Design of the control set in the framework of variational data assimilation

I.Y. Gejadze, Pierre-Olivier Malaterre

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

Design of the control set in the framework of variational data assimilation

I. Yu. GEJADZE [1], P.-O. MALATERRE
UMR G-EAU, IRSTEA-Montpellier, 361 Rue J.F. Breton, BP 5095, 34196, Montpellier, France.

Abstract. Solving data assimilation problems under uncertainty in basic model parameters and in source terms may require a careful design of the control set. The task is to avoid such combinations of the control variables which may either lead to ill-posedness of the control problem formulation or compromise the robustness of the solution procedure. We suggest a method for quantifying the performance of a control set which is formed as a subset of the full set of uncertainty-bearing model inputs. Based on this quantity one can decide if the chosen 'safe' control set is sufficient in terms of the prediction accuracy. Technically, the method presents a certain generalization of the 'variational' uncertainty quantification method for observed systems. It is implemented as a matrix-free method, thus allowing high-dimensional applications. Moreover, if the Automatic Differentiation is utilized for computing the tangent linear and adjoint mappings, then it could be applied to any multi-input 'black-box' system. As application example we consider the full Saint-Venant hydraulic network model SIC^2, which describes the flow dynamics in river and canal networks. The developed methodology seem useful in the context of the future SWOT satellite mission, which will provide observations of river systems the properties of which are known with quite a limited precision.

Keywords: control set design, uncertainty quantification, variational data assimilation, 1D hydraulic network model, automatic differentiation

1 Introduction

Methods of data assimilation (DA) have become an important tool for analysis of complex physical phenomena in various fields of science and technology. These methods allow us to combine mathematical models, data resulting from instrumental observations and prior information. In particular, variational approaches have proven to be particularly useful for solving high-dimensional DA problems arising in geophysical and engineering applications involving models governed by partial differential equations. The problems of variational DA (or 'deterministic inverse problems') can be formulated as optimal control problems (see, for example, [17, 19]) to find unknown model variables such as initial and boundary conditions, source terms (forcing), distributed and lumped coefficients. Equivalently, variational DA can be considered as a special case of the maximum *a posteriori* probability (MAP) estimator in a Bayesian framework [7]. Variational DA, implemented in the form of *incremental 4D-Var* [6], is currently a preferred method for operational forecasting in meteorology and oceanography (more recently also in the form of *ensemble 4D-Var*; see, for example, [20]). In other areas of geophysics it is used in glaciology [25, 14], river hydraulics [21, 11], reservoir modelling [29] and seismic inversion [5]. Variational DA is also useful in many engineering disciplines, such as heat transfer [3], for example.

[1]Corresponding author. Email address: igor.gejadze@irstea.fr

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

In many applications the choice of the control set seems rather obvious. For example, in short-range forecasting using global atmospheric or ocean models the initial state is controlled, whereas for longer forecasting periods one must also control the forcing term to remove the model bias. When the limited-area models are considered, the boundary conditions at open boundaries are usually controlled. However, there are applications when the control set composition is not so evident, for example, in hydraulic and hydrological modeling. These are important for understanding and monitoring the fresh water cycle, local and trans-boundary management in flood and drought context and evaluation of water balance in global scale [8]. A key role in this modelling plays information about river discharges. A distinctive feature of this problem is the likely presence of significant uncertainty in distributed source terms (lateral inflows and outflows) and in model parameters, such as bathymetry, friction, infiltration rate or in those defining behavior of hydraulic structures. Indeed, properties of many rivers are known with quite a limited precision, and even for once well-studied rivers they may evolve in time due to erosion, sedimentation or structures being erected or damaged. This uncertainty, if not taken into account, could degrade the estimated discharge accuracy very noticeably.

The usual way to tackle the (systematic) uncertainties is to include all uncertainty-bearing model inputs into the control set [27, 2]. An ultimate implementation of this idea results into the model error control concept or the *weak* DA formulation [15]. Unless the available computational resources are exceeded, working with such control set is not too difficult in the variational DA framework. However, there are clear reasons for limiting the number of control variables included into the control set. First we note that when a certain input is added into the control set, the corresponding constraints should be added to keep well-posedness of the problem formulation. In the framework of unconstrained minimization those are in the form of penalty terms added to the cost function. For some variables constructing such terms is possible, whereas for other variables the inequality constraints must be explicitly introduced, in which case the very nature of the minimization problem would be changed. Solving such problem requires notably more iterations which could be a serious drawback if the time when the results remain usable is limited.

There are even more delicate reasons. For example, if for a certain dynamical model solvability of the initial state control problem has been established, solvability of the joint state-parameter control problem is not warranted. Such problems are nonlinear even for a linear dynamical model, whereas for a nonlinear dynamical model the overall nonlinearity level would grow. This means losing convexity, decreasing the convergence radius around the global minimum, multiplying the local minima, etc. That is, the control problem becomes far more difficult to solve in practice. There is one more reason. In order to use the gradient-based unconstrained minimization we assume that the control-to-observation mapping exists and is continuous everywhere in a vicinity of the reference (true) value, i.e. the operator domain is dense around the truth and the initial guess belongs to this vicinity. In practice, some combinations of the model inputs may arise in the course of minimization such that the control-to-observation mapping does not exist. For example, in hydraulic modelling some combinations of bathymetry, friction and source terms may lead to local super-critical flow conditions. These conditions, however, are not supported by models which utilize the Preissmann discretization scheme [28]. In this case the model execution stops and the minimization process has to be restarted from a different point. This is an additional complication to the minimization procedure which is better to avoid if possible. Moreover, due to the problem nonlinearity, the unwanted combinations of

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

controls cannot be easily blocked using inequality constraints.

Taking into account all the above-mentioned reasons one may conclude that for certain DA problems the control set has to be chosen carefully. In one hand, it should provide the model predictions of a reasonable quality, on the other hand - guarantee the robustness and feasibility of the solution procedure. This is the meaning of the the notion '*control set design*' used in this paper.

In the Gaussian framework, the uncertainty quantification (UQ) method for observed systems (i.e. systems for which the posterior control estimates are available) includes two basic steps: a) computing the posterior covariance matrix of the control vector; b) computing the variance in chosen quantities of interest (QoI) using the posterior covariance and the control-to-QoI mapping. Under the assumption that the uncertainty propagation is well described by the tangent linear (TL) model (i.e. the nonlinearity of the mappings is mild or perturbations are small), the latter is actually used to represent the control-to-QoI mapping, whereas the posterior covariance is approximated by the inverse Hessian of the cost function (linearized or complete). Since such a UQ method relies on the same principles as variational DA, it seems reasonable calling it the *variational* UQ method. Combining the variational DA and variational UQ methods results into variational filtering [4]. Recent examples of the variational UQ method being applied to different problems can be found in [16, 14, 1].

The method suggested in this paper presents a generalization of the variational UQ method in the following respect. We divide the full set of the uncertainty-bearing model inputs in two parts. One part is considered as active controls (the active set), whereas the remaining inputs are fixed at their priors (the passive set). Next, we define a spatially distributed goal-function and its standard deviation (SD) as the *uncertainty measure*. Clearly, the passive set contributes to this measure both directly and via the posterior covariance of the active set. Our method allows both contributions (to the uncertainty measure) to be properly evaluated. We define a *sufficient* control set as a set for which this measure takes a value useful from the practical point of view. All possible active sets have to be examined, then ranked by the associated uncertainty measure level to reveal all sufficient control sets. The choice among these sets should be done in favour of those which will not corrupt the performance of the minimization algorithm.

The implementation of the method is matrix-free, hence it could be suitable for high-dimensional problems. Furthermore, if the Automatic Differentiation is used for producing the tangent linear and adjoint mappings, then our method could be applied to any multi-input black-box system. The suggested method is new and no references describing a similar development has been found. There might be an implicit relation to the IFP method suggested in [18], but the results of the latter are difficult to interpret in terms of the control set design.

The method has been implemented with the full Saint-Venant hydraulic network model SIC^2 (Simulation and Integration of Control for Canals) developed at IRSTEA-Montpellier [22]. More detailed description of SIC^2 software is provided in Appendix I. In certain practical configurations currently installed the model includes up to 10^4 computational sections, i.e. it can be considered as a high-dimensional model. Note that the flow behavior in rivers or canals is largely defined by the boundary conditions and source terms, i.e. the problem of estimating the time-dependent controls is of a major interest here. For this type of problems variational DA approach is superior to sequential estimation methods in terms of the estimation accuracy. We report on implementation of variational DA with SIC^2 in [11]. The latter work has been partly motivated by the future SWOT satellite mission, which is going to provide water level,

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

width and slope observations of some river networks, the properties of which are known with a great deal of uncertainty. This is where the control set design procedure may become particularly useful. In the presented paper the numerical experiments have been conducted using a relatively simple 'academic' benchmark. However, some interesting fundamental conclusions on the sufficient control set has been drawn. Let us underline that this is the first time when variational UQ has been applied to the full Saint-Venant hydraulic network model.

The paper is organized as follows. In §2 and §3 we introduce the basics of the variational UQ method for observed systems. A generalization of this method, which implies that the full set of the uncertainty bearing model inputs is divided into the active and passive sets, is described in §4. The matrix-free implementation of the generalized variational UQ method is given in §5 and §6. Next, in §7 and §8 we describe the hydraulic model used for validation and details of the model implementation. Results of numerical analysis for two test problems are presented in §9. The main results of the paper are summarized in the Conclusions. There are also two appendixes: Appendix I describes the SIC^2 software and Appendix II - conceptual steps of applying Automatic Differentiation for computing the gradient, the Hessian and the goal-function uncertainty.

2 Goal-function error in an observed system

Let us consider the numerical model which describes behavior of a system in terms of its state variables $X \in \mathcal{X}$. The full set of the model inputs $U \in \mathcal{U}$ shall be called the 'full control'. Thus, the model can be considered as a control-to-state mapping $\mathcal{M} : \mathcal{U} \to \mathcal{X}$

$$X = \mathcal{M}(U), \tag{2.1}$$

where $\mathcal{U}$ and $\mathcal{X}$ are the control and state spaces, correspondingly.

For modeling the system behavior the true full control vector $\bar{U}$ must be specified. Under the 'perfect model' assumption the following can be postulated: $\bar{X} = \mathcal{M}(\bar{U})$. In reality, some components of $\bar{U}$ contain uncertainties $\eta \in \mathcal{U}$. Thus, instead of $\bar{U}$ we use its best available approximation or background

$$U^* = \bar{U} + \eta, \tag{2.2}$$

where η is also called the background error. Because of the presence of η, the predicted state $X|U^*$, that is, X evaluated (or conditioned) on U^*, also contains an error $\delta X = X|U^* - X|\bar{U}$.

In many practical situations some functionals of the state are of major interest. They are usually called the Quantities of Interest (QoI). Thus, we introduce a vector of QoI, or the goal-function $G = \{G_i,\ i = 1, \ldots, K_G\} \in \mathcal{G}$, such that

$$G = D(X), \tag{2.3}$$

where $\mathcal{G}$ is a 'goal' space and $D : \mathcal{X} \to \mathcal{G}$ is a linear or nonlinear mapping. Because of the prediction error δX there exists the goal-function error

$$\delta G = D(X) - D(\bar{X}) = D(\mathcal{M}(U)) - D(\mathcal{M}(\bar{U})). \tag{2.4}$$

This error represents uncertainty in X in a practically valuable way.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

The state observing tools are represented by an observation operator $C : \mathcal{X} \to \mathcal{Y}$ in the form

$$Y = C(X) = C(\mathcal{M}(U)) := R(U), \tag{2.5}$$

where $R : \mathcal{U} \to \mathcal{Y}$ is a generalized control-to-observations mapping and $\mathcal{Y}$ is the 'observation' space. The true observations would be $\bar{Y} = R(\bar{U})$, however the actual observations usually contain noise ξ (observation uncertainty), i.e.

$$Y^* = \bar{Y} + \xi. \tag{2.6}$$

In many circumstances the level of the goal-function error δG which corresponds to the prior guess $U = U^*$ is not acceptable. The aim of data assimilation is to obtain $\hat{U} = U|Y^*$, i.e. an estimate of U conditioned on observations Y^*, which should be better than the prior U^* in the sense $\|\hat{U} - \bar{U}\| < \|U^* - \bar{U}\|$. We shall consider the system as fully/partially identifiable if the goal-function error

$$\delta G = D(\mathcal{M}(\hat{U})) - D(\mathcal{M}(\bar{U})) \tag{2.7}$$

falls (fully/partially) into the margins defined by certain practical requirements.

