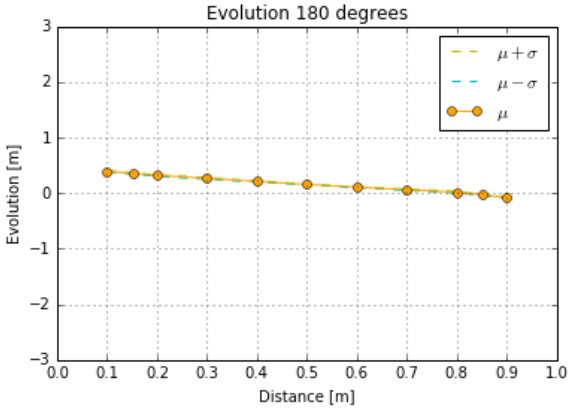
(a) Prior evolution at 90 degree



(b) Prior evolution at 180 degree



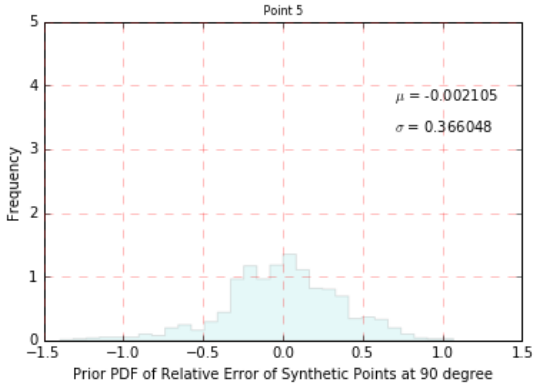
(c) Posterior evolution at 90 degree



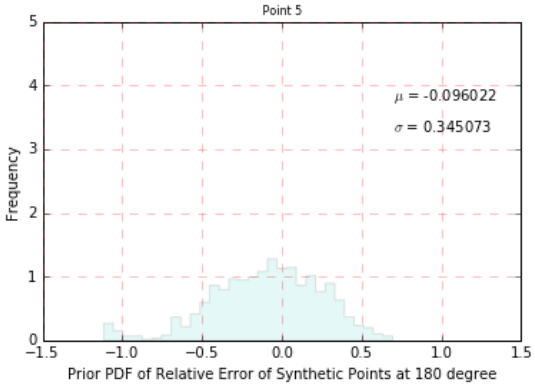
(d) Posterior evolution at 180 degree

Figure 4.20: Prior and Posterior of synthetic evolution at 90° and 180°

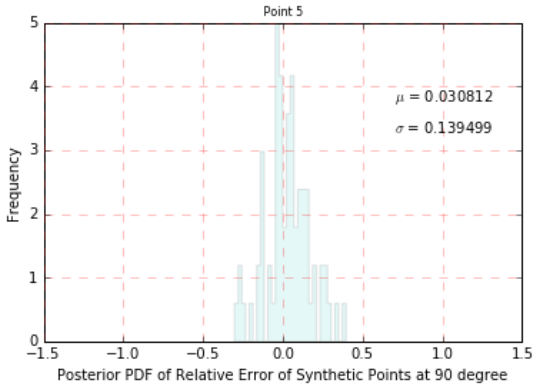
Prior and Posterior PDF of bed evolution at point 5. For instance, the relative errors of the measurement point number five from both bed evolution profiles is used, see figure 4.21. The prior and two posterior PDF after filtering with different allowed errors ϵ are used, namely 0.01 and 0.0009. In both cases, highly peaked PDFs appear on both solutions, showing that kurtosis values are apparently increased. In the second instance of the allowed error (0.0009) ϵ, the reduction of the relative error approaches highly to zero, showing that the proposed parameters minimize significantly the $L_2 - norm$.



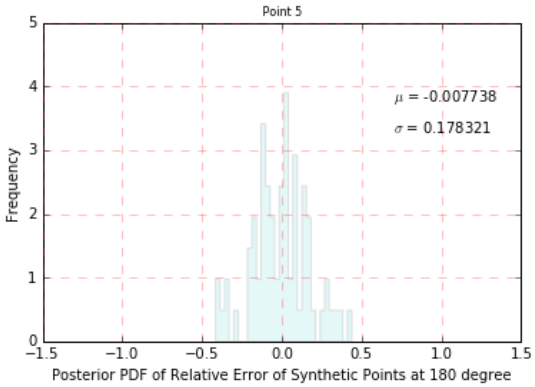
(a) Prior PDF at 90°



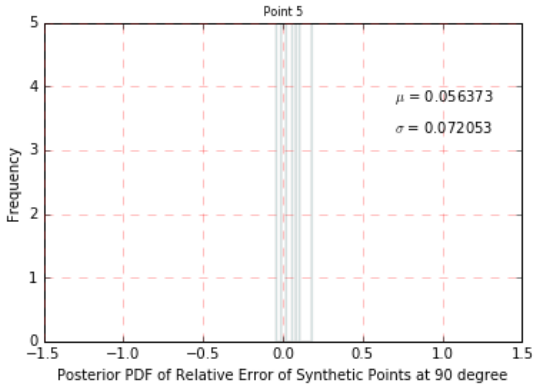
(b) Prior PDF at 180°



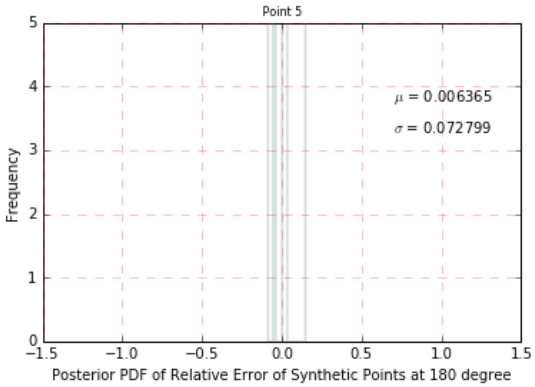
(c) Post. PDF at 90°, error of 0.01



(d) Post. PDF at 180°, error of 0.01



(e) Post. PDF at 90°, error of 9E-4



(f) Post. PDF at 180°, error 9E-4

Figure 4.21: Prior and Posterior PDF at point 5 of cross section 90° and 180°

4.4.2 Normally randomized parameters for Prior PDF

Prior Probability Distribution Function. In this case, the generation of 1000 normally randomized parameter sets with unknown values is proposed, where the mean for each PDF is the initial parameter values, and the standard deviation is a sixth of the range between the low and the upper bounds. The graphs of the normally distributed random Prior PDF are shown in figure 4.22.

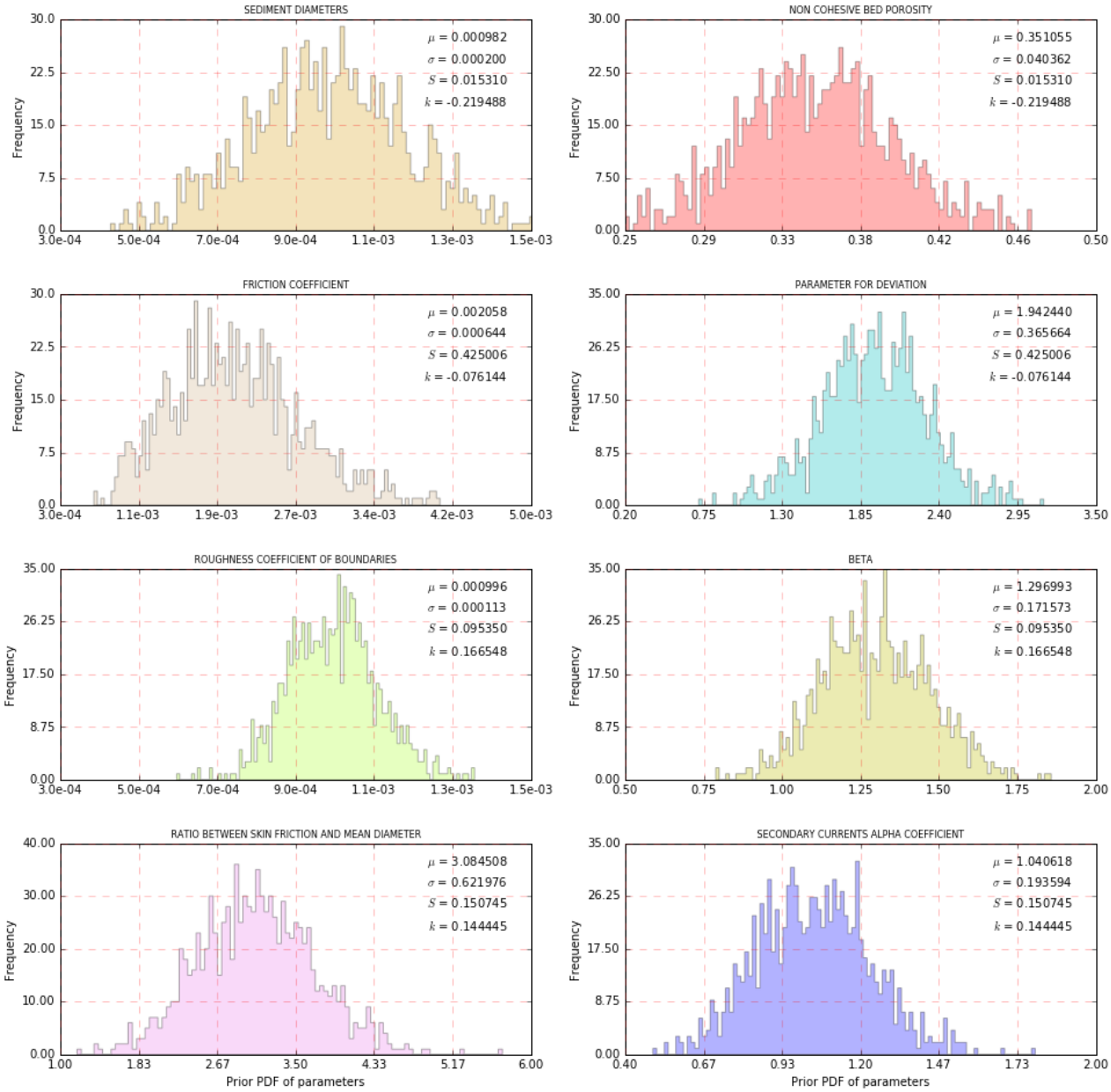


Figure 4.22: Prior PDF of the normally randomized parameters

Table 4.20: Statistics of prior PDF of normally randomized parameters

StatsPriorN	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.0010	0.0021	0.0010	3.0845	0.3511	1.9424	1.2970	1.0406
Median	0.0010	0.0020	0.0010	3.0846	0.0010	0.0020	0.0010	3.0846
Std.deviation	0.0002	0.0006	0.0001	0.6220	0.0404	0.3657	0.1716	0.1936
Skew	0.0153	0.4250	0.0954	0.1507	0.0153	0.4250	0.0954	0.1507
Kurt	-0.2195	-0.0761	0.1665	0.1444	-0.2195	-0.0761	0.1665	0.1444

Posterior Probability. From 1000 prior size, a total of two simulations is filtered to the posterior PDF, see table 4.21. The minimized objective functions obtain a value under 0.045, see table 4.21 and a good discrepancy under 25% 4.22. Therefore, the highly specific normal distribution Prior PDF brings only a few solutions around the mean value or the initial parameters, beyond this type or randomization is limited.

Also, some statistics concepts can be calculated, the table 4.23 presents that the zero skewness value dominates over all parameters, meaning the existence of non-extremely parameters values, due to the non-tailed PDF. The highly reduced standard deviation demonstrate no large different among solutions; yet is only two data. All kurtosis values are equal to -2.

Table 4.21: Solutions normally randomized parameter after filtering

Solutions	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	Norm
0	8.11E-04	8.18E-04	1.19E-03	4.123	0.319	1.130	1.235	1.038	0.04503
1	8.78E-04	8.90E-04	8.82E-04	2.614	0.370	1.041	1.281	1.414	0.03763

Table 4.22: Relative error of the solutions normally randomized parameter with global solution after filtering

solution ID	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	%
0	0.01	0.32	0.48	0.18	0.20	0.41	0.23	0.22	25.81
1	0.10	0.26	0.10	0.25	0.07	0.30	0.28	0.66	25.39

Table 4.23: Statistics of posterior PDF of parameters

StatsPostN	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
Mean	0.0008	0.0009	0.0010	3.3688	0.3446	1.0854	1.2577	1.2262
Median	0.0008	0.0009	0.0010	3.3688	0.0008	0.0009	0.0010	3.3688
Std.deviation	0.0000	0.0000	0.0002	0.7546	0.0255	0.0445	0.0231	0.1878
Skew	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Kurt	-2.0000	-2.0000	-2.0000	-2.0000	-2.0000	-2.0000	-2.0000	-2.0000

The normalized weights of the Prior PDF are displayed in figure 4.23, where the highest value is less than 0.02 meaning that non-single simulation fulfills the allowed error of 0.0009 ϵ since non-solution has a weight of one. The separations of the solutions indicate that moving just a fraction of the parameter value rises to a non-solution. Therefore, the

deterministic methods can fail, even though the numerical value is approximated to the optimal solution.

