Poncot, A.; Goeury, Cédric; Argaud, J.-P.; Zaoui, F.; Ata, Riadh; Audouin, Yoann

Optimal calibration of TELEMAC-2D models based on a data assimilation algorithm

Optimal calibration of TELEMAC-2D models based on a data assimilation algorithm

C. Goeury[1], A. Ponçot[2], J.-P. Argaud[2], F. Zaoui[1], R. Ata[1], Y. Audouin[1]
[1]EDF R&D National Laboratory for Hydraulics and Environment (LNHE)
[2]EDF R&D Performance and prediction of industrial risks for park simulation and studies (PERICLES)
6 quai Watier, 78401 Chatou, France
Email: cedric.goeury@edf.fr

***Abstract* — Numerical models are nowadays commonly used in fluvial and maritime hydraulics as forecasting and assessment tools for example. Model results have to be compared against measured data in order to assess their accuracy in operational conditions. Amongst others, this process touches on the calibration, verification and validation. In particular, calibration aims at simulating a series of reference events by adjusting some uncertain physically based parameters until the comparison is as accurate as possible. Calibration is critical to all projects based on numerical models as it requires a very large proportion of the project lifetime. In this study, the Python module TelApy of the TELEMAC system (www.opentelemac.org) has been used with the ADAO library of the SALOME platform (www.salome-platform.org) to automatically calibrate a 2D hydraulic model. based on a 3D-VAR data assimilation algorithm. The algorithm combines mathematical information contained in the hydraulic model and physical information coming from observations in order to estimate the model input parameters (bathymetry, bed friction, inflow discharge, tidal parameter, initial state and so on).**

Keywords: Shallow Water Model, Data Assimilation, Calibration

I. INTRODUCTION

Many problems in sciences and engineering require the estimation of unknown parameters that will produce a solution that best fits a finite set of indirect measurements. Examples include hydrology, oceanography and weather forecasting. This is also true for fluvial and maritime hydraulics where numerical models are used as forecasting and assessment tools for example.

In all case, model results have to be compared against measured data in order to ascertain their accuracy in operational conditions. Amongst others, this process touches on the model calibration, verification and validation tasks. In particular, model calibration aims at simulating a series of reference events by adjusting some uncertain physically based parameters until the comparison is as accurate as possible. Calibration is critical to all projects based on numerical models as it lasts over a non-negligible proportion of the project lifetime.

The objective of this work is to implement an efficient calibration algorithm, based on data assimilation and coupled with TELEMAC-2D, capable of processing measurements optimally, to estimate the partially known or missing parameters (bathymetry, bed friction, inflow discharge, tidal parameter, initial state, etc.).

Section II and III introduce the principle of the calibration algorithm and the software tools used for this work respectively. Section IV is dedicated to model results obtained from different cases. Finally, Section V, offers some conclusions and outlook.

II. CONTEXT AND PRINCIPLE

A. Context

Parameter estimation, a subset of the so-called inverse problems, consists of evaluating the underlying input data of a problem from its solution. For free surface flow hydraulics, parameters that are often unknown or difficult-to-assess include bathymetry, bed friction, inflow discharge, tidal parameter, initial state etc. The calibration of two typical projects will be used here for demonstration purposes, one schematic river application (fluvial configuration) and one actual coastal site application (maritime configuration):

- *Fluvial configuration.* In 2D hydraulic solvers such as TELEMAC-2D, the nature of the bottom of a waterway is modelled by a roughness coefficient. In some occasion, this coefficient also takes into account the friction of the walls as well as other phenomena such as turbulence. Automatic calibration is a reverse method which is used to find an "acceptable" friction coefficient (here assumed constant by zone) leading to a computed water level close to the measured water level for a fixed flow [1].
- *Maritime configuration.* Calibrating a hydrodynamic model for tide is typically an engaged and difficult process due to the tidal flow interaction between shoreline, islands, meteorological conditions,... and the lack of a reliable tidal observation stations. Thus, in addition to the friction coefficients, the tidal amplitudes at boundary locations and the water depth are considered in this work.

If done manually, model calibration is time consuming. Fortunately, the process can be largely automated to reduce human workload significantly. In the following sections, the automatic calibration task is explained.

B. Automatic calibration algorithm

Thereafter, all model parameters constitute the n-components of the control vector $X = (X_i)^T, \forall \mathrm{i} \in [1, \dots \mathrm{n}]$.

Automatic calibration is a parameter estimation or reverse method used to simulate a series of reference events by adjusting uncertain physically based parameters contained in the control vector X to produce a solution that is as accurate as possible. Therefore, the optimal search for the control vector takes a minimization form of an objective or cost function $J(X)$ given in (1).

$$\begin{cases} \mathrm{J(X)} = \mathrm{J_b} + \mathrm{J_0} \\ \mathrm{J_b} = \frac{1}{2}(\mathrm{X} - \mathrm{X_b})^{\mathrm{T}}\mathrm{B}^{-1}(\mathrm{X} - \mathrm{X_b}) \\ \mathrm{J_0} = \frac{1}{2}\big(\mathrm{Y} - \mathrm{H(X)}\big)^{\mathrm{T}}\mathrm{R}^{-1}\big(\mathrm{Y} - \mathrm{H(X)}\big) \end{cases} \quad (1)$$

where the components of X represents parameters to be estimated / calibrated, X_b represents the prior knowledge about the control vector X, Y is the observation vector, H is the so-called observation operator enabling the passage of the parameter space (where the vector X lives) to the observation space (where Y lives) such that $Y = H(X)$ and B, R are the background and observation error covariance matrices respectively.