Remark 1. In the above considerations the perfect model is assumed. This allows us to write $\bar{X} = \mathcal{M}(\bar{U})$ and, subsequently, $\bar{Y} = R(\bar{U})$. What if the model is not perfect? Let us consider, for example, a dynamic system

$$\frac{\partial \varphi}{\partial t} = \mathcal{F}(\varphi), \;\; t \in (0, T), \;\; \varphi|_{t=0} = u, \tag{2.8}$$

where $\mathcal{F}$ is a true spatial operator. This system is described by a model

$$\frac{\partial \varphi}{\partial t} = F(\varphi), \;\; t \in (0, T), \;\; \varphi|_{t=0} = u, \tag{2.9}$$

where F is an approximation (both in terms of physics and discretization) to $\mathcal{F}$. It is easy to see that the exact behavior of the system (2.8) can be modelled by equation

$$\frac{\partial \varphi}{\partial t} = F(\varphi) + \mu, \;\; t \in (0, T), \;\; \varphi|_{t=0} = u,$$

where $\mu = \mathcal{F}(\varphi) - F(\varphi)$ is the model error. Most certainly μ is not available for direct modelling, however, if considered as a part of the extended control vector $U = \{u, \mu\}$, it allows the mapping F to be considered 'perfect'. In this case the approach presented below is also applicable.

3 Goal-function error variance for full control

In the Bayesian framework the posterior probability density of U conditioned on observations Y^* is given by the Bayes formula

$$p(U|Y^*) = \frac{p(Y^*|U)p(U)}{p(Y^*)}. \tag{3.10}$$

Looking for the mode of the posterior density $p(U|Y^*)$, i.e. maximizing $p(U|Y^*)$, is the essence of variational data assimilation. Under the Gaussian assumption on the prior and observation

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

uncertainties, i.e. $\eta \sim N(0, B)$, $\xi \sim N(0, O)$, where B is the background error covariance and O - the observation error covariance, maximizing $p(U|Y^*)$ is equivalent to minimizing the cost-function

$$J(U) = \frac{1}{2}\|O^{-1/2}(R(U) - Y^*)\|^2_{\mathcal{Y}} + \frac{1}{2}\|B^{-1/2}(U - U^*)\|^2_{\mathcal{U}}. \tag{3.11}$$

Thus, the estimate $\hat{U}$ is obtained from the optimality condition

$$J'_U(\hat{U}) = 0. \tag{3.12}$$

For the operator $R(U)$ we define the tangent linear operator $R'(U)$ (Gateaux derivative) and its adjoint $(R'(U))^*$ [24] as follows:

$$R'_U(U)w = \lim_{t\to 0} \frac{R(U + tw) - R(U)}{t}, \tag{3.13}$$

$$(w, (R'_U(U))^* w^*)_{\mathcal{U}} = (R'_U(U)w, w^*)_{\mathcal{Y}}\,. \tag{3.14}$$

Given the above operator definitions, the full gradient of $J(u)$ in (3.12) can be expressed in the form:

$$J'_U(U) = (R'_U(U))^* O^{-1}(R(U) - Y^*) + B^{-1}(U - U^*). \tag{3.15}$$

Thus, the estimate $\hat{U}$ is the solution to the operator equation

$$(R'_U(\hat{U}))^* O^{-1}(R(\hat{U}) - Y^*) + B^{-1}(\hat{U} - U^*) = 0. \tag{3.16}$$

Let us consider an estimation error $\delta U = \hat{U} - \bar{U}$. We notice that

$$R(\hat{U}) - Y^* = R(\hat{U}) - (R(\bar{U}) + \xi) = R'_U(\tilde{U})\delta U - \xi,$$

where $\tilde{U} = \bar{U} + \tau\delta U,\ \tau \in [0, 1]$, and

$$\hat{U} - U^* = (\hat{U} - \bar{U}) - (U^* - \bar{U}) = \delta U - \eta.$$

Then, equation (3.16) yields the error equation

$$(R'_U(\hat{U}))^* O^{-1}(R'_U(\tilde{U})\delta U - \xi) + B^{-1}(\delta U - \eta) = 0. \tag{3.17}$$

Using the first order approximations $\hat{U} = \tilde{U} \approx \bar{U}$ we express δU as follows:

$$\delta U \simeq H^{-1}(\bar{U})((R'_U(\bar{U}))^* O^{-1}\xi + B^{-1}\eta), \tag{3.18}$$

where

$$H(\bar{U}) = (R'_U(\bar{U}))^* O^{-1} R'_U(\bar{U}) + B^{-1} \tag{3.19}$$

is the Hessian of an auxiliary control problem (not to be confused with the Hessian of the cost function (3.11)). We assume that H is positive definite and, hence, invertible. If the errors ξ and η truly satisfy the conditions $\eta \sim N(0, B)$, $\xi \sim N(0, O)$, then the estimation error covariance is

$$P = E[\delta U \delta U^T] \simeq H^{-1}(\bar{U}).$$

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

The above relationship is exact for linear R. For nonlinear R it is valid for small errors ξ and η.

Let us consider the goal-function error. For small errors equation (2.7) can be linearized as follows:

$$\delta G = D'_X(\bar{X})\mathcal{M}'_U(\bar{U})\delta U. \tag{3.20}$$

The error δU is not known by itself, but we may know its statistical properties, for example, let us assume $\delta U \sim N(0, V_{\delta U})$. Then

$$V_{\delta G} := E[\delta G \delta G^T] = D'_X(\bar{X})\mathcal{M}'_U(\bar{U})V_{\delta U}(\mathcal{M}'_U(\bar{U}))^*(D'_X(\bar{X}))^*. \tag{3.21}$$

The square roots of the diagonal elements of $V_{\delta G}$ describe the confidence interval for δG. For an unobserved system $V_{\delta U}$ is equal to the background (prior) covariance B. For an observed system (after data assimilation), the uncertainty in U is given by the estimation error covariance, i.e. $V_{\delta U} = P$.

Remark 2. The procedure of computing the inverse of the Hessian $H^{-1}(\cdot)$ in (3.19) is as follows. First, we define the projected Hessian

$$\tilde{H}(\bar{U}) = (B^{1/2})^* H(\bar{U}) B^{1/2} = I + (B^{1/2})^*(R'_U(\bar{U}))^* O^{-1} R'_U(\bar{U}) B^{1/2}. \tag{3.22}$$

It can be seen that all eigenvalues of $\tilde{H}(\bar{U})$ are greater than or equal to one. Furthermore, it has been observed that, for many practical DA problems, only a relatively small percentage of the eigenvalues are distinct enough from unity to contribute significantly to the Hessian. This suggests using limited-memory representations of the discrete Hessian, where this structure in the spectrum is exploited. Specifically, a few leading eigenvalue/eigenvector pairs λ_i, W_i are computed (typically using the Lanczos method as $\tilde{H}$ is available in operator-vector product form) and, for any power γ, $\tilde{H}^\gamma$ is replaced by the approximation

$$\tilde{H}^\gamma(\bar{U}) \simeq I + \sum_{i=1}^{L_H} (\lambda_i^\gamma - 1) W_i W_i^*. \tag{3.23}$$

Given $\tilde{H}$, the inverse Hessian can be easily recovered using

$$H^{-1}(\bar{U}) = B^{1/2} \tilde{H}^{-1}(\bar{U})(B^{1/2})^*.$$

4 Goal-function error variance for partial control

The theory considered so far is known and can be found in the literature (possibly, in a fragmented form). In what follows we present a new theory and a new implementation approach. That is, we have previously considered DA when the control vector U includes the full set of uncertainty-bearing model inputs. In practice, only a few selected inputs could be included (the *active set*), with the remaining inputs being fixed at their priors (the *passive set*).

Let us define the active set $U_a \in A$ of the full control vector, then $U_p = U \setminus U_a$, and the active set prior $U_a^* \in A$, then $U_p^* = U^* \setminus U_a^*$. Let us assume that the background error covariance B is block-diagonal, i.e. errors in different control variables are not correlated. This is often the

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

case naturally, otherwise the DA problem can be easily re-formulated in uncorrelated variables. Thus, the covariance B has the following structure:

$$B = \begin{pmatrix} B_a & 0 \\ 0 & B_p \end{pmatrix},$$

where B_a and B_p are the sub-matrices which correspond to the active and passive sets, correspondingly. The DA problem involving the active control set consists of minimizing the cost function

$$J(U_a) = \frac{1}{2}\|O^{-1/2}(R(U_a, U_p^*) - Y^*)\|_{\mathcal{Y}}^2 + \frac{1}{2}\|B_a^{-1/2}(U_a - U_a^*)\|_A^2. \tag{4.24}$$

Thus, the estimate $\hat{U}_a$ is obtained from the optimality condition

$$J'_{U_a}(\hat{U}_a) = 0. \tag{4.25}$$

Given the above operator definitions in (3.13) and (3.14), the gradient of $J(U_a)$ can be expressed in the form:

$$J'_{U_a}(U_a) = (R'_{U_a}(U_a, U_p^*))^* O^{-1}(R(U_a, U_p^*) - Y^*) + B_a^{-1}(U_a - U_a^*), \tag{4.26}$$

thus the estimate $\hat{U}_a$ must satisfy the operator equation

$$(R'_{U_a}(\hat{U}_a, U_p^*))^* O^{-1}(R(\hat{U}_a, U_p^*) - Y^*) + B^{-1}(\hat{U}_a - U_a^*) = 0. \tag{4.27}$$

Let us consider an estimation error $\delta U_a = \hat{U}_a - \bar{U}_a$. We notice that $\delta U_p = U_p^* - \bar{U}_p = \eta_p$. Then

$$R(\hat{U}_a, U_p^*) - Y^* = R(\hat{U}_a, U_p^*) - (R(\bar{U}_a, \bar{U}_p) + \xi) = R'_{U_a}(\tilde{U}_a, U_p^*)\delta U_a + R'_{U_p}(\bar{U}_a, \tilde{U}_p^*)\eta_p - \xi,$$

where $\tilde{U}_a = \bar{U}_a + \tau_1 \delta U_a$, $\tilde{U}_p^* = \bar{U}_p + \tau_2 \eta_p$, $\tau_{1/2} \in [0, 1]$ and

$$\hat{U}_a - U_a^* = (\hat{U}_a - \bar{U}_a) - (U_a^* - \bar{U}_a) = \delta U_a - \eta_a.$$

Then equation (4.27) yields the error equation

$$(R'_{U_a}(\hat{U}_a, U_p^*))^* O^{-1}(R'_{U_a}(\tilde{U}_a, U_p^*)\delta U_a + R'_{U_p}(\bar{U}_a, \tilde{U}_p^*)\eta_p - \xi) + B_a^{-1}(\delta U_a - \eta_a) = 0. \tag{4.28}$$

Using the first order approximations $\hat{U}_a = \tilde{U}_a \approx \bar{U}_a$ and $\tilde{U}_p^* = U_p^* \approx \bar{U}_p$ we express δU_a as follows:

$$\delta U_a \simeq H_a^{-1}(\bar{U})((R'_{U_a}(\bar{U}))^* O^{-1}\xi + B_a^{-1}\eta_a - R'_{U_a}(\bar{U})O^{-1}R'_{U_p}(\bar{U})\eta_p), \tag{4.29}$$

where

$$H_a(\bar{U}) = (R'_{U_a}(\bar{U}))^* O^{-1} R'_{U_a}(\bar{U}) + B_a^{-1} \tag{4.30}$$

is the Hessian of an auxiliary control problem formulated for the active control set.

Since the full input vector error after DA is

$$\delta U = (\delta U_a, \eta_p)^T, \tag{4.31}$$

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

its covariance takes the form

$$V_{\delta U} = E[\delta U \delta U^T] = \begin{pmatrix} V_{\delta U_a} & V_{\delta U_{ap}} \\ V_{\delta U_{pa}} & B_p \end{pmatrix}, \tag{4.32}$$

where

$$V_{\delta U_a} = E[\delta U_a \delta U_a^T] = H_a^{-1} + H_a^{-1}(R'_{U_a})^* O^{-1} R'_{U_p} B_p (R'_{U_p})^* O^{-1} R'_{U_a} H_a^{-1}, \tag{4.33}$$

$$V_{\delta U_{ap}} = E[\delta U_a \eta_p^T] = -H_a^{-1}(R'_{U_a})^* O^{-1} R'_{U_p} B_p, \tag{4.34}$$

$$V_{\delta U_{pa}} = E[\eta_p \delta U_a^T] = -B_p (R'_{U_p})^* O^{-1} R'_{U_a} H_a^{-1}. \tag{4.35}$$

All operators in (4.33)-(4.35) are taken at the point $\bar{U}$. The error covariance (4.32) must be used in (3.21) for computing the goal-function error covariance in case of partial control. Numerical tests show that using cross-terms $V_{\delta U_{ap}}$ and $V_{\delta U_{pa}}$ is absolutely vital for the method.