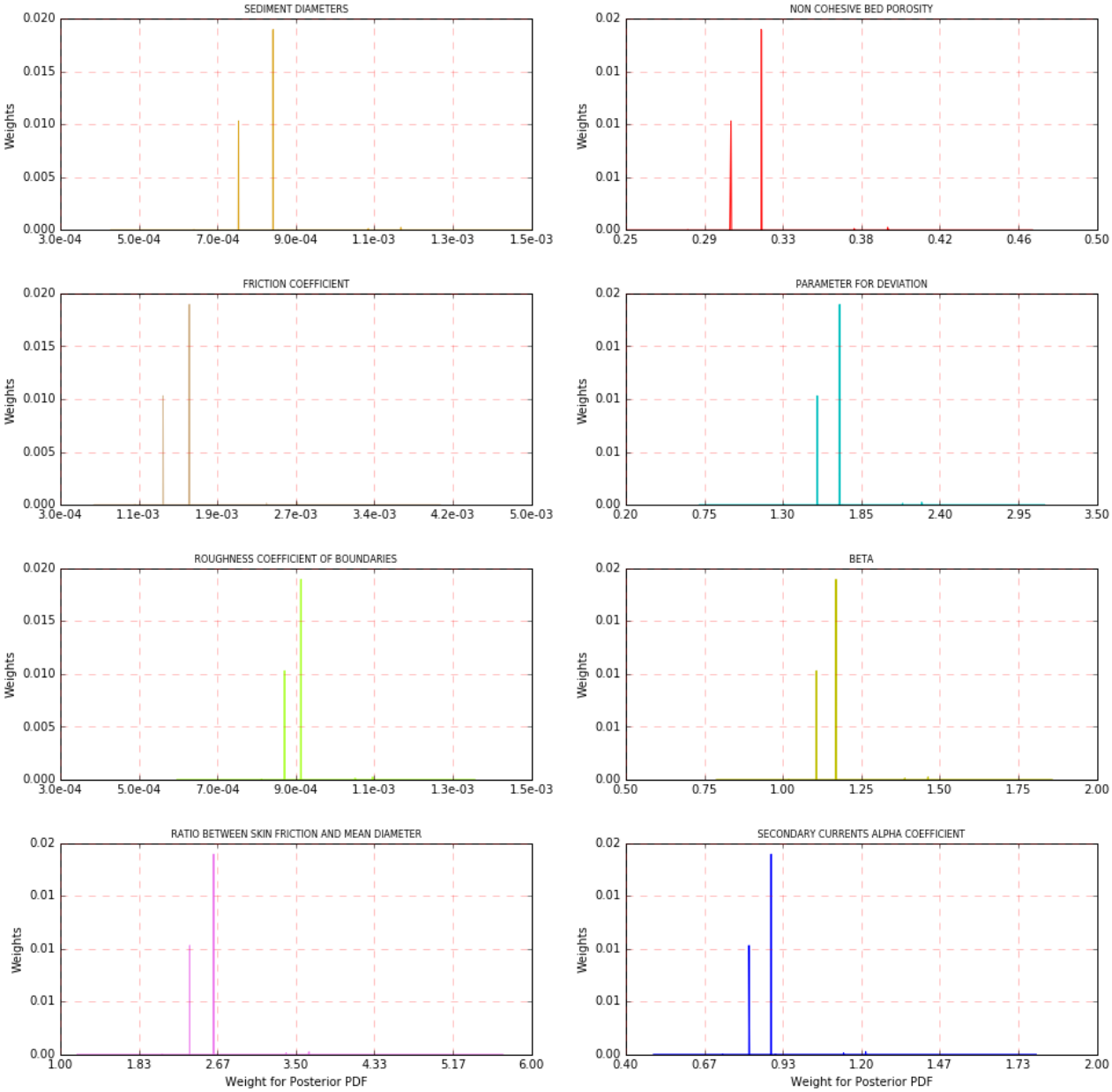


Figure 4.23: Weights of conditional PDF, normal distribution

4.5 Automated Calibration for the Bend Channel Model: *Real measurement data*

The knowledge of the hydro-morphological system behavior, the limitation, and the instabilities, as well, the performance, the accuracy, and the expectancies of each optimization method are the basis for the calibration of the *Bend Channel* numerical model with the real measurement data. These data are the bed evolution at 90° and 180° as input in the objective function.

4.5.1 Applied Deterministic Methods to the Bend Channel Model

The *objective function* is minimized, utilizing as input the real measurement data. The chosen methods are the SLSQP, the Root.LM, the Basin-Hopping, and the Different Evolution; they are limited to find a solution under 20 iterations.

SLSQP. For this method, different sceneries of the objective functions, which include the duplication of weight in the possible bending measurement points, the utilization of fewer measurement points, and the reduction of the number of parameters, are proposed:

Scenario 1: original minimization of an objective function that includes the evolution variable at 90 and 180 degrees, equal weight for every measurement point.

Scenario 2: previous situation plus a duplicate weight to the measurement points of 2, 4 and 8.

Scenario 3: minimize an objective function that only includes the measurement points of 2, 4 and 8.

Scenario 4: minimize an objective function with duplicate weight to the measurement points of 3, 5 and 8.

Scenario 5: minimize an objective function with duplicate the weights of the measurement points of 3, 5 and 8. Also, it only includes the characteristic diameter (P_0), the bed friction coefficient (P_1) and the boundary friction coefficient (P_2), which apparently are the most sensible parameters; the others remain as the initial parameter values.

Scenery 6: minimize an objective function with duplicated weighted the measurement points of 3, 5 and 8 and larger bounds of search.

Table 4.24: Solution of the real measurement

Scenary	Func. Calls	Jac.calls	Func.Eval.	Jac.Eval.	Iteration	Time (s)
1	344	20	164	20	20	9951
2	157	11	58	11	11	4990
3	182	11	83	11	11	5300
4	289	19	118	19	20	8148
5	58	8	25	8	8	1713
6	145	10	52	10	11	4419

Table 4.25: Solution of the real measurement

Scenary	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	Norm
1	1.10E-03	3.98E-03	9.99E-04	2.870	0.389	1.635	1.290	0.899	0.382
2	1.20E-03	3.18E-03	1.00E-03	3.019	0.415	1.312	1.307	0.998	0.529
3	7.42E-04	2.73E-03	1.00E-03	3.055	0.455	1.250	1.289	0.863	0.591
4	1.05E-03	4.91E-03	3.22E-04	2.536	0.493	1.403	1.292	0.752	0.370
5	1.06E-03	4.98E-03	1.00E-03	3.000	0.350	2.000	1.300	1.000	0.350
6	1.25E-03	3.13E-03	2.56E-03	2.912	0.338	1.759	1.301	0.811	0.421