This is a formulation of the optimal search of control vector X, which must satisfy the background error statistics (J_b), and the equivalent observation error (J_o). Eq. (1) is known as the traditional variational data assimilation cost function, called 3D-VAR [2].

Generally, optimisation methods can be used to solve minimisation problems. The former can be very different according to the form of the cost function to be minimised (convex, quadratic, nonlinear, etc.), its regularity and the dimension of the space studied. Many deterministic optimisation methods are known as gradient descent methods, among which is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi Newton method [3], [4].

For these methods, the estimation of the optimal control vector X involves minimising the objective function $J(X)$ (or finding its extremes), which requires the computation of its gradient with respect to X, defined as follows.

$$\nabla \mathrm{J(X)} = \mathcal{H}^{\mathrm{T}}\mathrm{R}^{-1}\big(\mathrm{Y} - \mathrm{H(X)}\big) \quad (2)$$

with $\mathcal{H}$ the adjoint of the observation operator H that is to say the partial derivatives of the operator's component with respect to its input parameters. It is noted that the observation operator H represents a call to the hydraulic solver, which implies that $\mathcal{H}$ represent a call to the adjoint of the hydraulics solver.

C. Constrained BFGS Quasi Newton method

The optimisation method mentioned above (BFGS) is based on the application of the Newton method to the gradient of the functional $J(X)$, which involves the computation of the first and second derivatives of the cost function. The main disadvantage of this type of approach is the computation of the second derivative $\nabla^2 J(X)$ (or Hessian) and using it to solve a linear system at every iteration of the Newton algorithm. For large problems, this is computationally out of reach.

An alternative is to use algorithms such as the Quasi-Newton algorithm which provides Hessian approximations that improve as the iterations progress, for a reasonable cost. The method chosen to perform this work is the so-called constrained Broyden Fletcher Goldfarb Shanno Quasi-Newton method (c-BFGS-QN).

The optimisation problem is formulated as follows (Eq. (3), (4) and (5)), starting with the constrained minimisation.

$$\begin{aligned} &\min_{\mathrm{X}} \mathrm{J(X)} \\ &\mathrm{X_i^{min}} \le \mathrm{X_i} \le \mathrm{X_i^{max}}, \ \forall \mathrm{i} \in [1, \dots \mathrm{n}] \end{aligned} \quad (3)$$

If X^k is the solution at stage k, a direction d^k is obtained by solving the minimization process defined in (4).

$$\min_{\mathrm{d^k}} \nabla \mathrm{J}\big(\mathrm{X^k}\big)^{\mathrm{T}}\,\mathrm{d^k} + \frac{1}{2}\mathrm{d^{kT}M d^k} \quad (4)$$

where $d_i = 0$ if $X_i \approx X_i^{min}$ or $X_i \approx X_i^{max}$ and M is the approximate Hessian matrix.

A line search is then performed along the direction d^k to find a new feasible solution X^{k+1}. Then M is modified by the BFGS formula as follows.

$$\begin{aligned} \mathrm{M^{k+1}} = \mathrm{M^k} + \\ \left(1 + \frac{\gamma^{\mathrm{kT}}\mathrm{M^k}\gamma^{\mathrm{k}}}{\delta\mathrm{X^{kT}}\gamma^{\mathrm{k}}}\right)\frac{\delta\mathrm{X^k}\delta\mathrm{X^{kT}}}{\delta\mathrm{X^{kT}}\gamma^{\mathrm{k}}} - \frac{1}{2}\frac{\delta\mathrm{X^{kT}}\big(\mathrm{M^k} + \mathrm{M^{kT}}\big)\gamma^{\mathrm{k}}}{\delta\mathrm{X^k}\gamma^{\mathrm{k}}} \end{aligned} \quad (5)$$

with $\delta X^k = X^{k+1} - X^k$ and $\gamma^k = \nabla J(X^{k+1}) - \nabla J(X^k)$

Using a constrained optimisation method makes it possible to impose boundaries during the research process of the model parameters guaranteeing their physical values. Because the inverse problem (1) is often ill-posed and unstable with available data corresponding to more than one solution, small changes in model results can lead to very different estimates for the input (calibration) parameters. These problems are related to the issue of "parameter identifiability" [5].Still, the chosen optimisation method involves computing the adjoint $\mathcal{H}$ of the observation operator H (or the partial derivatives of the operator with respect to its input parameters). In this work, the partial derivatives are approximated by using a classical finite differences method. While this remains a simple solution, numerically sensitive and computationally costly, the observation operator can be written to make use of multiprocessor parallelism in order to provide an automatic calibration algorithm efficient in term of computational cost, practical for industrial applications.