5 Implementation with high-dimensional models

Let us first consider the formula for computing $V_{\delta G}$ (3.21). The operator-vector products $D'_X(\bar{X})\mathcal{M}'_U(\bar{U}) \cdot v$ and $(\mathcal{M}'_U(\bar{U}))^*(D'_X(\bar{X}))^* \cdot v$ are computed by calling the tangent linear and adjoint models of the corresponding mappings D and $\mathcal{M}$. Having the covariance-vector product $V_{\delta U} \cdot v$ defined in (4.32), the covariance-vector product $V_{\delta G} \cdot v$ can be used for the eigenvalue analysis of matrix $V_{\delta G}$. That is, its L_G largest eigenvalues $\lambda_{G,i}$ and the corresponding eigenvectors $W_{G,i}$ can be computed by the Lanczos method and used for constructing the limited-memory representation of $V_{\delta G}$ in the form

$$V_{\delta G} = \sum_{i=1}^{L_G} \lambda_{G,i} W_{G,i} W_{G,i}^T. \tag{5.36}$$

If the elements of vector δG are strongly correlated, the number of eigenpairs required for meaningful representation of $V_{\delta G}$ (and its diagonal elements, in particular) could by surprisingly small as compared to N_G (the dimension of vector G). The same is true for the number of Lanczos iterations needed for evaluating those eigenpairs.

Now we try to define $V_{\delta U} \cdot v$ without explicitly assembling the matrix $V_{\delta U}$, the components of which are presented in (4.33)-(4.35). Note that, if necessary, the operator-vector products $R'_{U_a} \cdot v$, $(R'_{U_a})^* \cdot v$, $R'_{U_b} \cdot v$ and $(R'_{U_b})^* \cdot v$ can be computed by calling the tangent linear and adjoint models of mapping R. However, we will use a different approach.

Let us consider the complete Hessian in (3.19). If the full control vector U is partitioned into the active and passive sets, i.e. $U = (U_a, U_p)^T$, then the Hessian matrix can also be partitioned as follows:

$$H = \begin{pmatrix} H_a & H_{ap} \\ H_{pa} & H_p \end{pmatrix} = \begin{pmatrix} B_a^{-1} + (R'_{U_a})^* O^{-1} R'_{U_a} & (R'_{U_a})^* O^{-1} R'_{U_p} \\ (R'_{U_p})^* O^{-1} R'_{U_a} & B_p^{-1} + (R'_{U_p})^* O^{-1} R'_{U_p} \end{pmatrix}. \tag{5.37}$$

Taking into account this partition, the expressions (4.33)-(4.35) can be rewritten in the form

$$V_{\delta U_a} = H_a^{-1} + H_a^{-1} H_{ap} B_p H_{pa} H_a^{-1}, \tag{5.38}$$

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

$$V_{\delta U_{ap}} = -H_a^{-1} H_{ap} B_p, \tag{5.39}$$

$$V_{\delta U_{pa}} = -B_p H_{pa} H_a^{-1}. \tag{5.40}$$

As mentioned in Sec.3, usually we compute and save in memory the eigenpairs of the projected Hessian $\tilde{H}$, which has the following partition

$$\tilde{H} = (B^{1/2})^* H B^{1/2} = \begin{pmatrix} \tilde{H}_a & \tilde{H}_{ap} \\ \tilde{H}_{pa} & \tilde{H}_p \end{pmatrix} = \begin{pmatrix} (B_a^{1/2})^* H_a B_a^{1/2} & (B_a^{1/2})^* H_{ap} B_p^{1/2} \\ (B_p^{1/2})^* H_{pa} B_a^{1/2} & (B_p^{1/2})^* H_p B_p^{1/2} \end{pmatrix}. \tag{5.41}$$

From (5.41) we derive:

$$H_a^{-1} = B_a^{1/2} \tilde{H}_a^{-1} (B_a^{1/2})^* \tag{5.42}$$

$$H_{ap} = (B_a^{-1/2})^* \tilde{H}_{ap} B_p^{-1/2}, \;\; H_a^{-1} H_{ap} = B_a^{1/2} \tilde{H}_a^{-1} \tilde{H}_{ap} B_p^{-1/2} \tag{5.43}$$

$$H_{pa} = (B_p^{-1/2})^* \tilde{H}_{pa} B_a^{-1/2}, \;\; H_{pa} H_a^{-1} = (B_p^{-1/2})^* \tilde{H}_{pa} \tilde{H}_a^{-1} (B_a^{1/2})^*. \tag{5.44}$$

By substituting expressions for H_a^{-1}, $H_a^{-1} H_{ap}$ and $H_{pa} H_a^{-1}$ into (5.38)-(5.40) we obtain

$$V_{\delta U_a} = B_a^{1/2} \tilde{H}_a^{-1/2} (I_a + \tilde{H}_a^{-1/2} \tilde{H}_{ap} \tilde{H}_{pa} \tilde{H}_a^{-1/2}) \tilde{H}_a^{-1/2} (B_a^{1/2})^*, \tag{5.45}$$

$$V_{\delta U_{ap}} = -B_a^{1/2} \tilde{H}_a^{-1} \tilde{H}_{ap} (B_p^{1/2})^*, \tag{5.46}$$

$$V_{\delta U_{pa}} = -B_p^{1/2} \tilde{H}_{pa} \tilde{H}_a^{-1} (B_a^{1/2})^*. \tag{5.47}$$

Assuming $v = (v_a, v_p)^T$,

$$V_{\delta U} \cdot v = \begin{pmatrix} V_{\delta U_a} \cdot v_a + V_{\delta U_{ap}} \cdot v_p \\ V_{\delta U_{pa}} \cdot v_a + B_p \cdot v_p \end{pmatrix}, \tag{5.48}$$

where the operators $V_{\delta U_a}$, $V_{\delta U_{ap}}$ and $V_{\delta U_{pa}}$ are defined in (5.45)-(5.47). Implementation of the above formulas requires, in turn, the operator-vector products $\tilde{H}_{pa} \cdot v_a$, $\tilde{H}_{ap} \cdot v_p$ and $\tilde{H}_a^{\gamma} \cdot v_a$ for $\gamma = -1, -1/2$.

Let us assume that the eigenpairs λ_i, W_i of $\tilde{H}$ are available and, therefore, $\tilde{H}_{pa} \cdot v_a$, $\tilde{H}_{ap} \cdot v_p$ and $\tilde{H}_a \cdot v_a$ are somehow defined (see §6). The latter allows the leading eigenvalue/eigenvector pairs $\lambda_{a,i}$, $W_{a,i}$ of $\tilde{H}_a$ to be evaluated by the Lanczos method. Then, according to (3.23)

$$\tilde{H}_a^{\gamma}(\bar{u}) \simeq I_a + \sum_{i=1}^{L_a} (\lambda_{a,i}^{\gamma} - 1) W_{a,i} W_{a,i}^*. \tag{5.49}$$

The above formula provides $\tilde{H}_a^{\gamma} \cdot v$ for $\gamma = -1, -1/2$ required in (5.45)-(5.47). Below we summarize the steps of computing $V_{\delta G}$, for all K possible active sets:

Algorithm 1

1. compute by the Lanczos method and store in memory:
$\{\lambda_i, W_i\}$, $i = 1, \ldots, L_H$ of $\tilde{H}(\bar{U})$ in (3.22)
2. for $k = 1, \ldots, K$
   a. compute by the Lanczos method and store in memory:
   $\{\lambda_{a,i}, U_{a,i}\}$, $i = 1, \ldots, L_a$ of $\tilde{H}_a(\bar{U})$ defined via $\{\lambda_i, W_i\}$
   b. compute by the Lanczos method and store in memory:
   $\{\lambda_{g,i}, U_{g,i}\}$, $i = 1, \ldots, L_g$ of $V_{\delta G}$ defined in (3.21), using $V_{\delta U} \cdot v$ defined in (5.48)
end k

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

Remark 3. Step 1 enables $\tilde{H}_a \cdot v_a$, $\tilde{H}_{pa} \cdot v_a$ and $\tilde{H}_{ap} \cdot v_p$ to be evaluated when necessary, see §6 for details. After step 2a, for any chosen active control set we are able to compute $H_a^\gamma \cdot v_a$ using (5.49) and, correspondingly, $V_{\delta U} \cdot v$ using (5.48). At steps 1 and 2b we solve the sequence of the tangent linear and adjoint models, which is, in case of using models based on partial differential equations, the most expensive part in terms of the CPU time. Step 2a requires algebraic computations only. The diagonal elements of $V_{\delta G}$ can be retrieved on the basis of representation (5.36). Square roots of these elements (standard deviation) are the sought outcome of the algorithm above.

6 Active set algebra

Let us assume that the 'active control' status has been assigned to some elements of the full control vector U. In practical implementation there is no need ordering U into the partition $U = (U_a, U_p)^T$. Instead, the active and passive elements have to be correspondingly labeled, then a special algebra can be applied.

Let us introduce mappings between the full, active and passive set vectors. First, we create integer arrays K_a of size N_a and K_p of size N_p, containing ordinal numbers of the active set elements of U and the passive set elements of U, correspondingly. The full-to-active set mapping L_a is defined as

$$v_a = L_a v : \quad v_{a,i} = v_{K_a(i)},\ i = 1, \ldots, N_a.$$

The active-to-full set control mapping L_a^* (adjoint to L) injects values of the active set vector v_a into the corresponding locations in the full set vector v, i.e.

$$v = L_a^* v_a : v_k = \left\{ \begin{array}{cc} v_{a,i}, & k = K_a(i) \\ 0, & k \neq K_a(i) \end{array} \right. ,\ k = 1, \ldots, N_a + N_p.$$

Similarly, using K_p, we define the full-to-passive set mapping L_p and the passive-to-full set mapping L_p^*. Now we can finally define

$$V_{\delta U} \cdot v = L_a^* V_{\delta U_a} L_a v + L_p^* V_{\delta U_{pa}} L_a v + L_a^* V_{\delta U_{ap}} L_p v + L_p^* B_p L_p v \tag{6.50}$$

where the blocks $V_{\delta U_a}$, $V_{\delta U_{ap}}$ and $V_{\delta U_{pa}}$ are given by (5.45)-(5.47).

Since we refuse ordering elements of U, the original structure of H (or $\tilde{H}$) is no longer in the form (5.37), but the latter could be achieved using some row and column permutations. The purpose is to enable evaluating $\tilde{H}_a \cdot v_a$, $\tilde{H}_{pa} \cdot v_a$ and $\tilde{H}_{ap} \cdot v_p$ using $\tilde{H}$ in its given (non-ordered) form.