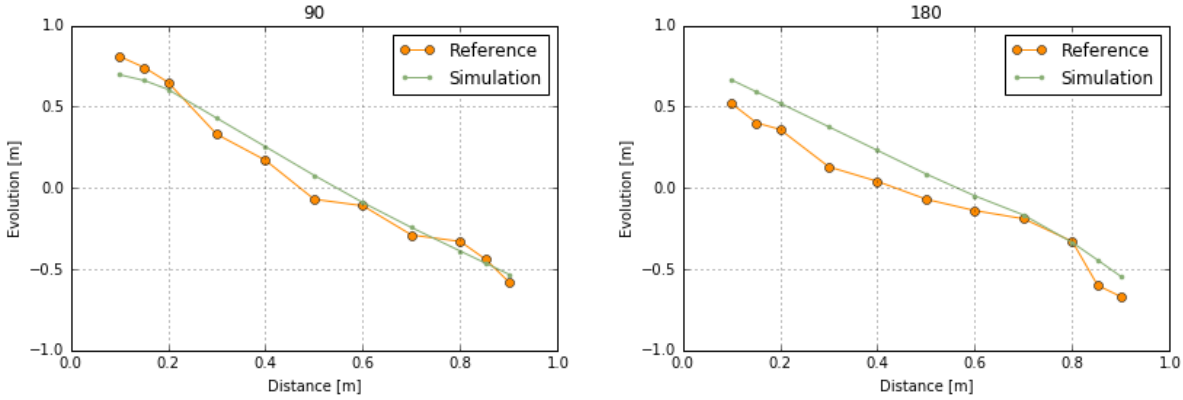


Figure 4.24: Proposed solution for scenario 01

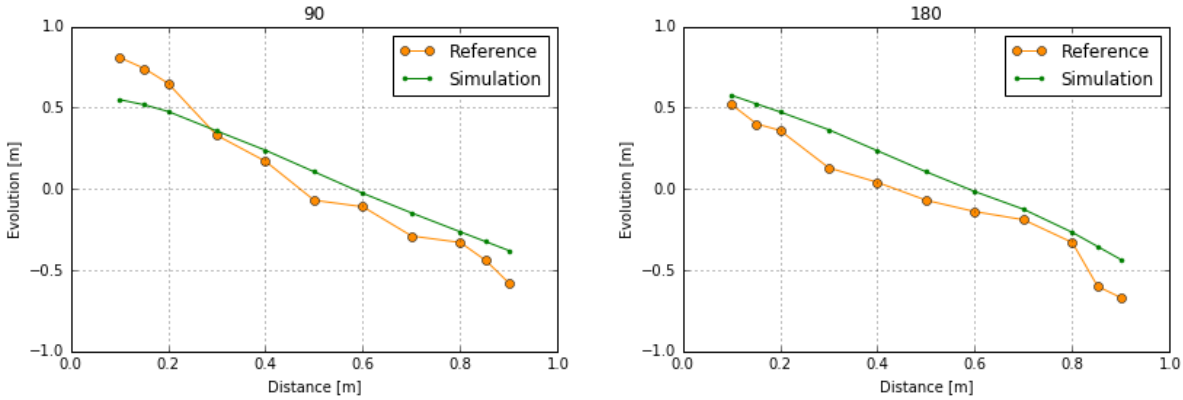


Figure 4.25: Proposed solution for scenario 02

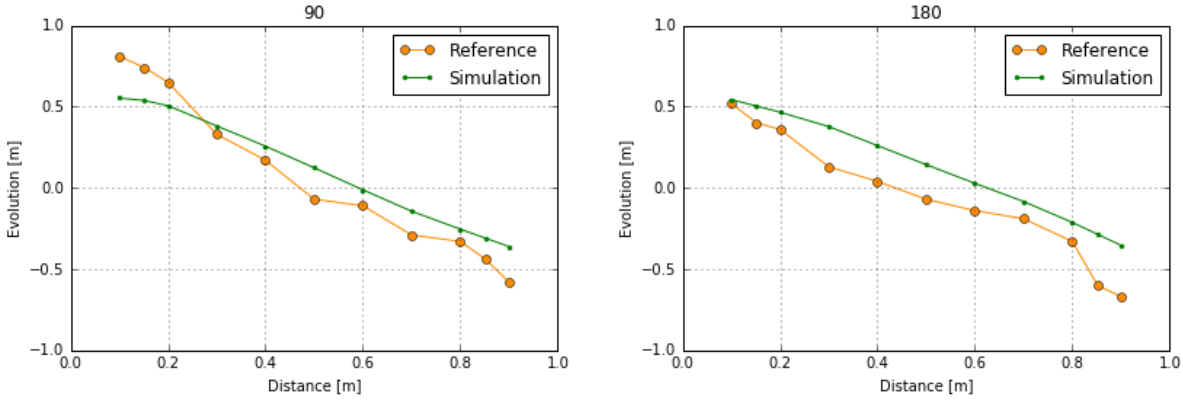


Figure 4.26: Proposed solution for scenario 03

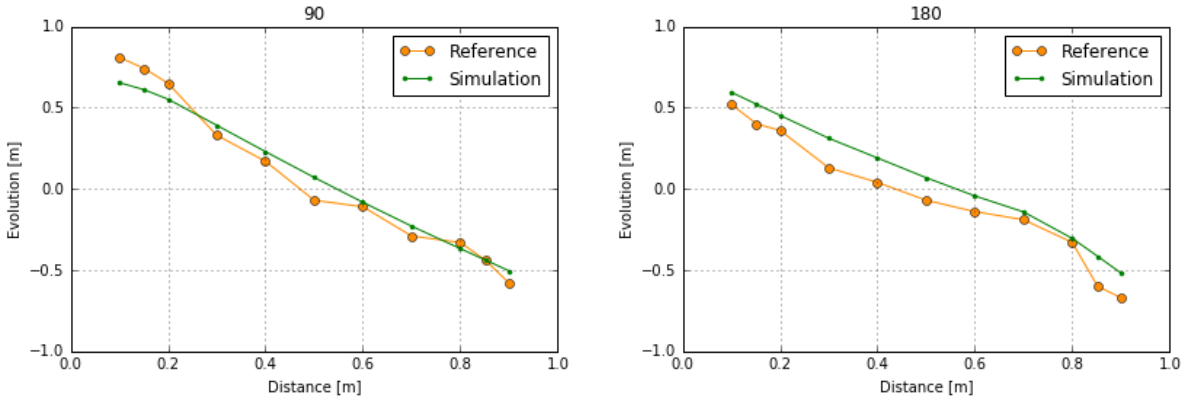


Figure 4.27: Proposed solution for scenario 04

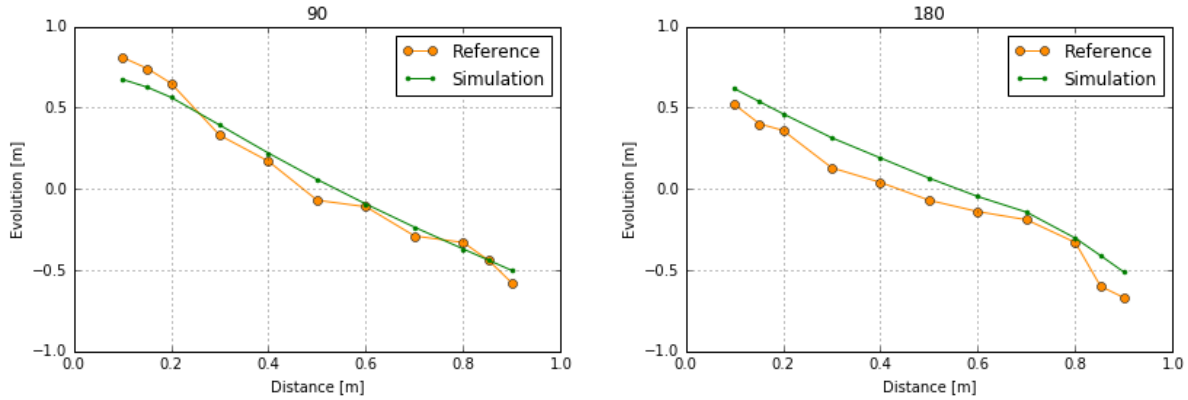


Figure 4.28: Proposed solution for scenario 05

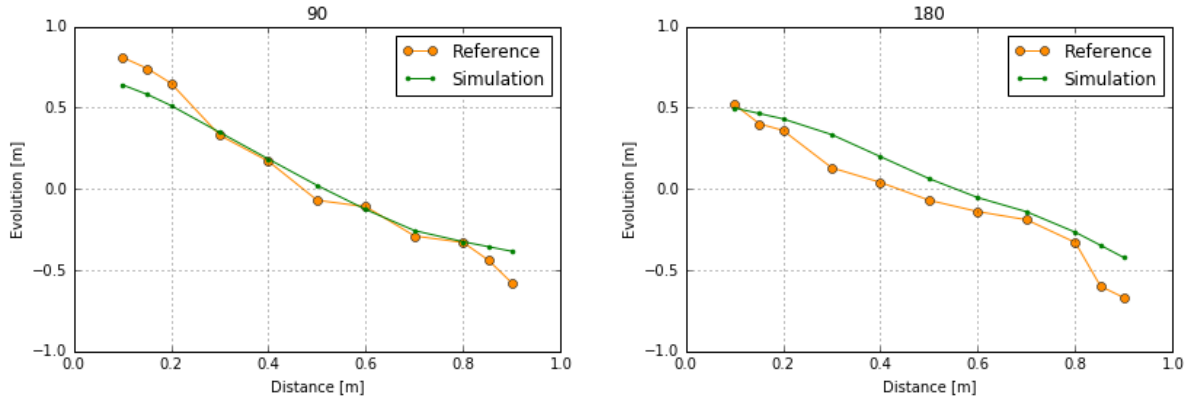


Figure 4.29: Proposed solution for scenario 06

Afterward, the solution of different weighted objective functions and their L_2 misfit are compared in table 4.25; even though, they are not the perfect minimized solution. It is clear that the Characteristic sediment diameter (P_0), the Bed friction coefficient (P_1) and the Boundary friction coefficient (P_2) are most difficult to calibrate. The sensitivity of these parameters is visible on the result of the derivative, usually presenting high slope. Meanwhile, the others parameter's derivatives get intermediately around a zero value. Demonstrating that is not necessary increasing the problem dimensionality with parameters of low sensitivity. Nicely, the scenery with fewer parameters to being calibrated is the one with best-minimized objective function and few iterations. The resolution of the grid should be higher than the resolution of the measurement points to improve results of the calibration.

Other optimization methods. Other well performance optimization methods are applied to the objective function with the real measurement data of the *Bend Channel*

model for investigation, as visible in tables 4.26 for the proposed solutions and 4.27 for the performance of the optimization methods, the scenario one is used as the objective function.

The combination of the Basin-Hopping method for global search plus the Nelder-Mead method for refining perform a considerably reduced norm of 0.336, see figure 4.30. However, at expense of the highly expensive computational cost of 1766 function evaluations. The acceptance test, which includes the physical bounds, increases the number of evaluations functions, to allow realistic solutions.

The Differential Evolution method performs a final solution of the L_2 to 0.3114, at the expense of the 20 iterations, and a large number of rejected evaluated functions. It is the best method's performance in this group; however, it requires a significant amount of evaluated functions.

The Levenberg Marquardt application, unfortunately, for the real non-smooth problems, fails due to the misconception of finding a local minimum. All gradients present zero values, due to output behavior in TELEMAC2D-SISYPHE, in the case of the non-executable combination of parameters; delivering the previous well-computed evaluation function for subsequent calculations; even though, it is used different parameters sets. The non-bounded application proposes non-meaningful parameters values after the first Jacobian matrix evaluation shows a large gradient for most sensitive parameters.

The Nelder-Mead, cheap computational performance is investigated for the real case, the weak reduction of the norm shows that non-suitable for present type of problems.

The curve-fit profiles of the Differential Evolution and the Basin-Hoping methods are presented in figures 4.30 and 4.31.

Table 4.26: Solutions by methods for real measurements

Methods	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	Norm
SLSQP	1.10E-03	3.98E-03	9.99E-04	2.870	0.389	1.635	1.290	0.899	0.382
Basin Hoppin .+Nelder-mead	1.12E-03	2.51E-03	6.58E-04	2.203	0.272	2.634	1.520	1.131	0.336
Differential Evolution	1.30E-03	4.14E-03	4.70E-04	3.431	0.327	2.156	1.091	0.872	0.314
LM	4.70E-04	-9.20E-04	5.76E-03	-0.213	0.442	6.095	27.545	3.991	0.547
Nelder mead	1.02E-03	2.46E-03	9.90E-04	2.919	0.350	1.722	1.255	1.042	0.519

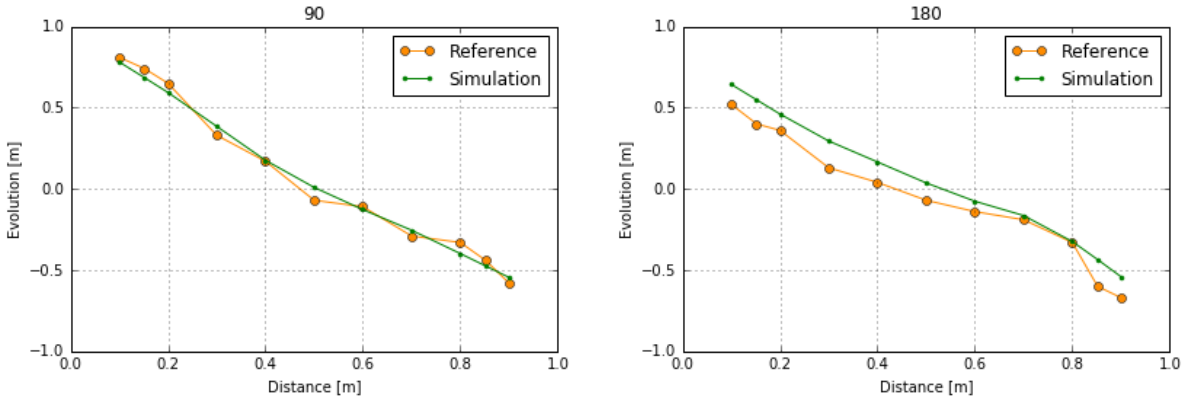


Figure 4.30: Basin Hopping proposed solution

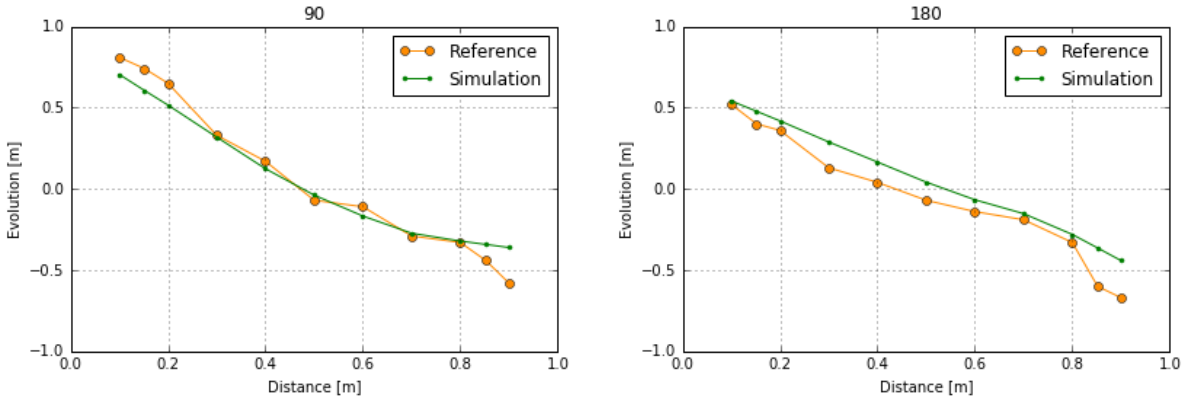


Figure 4.31: Differential Evolution proposed solution

Table 4.27: Performance of methods applied to real measurements

Methods	Func. Calls	Jac.calls	Func.Eval.	Jac.Eval.	Iteration	Time (s)
SLSQP	344	20	164	20	20	9951
Basin Hoppin	1766	-	1766	-	4	49907
.+Nelder-mead	323	-	323	-	154	
Differential Evolution	1680	-	1680		20	51302
LM	31	3	2	2	3	648
Nelder-Mead	34	-	20	-	20	1031

4.5.2 Applied Stochastic Method to the Bend Channel Model

The Bayesian Inference is now applied to the objective function that includes the real measurement data. Previously, the prior PDF of the parameters in figure 4.18 can be compared with the new posterior PDF of real measurements on 4.32, after the filtering process. One posterior solution out of the 1000 prior PDF is obtained with a non-suitable norm of 0.4446.

Deviation of bed evolution profiles. The deviation of the bed evolution consists of the real measurement profiles as mean value and its standard deviation. A zig-zag shape of the profiles performs, compared with the relative soft of 4.20. The non-significant decrement of the norm can be seen on the deviation after posterior, meaning that 1000 prior data is not enough for this type of problems. Furthermore, this prior data is uniformly distributed random numbers, meaning that other behavior can be expected in another kind of distributions.

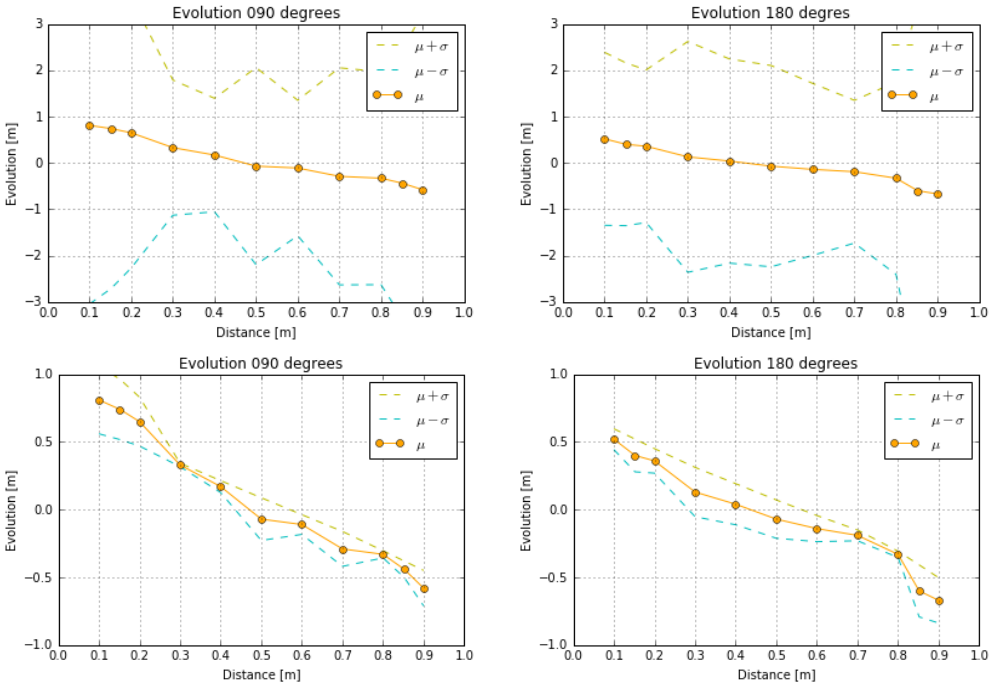


Figure 4.32: Deviation of posterior PDF from measurement data

Table 4.28: Solution of Posterior for real measurement data

P:	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	Norm
Posterior Real	8.88E-04	4.95E-03	7.22E-04	2.3622	0.2823	1.3776	1.3773	0.9180	0.4446

5 Summary and Conclusions

The present chapter entails the achievements of our investigation, some recommendations for the improvement of the automated calibration and finally the summary by chapters with the most relevant conclusions.