III. SOFTWARE TOOLS

The automatic calibration algorithm presented in the previous section (section II) combines different skills such as optimisation, numerical analysis, parameter estimation, and

free surface flow hydraulics. The software implementation of the algorithm has to be based, therefore, on open source and flexible architecture with reusable components. This study was performed by coupling the hydrodynamic solver TELEMAC-2D and the data assimilation library ADAO within the SALOME platform, through the component TelApy of the TELEMAC system.

A. The SALOME platform

SALOME is an open source platform (www.salome-platform.org) for pre and post processing of numerical simulations, enabling the chaining or the coupling of various software tools and codes. SALOME is developed by EDF, the CEA and OPENCASCADE S.A.S. under the GNU LGPL license. It is based on an open and flexible architecture with reusable components, which can be used together to build a computation scheme assembling each module or external codes together through specific communication protocols. In our case, the TELEMAC-2D model is driven through the TelApy component and dynamically linked to ADAO within SALOME (See Fig. 1).

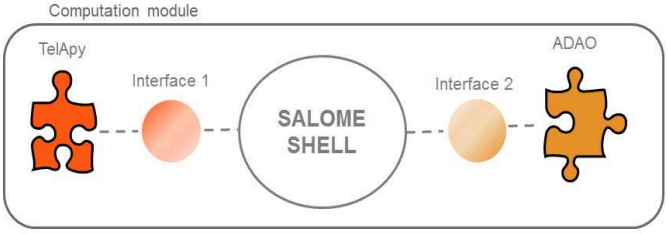

Figure 1. *The SALOME composition linking TelApy to ADAO*

B. The TelApy component of the TELEMAC system

The recently implemented TelApy component is distributed with the open source TELEMAC system (www.opentelemac.org). It aims at providing python source code that wraps and controls a TELEMAC simulation through a Fortran API (Application Program Interface) [6]. The API's main goal is to have control over a simulation while running a case. For example, it allows the user to hold the simulation at any time step, retrieve some variables and / or change them. The links between the various interoperable scientific libraries available within the Python language allows the creation of an ever more efficient computing chain able to more finely respond to various complex problems. The TelApy component has the capability to be expended to new types of TELEMAC simulations use including high performance computing for the computation of uncertainties, other optimization methods, coupling, etc.

C. The Data Assimilation library ADAO

The ADAO library provides modular data assimilation and optimization features within the SALOME platform [2]. It can be coupled with other modules or external simulation codes while providing a number of standard and advanced data assimilation or optimization methods. The ADAO library also covers a wide variety of practical applications, from real engineering to quick experimental methodologies. Its architechture and numerical scalability gives way to extend the field of application.

IV. Applications

In the previous section, the software tools used to implement the computing chain of the automatic calibration algorithm have been presented. In order to demonstrate its application some case are presented in the following section.

A. Fluvial configuration

Within this fluvial configuration, the implementation within SALOME of the automatic calibration algorithm is applied to two of the standard TELEMAC-2D test cases: the so-called "estimation" and "verysimple" test cases. The aim of this configuration is to find optimal friction coefficients based on a numerically generated synthetic data from the so-called "identical-twin-experiment", in which true state is known.

1) Parameter estimation: friction coefficient

Friction comes into the momentum equations of the shallow water equations and is treated in a semi-implicit form within TELEMAC-2D [7]. The two components of friction force are given in Eq. (6).

$$\begin{cases} F_x = -\frac{u}{2h} C_f \sqrt{u^2 + v^2} \\ F_y = -\frac{v}{2h} C_f \sqrt{u^2 + v^2} \end{cases} \quad (6)$$

where h is the water depth, C_f a dimensionless friction coefficient and u and v are the horizontal x and y components of the current velocity.

The roughness coefficient often takes into account the friction by the walls on the fluid or other phenomena such as turbulence. Thus it is difficult to define directly from available data and must be adjusted using the water surface profiles measured for a given flow rate.

2) Test case "estimation"

This first test case is based on a schematic rectangular channel of varying mesh resolution. The channel is 500 m long and 100 m wide. The finite element mesh consists of 551 triangles (Fig. 2).

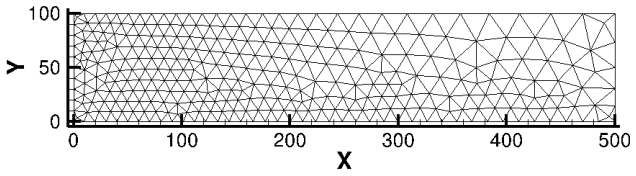

Figure 2. *Mesh and layout of the test case "estimation"*

Upstream and downstream of the model, the boundary conditions used are the imposed flow rate and water level, respectively set to 50 m^3s^{-1} and 1 m. A single Strickler friction value is set for the entire domain.