Let us define the following operator-vector products:

$$H_{aa}^\circ \cdot v = \left\{ \begin{array}{cc} \sum_{j \in K_a} H_{i,j} v_j, & i \in K_a \\ 0, & i \notin K_a \end{array} \right. , \tag{6.51}$$

$$H_{ap}^\circ \cdot v = \left\{ \begin{array}{cc} \sum_{j \notin K_a} H_{i,j} v_j, & i \in K_a \\ 0, & i \notin K_a \end{array} \right. , \tag{6.52}$$

$$H_{pa}^\circ \cdot v = \left\{ \begin{array}{cc} 0, & i \in K_a \\ \sum_{j \in K_a} H_{i,j} v_j, & i \notin K_a \end{array} \right. , \tag{6.53}$$

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

$$H^{\circ}_{pp} \cdot v = \begin{cases} 0, & i \in K_a \\ \sum_{j \notin K_a} H_{i,j} v_j, & i \notin K_a \end{cases} . \tag{6.54}$$

If H is given in the limited-memory form (3.23), then

$$H_{i,j} = e_i^T H e_j = I_{i,j} + \sum_{k=1}^{L} (\lambda_k - 1) W_{k,i} W_{k,j}. \tag{6.55}$$

By substituting $H_{i,j}$ into (6.51)-(6.54) one obtains:

$$H^{\circ}_{aa} \cdot v = \begin{cases} v_i + \sum_{k=1}^{L} (\lambda_k - 1) \sum_{j \in K_a} W_{k,i} W_{k,j} v_j, & i \in K_a \\ 0, & i \notin K_a \end{cases} , \tag{6.56}$$

$$H^{\circ}_{ap} \cdot v = \begin{cases} v_i + \sum_{k=1}^{L} (\lambda_k - 1) \sum_{j \notin K_a} W_{k,i} W_{k,j} v_j, & i \in K_a \\ 0, & i \notin K_a \end{cases} , \tag{6.57}$$

$$H^{\circ}_{pa} \cdot v = \begin{cases} 0, & i \in K_a \\ v_i + \sum_{k=1}^{L} (\lambda_k - 1) \sum_{j \in K_a} W_{k,i} W_{k,j} v_j, & i \notin K_a \end{cases} , \tag{6.58}$$

$$H^{\circ}_{pp} \cdot v = \begin{cases} 0, & i \in K_a \\ v_i + \sum_{k=1}^{L} (\lambda_k - 1) \sum_{j \notin K_a} W_{k,i} W_{k,j} v_j, & i \notin K_a \end{cases} . \tag{6.59}$$

Let us note that in the above operator definitions the full set vector v is used (as input and output), whereas we need $\tilde{H}_a \cdot v_a$ for computing $\tilde{H}_a^{-1}$, and $\tilde{H}_{pa} \cdot v_a$ and $\tilde{H}_{ap} \cdot v_p$ in formulas (5.45)-(5.47). Therefore, we use operators $\tilde{H}^{\circ}$ together with mappings L_a, L_a^*, L_p, L_p^* in the following way:

$$\tilde{H}_a \cdot v_a = L_a \tilde{H}^{\circ}_{aa} L_a^* \cdot v_a, \tag{6.60}$$

$$\tilde{H}_{ap} \cdot v_p = L_a \tilde{H}^{\circ}_{ap} L_p^* \cdot v_p, \tag{6.61}$$

$$\tilde{H}_{pa} \cdot v_a = L_p \tilde{H}^{\circ}_{pa} L_a^* \cdot v_a. \tag{6.62}$$

7 Validation

Estimating river discharges from *in-situ* and/or remote sensing data is a key component for evaluation of water balance at local and global scales and for water management. A distinctive feature of the river discharge estimation problem is the likely presence of significant uncertainty in parameters defining basic properties of a hydraulic model, such as bathymetry (surface topography), friction, infiltration level, etc. There are also unaccounted lateral tributaries/offtakes and storage areas.

Since the discharge estimation problem is considered, the active set must undoubtedly include the inflow discharge at a chosen upstream location (the inlet). Would it be a sufficient control set? If not, what other model inputs should be included into the active set to reduce the impact of uncertainties? Do we have enough data? Indeed, *in-situ* measurements of water elevation and discharge are relatively rare on most rivers because of limited accessibility and associated costs, whereas the satellite data can be sparse in time and far less accurate. Therefore, designing the control set is a key issue for solving the river discharge estimation problem. That explains our choice of the application.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

7.1 Model statement

The hydraulic network is represented by a set of closed-line segments or 'reaches' connected at nodes N_k, see Fig.1. The spatial discretization along reach number i produces a set of co-ordinates $x_{i,j}$, also called longitudinal abscissas, each having the associated global index k and its own position vector $\vec{r}_k = (x'_k, y'_k, z'_k)$ in the global co-ordinate system (bathymetry). Given $\vec{n}_k$ is a pre-dominant flow direction at $x_{i,j}$, a hydraulic cross-section $S_{i,j}$ is defined by a set of points on a plane $\vec{n}_k \cdot (\vec{r} - \vec{r}_k) = 0$ describing the bed profile, which are evaluated from a design sketch or from a topographical survey. For each section this data allows us to compute for any given water level line Z: the wetted area function $A(Z, p_g)$, the wetted perimeter function $P(Z, p_g)$, the hydraulic radius function $R(Z, p_g)$ and the top width function $L(Z, p_g)$, where p_g are geometric parameters of the corresponding cross-section. For a given reach, p_g is a function of the longitudinal abscissa x.

For a 'regular' section, the shallow water flow in the longitudinal direction x is described by the Saint-Venant equations:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = Q_L, \tag{7.63}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial Q^2/A}{\partial x} + gA\frac{\partial Z}{\partial x} = -gAS_f + C_k Q_L v, \tag{7.64}$$

$$t \in (0, T],$$

where $Q(x,t)$ is the discharge, $Z(x,t)$ is the water level, $v(x,t) = Q/A$ is the mean velocity, $Q_L(x,t)$ is the lateral discharge, $C_k(x)$ is the lateral discharge coefficient and S_f is the friction term dependent on the Strickler coefficient $C_s(x)$ and on the hydraulic radius $R(Z, p_g)$:

$$S_f = \frac{Q|Q|}{C_s^2 A^2 R^{4/3}}.$$

The initial condition for equations (7.63)-(7.64) is

$$Z(x,0) = Z_0(x), \quad Q(x,0) = Q_0(x). \tag{7.65}$$

For an internal node we consider the mass balance equation alongside the condition of local elevations or 'heads' ($H = v^2/2g + Z$) equality, for all connected reaches. On the network example presented in Fig.1 these equations are

$$q_1 = -Q|_{S_{1,k1}} - Q|_{S_{2,k2}} + Q|_{S_{3,1}}, \tag{7.66}$$

$$Z|_{S_{1,k1}} = Z|_{S_{3,1}}, \;\; Z|_{S_{2,k2}} = Z|_{S_{3,1}}, \tag{7.67}$$

or

$$H|_{S_{1,k1}} = H_{S_{3,1}}, \; H|_{S_{2,k2}} = H_{S_{3,1}}, \tag{7.68}$$

where q_1 is the offtake or tributary at node N_3.

Boundary conditions are defined at boundary nodes. For the upstream nodes we usually use the inflow discharge $\mathcal{Q}(t)$ or elevation $\mathcal{Z}(t)$, for example [2]:

$$Q(t)|_{S_{1,1}} = \mathcal{Q}_1(t) \quad \vee \quad Z(t)|_{S_{1,1}} = \mathcal{Z}_1(t), \tag{7.69}$$

[2] In the text below $\vee$ stands for logical 'or' and $\wedge$ for logical 'and'

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

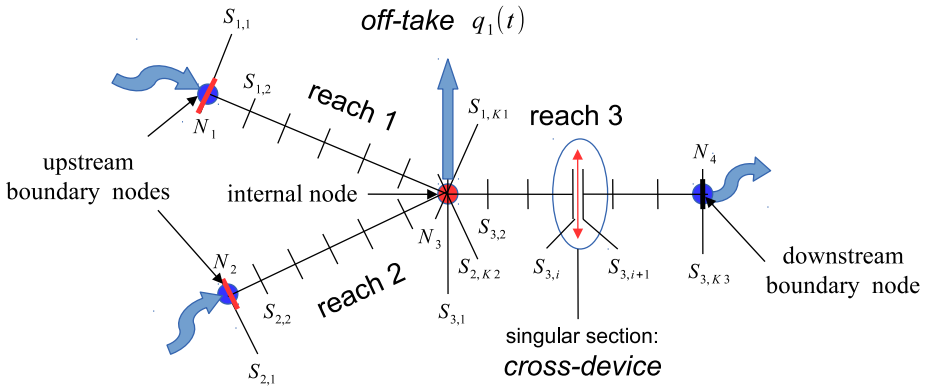

Figure 1: River or canal network conceptual scheme

whereas for the downstream nodes it is the elevation $\mathcal{Z}(t)$ or the rating curve $Q = f(Z, p_{rc})$, where p_{rc} are the rating curve parameters, for example:

$$Z(t)|_{S_{3,k3}} = \mathcal{Z}_3(t), \quad \vee \quad Q|_{S_{3,k3}} = f(Z|_{S_{3,k3}}, p_{rc}). \tag{7.70}$$

We also consider a singular section, which consists of the collocated upstream and downstream sections. It is mainly used to represent artificial structures (cross-devices), such as gates, weirs, bridges etc., where the Saint-Venant are replaced by ad-hoc alternative equations. For the singular section we consider the mass balance equation alongside the equation relating the elevations (or 'heads'), for example:

$$Q|_{S_{3,i}} - Q|_{S_{3,i+1}} = 0 \tag{7.71}$$

$$Q|_{S_{3,i}} = \mathcal{F}(Z|_{S_{3,i}}, Z|_{S_{3,i+1}}, C_d|_{S_{3,i}}), \tag{7.72}$$

where C_d is the cross-device discharge coefficient.

Let $\mathcal{U}$ be a space of all possible input variables for the model (7.63)-(7.72), including some parameters p_{nm} of the implemented numerical scheme, such as the Preissmann implicitation coefficient, for example. Let us also assume that the specified network configuration includes K_{bn} boundary nodes, K_{in} internal nodes, K_{ss} singular sections, K_r reaches and $K_s(i)$ sections, $i = 1, \dots K_r$. Then, the full control vector $U \in \mathcal{U}$ looks as follows:

$$U = (Z_0,\ Q_0,\ \mathcal{Z},\ \mathcal{Q},\ q,\ Q_L,\ C_s,\ C_k,\ C_d,\ p_{rc},\ p_g,\ p_{nm})^T, \tag{7.73}$$

where by (Z_0, Q_0) we mean a set of initial conditions for all reaches, i.e.

$$(Z_0, Q_0) = \{(Z_0(x_{i,j}), Q_0(x_{i,j})),\ i = 1, \cdots K_r,\ j = 1, \cdots K_s(i)\},$$

by $(\mathcal{Z}, \mathcal{Q})$ - a set of inflow discharges or elevations at all boundary nodes, i.e.

$$(\mathcal{Z}, \mathcal{Q}) = \{(\mathcal{Z}_k(t) \vee \mathcal{Q}_k(t)),\ k = 1, \cdots K_{bn}\}$$

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

by q - a set of all offtakes/tributaries, i.e. $q = \{q_k(t),\ k = 1, \cdots K_{in}\}$, by p_g - a set of geometric parameters for all sections, i.e. $p_g = \{p_g(x_{i,j}),\ i = 1, \cdots K_r,\ j = 1, \cdots K_s(i)\}$, and, similarly, for the remaining variables in (7.73), each having its own dimension. For given U, by solving the model equations (7.63)-(7.72) simultaneously for all network reaches, we obtain the state $X \in \mathcal{X}$ such that

$$X = (Z, Q)^T = \{(Z(x_{i,j}, t), Q(x_{i,j}, t))^T,\ i = 1, \cdots K_r,\ j = 1, \cdots K_s(i),\ t \in [0, T]\}. \quad (7.74)$$

Let us assume that the state is observed in the form (2.5). In particular, the water surface elevation measured by the gauge stations, located at the specified sections of the specified reaches, may be available. We shall denote by I_o the array defining the indices of these sections and reaches. Usually, such measurements are recorded with sufficiently small time step, so we can treat them as nearly continuous in time. Then, the observation operator is as follows:

$$Y = C(Z, Q) = \{Z(x_{i,j}, t), (i, j) \in I_o\}. \quad (7.75)$$

Thus, the particular form of vectors U, X and Y is now defined.

The above hydraulic equations are implemented in SIC^2, which is the full nonlinear Saint-Venant hydraulic network model. The basic features of this model are presented in Appendix I. The routine which maps U into Y represents operator R. The tangent linear model (TLM) and the adjoint model, which represent operators R' and $(R')^*$, respectively, are produced by means of the AD engine TAPENADE [12] applied to the main computational routine of the SIC^2 package (the forward model). An outline of conceptual steps, needed for producing the routines which calculate the Hessian-vector product and the $V_{\delta G}$-vector product, is presented in Appendix II.

Remark 4. Let us note that at different steps of Algorithm 1 the tangent linear and adjoint models are involved. However, these models may not be available in certain circumstances. Thus, the 'derivative-free' implementation of the presented method should be developed in the future. In particular, the Hessian-vector product could possibly be defined using the simultaneous perturbation gradient approximation (SPGA) approach [13], whereas the covariance $V_{\delta G}$ can be constructed by using the eigenpairs of $V_{\delta u}$ (defined in (5.48)) as 'sigma-points', in a manner considered in [26]. An additional benefit of such approach would be the nonlinear posterior uncertainty propagation.

7.2 Particular choice of goal-functions (QoI)

The goal-functions (QoI) are usually some functionals of the state trajectory. In hydraulics, certain quantities useful in the flood risk assessment may be of interest, such as the maximum water surface elevation above a given (safe) threshold at some locations, for example. Thus, we consider the goal-function in the form

$$G = D(Q(x, t), Z(x, t)) = (G_Q(x), G_Z(x))^T,$$

where

$$G_Q(x) = \int_{t_1}^{t_2} |Q(x, t) - Q_*(x)| dt, \quad (7.76)$$

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

$$G_Z(x) = \int_{t_1}^{t_2} |Z(x,t) - Z_*(x)| dt, \tag{7.77}$$

where Q_* and Z_* are some reference discharge and elevation levels, and t_1, t_2 define the time window of interest. One may consider multiple time windows or the values $G_Q(x)$ and $G_Z(x)$ at chosen time instants, including $t \geq T$. In the latter case the goal-function represents the forecast.