Achievements. *Achievement of goal .1:.* Above all the deterministic methods, the SLSQP (Sequential least Squares Programming) obtains a score of 80%, without failing any of the criteria. This method requires the construction of the Jacobian matrix and includes the possibility of the constraints function which is essential for the sediment sorting in future investigations. Its uncertainties analysis demonstrates that the probability of finding a local solution by utilizing different initial parameter sets is 64%. In the same way, the methods' robustness is high for the ones that achieve a low total discrepancy; this can be seen with the significant Kurtosis value per parameter at a small range of the relative error. The SLSQP, the Basin-Hopping, the Difference Evolution, and the Root Levenberg-Marquardt converge under the 20 iterations, some of them with a physical meaning and a low minimized objective function. *Achievement of goal .2:* The mathematical landscape of parameters functions shows a highly ill-posed problem which demonstrates the existence of multiple local minima. *Achievement of goal .3:* The main features of the Automated Calibration Routine in Python 2.7 are listed in Annex A.

Recommendations. Highly improvements are expected after objectively proposing some changes, principally in the construction of the *objective function* that includes the variables type of the measurement data and their weights. Moreover, other combinations of the correction factors, which assist the governing equations and include the parameters to be calibrated, can improve the results, as well, as the selection of the optimization method.

Objective function: Its construction can be upgraded by a proper selection of the hydrodynamic or hydro-morphological variables as the measurement and simulated data, either

the bed evolution or the water depth present similar performance. However, the combination of both variables at different locations covers the problem better. High weights for most careful observation points offer the most realistic behavior. On the other hand, the *Euclidean Norm* can be replaced by the $p-norm$:

$$L_p = \left(\sum_{i=1}^{n} |f_{x_i} - y_{obs}|^p \right)^{1/p},$$

where p can be a number between zero and infinity, selection depends on compromise.

Relaxed-real-profile: In practice, the real observation profiles are nonsmooth enough, making the methods complicate themselves due to those irregularities. For the calibration improvements, the replacement of the real measurement data is proposed for a possible, recognizable and easy relaxed-real-profile of variables only for optimization purposes. In fact, the final conclusions must be done with the measurement data. Other way of improvement is that way, the resolution of the grid of the domain has to be larger than the resolution of the measurement data.

Parameters selection: Not all parameters improve the optimization process drastically that is visible with the derivatives of the gradient-based methods; thus it is not practical to increase the dimensionality of the mathematical problem for including non-significant parameters. The most relevant parameters are the characteristic sediment diameter, the friction coefficient, the porosity, and the parameter of deviation; all the others practically do not reduce the objective function with their deltas.

Method selection: The deterministic and the stochastic methods' efficiency depends on the nature of the problem. The deterministic method performs well for defined and well-posed problems, relatively rare in nature. On the other hand, the stochastic methods are feasible when extensive prior data are given, so river systems, that have stored and accumulated the results of their model's simulations, can implement the Bayesian Inference Theory.

Storage of simulations: To create the prior information of a model, it is necessary to begin saving the data of every simulation. It is likely that the Bayesian Theorem is not suitable intermediately; however, it is a good strategy to use the deterministic methods to calibrate a model. Meanwhile, the run data is saved. Afterward, the information is enough to create a prior PDF function; it could switch from the deterministic to the stochastic calibration. Also, to create specific prior data, instead of generating it with uniformly distributed parameters, it is also reasonable to try another kind of distribution functions.

Summary and Conclusions. The *Automated Calibration for Riverflow models* is a possibility to get a precise, fast and accurate prediction of the nature system behavior, to characterize and quantify the water resources for the future human use. The present study covers not only the hydrodynamics of river flow but also the sediment transport components. The research literature is gathered to find the authors' contributions and the possible obstacles that are taken as challenges. The principal goal is to propose a robust and feasible optimization method whose solutions are physical meaningful in the hydro systems.

The current investigation is based on the hydro-morphological model described in Yen and Lee (1995) from the experimental and the numerical point of view. This model brings the understanding of a 2-dimensional controlled bend channel system; nature is not of that kind. The experimental model description is based on the geometry, the applied hydrograph, and the obtained measurement data after the experiment, specifically the bed evolution (m) over cross sections 90° and 180°. The numerical model relates all the utilized governing equations to be solved by TELEMAC2D-SISYPHE and the implemented spatial and temporal discretization scheme, as well the output variable of the model.

The mathematical definition of the optimization methods includes three types, first the deterministic, which is subdivided in the gradient and free-gradient based point of view, second the genetic methods and third the stochastic method. The gradient-based methods usually produce high computational costs due to the first and second derivative demand; contrary to the simple approach but commonly not self-adaptive of free-gradient based methods. The genetic and the stochastic methods pursue a practical solution after testing evaluated functions and rejecting process; this substitutes the Jacobian and Hessian matrices' information from the gradient-based methods. The methods must minimize or find the root of an objective function. In this case, it is defined by the *Euclidean norm* which is the square-root of the weighted sum of residuals between simulated and observed data.

An *objective function OF* is designed, that includes the synthetic measurement, which replaces the real measurement data, to test not only the mathematical optimization method but also the response and the efficiency of the numerical model due to the inserted knowledge of the global solution. The mentioned objective function is minimized by each of the optimization methods under the same conditions, meaning the same initial parameter sets, the goal tolerance and a maximal number of iterations and other particular arguments of each. The different characteristics as the computational performance, the minimization

capacity of the objective function, the discrepancies with the synthetic global solution, and others are the criteria to grade each method.

The SLSQP method obtains the highest evaluation out of all the assessed deterministic methods. The uncertainty analysis demonstrates the existence of multiple local minima since the existence of low minimized OF value along with a huge discrepancy with the synthetic global solution, for instances: the root method Levenberg-Marquardt and the Differential Evolution. The SLSQP method finds extreme solution values, even though it keeps low discrepancies with the global synthetic parameter solution.

Afterward, using the highest rated deterministic method, other possibilities of the objective functions are investigated, combining the different weights of the real measurement data and using more than one variable. Therefore, not only the bed evolution but also the water depth real considered in the objective function. On the other hand, the stochastic method evaluation based on the Bayesian Inference Theory implemented via the Bootstrap filter is used to find all parameter sets. The threshold between the simulated and the synthetic variable, in this case, the bed evolution, comply within a limit error. This filtering process is done from 1000 model simulations, denominated the Prior PDF. The solution or the Posterior PDF is contained after filtering; it is visible that the ranges of both mentioned PDFs are similar for large errors and shorter for a small error, meaning that multiple local solutions are further found in the evaluated stochastic method. The multiple local minimum assessments are done from the different combinations of parameter pair landscapes, where channels, plateaus, and irregular surface are presented. Then, the real measurement data obtained after the physical experiment is utilized to construct different sceneries of the weighted objective function to be minimized by the best grade deterministic methods and the Bayesian Inference Theory. The irregular cross section of the real measurements and the and low resolution of the grid prevent the methods from finding fast a solution.

Finally, a memorable quote of Heraclitus ingenuously explains the calibration issue, referring to the highly difficult process of finding the global solution for the same river system with different optimization methods. By analogy of an optimization method, a man steps by a river and every time he goes at the river with a different set of initial parameters, he will find a different solution, if he is not robust enough.

> "No man ever steps in the same river twice, for it is not the same river and he is not the same man."

ANNEX A: Automated Algorithm Python Routine

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jun 16 21:14:51 2016
@author:
"""
from __future__ import division
import matplotlib.pyplot as plt
import numpy as np
import subprocess
import os
import parsers.parserSELAFIN as parserSlf
import sys
import matplotlib.tri as tri
from  scipy.optimize import minimize, leastsq , basinhopping, brute,
    differential_evolution, root
print """    _          _                                  _              _
   / \   _   _| |_ ___   _ __ ___   __ _| |_ ___  __| |
  / _ \ | | | | __/ _ \ | '_ ` _ \ / _` | __/ _ \/ _` |
 / ___ \| |_| | || (_) | | | | | | (_| | ||  __/ (_| |
/_/   \_\\__,_|\__\___/|_| |_| |_|\__,_|\__\___|\__,_|

  ____          _ _ _                    _   _
 / ___|__ _| (_) |__   _ __ __ _| |_(_) ___   _ __
| |   / _` | | | '_ \ | '__/ _` | __| |/ _ \ | '_ \
| |__| (_| | | | |_) | | | (_| | |_| | (_) | | | |
 \____\__,_|_|_|_.__/|_|  \__,_|\__|_|\___/|_| |_|
 """
__authors__="Rebekka Kopmann, Leopold Stadler", "Betsaida Fernandez"

float_formatter = lambda x: "%.5f" % x
np.set_printoptions(formatter={'float_kind':float_formatter})
def FilePath(InputFile):
    return os.path.abspath(InputFile)
def UpdateParameter(File_path, lookup, nValue ):
    """Replacement of a parameter from the Cas.file"""
    File_in=open(File_path,'r')
    filedata=File_in.readlines()
    for j in range(len(lookup)):
        for line in filedata:
                if lookup[j] in line:
                    oLine=line
```

```
                    i=filedata.index(oLine)
                    filedata[i]=lookup[j] +'\t = %f \n' %nValue[j]
        File_out=open(File_path,'w')
        File_out.writelines(filedata)
        File_out.close()
        File_in.close()
    return
def Telemac2dRunner(InputFile):
    """ Telemac2d Runner takes the program and configuration file path
    to model an given tel.cas file"""
    program_path= "C:\opentelemac-mascaret\\v7p1\scripts\python27\\
       telemac2d.py" #
    config_file_path="--configfile=C:\opentelemac-mascaret\\v7p1\configs
       \\systel_v7p1.cfg"
    config_name3="--configname=win7bigbee"
    File_path=FilePath(InputFile)
    Process=subprocess.call(['python.exe',program_path,
       config_file_path,config_name3,File_path])
    return Process

def getPositionOfVariable(varnames,searchName):
   for i,name in enumerate(varnames):
      if name.strip() == searchName:
         return i
   sys.exit()
   return

def profileRef(CoordinateFile):
    data_profile =  np.genfromtxt(CoordinateFile,names=["x","y","z"])
    profileRef_X = data_profile["x"]
    profileRef_Y = data_profile["y"]
    profileRef_Z = data_profile["z"]
    ProfileREF=np.array([profileRef_X,profileRef_Y,profileRef_Z])
    return ProfileREF

def SpecificPointsRefZ(ProfileREF,SP):
    SpecificPointsZ=np.array([ProfileREF[2,SP[0]], ProfileREF[2,SP[1]],
       ProfileREF[2,SP[2]]])
    return SpecificPointsZ

def EvolutionCut(ProfileREF):
    float_formatter = lambda x: "%.5f" % x
    np.set_printoptions(formatter={'float_kind':float_formatter})
    slf = parserSlf.SELAFIN(SipypheFile)
    mesh = np.array(slf.IKLE3)
    triang = tri.Triangulation(slf.MESHX,slf.MESHY,mesh)
    varPos = getPositionOfVariable(slf.VARNAMES,'EVOLUTION')
    timePos = len(slf.tags["times"]) -1
    values = slf.getVALUES(timePos)
    h0_factor = 0.0544
    evolution = values[varPos]
    evolution_interpolator = tri.LinearTriInterpolator(triang,evolution)
    evolutionCut = evolution_interpolator.__call__(ProfileREF[0],
       ProfileREF[1])/h0_factor
    evolutionGra = evolution_interpolator.gradient(ProfileREF[0],
       ProfileREF[1])
    return evolutionCut, evolutionGra
```

```
93  def WaterDepth(ProfileREF):
        float_formatter = lambda x: "%.5f" % x
95      np.set_printoptions(formatter={'float_kind':float_formatter})
        slf = parserSlf.SELAFIN(SipypheFile)
97      mesh = np.array(slf.IKLE3)
        triang = tri.Triangulation(slf.MESHX,slf.MESHY,mesh)
99      varPos = getPositionOfVariable(slf.VARNAMES,'WATER DEPTH')
        timePos = len(slf.tags["times"]) -1
101     values = slf.getVALUES(timePos)
        h0_factor = 0.0544
103     WaterDepth = values[varPos]
        WD_interpolator = tri.LinearTriInterpolator(triang,WaterDepth)
105     WaterDepthCut = WD_interpolator.__call__(ProfileREF[0], ProfileREF
            [1])/h0_factor
        WaterDepthGrad = WD_interpolator.gradient(ProfileREF[0], ProfileREF
            [1])
107     return WaterDepthCut, WaterDepthGrad

109 def Plot(title, ProfileREF,evolutionCut, Threshold, CrossAxis, legend):
        distance_profile = ProfileREF[CrossAxis] - Threshold
111     plt.plot(distance_profile, ProfileREF[2], "o-.",color="sienna",
            label = legend)
        plt.plot(distance_profile, evolutionCut,".-",color="orange", label =
             "Simulation")
113     plt.xlabel("Distance [m]")
        plt.ylabel("Evolution [m]")
115     plt.grid()
        plt.xlim([0.0,1.0])
117     plt.ylim([-1.0,1.0])
        plt.legend()
119     plt.title(title)
        plt.show()
121     return