Regarding the automatic calibration, the initial guess of the Strickler is set to 15 $m^{1/3}s^{-1}$. The "observations" used to calibrate the model are synthetic water depth generated numerically with a Strickler coefficient of 35 $m^{1/3}s^{-1}$. The objective of this test case is to recover the synthetic value defined as the true state using the automatic calibration algorithm chain. The observation considered in this case are

the water depth on all mesh nodes at different times in second $T = \{2{,}000; 4{,}000;\ 6{,}000;\ 8{,}000; 10{,}000\}$.

Regarding the optimization search , the friction parameter is constrained between 5 $m^{1/3}s^{-1}$ and 90 $m^{1/3}s^{-1}$ and the differential increment is set to 10^{-4} for the computation of the observation operator approximated derivatives.

Fig. 3 shows the Strickler coefficient and the associated cost function evolution as a function of the number of iterations of the automatic minimization algorithm.

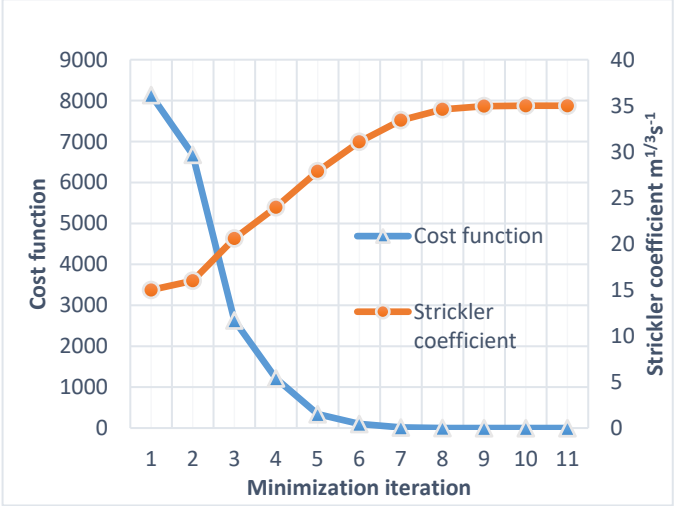

Figure 3. *Value of the cost function and the Strickler coefficient according to number of algorithmic calibration iterations*

Regarding the convergence speed of the automatic calibration method, Fig. 3 highlights a fast convergence and accuracy for the automatic calibration method developed in this work. The calibration tool finds an optimal solution in about 10 iterations with a cost function less than 10^{-2} corresponding to a Strickler coefficient of 34.998 $m^{1/3}s^{-1}$. This test case demonstrates that the data assimilation chain is well implemented.

3) Test case "verysimple"

For real applications, there are often a few measurements available at different locations. Additionally, bed properties often varies over the entire domain (rock, vegetation, sand, mud, etc.), for which a unique friction coefficient may not be appropriate.

This test case defines a limited number of zones (as many as the measurements), within which the friction coefficient is a constant parameter, independent of one another.

The "verysimple" test case is a rectangular channel of 50 m long and 1 m wide with a constant slope of 0.1 degree. The TELEMAC-2D model, constituted by a triangular mesh of some 5010 nodes (Fig. 4), has a constant discharge upstream and a water level downstream imposed, respectively set to 2.38 m^3s^{-1} and 1 m.

In this test case, 5 friction zones of equal sizes, numbered 1 to 5, are considered every 10 m of the channel. At the middle of each friction zone, one observation node is considered.

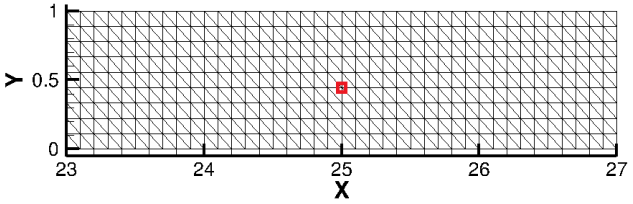

Figure 4. *Zoom of the test case "verysimple"(■observation node)*

Similar to the previous schematic application, the "observations" are water depth synthetic data generated numerically. The initial guess of Strickler (in $m^{1/3}s^{-1}$) is set to $[K_1^b = 25; K_2^b = 30; K_3^b = 35; K_4^b = 32; K_5^b = 28]$, whereas the value used to compute the synthetic data is $[K_1^t = 56.107 = K_2^t = K_3^t = K_4^t = K_5^t]$ (where the t and b exponents denote respectively true and background state).

This value of Strickler coefficient corresponds to a sub-critical flow with constant depth over the full length of the channel. The observation considered in this case are the water depth on observation nodes at different times in second $T = \{5{,}000; 10{,}000;\ 15{,}000;\ 20{,}000; 30{,}000\}$.

As for the previous test case, the optimal friction parameters are constrained between 5 $m^{1/3}s^{-1}$ and 90 $m^{1/3}s^{-1}$ and the differential increment is imposed to 10^{-4} for the computation of the observation operator approximated derivatives.

Fig. 5 shows the Strickler coefficients and the associated cost function evolution as a function of the number of minimization algorithm iterations.