8 Model implementation details

8.1 Initial condition treatment

The initial condition (Z_0, Q_0) is a model state at $t = 0$. As such it must be a model solution consistent with the parameters which define the fundamental properties of the model, such as bathymetry, friction, rating curve parameters and cross-device coefficients, and also with the previous values of time-dependent controls. Changing arbitrarily some of those parameters while keeping the initial condition intact leads to severe shocks in the flow fields at the initial time period. Furthermore, the difference between the observations and the model predictions during the initial time period dominates the gradient. The corresponding updates being introduced into the nonlinear system may lead to unsupported flow conditions. Even if the initial condition is consistent with the other parameters at the start of the iterative process, independent updates of time-dependent controls and parameters may lead to inconsistency again.

The way to deal with this issue is as follows. We notice that the influence of the initial state on the flow is very limited in time, then it is dominated by actuators (boundary conditions and source terms). Therefore, we postulate that (Z_0, Q_0) is a **steady state** flow solution consistent with the initial value of the time-dependent controls and time-independent controls. This state is approached by performing a relaxation model run. By doing so we stop considering (Z_0, Q_0) as an independent control, but it becomes a unique function of other controls.

8.2 Spline approximation of time-dependent controls

The time-dependent controls, such as the inflow discharge $\mathcal{Q}(t)$, water elevation $\mathcal{Z}(t)$ at boundaries, offtakes/tributaries $q_i(t)$ and the lateral discharge $Q_L(x,t)$ are approximated in time by cubic splines. Thus, the control points for the chosen control variable can be arbitrarily distributed in time. Given the values of control at these points the spline coefficients are constructed. The control values at time instants required for the model numerical integration (usually at $t = i \times dt$, where dt is the time step and i is the time index) are evaluated as the corresponding spline values. This allows the number of control nodes to be significantly smaller than the number of integration time steps, which is useful given that the simulation period can be fairly long. Besides, the control nodes can be distributed more densely in the areas of fast dynamics and more sparsely in the areas of slow dynamics. This approach can also be considered as a preliminary regularization.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

8.3 Defining the background covariance

The controls in U can be divided into three groups: time dependent controls, e.g. the inflow discharge $\mathcal{Q}(t)$, spatially distributed controls, e.g. the Strickler coefficient $C_s(x)$ or bed elevation $z(x) \in p_g$, and lumped parameters, e.g. the cross-device discharge coefficients C_d or rating curve parameters p_{rc}. For the lumped parameters we can only prescribe the variance, for the distributed controls the covariance matrix must be specified.

Here we present a slightly modified version of the approach described in [10]. In solving ill-posed inverse problems the solution is often considered to be a smooth function which belongs to a Sobolev space of certain order, e.g. W_2^2. Let $f(x),\ x \in [a,b]$ be a one-dimensional function of x and let us introduce two positive weight functions $\beta(x), \gamma(x)$. We define the norm of $f(x)$ in W_2^2 as follows:

$$\|f(x)\|^2_{W_2^2[a,b]} = \int_a^b \left[\beta(x) f^2(x) + \beta(x) \left(\frac{\partial}{\partial x} \left(\gamma(x) \frac{\partial f(x)}{\partial x} \right) \right)^2 \right] dx.$$

To evaluate this integral numerically we discretize $f(x)$ using a set of uniformly distributed nodes $\{x_i = (i-1)\Delta x,\ i = 1, \dots, m\}$ and substitute the integral by the sum

$$\|f(x)\|^2_{W_2^2[m]} = \Delta x \sum_{i=1}^{m} \beta(x_i) f^2(x_i) + \Delta x \sum_{i=1}^{m} \beta(x_i) \left(\frac{\partial}{\partial x} \left(\gamma(x) \frac{\partial f(x)}{\partial x} \right) \right)^2 \Bigg|_{x=x_i}. \tag{8.78}$$

Numerical implementation of the second term depends on the boundary conditions imposed on $f(x)$; in this paper we use the 'natural' boundary conditions, i.e. $f''(a) = f''(b) = 0$.

In practice, we consider a discrete function $\bar{f}(\bar{x}_i)$, where $\bar{x}_i,\ i = 1, \dots, \bar{m}$ are arbitrarily distributed nodes. Therefore, an operator G which maps $\bar{f}$ into f must be constructed. Because we need the second derivative of $f(x)$, the cubic spline approximation of $f(x_i)$ is sufficient. The inverse of the covariance matrix B must satisfy the following condition

$$\|B^{-1/2} \bar{f}\|_{L_2[\bar{m}]} = \|G(\bar{f})\|_{W_2^2[m]}. \tag{8.79}$$

Assuming that $\bar{f}$ is reasonably close to the prior $\bar{f}_b$, the elements $B^{-1}_{i,j}$ can be obtained by the following formula:

$$B^{-1}_{i,j} \approx \left. \frac{\partial^2 (\|G(\bar{f})\|^2_{W_2^2[m]})}{\partial \bar{f}_i \partial \bar{f}_j} \right|_{\bar{f}=\bar{f}_b}.$$

The code for computing the elements $B^{-1}_{i,j}$ is obtained by applying the Automatic Differentiation (direct mode) twice to the subroutine evaluating $\|G(\bar{f})\|_{W_2^2[m]}$. The matrix B^{-1} is symmetric and narrow-banded. It can be easily factorized using Choleski decomposition:

$$B^{-1} = B^{-1/2} (B^{-1/2})^T.$$

Given the factor $B^{-1/2}$, the product $v = B^{1/2} w$ is defined via solving the equation $B^{-1/2} v = w$. Since $B^{-1/2}$ is a triangular banded matrix, the solution procedure is simply a backward sweep. In the covariance matrix obtained by this method the functions $\beta(x)$ and $\gamma(x)$ define the local variance and the correlation radius, respectively. For any time-dependent control, the covariance is generated for the full time domain. It is slightly more complicated for the spatially distributed controls due to the presence of nodes and singular sections. While the method allows the variable variance and correlation level along the abscissa x, in the numerical examples considered below these are uniform everywhere except near the boundaries.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

8.4 Miscellaneous

a) Bathymetry. In Sect.2 the bathymetry is formally defined by geometric parameters p_g, entering the functions $A(Z, p_g)$ and $P(Z, p_g)$. Here we present a more detailed description. For each section n, the elevation $z_b(n)$ of the lowest point of the cross-section shape with respect to a chosen reference horizontal level is given. The function $z_b(n)$ is referred to as the bed elevation. Other parameters describe the cross-section shape itself. The dilation coefficient $b(n)$ is introduced to allow the cross-section width to be modified. This is achieved by multiplying on $(1 + b(n))$ all horizontal dimensions of the corresponding cross-section. For example, in the case of trapezoidal cross-section shape, the trapezoid bases are scaled. Subsequently, this affects the functions $A(Z, p_g)$ and $P(Z, p_g)$. The functions $z_b(n)$ and $b(n)$ are considered as the generalized bathymetry controls.
b) Identical twin experiment. In this paper the identical twin experiment approach is adopted: given a reference ('true') value of the control vector $\bar{U}$, for a chosen observation scheme the model predictions at the specified points (in space and time) are considered as 'exact' observations; after being corrupted by noise these observations are considered as 'noisy'. The task is to estimate the control vector using either 'exact' or 'noisy' data and to evaluate the estimation error $dU = \hat{U} - \bar{U}$.

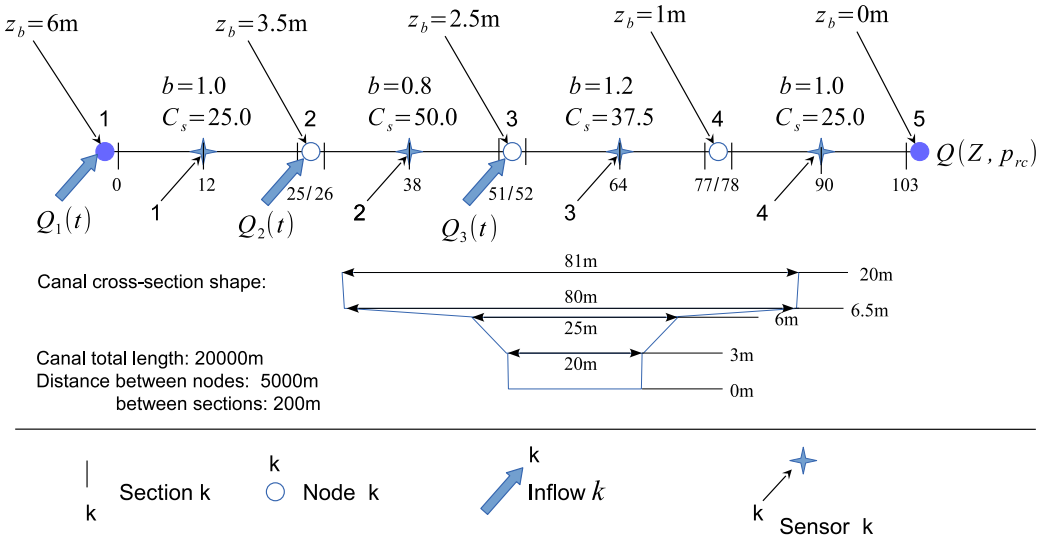

Figure 2: Testing configurations.

9 Numerical results

For numerical tests we use an idealized benchmark presented in Fig.2. The canal is composed of four consecutive $5km$-long reaches bounded by nodes (shown in 'circles'). The canal inlet is at *node 1*, its outlet - at *node 5*. A reach is discretized into 25 equidistantly positioned computational sections having the same cross-sectional shape. The internal nodes contain

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

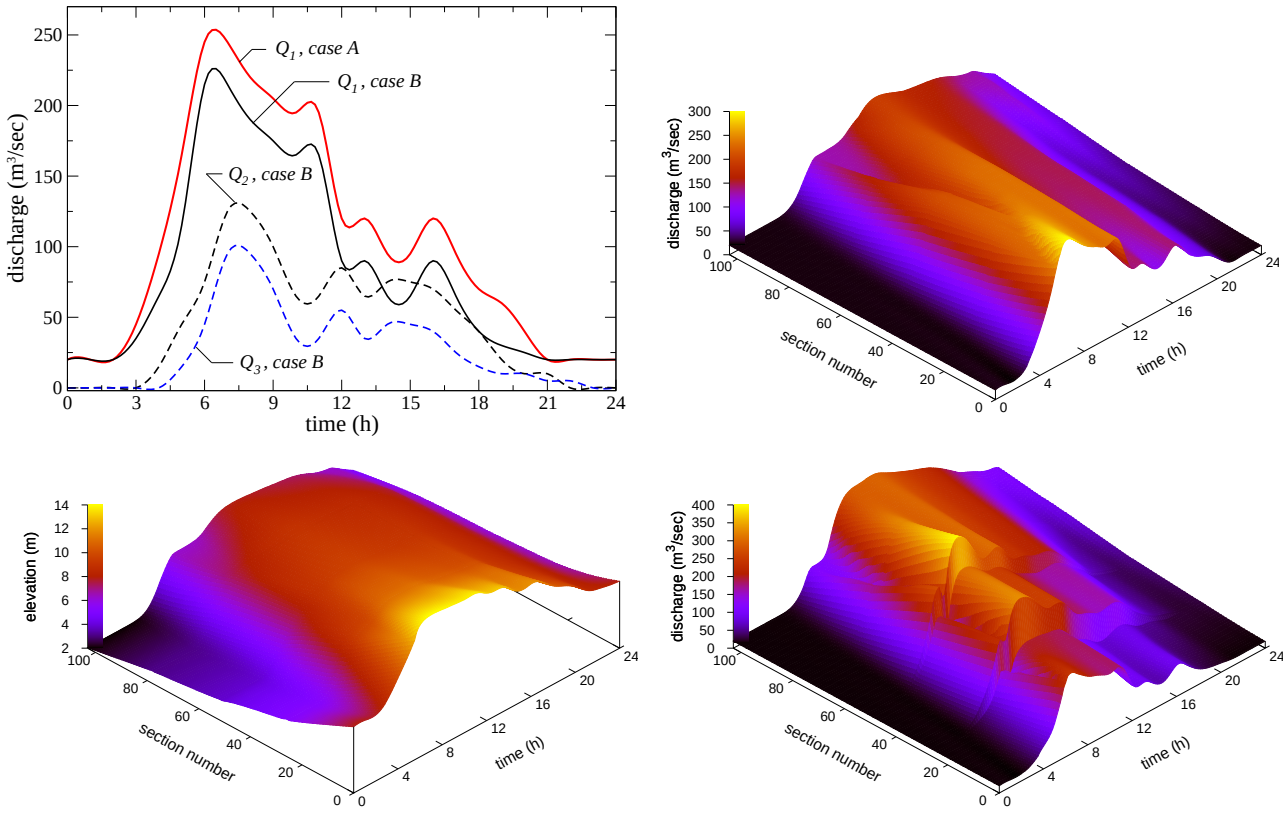

Figure 3: - Inflow discharges Q_i (upper/left), discharge field evolution: case A (upper/right) and case B (lower/right), elevation field evolution: case A (lower/left).

co-located boundary sections of the connected reaches. The bed elevation z_b value for each node (and for the boundary sections involved) is presented in the figure; between nodes z_b changes linearly. The Strickler coefficient C_s and the dilation coefficient b are constant along the reach. The values of both are also presented in the figure. The boundary conditions are as follows: the inflow discharge $\mathcal{Q}(t) = Q_1(t)$ at *node 1* and the rating curve $Q(Z, p_{rc})$ at *node 5*. The tributaries are represented by the discharges $Q_2(t)$ and $Q_3(t)$ at *node 2* and *node 3*, respectively. We consider four surface elevation sensors, each located in the middle of the corresponding reach (shown in 'stars').