123 counter =[0]
    def Jacobian(x ):
125     """Manual Jacobian function, customized deltas per parameter"""
        Deltas=np.zeros((n,))
127     Deltas[0]=(np.abs(bounds[0][1]-bounds[0][0]))/18
        Deltas[1]=(np.abs(bounds[1][1]-bounds[1][0]))/48
129     Deltas[2]=(np.abs(bounds[2][1]-bounds[2][0]))/24
        Deltas[3]=(np.abs(bounds[3][1]-bounds[3][0]))/12
131     Deltas[4]=(np.abs(bounds[4][1]-bounds[4][0]))/24
        Deltas[5]=(np.abs(bounds[5][1]-bounds[5][0]))/6
133     Deltas[6]=(np.abs(bounds[6][1]-bounds[6][0]))/6
        Deltas[7]=(np.abs(bounds[7][1]-bounds[7][0]))/6
135     fun0=Function(x)
        m=fun0.size
137     JAC=np.zeros((m,n),  dtype='float')
        for i in range(n):
139         dx=np.zeros((n,), dtype='float')
            dx[i]=   Deltas[i]
141         dy =  Function(x+dx)  -fun0
            dy_dx=np.true_divide(dy  ,dx[i])
143         JAC[:,i]= dy_dx
        if m==1:
145         JAC=np.hstack(JAC)
        return JAC
```

```
147 #-------------------------------------------------------------------------
    #        Optimization Algorithm Minimize Methods
149 #-------------------------------------------------------------------------

151 def Simplex():
        options=      {'disp': True, 'maxiter': nite,   'ftol': ftol,'
            return_all': True }
153     return minimize(Function, Initial_Parameters, method='Nelder-Mead',
            options=options)
    def PolakandRibiere():
155     options={'disp': True, 'gtol': gtol,  'return_all': True, 'maxiter':
             nite, 'norm': 2.0}
        return minimize(Function, Initial_Parameters, jac=Jacobian, method='
            CG', options=options)
157 def LBFGSB():
        optionsLBFGSB={'disp': True, 'maxiter': nite,   'ftol': ftol,  '
            maxcor': 10, 'iprint': 2}
159     return  minimize(Function, Initial_Parameters,  method='L-BFGS-B',
            jac=Jacobian, bounds=bounds, options=optionsLBFGSB)
    def BFGS():
161     optionsBFGS=  {'disp': True, 'maxiter': nite,   'return_all': True,
            'norm': 0.00001 }
        return minimize(Function, Initial_Parameters,  method='BFGS', jac=
            Jacobian,options=optionsBFGS)
163 def NewtonCG():
        optionsNewton={'disp': True, 'maxiter': nite,   'return_all': True }
165     return minimize(Function, Initial_Parameters,  tol=tol, method='
            Newton-CG',jac=Jacobian, hess=None, options=optionsNewton)
    def Trun_Newton():
167     options={'disp': True,'maxiter': nite}
        return minimize(Function, Initial_Parameters, tol=tol, method='TNC',
            jac=Jacobian, bounds=bounds,   options=options) # sin la jacobina
             , ni con oraciones
169 def SLSGP ():
        optionsSLSGP ={'disp': True, 'maxiter': nite, 'iprint':1, 'ftol':
            ftol}
171     const=({'type': 'ineq', 'fun': lambda x:np.array([-x[0]  + x[1]])},
                {'type': 'ineq', 'fun': lambda x:np.array([5*x[0]- x[1]])})
173     return minimize(Function, Initial_Parameters, method='SLSQP',jac=
            Jacobian,  tol=tol,constraints=const, bounds=bounds, options=
            optionsSLSGP)
    def Trust_ncg():
175     return minimize(Function, Initial_Parameters, method='trust-ncg',
            jac=Jacobian, hess=hessian, tol=tol, options={})
    #-------------------------------------------------------------------------
177 #        Optimization methods
    #-------------------------------------------------------------------------
179 def Basinhopping():
        boundmax =np.zeros((8,))
181     for i in range( len(bounds) ):
            boundmax[i]=(bounds[i][1])
183     boundmin =np.zeros((8,))
        for i in range( len(bounds) ):
185         boundmin[i]=(bounds[i][0])
        AcceptedParametersSET=[]
187     def print_fun(x, f, accepted):
            if  accepted==True:
189             AcceptedParametersSET.append(x)
```

```
        print("at minimum %.4f accepted %d" % (f, int(accepted)))

    class MyBounds(object):
        def __init__(self, xmax=boundmax, xmin= boundmin):
            self.xmax = np.array(xmax)
            self.xmin = np.array(xmin)
        def __call__(self, **kwargs):
            x = kwargs["x_new"]
            tmax = bool(np.all(x <= self.xmax))
            tmin = bool(np.all(x>= self.xmin))
            print tmax, tmin
            return tmax and tmin

    class MyTakeStep(object):
       def __init__(self, stepsize=0.5):
           self.stepsize = stepsize
       def __call__(self, x):
           s = self.stepsize
           x[0] += np.random.uniform(0.5*s*bounds[0][0], 0.5*s*bounds
              [0][1])
           x[1] += np.random.uniform(s*bounds[1][0], s*bounds[1][1])
           x[2] += np.random.uniform(s*bounds[2][0], s*bounds[2][1])
           x[3] += np.random.uniform(s*bounds[3][0], s*bounds[3][1])
           x[4] += np.random.uniform(s*bounds[4][0], s*bounds[4][1])
           x[5] += np.random.uniform(s*bounds[5][0], s*bounds[5][1])
           x[6] += np.random.uniform(s*bounds[6][0], s*bounds[6][1])
           x[7] += np.random.uniform(s*bounds[7][0], s*bounds[7][1])
           print "Hey"
           return x
    mytakestep = MyTakeStep()
    mybounds = MyBounds()
    return basinhopping(Function, Initial_Parameters,T=0.1, niter=1,
       disp=True, niter_success=1, minimizer_kwargs={"method":"Nelder-
       Mead"}, accept_test=mybounds, take_step=mytakestep)

def LeastSquare():
    return leastsq(Function, Initial_Parameters,full_output=True, factor
       =10,ftol=0.0001, Dfun=Jacobian, maxfev=nite*(10))

def Root_Levenberg_Marquardt ():
    #diag=[1.0001, 1.0002, 1.0004, 1.5, 1.1, 1, 0.5,0.1 ]
    optionsLM={'diag': None, 'factor': 10, 'eps': eps,  'maxiter': 50,'
       xtol':xtol, 'ftol':ftol}
    return root(Function, Initial_Parameters, method='lm', jac=Jacobian,
          options=optionsLM)
def Diff_Evol():
    return differential_evolution(Function, bounds, strategy='best1bin',
        maxiter=nite, popsize=10, mutation=(0.5, 1.5), tol=0.001,
       recombination=0.7, seed=None, callback=None, disp=True, polish=
       False, init='latinhypercube' )
def BruteForce():
    return brute(Function, bounds,  Ns=3, full_output=True, disp=True,
       finish= None)

#-------------------------------------------------------------------------
#       Objective Functions
#-------------------------------------------------------------------------
```

```
def Function(Parameters):
    """ Objective Function """
    UpdateParameter(InputTelFilePath,ParameterName_, Parameters)
    UpdateParameter(InputSisFilePath,ParameterName_, Parameters)
    Telemac2dRunner(InputTelFile)
    #SP=[1,4,-2]
#    SpecificPointsSimulationZ0=np.array([evolutionCut0[SP[0]],
    evolutionCut0[SP[1]], evolutionCut0[SP[2]]], SP)
#    SpecificPointsSimulationZ=np.array([evolutionCut[SP[0]],
    evolutionCut[SP[1]], evolutionCut[SP[2]]])
#    norm30=np.sqrt((sum((SpecificPointsSimulationZ0-SpPoints0)**2)))
#    norm3=np.sqrt((sum((SpecificPointsSimulationZ-SpPoints)**2)))

    evolutionCut090, evolutionGra090=EvolutionCut(ProfileREF_090)
    evolutionCut180, evolutionGra180=EvolutionCut(ProfileREF_180)

    #Plot("90", SynthProfile_090,evolutionCut090, 46.1371, 1, "
        Sythetic")
    #Plot("180", SynthProfile_180,evolutionCut180, 104.682, 0, "Sythetic
        ")
    #normSyn10=np.sqrt((sum((evolutionCut090-SynthProfile_090[2])**2)))
    #normSyn11=np.sqrt((sum((evolutionCut180-SynthProfile_180[2])**2)))

    Plot("90", ProfileREF_090,evolutionCut090, 46.1371, 1, "Reference"
        )
    Plot("180", ProfileREF_180,evolutionCut180, 104.682, 0, "Reference")
    normSyn10=np.sqrt((sum((evolutionCut090-ProfileREF_090[2])**2)))
    normSyn11=np.sqrt((sum((evolutionCut180-ProfileREF_180[2])**2)))

    RootVector=evolutionCut090-SynthProfile_090[2]
    global norm
    norm=0.5*normSyn10+0.5*normSyn11
    if Type=="root":
        OF=RootVector
    elif Type =="minimize":
        OF=norm.copy()
    else:
        OF=norm.copy()
    return  OF


#-----------------------------------------------------------------------
#        Example: Yen and Lee
#-----------------------------------------------------------------------
InputTelFile='tel.cas'
InputSisFile='sis.cas'
ResultFile = 'tel.res'
SipypheFile='sis.res'

InputTelFilePath=FilePath(InputTelFile)
InputSisFilePath=FilePath(InputSisFile)

ParameterName_=['SEDIMENT DIAMETERS', 'FRICTION COEFFICIENT', 'ROUGHNESS
    COEFFICIENT OF BOUNDARIES', 'RATIO BETWEEN SKIN FRICTION AND MEAN
    DIAMETER', 'NON COHESIVE BED POROSITY', 'PARAMETER FOR DEVIATION', '
    BETA','SECONDARY CURRENTS ALPHA COEFFICIENT' ]
Initial_Parameters=np.zeros((len(ParameterName_),),dtype='float')
Initial_Parameters[0]=0.001
```

```
Initial_Parameters[1]=0.002
Initial_Parameters[2]=0.001
Initial_Parameters[3]=3
Initial_Parameters[4]=0.35
Initial_Parameters[5]=2
Initial_Parameters[6]=1.3
Initial_Parameters[7]=1
n=Initial_Parameters.size
#-------------------------------------------------------------------------
#       Profile Reference DAta
#-------------------------------------------------------------------------
CoordinateFile=["yen_90profilexy-koor.dat", "yen_180profilexy-koor.dat"]
ProfileREF_090=profileRef(CoordinateFile[0])
ProfileREF_180=profileRef(CoordinateFile[1])

#SpPoints090=SpecificPointsRefZ(ProfileREF_090)
#SpPoints180=SpecificPointsRefZ(ProfileREF_180)

#-------------------------------------------------------------------------
#        Synthetic Data
#-------------------------------------------------------------------------
SyntheticParameters=np.array( [0.00080, 0.00120, 0.00080,3.5, 0.4, 0.80,
    1.0, 0.85])
DataSynt=np.load('04profile_SyntheticData.npz')
DataSynt.files
SynthProfile_090=DataSynt['SynEvoCut_090']
SynthProfile_180=DataSynt['SynEvoCut_180']
#-------------------------------------------------------------------------
#         Calibration
#-------------------------------------------------------------------------
tol=1e-04 , Iterations=20, gtol=1e-04 , xtol= 1e-03
ftol=1e-03 , eps= 1e-08, nite=20
bounds=((0.0003,0.0015),(0.0003,0.005),(0.0003,0.001),(1,5),(0.25,0.5)
   ,(0.2,2.5),(1,2),(0.75,1))