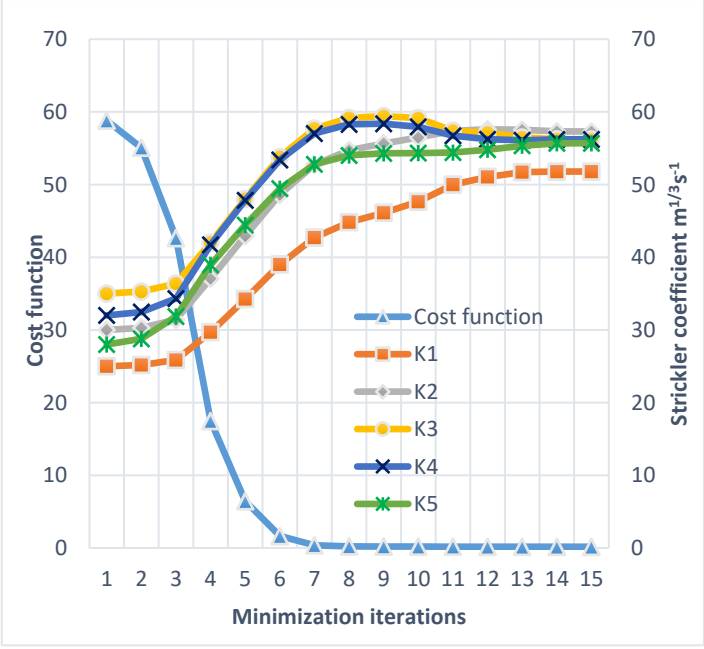

Figure 5. *Value of the cost function and the Strickler coefficients according to number of algorithmic calibration iterations*

As shown in the previous test case, the optimization process presents a fast convergence. In fact, the optimal solution is reached in about 15 iterations with a set of Strickler

coefficient values in $m^{1/3}s^{-1}$ such as $\left[K_1^{opt} = 51.79; K_2^{opt} = 57.3; K_3^{opt} = 56.21; K_4^{opt} = 56.215; K_5^{opt} = 55.69\right]$ corresponding to a cost function value less than 10^{-1}.

Furthermore, the results shown in Fig. 5 emphasise the complexity of determining a set of optimal friction coefficients. Indeed, the uniqueness of a solution of this type of problem is not mathematically proven and different sets of parameters can give analogous results. In fact, even in the framework of a simplify test case, the optimal friction parameters stabilize close but not exactly at the true state value. Thus it is important to set bounds on the search for optimal friction coefficients to avoid all the outliers and non-physical values [1]. This is why the automatic calibration method developed here uses a constrained optimisation approach.

B. Maritime configuration

A real maritime configuration is presented and calibrated using measurement data. Contrarily to the fluvial configuration, the identification of the most influential parameters was carried out through a sensitivity analysis. Indeed, it is essential to understand the relationship between the modelling inputs and the simulated variables which describe the system's dynamic. Subsequently, the automated calibration method was used for the estimation of those influential parameters.

1) Context and available data

The Alderney Race (also as known as "Raz Blanchard") is located between the Island of Alderney, UK, and the western tip of the Cotentin peninsula in Normandy, France. The maritime model includes Alderney and the tip of the Cotentin peninsula and covers an area roughly 55 km x 35 km. The finite element mesh is composed of 17,983 nodes and 35,361 triangular elements (Fig. 6). The mesh size varies from 100 m, at the shoreline and within the areas of interest, to 1.8 km offshore (western and northern sectors of the model).

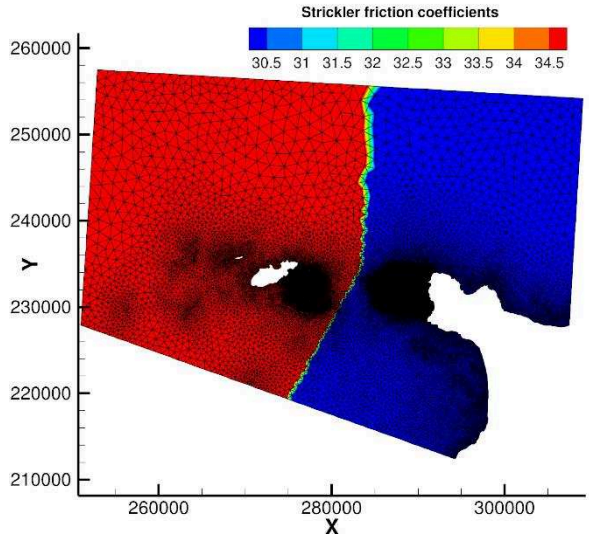

Figure 6. *Model mesh and friction coefficients (Lambert 1 North coordinate system)*

The boundary conditions of the model have been set up using depth-averaged velocities and water levels from the TPXO dataset (8 primary, 2 long-period and 3 nonlinear constituents). The TPXO dataset is an accurate global models of ocean tides based on a best-fit of tidal levels measured along remote sensing tracks from the TOPEX/POSEIDON satellite project in operation since 2002. Moreover, velocities and water depths are imposed along the marine borders of the model using Thompson-type boundary conditions that allows internal waves to leave the domain with little or no reflection.