The covariance matrix $V_{\delta U}$ includes the diagonal blocks $V_{\delta U_i}$, each being associated with the corresponding control variable U_i of the control vector U. Thus, we define the *standard deviation* (SD) vector $\sigma[\delta U_i]$, such that its elements are the square-roots of the diagonal entries of $V_{\delta U_i}$. Similarly, the covariance matrix $V_{\delta G}$ includes two diagonal blocks $V_{\delta G_Q}$ and $V_{\delta G_Z}$, associated with different goal-functions (QoI) in (7.76) and (7.77); the corresponding SD vectors are denoted $\sigma[\delta G_Q]$ and $\sigma[\delta G_Z]$. All results below are presented in terms of $\sigma[\cdot]$.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

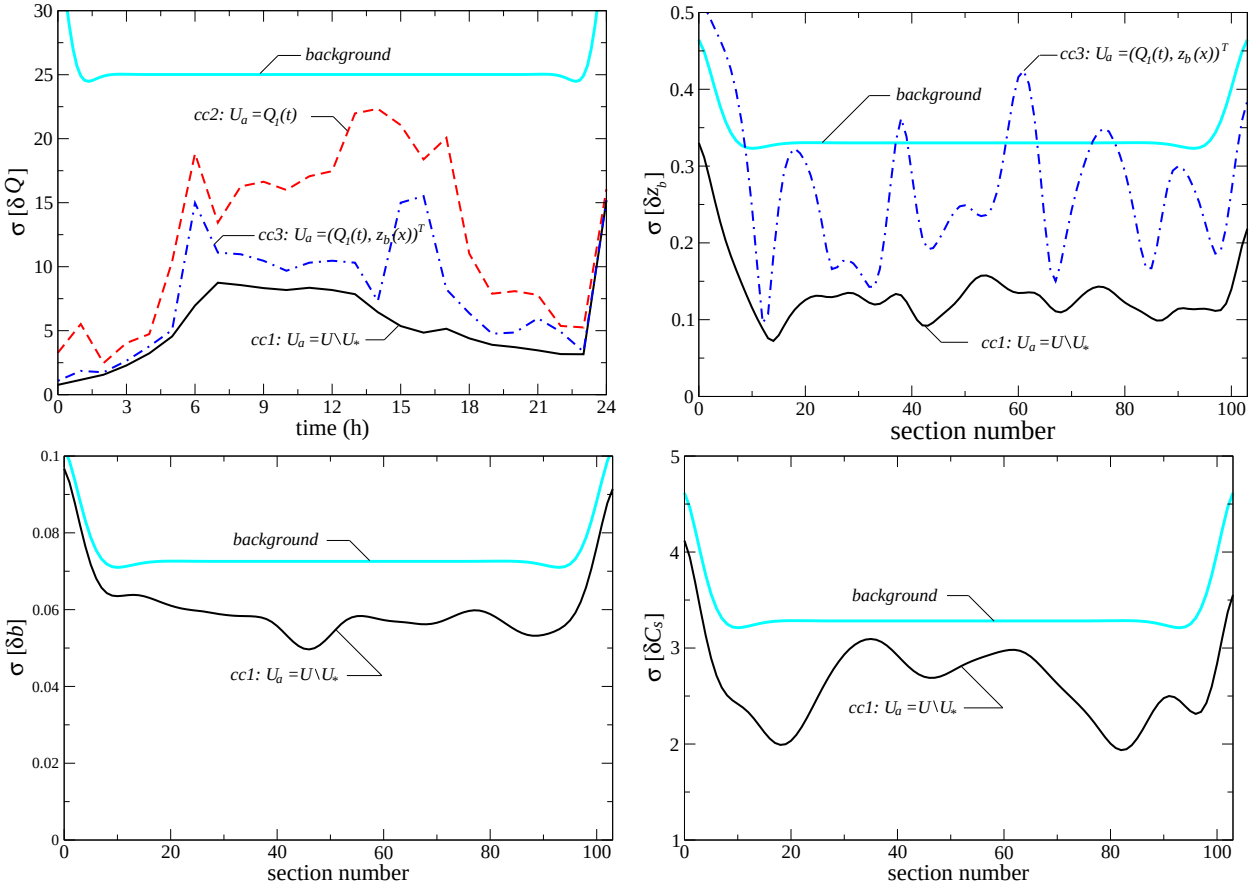

Figure 4: Standard deviation of control estimates.

9.1 Case A

In this case we consider the inflow discharge $\mathcal{Q}(t) = Q_1(t)$ as a driving condition (actuator), whereas $Q_2(t) = Q_3(t) = 0$. The uncertainty-bearing approximations of the geometry-defining parameters $z_b(k)$ and $b(k)$ and the Strickler coefficient $C_s(k)$ are available for $k = 1, \ldots, K_s$, where K_s is the total number of sections. Then, the full control vector is

$$U = (Q_1(t), z_b(k), b(k), C_s(k), U_*)^T, \ k = 1, \ldots, K_s,$$

where U_* stands for all remaining model inputs which contain no uncertainty. The 'true' value of the inflow discharge $Q_1(t)$ is presented in Fig.3, upper/left subplot. The 'true' discharge field $Q(t,k)$ and of the water surface elevation field $Z(t,k)$ are presented in upper/right and lower/left subplots, correspondingly.

Data from all four sensors is used. Here we consider the following active control sets:
*cc*1 - full control case: i.e. $U_a = U \setminus U^*$;
*cc*2 - partial control case: inflow discharge only, i.e. $U_a = \mathcal{Q}(t)$;
*cc*3 - partial control case: inflow discharge and bed elevation, i.e. $U_a = (\mathcal{Q}(t), z_b(k))^T$;
*cc*4 - partial control case: inflow discharge and Strickler coefficient, i.e. $U_a = (\mathcal{Q}(t), C_s(k))^T$.

Numerical results for case A are summarized in Fig.4 and Fig.5. In Fig.4 we show the background (prior) SD and the SD of the estimates (posterior) for all components of the control vec-

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

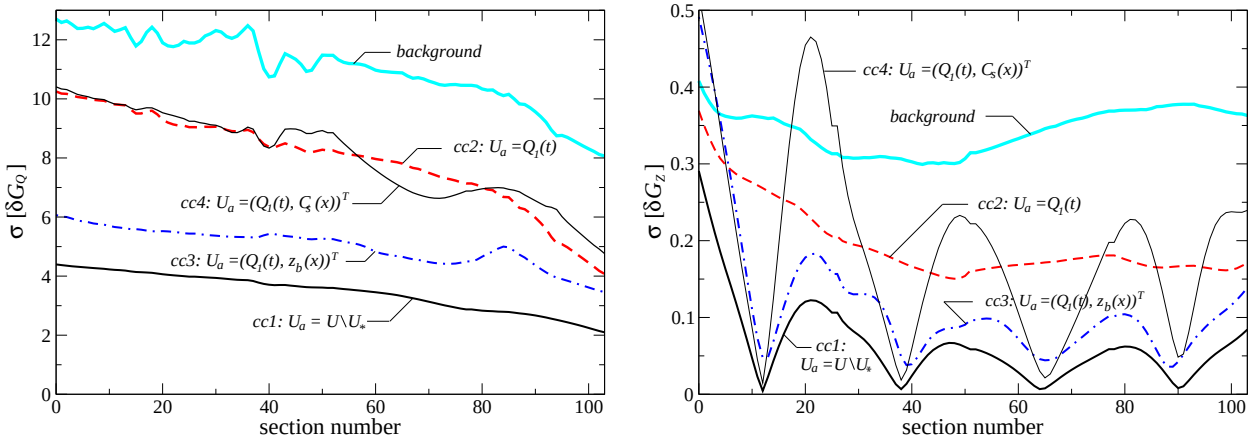

Figure 5: Standard deviation of the goal-vector (QoI).

tor, i.e.: $\sigma[\delta Q_1]$ (upper/left), $\sigma[\delta z_b]$ (upper/right), $\sigma[\delta b]$ (lower/left) and $\sigma[\delta C_s]$ (lower/right). In Fig.5 we show the SD of the goal-vectors: $\sigma[\delta G_Q]$ (lower/left) and $\sigma[\delta G_Z]$ (lower/right) for different control cases $cc1 - cc4$.

These figures reveal the following interesting features:

1. In the full control case $cc1$, the posterior SD are strictly smaller than the background SD, for all control variables (compare the curves marked $cc1$ to those marked $cc0$). This difference is usually referred as the 'uncertainty reduction'. The curves marked $cc1$ show the minimum SD level that can be achieved with the given observations. Since estimating the discharge is the major task, let us pay attention to Fig.4(upper/left) and Fig.5(left) subplots. For the partial control case $cc2$ (the inflow discharge only), $\sigma[\delta Q_1]$ and $\sigma[\delta G_Q]$ are presented in dashed lines. One can see that the uncertainty reduction achieved in this case makes only a fraction (30-40%) of the one achieved in the full control case. Thus, controlling the discharge only is hardly sufficient.

2. The control set can be extended, however this must be done with caution. It seems reasonable to add such control variable that the resulting uncertainty in the goal-function (QoI) would be most essential. At the same time this extension should not affect the reliability of the minimization process. By comparing results for different active control sets one can see that such component does exist: it is the bed elevation z_b, see the results in dash-dotted line, case $cc3$. For comparison we also present the partial control case $cc4$, where instead of z_b we use C_s. One can notice that, in terms of $\sigma[\delta G_Q]$, the effect of inclusion C_s into the active set is negligible. Some improvement can be seen in terms of $\sigma[\delta G_Z]$ in the vicinity of sensors. It has been repeatedly observed that including both z_b and C_s into the active control set leads to unsupported combinations of controls, see [11]. Thus, the sufficient control set is given by vector $U_a = (\mathcal{Q}(t), z_b(k))^T$.

3. It is difficult to judge whether or not the control set is sufficient looking at the posterior uncertainty in controls $V_{\delta U_a}$. For small errors the goal-functions (QoI) are certain combinations of responses associated to different control variables, therefore an important role belongs to correlations. For example, in case $cc3$ $\sigma[\delta z_b]$ does not look much reduced, but both $\sigma[\delta G_Q]$ and $\sigma[\delta G_Z]$ are sufficiently good. This is because the error associated with C_s and b are partly

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

absorbed by the bed elevation control. However, the estimate of z_b as such has a little practical use. For example, it can hardly be considered as an improved background in the subsequent DA cycles.

9.2 Case B

In this case the inflow discharges $\mathcal{Q}(t) = Q_1(t)$, $Q_2(t)$ and $Q_3(t)$ are considered as driving conditions. As opposed to case A we assume that $z_b(k)$, $b(k)$ and $C_s(k)$ are known precisely. Then, the full control vector is

$$U = (Q_1(t), Q_2(t), Q_3(t), U_*)^T,$$

where U_* stands for all model inputs which contain no uncertainty. The 'true' values of $Q_1(t)$, $Q_2(t)$ and $Q_3(t)$ are presented in Fig.3, upper/left subplot. The corresponding 'true' discharge field $Q(t,k)$ is presented at lower/right subplot.