Type="minimize"
simplex=  Simplex()
np.savez('simplex', simplex=simplex)
```

ANNEX B: PDFs of the Solutions Parameters after Uncertainty Analysis

The relative error data between the global solution and the proposed solution, calculated in the section 4.3.3, are used also to create the PDF of the parameters.

Nelder-Mead method probability distributed functions. The PDFs of the parameter solutions proposed by 500 optimizations with the Nelder-Mead method present mostly spread distribution proper of an negative kurtosis, see following figure :

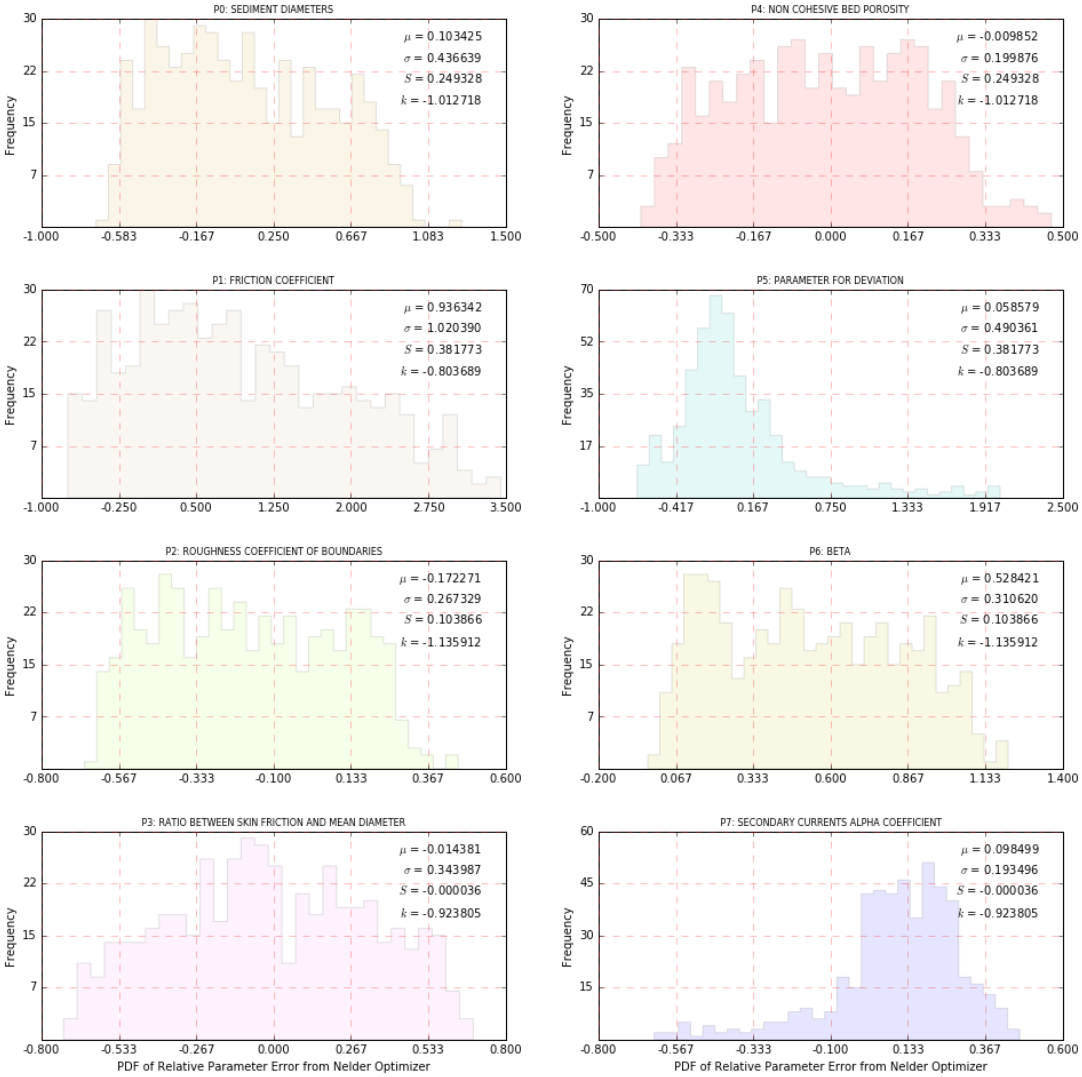


Figure B.1: PDFs of the solution parameters after 500 optimizations with the Nelder-Mead method

Polak and Ribièri method probability distributed function. The solutions are highly peaked for the physical parameter of the Characteristic sediment diameter P_0 and bed and lateral friction coefficient P_1 and P_2. However, they have extremely large discrepancies with the global solution, at seen the large relative errors. The other parameters have distributed solutions around the global solutions.

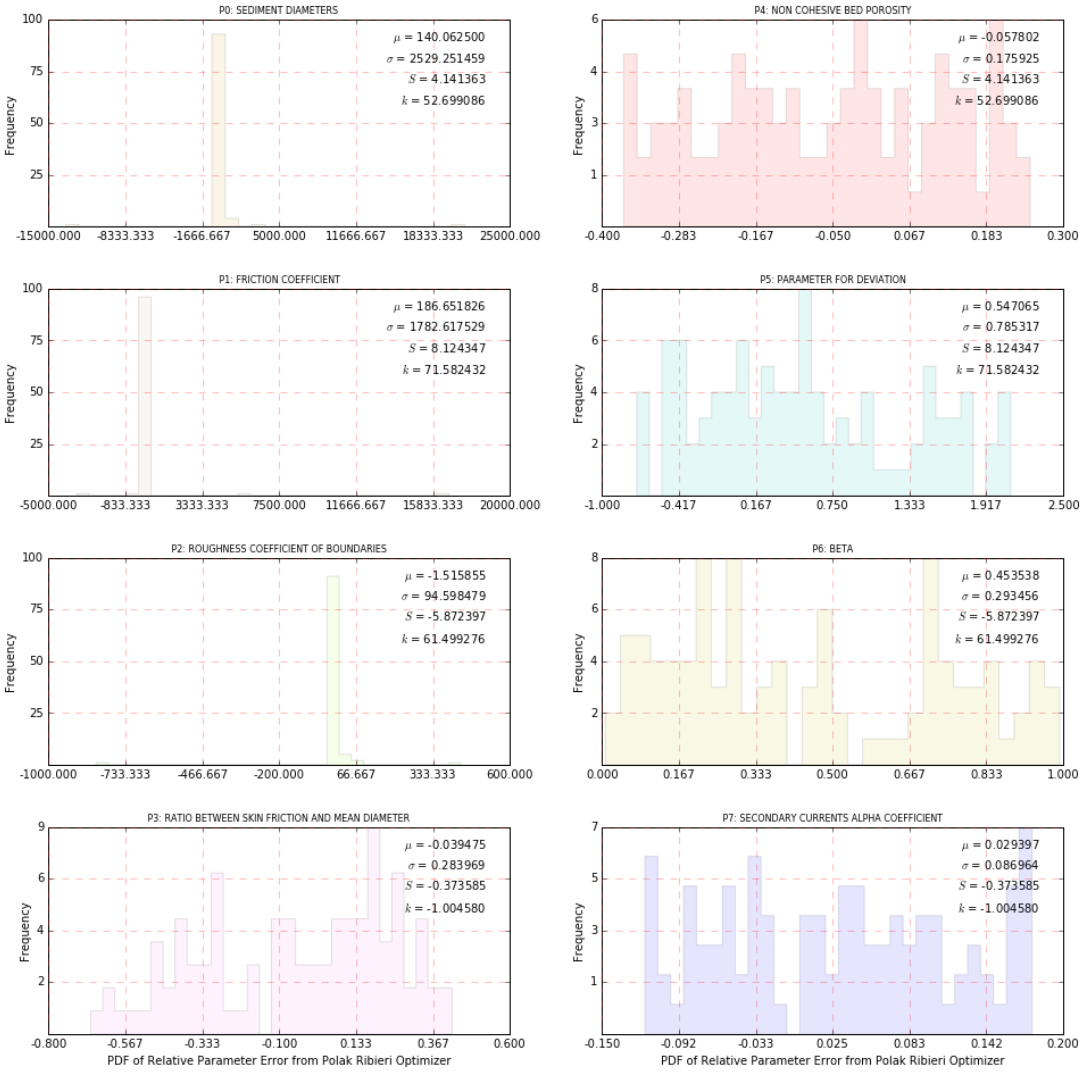


Figure B.2: PDFs of the solution parameters after 100 optimizations with the Polak and Ribièri method

L-BFGS-B method probability distributed function. The parameter solutions of 100 optimizations with the L-BFGS-B method present that the relative error have some isolate peaks aside other uniform solutions for the Characteristic sediment diameter P_0, the Roughness coefficient of boundaries P_2, the non-cohesive porosity P_4 and for the Secondary currents alpha coefficient P_7.

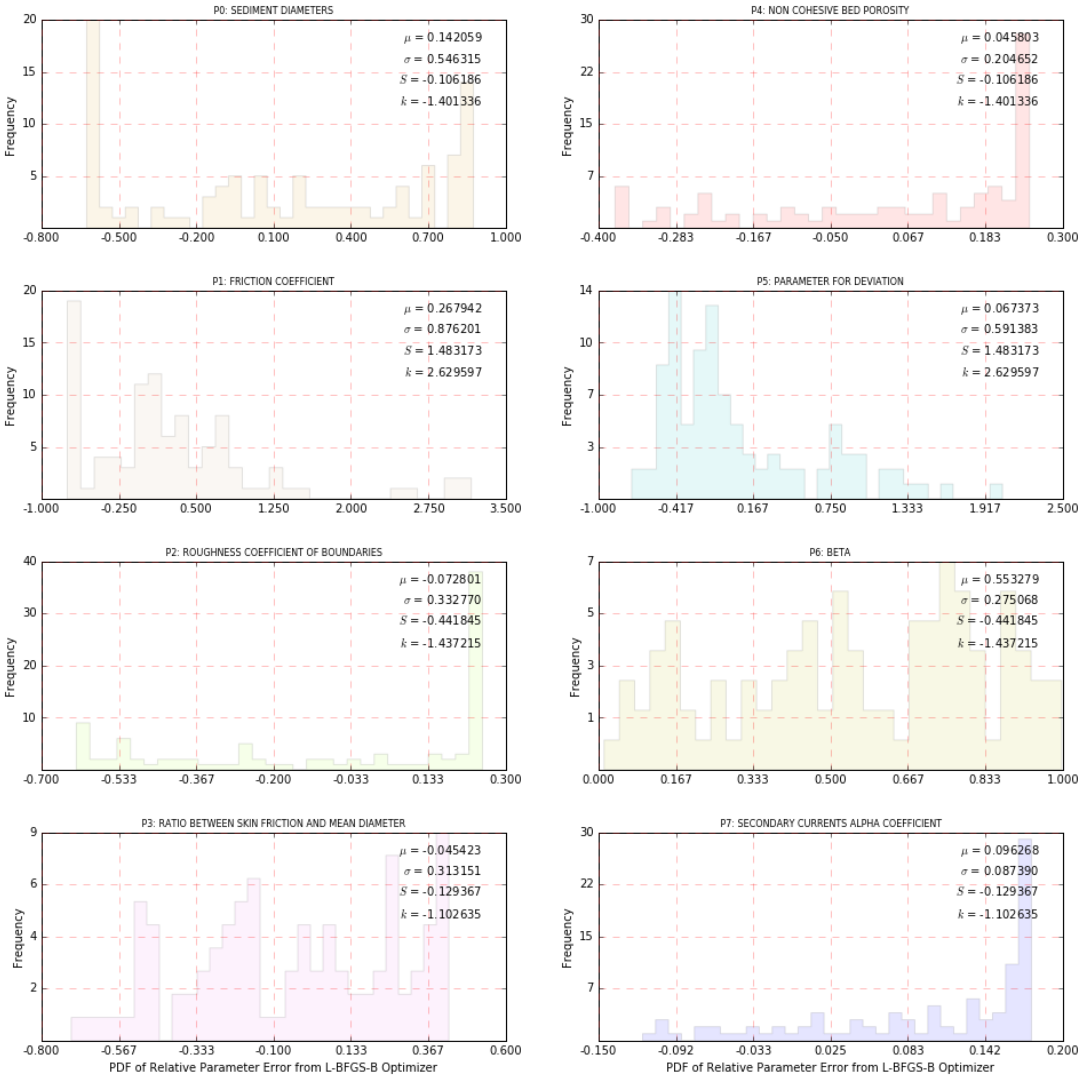


Figure B.3: PDFs of the solution parameters after 100 optimizations with the L-BFGS-B method

SLSQP method probability distributed functions. The relative error of the parameter solutions proposed by 900 optimizations with the SLSQP method and uniformly distributed random parameters as initial values per optimization, is following presented:

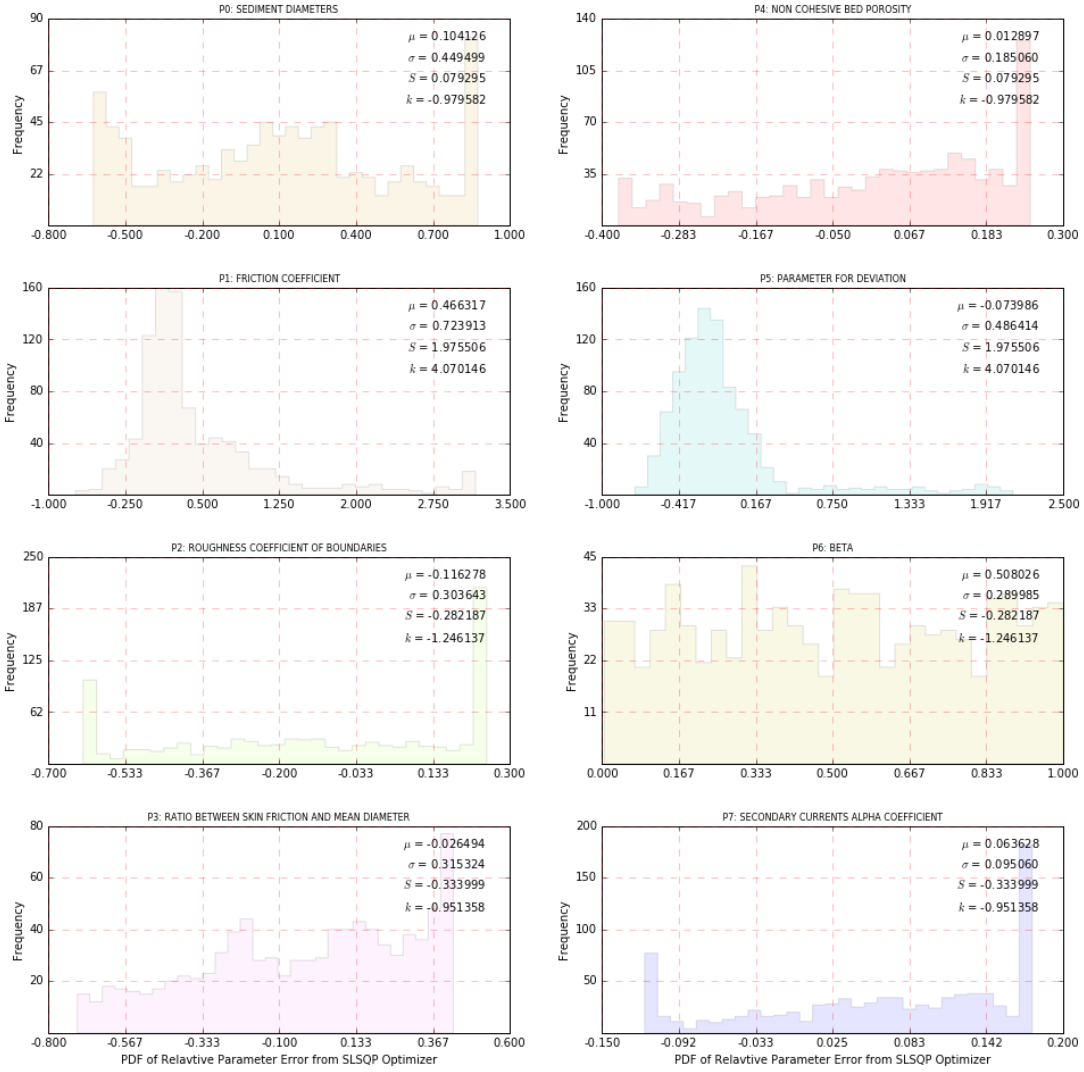


Figure B.4: PDFs of the solution parameters after 900 optimizations with the SLSQP method

Levenberg Marquardt method probability distributed functions. The PDFs of the parameter solutions proposed by 100 optimization with the Levenberg-Marquardt method, this method achieve zero values in the physical parameter range of the Characteristic sediment diameter P_0 and the Friction coefficient P_1, in the correction parameters proposes values out of boundaries. Remarkably this method insists in the same local minimum, regardless all different initial parameter sets, see in the PDFs the high kurtosis values and peaked distributions. :

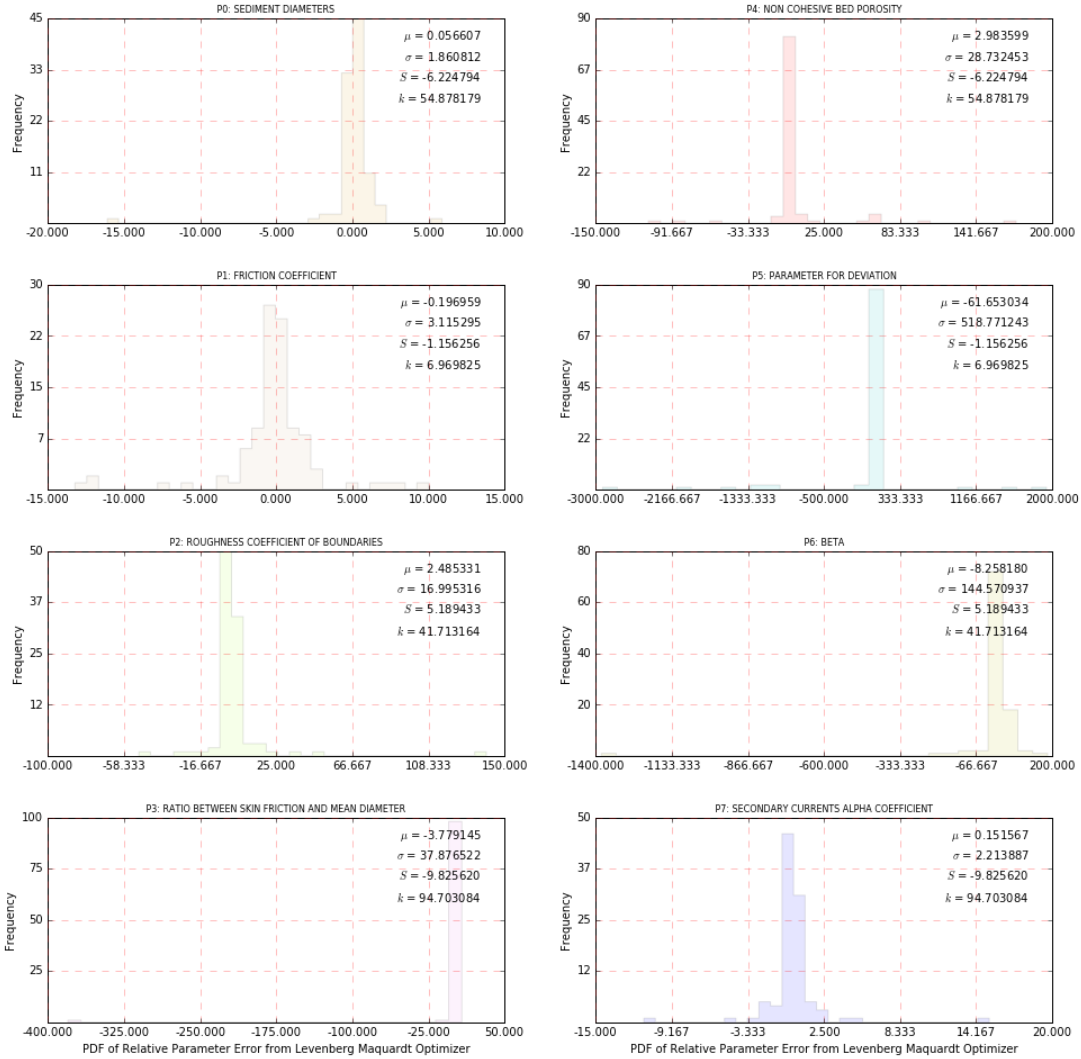


Figure B.5: PDFs of the solution parameters after 100 optimizations with the Levenberg Marquardt method

Differential Evolution method probability distributed functions The PDFs of the parameter solutions proposed by 30 optimizations with the Differential Evolution method present gaps in between the solutions. Clearly, the local minima are separated, this method highly reduces the $L_2 - norm$ at expense of large computation:

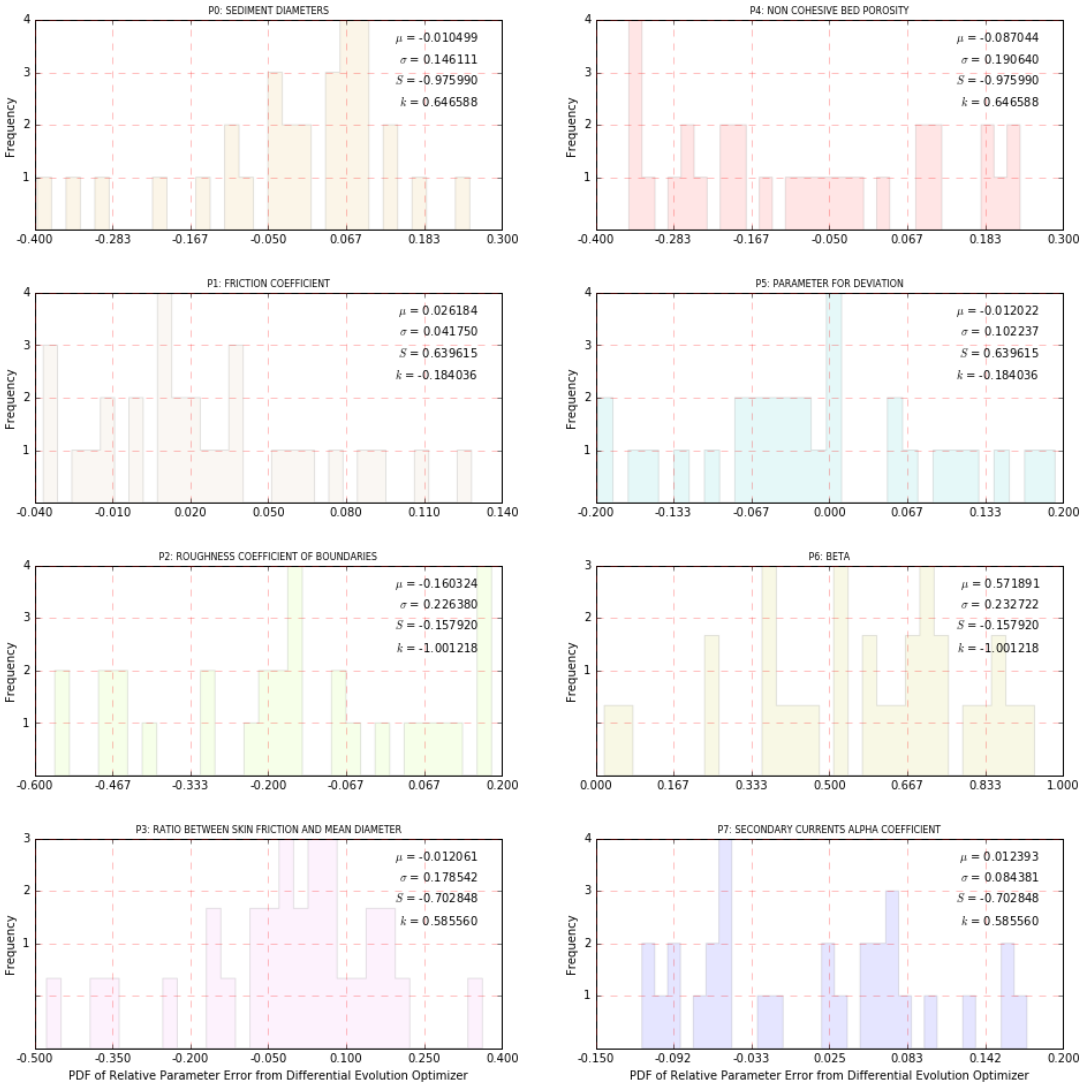


Figure B.6: PDFs of the solution parameters after 30 optimizations with the Differential Evolution method

ANNEX C: Calibration of River Rhine

The bed morphological evolution of river Rhine is taken after 500 seconds, from kilometer 738.50 to kilometer 749.30, a total distance of 10.8 kilometers. Each measurement point is located in the center of an area which dimensions correspond to a length of 100 meters and the width of the fairway stretch. The measurement value is the averaged bed morphological evolution over the analogous area.

For simplicity, a twin experiment (generated synthetic measurement data) is performed to create an existence global solution. The initial $L_2 - norm$ rises to 17.57 E-4. The elapsed time for each run of TELEMAC2D-SISYPHE is approximately one minute.

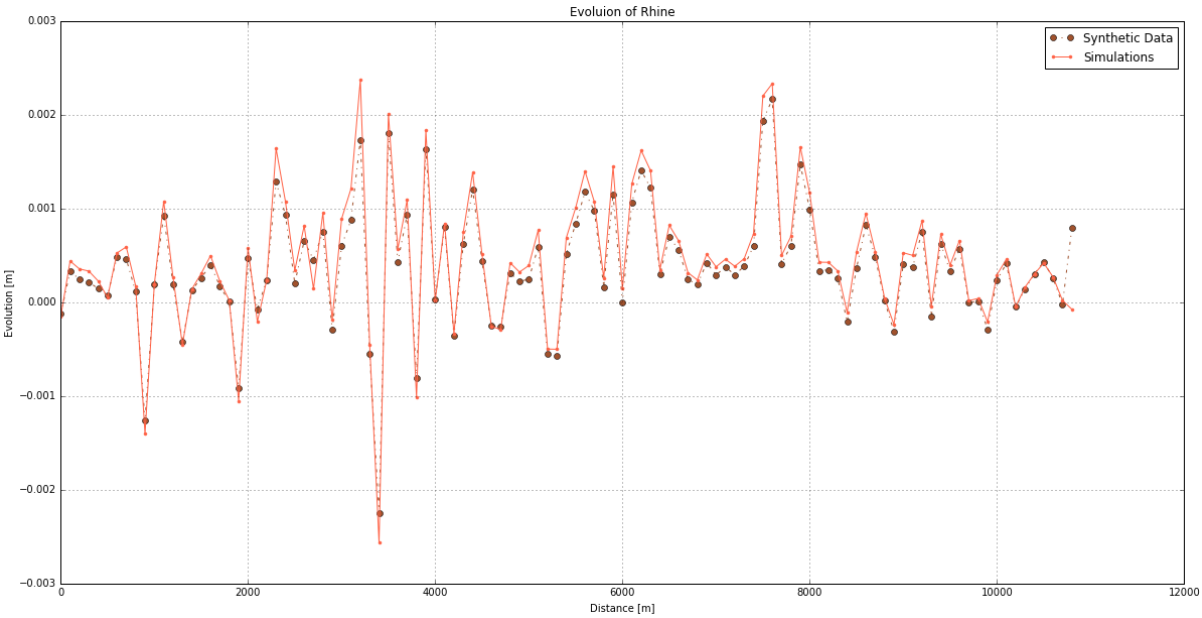


Figure C.1: The synthetic measurement of the bed evolution in River Rhine after 500 seconds, simulated with the initial parameters

Table C.1: Synthetic and initial parameters for River Rhine's calibration