Several measurement campaigns were carried out to the west of Cap de la Hague. For this study, only the results of one of these (a campaign lasting six months) are used over a 5 days period from October 15th to 20th. This six-month campaign was carried out during summer of 2009 (from the end of July to the end of January). Two ADCPs were deployed to measure flow velocity (magnitude and direction) and water depth with 1 measurement every 10 minutes and one hour respectively.

2) Sensitivity analysis

Calibrating a hydrodynamic model (here for a real tidal site) is typically an engaged and difficult process due to the complexity of the flows and their interaction with the shoreline, the bathymetry, islands, etc. Thus, it is essential to understand in depth the relationship between the modelling calibration parameters and the simulated state variables which are compared to the observations.

In this case, the identification of the most influential input parameters by sensitivity analysis has been led to target the calibration parameters when observations are available. In particular, both friction and tidal amplification were highlighted.

a. Friction parameter

As shown in Fig. 6, the model is composed of two friction zones, roughly along the French-UK border, where the Strickler friction coefficients are imposed to $K_1 = 30$ and $K_2 = 35$ m1/3s-1. The approach here is similar to that used in the schematic fluvial configuration.

b. Tidal amplification parameter

Tidal characteristics are imposed using a database of harmonic constituents to force the open boundary conditions. For each harmonic constituent, the water depth h and horizontal components of velocity u and v are calculated, at point M and time t by Eq. (7).

$$\begin{cases} \mathrm{F(M,t)} = \sum_i \mathrm{F_i(M,t)} \\ \mathrm{F_i(M,t)} = \\ \qquad \mathrm{f_i(t) A_{F_i}(M) \cos\left(\frac{2\pi t}{T_i} - \varphi_{F_i}(M) + u_i^0 + v_i(t)\right)} \end{cases} \quad (7)$$

where F is either the water level (referenced to mean sea level) z_S or one of the horizontal components of velocity u or v, i refers to the considered constituent, T_i is the period of the constituent, A_{F_i} is the amplitude of the water level or one of the horizontal components of velocity, $\phi_{F_i}$is the phase, $f_i(t)$ and $v_i(t)$ are the nodal factors and u_i^0 is the phase at the original time of the simulation.

The water level and velocities of each constituent are then summed to obtain the water depths and velocities for the open boundary conditions (8).

$$\begin{cases} h = \alpha \sum z_{Si} - z_f + z_{mean} \\ u = \beta \sum u_i \\ v = \beta \sum v_i \end{cases} \quad (8)$$

where z_f is the bottom elevation and z_{mean} the mean reference level.

In Eq. (8), the tidal amplitudes multiplier coefficient of tidal range and velocity, respectively α and β, at boundary locations and the sea level z_{mean} are assumed to be the tidal calibration parameters [8].

c. Analysis of variance

The sensibility analysis aims at quantifying the relative importance of each input parameter of a model. The variance-based methods aim at decomposing the variance of the output to quantify the participation of each variable. Generally, these techniques compute sensitivity indices called Sobol Indices. The definition of Sobol Indices is a result of the ANOVA (Analysis Of VAriance) variance decomposition. In fact, given a set of independent uncertain parameters $X = (X_1, \dots, X_n)$, the variance of the output $Y = M(X)$ can be expressed based on the total variance theorem by Eq. (9).

$$Var[Y] = \sum_{i=1}^{p} V_i(Y) + \sum_{i<j} V_{ij}(Y) + \cdots + V_{12\dots p}(Y) \quad (9)$$

where:

$$\begin{cases} V_i(Y) = Var[E(Y|X_i)] \\ V_{ij}(Y) = Var\left[E(Y|X_i, X_j)\right] - V_i(Y) - V_j(Y) \\ \vdots \end{cases} \quad (10)$$

The term $E(Y|X_i)$ represents the conditional expectation of the output Y under the assumption that the uncertain variable X_i remains constant. The resulting decomposition of the variance can then be used to compute the sensitivity indices called Sobol Indices. In the framework of this study, only the first order S_i and the total S_{Ti} Sobol sensitivity indices are studied. The definition of these two indices is given by Eq. (11).

$$\begin{cases} S_i = Var(E[Y|X_i]) \Big/ Var(Y) \\ S_{Ti} = Var(E[Y|X_{-i}]) \Big/ Var(Y) \end{cases} \quad (11)$$

where X_i is the uncertain variable i and X_{-i} refers to the set of uncertain input factors excluding X_i.

d. Polynomial Chaos method

In this work, a polynomial chaos expansion has been carried out to estimate Sobol sensitivity indices. This technique consists in looking for a functional representation of the output response of the system in the form of the development described by Eq. (12).

$$M(X(\zeta)) \approx M^{PC}(X(\zeta)) = \sum_{|j|<d} y_j \phi_j(\zeta) \quad (12)$$

where $\{\phi_j, j \in \mathbb{N}^p\}$ is a multivariate polynomial basis, d the maximum polynomial order and y_j adequate coefficients for the estimation of the model's response that can be determined using projection or least square methods.