Only observations made by *sensors 1* and *3* (see Fig.2) are used in DA. *Sensors 2* and *4* are removed on purpose. In the original sensor configuration we have at least one sensor located between two nodes where the inflow discharge is imposed. Since flow perturbations assuredly propagate from upstream to downstream, all Q_i can be resolved in this case. However, without data from *sensor 2* $Q_2(t)$ and $Q_3(t)$ can be resolved only if the flow perturbations from *node 2* could reach *sensor 1*, which is located upstream. This is a more interesting case to investigate with our method.

Here we consider the following active control sets:
*cc*1 - full control case: i.e. $U_a = U \setminus U^* = (Q_1(t), Q_2(t), Q_3(t))^T$;
*cc*2 - partial control case: $U_a = (Q_1(t), Q_2(t))^T$;
*cc*3 - partial control case: $U_a = (Q_1(t), Q_3(t))^T$;
*cc*3* - partial control case: $U_a = (Q_1(t), Q_3(t))^T$.

Numerical results for case B are summarized in Fig.6. Here, the upper/left subplot shows the background (prior) SD and the SD of the estimates (case *cc*1), for all $Q_i(t)$. As in case A, the posterior SD are strictly smaller than the background SD. One can notice that the uncertainty reduction in the estimates of $Q_2(t)$ and, especially, in $Q_3(t)$ is significantly smaller than in $Q_1(t)$, which is due to the absence of observations between *node2* and *node3*. However, the uncertainty reduction in the goal-vector (see lower subplots, case *cc*1) looks far more significant than one could expect looking at $\sigma[\delta Q_i]$. This proves that even though $Q_2(t)$ and $Q_3(t)$ are not well resolved, the sums $Q_1(t) + Q_2(t)$ and $Q_1(t) + Q_2(t) + Q_3(t)$ which dominate the flow behavior between nodes *2* and *3* and downstream *node3*, correspondingly, are well estimated.

The upper/right subplot shows results for different active control sets. Since no observations between nodes *2* and *3* are used, we try to substitute two tributaries by one 'effective' discharge. These are the partial control cases *cc*2 and *cc*3. It is interesting to note that in case *cc*2 $\sigma[\delta Q_2]$ is even larger than the background value, i.e. the uncertainty has increased. However, it terms of $\sigma[\delta G_Q]$ and $\sigma[\delta G_Z]$ the uncertainty is reduced everywhere, however the control set *cc*3 provides better reduction for *reach 2*. A simple practical conclusion from this behavior is as follows: all discharges (tributaries/offtakes, lateral inflows) located between two sensors can be combined into one lumped discharge imposed at a node nearest to the downstream sensor.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

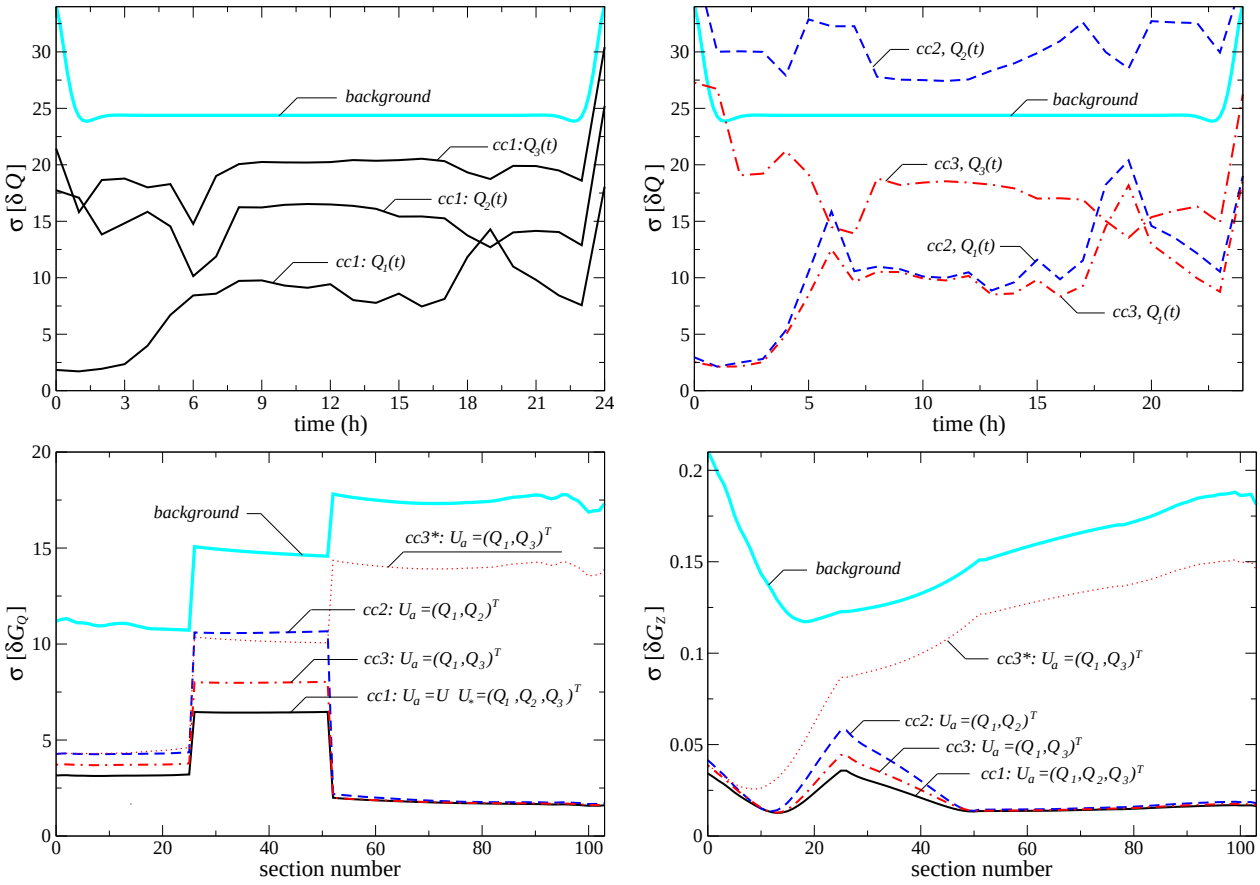

Figure 6: Eigenvalues of operators.

The lower subplots also demonstrate case $cc3^*$. This case is different from case $cc3$ in a way that the cross-correlation terms in (4.32) are neglected when $V_{\delta G}$ is computed by (3.21). This example shows that taking into account cross-correlations is absolutely vital.

In this paper the control set performance is quantified by $\sigma[\delta G_Q]$ and $\sigma[\delta G_Z]$. However, we have more information since the truncated eigenvalue decompositions of matrices $V_{\delta G_Q}$ and $V_{\delta G_Z}$ are available. This allows more delicate analysis to be performed. For example, the eigenvectors which correspond to the largest eigenvalues of $V_{\delta G_Q}$ may reveal the most dangerous combinations of control uncertainties. The task then would be to block such combinations by introducing the appropriate penalty term. The spectrum of $V_{\delta G_Q}$ is presented in Fig.7, where we notice indeed a few largest eigenvalues, well separated from the rest of the spectrum.

We can also see in Fig.8 the structure of correlations in $V_{\delta G_Q}$ and $V_{\delta G_Z}$. Such information could be useful for a general identifiability analysis, though this issue is not investigated further in this paper. Finally, in Fig.9 we present the comparison of some results obtained for different number of eigenpairs involved in the limited-memory representation of H and $V_{\delta G}$. Note that the full H has size $m = 384$. One can see that even with a relatively small number of eigenpairs both the magnitude and the shape of $V_{\delta G_Q}$ and $V_{\delta G_Z}$ are well captured.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

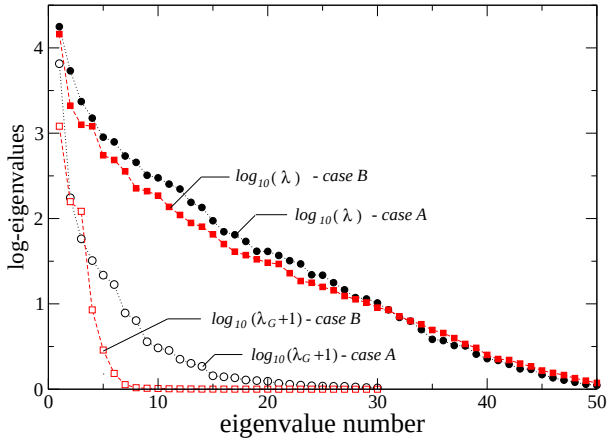

Figure 7: Eigenvalues of operators $\tilde{H}$ and $V_{\delta G}$.

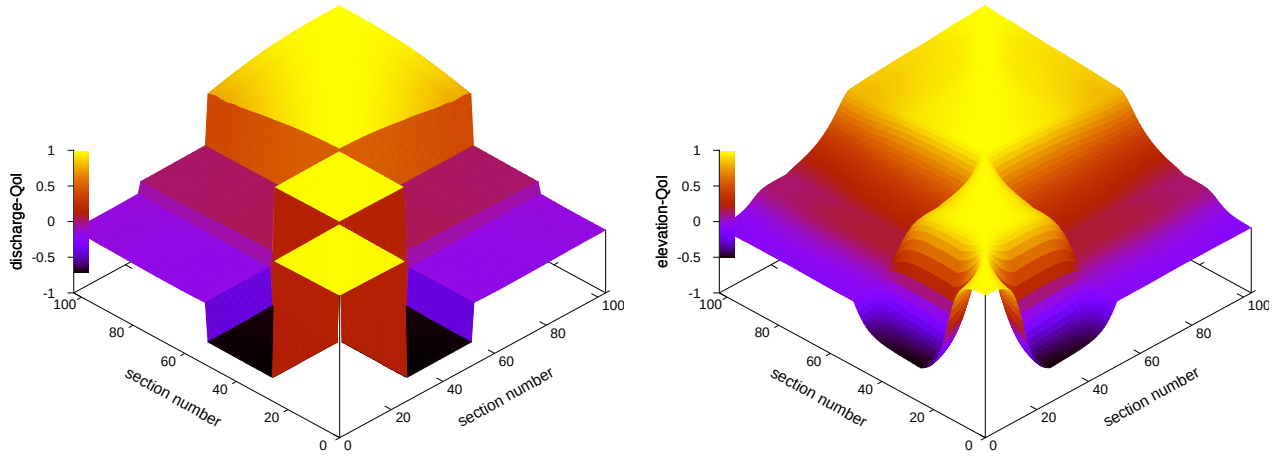

Figure 8: - Correlations in QI

Conclusions

In this paper we introduce the *control set design* concept. The need for such procedure arises in solving DA problems for models with multiple heterogeneous inputs containing significant uncertainties. In one hand, it looks appealing to include all the uncertainty-bearing inputs into the (active) control set. On the other hand, there are different reasons against such a straightforward approach. Some of them are discussed in Introduction. In order to design the control set one must be able to quantify its performance in terms of the uncertainty level in specially chosen goal-functions (QoI). Those sets for which this level is acceptable from the practical point of view are called 'sufficient'.

Technically, our method is a generalization of the standard variational UQ method. That is, the full set of the uncertainty-bearing model inputs is divided into the active and passive sets, each affecting the goal-function uncertainty covariance in different ways. The implementation

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

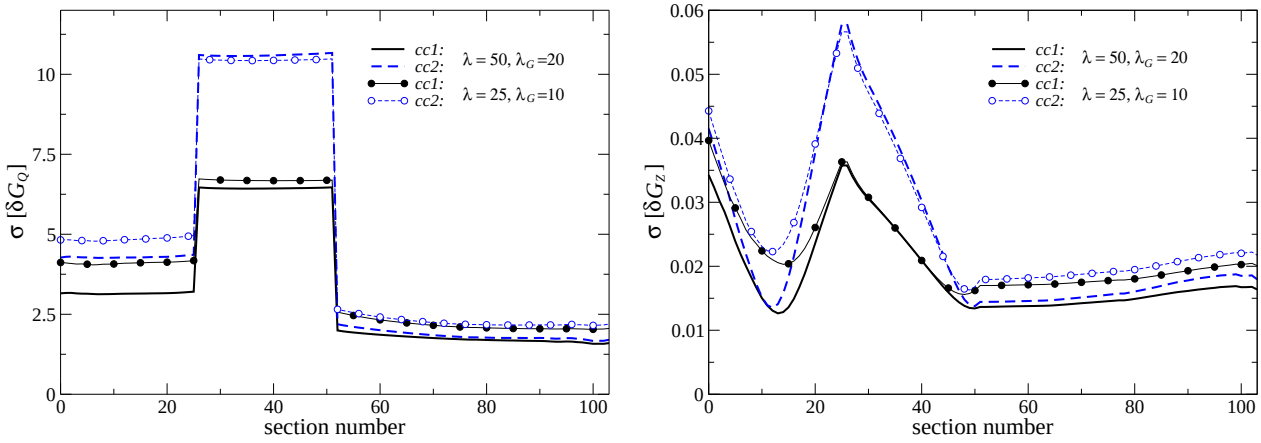

Figure 9: Standard deviation of the goal-vector built using reduced number of eigenvalues

is 'matrix free' in the sense that the limited-memory representations of the operators are used. These are constructed by means of the Lanczos procedure. Because of this, the method may be suitable for high-dimensional problems, though it still depends on the eigenvalue distribution of the operators involved. Let us note that the variational UQ method is only valid for mildly nonlinear models. For strongly nonlinear high-dimensional models the reduced-order modelling option has to be considered.