Name	Synthetic	Initial	Low Bound	Upper Bound	Unit
Non cohesive bed porosity	0.3560	0.400	0.25	0.50	-
MPM coefficient	7.5640	8.000	1	10.00	-
Parameter for deviation	2.2100	1.700	0.25	2.50	-
Secondary currents alpha coefficient	0.9180	0.800	0.75	1.00	-

The optimization method to minimized the Euclidean Norm is the SLSQP method. The chosen tolerance is extremely small 1E-120, since the initial misfit is already small. The maximal number of iterations for experimental reasons is 100. The Jacobian matrix is composed by the derivative of the parameters along the investigating points of the objective function; the delta of each parameter is taken as the range between of upper and lowest bound's limits divided by 4800. Previous large number is used considering that big changes are presented in small parameter steps.

$$\frac{df}{dP} = \frac{f_{k+1} - f_k}{P_{k+1} - P_k} \tag{5.1}$$

$$P_{k+1} - P_k = \frac{UpperBound - lowerBound}{4800} \tag{5.2}$$

where f_k is the previous evaluated objective function, f_{k+1} is the actual evaluated objective function, P_k is the previous parameter, and P_{k+1} is the actual parameter.

The parameter solution is presented in the table C.2. In this table is also shown the total discrepancy with the global solution that rises an averaged value of 11.40%. The final bed morphological evolution corresponding to the solution parameter set can be seen in the figure C.2.

Table C.2: Proposed parameter solution and discrepancies with the global solution

Name	Synthetic	Solutions	Discrepancy (%)
Non cohesive bed porosity	0.3560	0.3052	14.26
MPM coefficient	7.5640	7.9297	4.83
Parameter for deviation	2.2100	1.9246	12.91
Secondary currents alpha coefficient	0.9180	0.7935	13.57

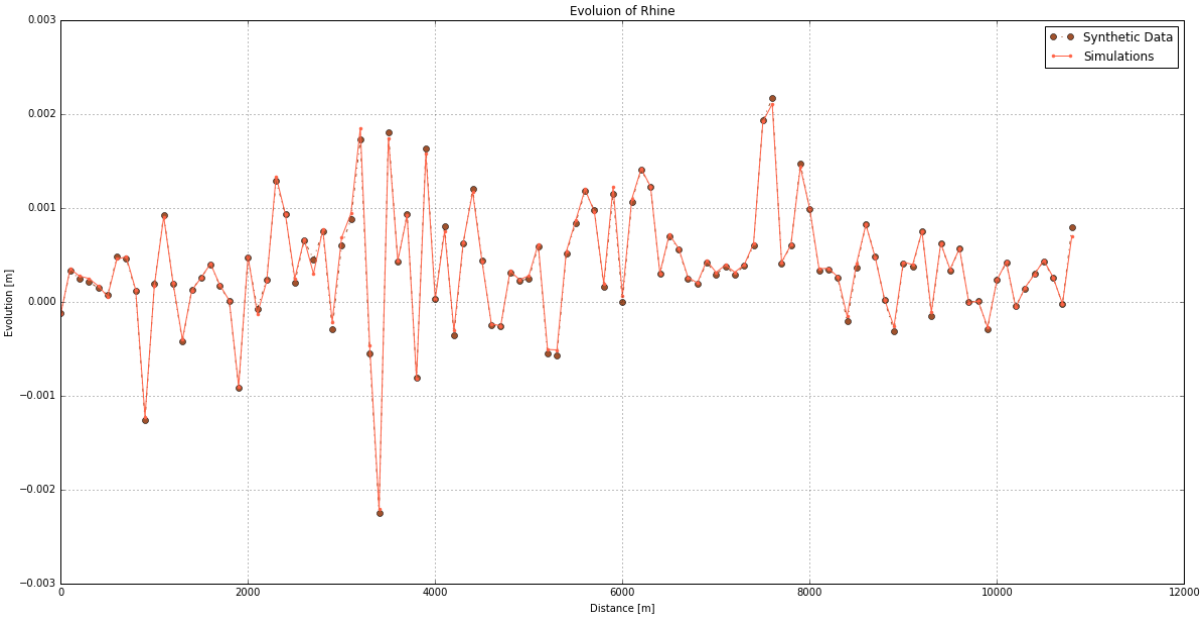


Figure C.2: The synthetic measurement of the bed evolution in River Rhine after 500 seconds, simulated with the solution parameters

The final $L_2 - norm$ is 3.7932 E-04. The reduction of the initial norm is a significant percent of 78.41% at 32 iterations which corresponds to the number of evaluated Jacobian matrices, see table C.3.

Table C.3: Computational performance of the SLSQP in the River Rhine

Performance	Value
Iterations	32
Functions calls	344
Jacs calls	32
Elapsed time	15 h

The figure C.3 presents the real measurement data of the bed evolution after 2 years and the simulated bed evolution with the initial parameters; large erosion depth and sedimentation accumulation are visible. It is a challenge for any optimization method to calibrate a real river system due to the long required computation time and non-smooth hydro-morphological problem

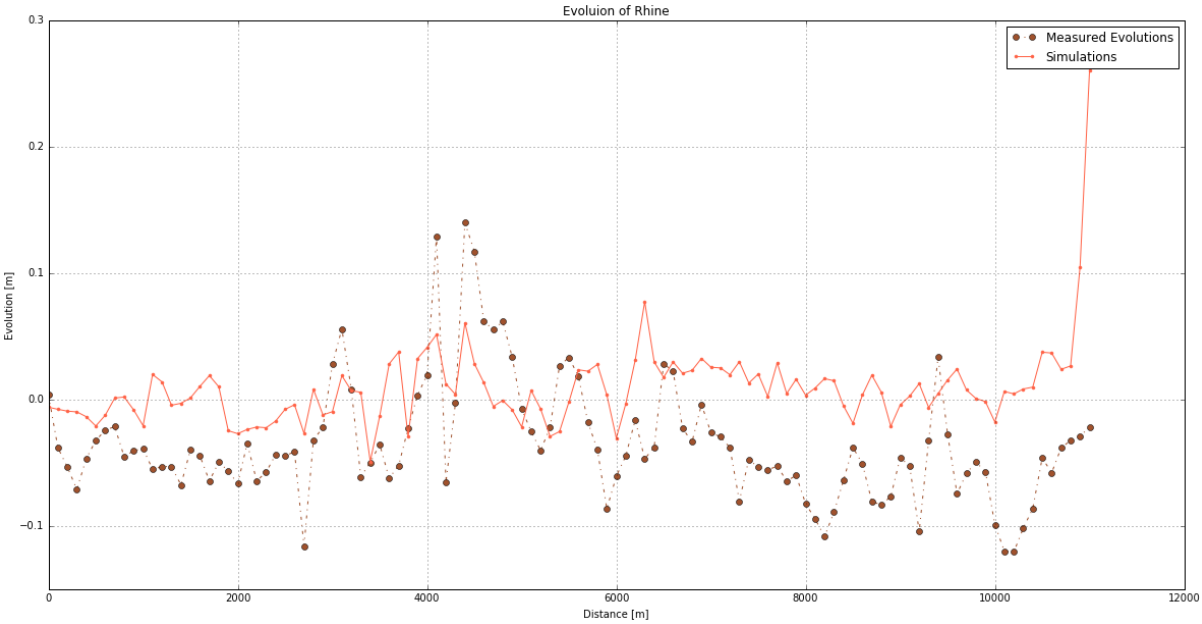


Figure C.3: The real measurement of the bed evolution in River Rhine after 2 years, simulated with the initial parameters

Bibliography

Anderson, E. J., Schwab, D. J., and Lang, G. A. Real-time hydraulic and hydrodynamic model of the st. clair river, lake st. clair, detroit river system. *Journal of Hydraulic Engineering*, 136(8):507–518, 2010.

Broyden, C. G. The convergence of a class of double-rank minimization algorithms 2. the new algorithm. *IMA Journal of Applied Mathematics*, 6(3):222–231, 1970.

Cheng, C.-T., Ou, C., and Chau, K. Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall–runoff model calibration. *Journal of hydrology*, 268(1):72–86, 2002.

Cunge, J. A. Of data and models. *Journal of Hydroinformatics*, 5(2):75–98, 2003.

Dai, Y.-H. A perfect example for the bfgs method. *Mathematical Programming*, 138(1-2): 501–530, 2013.

Duan, Q., Gupta, V. K., and Sorooshian, S. Shuffled complex evolution approach for effective and efficient global minimization. *Journal of optimization theory and applications*, 76(3):501–521, 1993.

Dung, N. V., Merz, B., Bárdossy, A., Thang, T. D., and Apel, H. Multi-objective automatic calibration of hydrodynamic models utilizing inundation maps and gauge data. *Hydrology and Earth System Sciences*, 15(4):1339–1354, 2011.

Engelund, F. Instability of erodible beds. *Journal of Fluid Mechanics*, 42(02):225–244, 1970.

Fletcher, R. A new approach to variable metric algorithms. *The computer journal*, 13(3): 317–322, 1970.

Flokstra, C. and Koch, F. Numerical aspects of bed level predictions for alluvial river bends. *Delft Hydraulics Laboratory. Netherlands, Publication*, 1981.