In this work, the coefficients y_j are determined using the leat-square approach. In the following, the random variables in X defined in the input physical space are rescaled in the standard probabilistic space noted $\zeta \equiv \zeta(\omega) = [\zeta_1, \dots, \zeta_p]^T$ (0 mean and unit variance), to which the Polynomial Chaos framework applies. The fruitful link between Polynomial Chaos expansion and the formalism of Sobol indices has been established by [9].

e. Results

In this paper, we investigate the effect of three sources of uncertainty: the friction coefficients (K_1 and K_2), the tidal amplification coefficients along the marine boundaries (α and β) and the reference mean water level z_{mean}. All input parameters are described by uniform probability density functions such that $K_1, K_2 = U[15; 50]$ in $m^{1/3}s^{-1}$; $\alpha, \beta = U[0.7; 1.3]$ and $z_{mean} = U[-5; 5]$ in m.

To handle the sensitivity analysis with the polynomial chaos expansion, it is important to run a lot of simulations in order to have reliable results. In this work, around 2,000 Monte-Carlo computations have been carried out based on TELEMAC-2D through the SALOME platform described in [10]. Since, the sensitivity analysis results obtained for the water depth and velocity variables are similar to each other with a ranking variable without any time dependency, only the results for water depth at the last time step are presented here. Fig. 7 displays the first order and the total Sobol sensitivity indices (respectively S_i and S_{Ti}) obtained at the ADCP measurement point with a polynomial chaos expansion of degree 5.

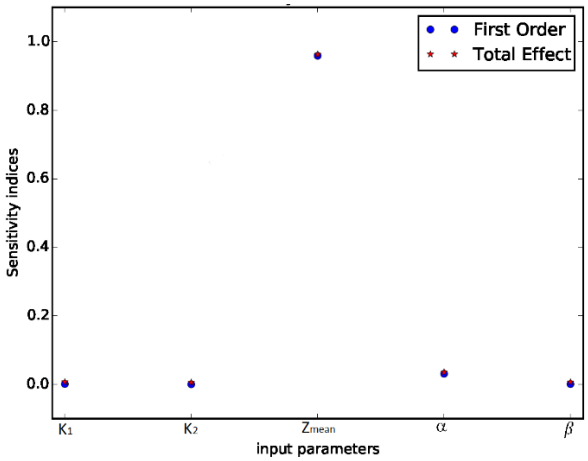

Figure 7. *First order and total Sobol sensitivity indices (respectively S_i and S_{Ti}) obtained by a degrees 5 chaos polynomial expansion*

As shown, the sea level calibration factor is by far the most influencing variable. This uncertain variable explains more than 95% of the output variation. Then, the tidal amplitude

multiplier coefficient α explains some percent of the output variation. The other variables can be considered negligible in comparison. These results depend, of course, on the hypothesis on the input random variables and especially on the choice of their distributions. Consequently, the calibration of the model is focused on these tidal parameters.

3) Tidal parameters estimation

The initial guess of the tidal parameters is set to the value $[z_{mean} = -1.6,\ \alpha = 1,\ \beta = 1]$ as prescribed by the original study with this model. The constrains of the search taken from the sensitivity analysis (section IV.B.2.e). Similarly to the fluvial configuration, the differential increment is set to 10^{-4} for the computation of the observation operator approximated derivatives.

The automatic calibration algorithm finds an optimal solution in about 16 iterations with the following set of parameters $[Z_{mean} = -0.995, \alpha = 1.112, \beta = 1.106]$.

Fig. 8 and Fig. 9 display the results of the automatic calibration over a 5 days period. As shown in Fig. 8, the water surface profiles calculated are much closer to the ADCP measurements than the original model calibration. The final results emphasises the efficiency of the automatic calibration tool in the framework of a maritime configuration.

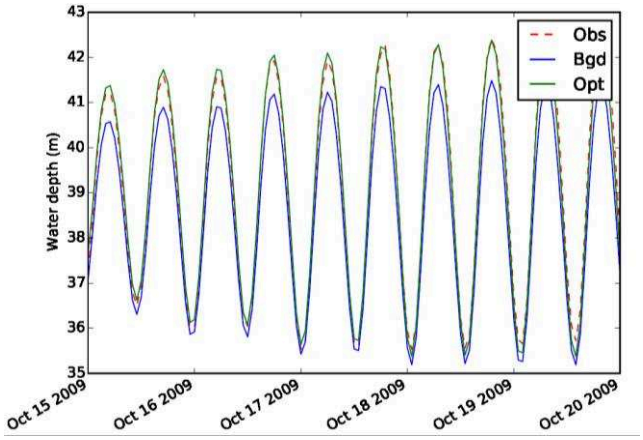

Figure 8. *Comparison of the water depth evolution with and without calibration (respectively - Opt and - Bgd) with respect to the ADCP measurements (- Obs)*