Galland, J.-C., Goutal, N., and Hervouet, J.-M. Telemac: A new numerical model for solving shallow water equations. *Advances in Water Resources*, 14(3):138–148, 1991.

Goldfarb, D. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.

Gupta, H. V., Sorooshian, S., and Yapo, P. O. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, 34(4):751–763, 1998.

Han, S.-P. Dual variable metric algorithms for constrained optimization. *SIAM Journal on Control and Optimization*, 15(4):546–565, 1977.

Henao, W. *An L-BFGS-B-NS Optimizer for Non-Smooth Functions*. PhD thesis, Courant Institute of Mathematical Sciences, 2014.

Hervouet, J.-M. *Hydrodynamics of free surface flows: modelling with the finite element method*. John Wiley & Sons, 2007.

Jones, E., Oliphant, T., Peterson, P., et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed 2016-08-26].

Kraft, D. et al. *A software package for sequential quadratic programming*. DFVLR Obersfaffeuhofen, Germany, 1988.

Krzysztofowicz, R. Bayesian theory of probabilistic forecasting via deterministic hydrologic model. *Water Resources Research*, 35(9):2739–2750, 1999.

Lawson, C. L. and Hanson, R. J. Solving least squares problems. 1974.

Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

Liu, D. C. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Liu, Y., Batelaan, O., De Smedt, F., Poórová, J., and Velcická, L. Automated calibration applied to a gis-based flood simulation model using pest. *Floods, from defence to management*, pages 317–326, 2005.

Madsen, H. Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. *Journal of hydrology*, 235(3):276–288, 2000.

Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

Moré, J. J. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

Nelder, J. A. and Mead, R. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Nowak, W. and Cirpka, O. A. A modified levenberg–marquardt algorithm for quasi-linear geostatistical inversing. *Advances in water resources*, 27(7):737–750, 2004.

Oladyshkin, S., Class, H., and Nowak, W. Bayesian updating via bootstrap filtering combined with data-driven polynomial chaos expansions: methodology and application to history matching for carbon dioxide storage in geological formations. *Computational Geosciences*, 17(4):671–687, 2013.

Powell, M. J. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical analysis*, pages 144–157. Springer, 1978.

Schittkowski, K. The nonlinear programming method of wilson, han, and powell with an augmented lagrangian type line search function. *Numerische Mathematik*, 38(1): 83–114, 1982.

Seong, C., Her, Y., and Benham, B. L. Automatic calibration tool for hydrologic simulation program-fortran using a shuffled complex evolution algorithm. *Water*, 7(2): 503–527, 2015.

Shanno, D. F. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.

Soulsby, R. *Dynamics of marine sands: a manual for practical applications*. Thomas Telford, 1997.

Storn, R. and Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

Sun, N.-Z. and Sun, A. *Model Calibration and Parameter Estimation: For Environmental and Water Resource Systems*. Springer, 2015.

Talmon, A., Struiksma, N., and Van Mierlo, M. Laboratory measurements of the direction of sediment transport on transverse alluvial-bed slopes. *Journal of Hydraulic Research*, 33(4):495–517, 1995.

Tassi, P. and Villaret, C. Sisyphe v6. 3 user's manual. Technical report, Tech. Rep. EDF R & D, 2014.

Varoquaux, G., Gouillart, E., Vahtras, O., Haenel, V., Rougier, N. P., Gommers, R., Pedregosa, F., Jędrzejewski-Szmek, Z., Virtanen, P., Combelles, C., et al. Scipy lecture notes. 2015.

Wales, D. J. and Doye, J. P. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

Wasantha Lal, A. Calibration of riverbed roughness. *Journal of Hydraulic Engineering*, 121(9):664–671, 1995.

Willis, R. and Yeh, W. W.-G. Groundwater systems planning and management. 1987.

Wu, W. *Computational river dynamics*. CRC Press, 2007.

Xu, C.-y. Hydrologic models. *Uppsala University, Department of Earth Sciences and Hydrology*, 2002.

Yen, C.-l. and Lee, K. T. Bed topography and sediment sorting in channel bend with unsteady flow. *Journal of Hydraulic Engineering*, 121(8):591–599, 1995.