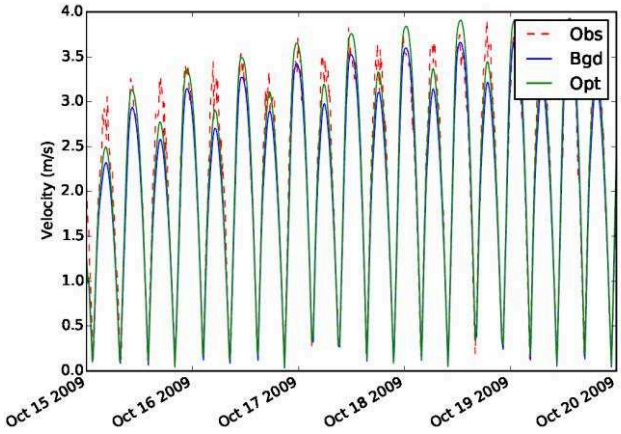

Figure 9. *Comparison of the water depth evolution with and without calibration (respectively - Opt and - Bgd) with respect to the ADCP measurements (- Obs)*

As expected, the velocity results are also closer to the measures but some differences can be observed (Fig. 9). These differences can be explained given the degree of uncertainty in the ADCP instrument or the influence of the bathymetry on the 3D structure of the flow with turbulent structure not well represented with 2D hydraulic model.

Finally, the computation time is a crucial point from operational point of view. Thus, the algorithmic calibration tool implemented in this work has been written to make use of multiprocessor parallelism in order to be efficient and compatible with industrial needs. Tab. 1 below summarizes the computation time in scalar and parallel modes obtained with processor Intel®Xeon®CPU E5-2620v3@2.40Ghz.

Number of processors	CPU time
1	9h6min26s
8	1h40min38s

TABLE 1. COMPARISON OF CPU TIME

Together with the chosen software tools and platform, the implementation of the automatic calibration algorithm remains efficient in term of computational cost for practical and industrial applications.

V. CONCLUSIONS AND OUTLOOK

Numerical models are nowadays commonly used in fluvial and maritime hydraulics as forecasting and assessment tools. Model results have to be compared against measured data in order to assess their accuracy in operational conditions. Amongst others, this process touches on model calibration, verification and validation. In particular, calibration aims at simulating a series of reference events by adjusting some uncertain physically based parameters until the comparison is as accurate as possible. However, if done manually, model calibration is time consuming. Fortunately, the process can be largely automated to reduce human workload significantly.

This article presented an automatic calibration algorithm and it implementation as a series of coupled tools. The optimal search for the parameters takes the form of minimising a cost function, which led to the implementation of a constrained BFGS Quasi-Newton method. Using a constrained optimisation method (setting bounds over the parameters) helped in finding an optimal friction coefficient in a complex, sometimes ill-posed problem for which different sets of parameters can provide analogous results.

In order to demonstrate the applicability of the automatic calibration method to converge to a known solution ("identical twin experiment"), tests on two fluvial schematic test cases were carried out. A real maritime configuration was also further calibrated against ADCP measurement with better results than originally obtained. Computation time also emphasised the efficiency of the automatic calibration tool in the framework of TELEMAC-2D.

Future works will include the replacement of the classical finite differences method, for the gradient computation of the cost function in the optimisation method, by a gradient based on algorithmic differentiation.

REFERENCES

[1] F. Demangeon, C. Goeury, F. Zaoui, N. Goutal, V. Pascual, L. Hascoet, "Algorithmic differentiation applied to the optimal calibration of a shallow water model", La Houille Blanche, n°4, pp.57-65, 2016.

[2] J.-P. Argaud, "User Documentation, in the SALOME 7.5 platform, of the ADAO module for Data Assimilation and Optimization", EDF R&D report, 2016.

[3] J. L. Morales, J. Nocedal, "L-BFGS-B: Remark on Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization", ACM Transactions on Mathematical Software, 38(1), 2011.

[4] C. Zhu, R. H. Byrd, J. Nocedal, "L-BFGS-B, FORTRAN routines for large scale bound constrained optimization," ACM Transactions on Mathematical Software, 23(4), pp. 550-560, 1997.

[5] I. M. Navon, "Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography", Dynamics of Atmospheres and Oceans, 27(1-4), pp. 59-79, 1998.

[6] C. Goeury, Y.Audouin, F. Zaoui, "User documentation v7p3 of TelApy module", 2017

[7] J-M. Hervouet, "Hydrodynamics of Free Surface Flows", Wiley, 2007, pp. 83–130.

[8] C. T. Pham, F. Lyard, "Use of tidal harmonic constants databases to force open boundary conditions in TELEMAC, Proceedings of the 19th Telemac-Mascaret User Club, 2012.

[9] B. Sudret, "Uncertainty propagation and sensitivity analysis in mechanical models. Contributions to structural reliability and stochatstic spectral methods", Accreditation to supervise research report, 2007.

[10] C. Goeury, T. David, R. Ata, S. Boyaval, Y. Audouin, N. Goutal, A.-L. Popelin, M. Couplet, M. Baudin, R. Barate, "Uncertainty Quantification on a real case with TELEMAC-2D", Proceedings of the 22nd Telemac-Mascaret User Club, 20