The method has been applied in the area of river hydraulics, using SIC^2 model. To the best of our knowledge, this is the first time when: a) the variational UQ method has been applied to an observed hydraulic system; b) the hydraulic system has been investigated in the context of the control set design. Two tests have been considered. In the first one the inflow discharge estimation problem under uncertainty in bathymetry (bed elevation and dilation coefficient) and the bed roughness (Strickler coefficient) is considered. The numerical experiments show that, for the chosen goal-functions, the sufficient control set must include the inflow discharge and bed elevation, whereas the Strickler coefficient and the dilation coefficient should not be controlled. It is known from the hydraulic theory that, in a steady-state regime, the bed slope and friction are related via the Manning equation. Therefore, they cannot be resolved in the steady-state (equifinality). In the transient regime, controlling both may still cause difficulties during the minimization procedure. These have been repeatedly encountered as reported in [11]. Using our method the accuracy loss due to not considering the Strickler and dilation coefficients as control variables has been accessed.

In the second test we consider the problem with several unknown inflow discharges. It is assumed that all other model inputs contain no uncertainty. In practice, knowing all tributaries seems impossible, particularly during the 'rain' season. Thus, they can be considered as spatially distributed time-dependent source terms in the continuity equation. The experiments have shown that all inflows in the area between two surface elevation sensors can be simulated by one lumped time-dependent inflow discharge imposed at a node which is nearest to the downstream sensor. Again, the accuracy loss due to this simplification has been properly accessed.

The suggested method offers an additional dimension in design of DA systems. Indeed, it is

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

more usual to talk about design of observations. More generally, one could talk about optimal experimental design, which also includes the possibility to influence the system response. In hydraulics this can be done using devices (gates, weirs, etc.) The future work may include a few directions. One of them is the design of integrated controls, i.e. controls which may include certain combinations of the existing model inputs or invariants (e.g. characteristic variables). Another direction is 'globalization' of the method and its use in the framework of stochastic UQ methods (e.g. MCMC) to accelerate the latter.

Acknowledgments

The authors thank Jean-Pierre Baume and David Dorchies, the members of the G-EAU team, IRSTEA-Montpellier, for their help with the SIC^2 software acquaintance, and Prof. Victor Shutyaev, Institute of Numerical Mathematics, Russian Academy of Science, Moscow, for useful discussions on the methodological issue.

Appendix I. Short description of the SIC^2 model

SIC^2 is a hydrodynamic model that has been developed at IRSTEA (CEMAGREF) for more than 30 years. It is an industrial software distributed to different type of users, including consultant companies, irrigation canal managers, engineering schools and universities all over the world (France, Spain, Italy, Portugal, Netherlands, England, Germany, Morocco, Tunisia, Egypt, Senegal, USA, Mexico, Pakistan, Iraq, Sri Lanka, Vietnam, China, etc). It has many innovative features, that make it the leader among this type of software, for some specific applications including irrigation canal design, irrigation canal manual or automatic control, and data assimilation.

The basic features of the SIC^2 model are as follows:
a) the model is based on the full Saint-Venant 1D non-linear partial derivative equations;
b) the model is discretized using the semi-implicit Preissmann scheme [28], for its brief description see Appendix III;
c) two-step solution approach is used: the boundary conditions for the reaches are computed first, then the water profiles in the reaches are recovered. The second step can be potentially implemented in a parallel setting;
d) in the version of SIC^2 used, only subcritical flows are allowed in the unsteady mode, but local critical and supercritical flows can be managed within the cross-devices [3];
e) the canal can be composed of a minor, medium (with a different Strickler coefficient) and major bed (can be used as a storage area during the canal overflow events) and ponds at nodes. The minor - medium bed interactions are modelled using the Debord formula, validated on large laboratory experiments, giving better results than the more classical Divided Channel Method [9];
f) the model allows the pressurized flow conditions using the Preissmann slot approach;
g) the model has two separate modules: one calculating real steady flow solutions, even in

[3] In the recent versions of SIC^2 the supercritical flow regime is supported in both steady and unsteady flow calculation, implementing ideas developed in [30]

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

branched and looped networks, without a priori knowledge of flow repartition, and one calculating unsteady flow solutions on the same type of networks. The steady flow module is able to manage any well posed boundary conditions, such as water levels, discharges and rating curves, at either upstream, downstream or intermediate boundary conditions.

One original and unique feature of SIC^2 is to be able to describe any operational rule or algorithm either of feedforward or feedback type, moving any dynamical cross or lateral device (gate, weir, pump, etc) using any measurement over the hydraulic system. This allows to design, test and optimize management rules on irrigation canal, or on rivers having dynamical devices (dams, hydroelectric power plants, moving weirs, etc). Some predefined algorithms are already available into a library (ex: PID), even some of them with auto-tuning procedures (ex: ATV). More advanced algorithms can be implemented using several programming languages (ex: MatLab, Scilab, Fortran, WDLangage) taking advantage of an embedded interface of these languages into SIC^2. Using this feature some very advanced MIMO (Multi Input, Multi Output) automatic controllers have been tested such as LQG, ℓ_1, H_∞ [23].

Another original and unique feature of SIC^2 is its capability to model complex hydraulic structures that are encountered on irrigation canals, such as hydrodynamic gates (AMIL, AVIS, AVIO, Mixte gates). Also, the modelling of more classical devices such as gates and weirs are modelled in such a way that it allows all possible flow conditions and all continuous transitions between these conditions.

For a detailed description of the model see [22] and the User's Manual at the website: http://sic.g-eau.net.

Appendix II. Details of using the AD

A1. Computing the Hessian-vector product

The task is to produce a code for computing the Hessian-vector product $\tilde{H} \cdot v$ by (3.22). We start from subroutine $forward(U, U^*, Y^*, J(U))$, which calls subroutines $model(U, Y^*, J(U))$ and $costB(U, U^*, J(U))$. The latter evaluates the background term in the cost function (3.11). Subroutine $costO(C(X|_t), Y^*, J(U))$, which evaluates the residual term in (3.11), is called from inside the time loop in $model()$.

First, we create subroutine $forwardH(U, Y^*, J(U))$ by removing *call* $costB(U, U^*, J(U))$ from $forward()$. By differentiating $forwardH(\cdot)$ (output $J(U)$ with respect to input U, the 'tangent' mode) we get the following TL subroutines:
$forwardH\_d\,(U, U\_d, Y^*, J(U), J(U)\_d)$,
$model\_d(U, U\_d, Y^*, J(U), J(U)\_d)$,
$costO\_d(C(X|_t), C(X|_t)\_d, Y^*, J(U), J(U)\_d)$.
In $forwardH\_d()$, the input variable U_d is the vector of perturbations in U, the output variable $J(U)\_d$ is the associated perturbation in the cost-function $J(U)$. In $model\_d()$ the sequence of operators involved with computing the state X totally replicates the one in $model()$, whereas the state perturbation X_d is computed alongside the state X. Similarly, by differentiating $forwardH()$ under the 'reverse' mode we get the adjoint subroutines:
$forwardH\_b\,(U, U\_b, Y^*, J(U), J(U)\_b)$,
$model\_b(U, U\_b, Y^*, J(U), J(U)\_b)$,
$costR\_b\ (C(X|_t), C(X|_t)\_b, Y^*, J(U), J(U)\_b)$.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

In $forwardH\_b$, the input variable $J(U)\_b$ gives the scale of adjoint perturbation, the output variable U_b contains the adjoint sensitivities $J'_U(U)$.

The structure of $forwardH\_b\,()$ is as follows. First, it calls $model()$, then $model\_b()$. In turn, $model\_b()$ consists of two blocks. The first block simply replicates the sequence of operators in $model()$ with the difference that the system's trajectory is 'pushed' into memory, when necessary. This trajectory is 'popped' out from memory and used in the second block of $model\_b()$, which actually implements the adjoint model. The first call to $model()$ in $forwardH\_b\,()$ is, therefore, redundant and must be removed.

The Hessian-vector product $\tilde{H}\cdot v$ in (3.22) is defined by the successive solution of the tangent linear and adjoint models. The information exchange between the two models takes place in the observation space $\mathcal{Y}$. The code for computing $\tilde{H}\cdot v$ could be constructed on a basis of $model\_b()$. One approach would be to insert manually the code lines from $model\_d$ involved with computing X_d at appropriate locations in the first block of $model\_b$, which must be identified from $model\_d$. Another approach would be to substitute the first block in $model\_b()$ by the operator sequence from $model\_d()$, in which case one must introduce the 'push' operators at appropriate locations as in the original version of $model\_b()$. Then, the information transfer from the TL to adjoint model has to be arranged. Unfortunately, both approaches require a very substantial manual post-processing of $model\_b()$.

Since the manual interventions have to be minimized, a better way would be to generate an approximation to the desired modification of $model\_b$ by the AD tool. One possible approach is presented below. First, we create $modelH\_d()$, that is different from $model\_d()$ in a way that is calls $costO()$ instead of $costO\_d()$. Next, we modify $forwardH\_d()$, so that it calls $modelH\_d()$ instead of $model\_d()$. The adjoint code is obtained by differentiating $forwardH\_d()$ (output $J(U)$ with respect to input U, the 'reverse' mode). As a result we get the adjoint subroutines:
$forwardH\_d\_b\,(U, U\_b, U\_d, Y^*, J(U), J(U)\_b, J(U)\_d)$,
$modelH\_d\_b(U, U\_b, U\_d, Y^*, J(U), J(U)\_b, J(U)\_d)$,
$costO\_b(C(X|_t), C(X|_t)\_b, Y^*, J(U), J(U)\_b)$.

The following modifications in $modelH\_d\_b()$ must be introduced to provide the information exchange from the TLM to the adjoint:
1. add $push(C(X|_t)\_d)$ right after the existing $push(C(X|_t))$ in the first block;
2. add $pop(C(X|_t)\_d)$ just before the existing $pop(C(X|_t))$ in the second block;
3. in the existing *call* $costO\_b(C(X|_t), C(X|_t)\_b, Y^*, J(U), J(U)\_b)$ substitute $C(X|_t)$ by $C(X|_t)\_d$, and use $Y^* \equiv 0$.

Then, calling $forwardH\_d\_b\,()$ with $U\_d = v$ we obtain $(R'_U(\bar{U}))^* O^{-1} R'_U(\bar{U})\cdot v = U\_b$, which is a key part of (3.22).

A2. Computing the covariance-vector product

We start from the forward subroutine $forwardG(U, G(U))$, which calls $modelG(U, G(U))$. The latter includes call to $costG(D(X|_t), G(U))$, which computes the QI-vector $G(U)$. Otherwise, it is exactly the same as the previously considered $model(U, Y^*, J(U))$. Here we follow the approach presented in Sec. 9.2. Thus, the first step is to generate the TL model. By differentiating $forwardG$ using the 'tangent' mode (output $G(U)$ (vector!) with respect to input U) we get the TL subroutines:
$forwardG\_d\,(U, U\_d, G(U), G(U)\_d)$,

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

$modelG\_d(U, U\_d, Y^* \equiv 0, G(U), G(U)\_d)$,
$costG\_d(D(X|_t), D(X|_t)\_d, G(U), G(U)\_d)$.
We modify $modelG\_d()$ by substituting *call* $costG\_d(D(X|_t), D(X|_t)\_d, G(U), G(U)\_d)$ by *call* $costG(D(X|_t), G(U))$. The adjoint code is obtained by differentiating $forwardG\_d()$ using the 'reverse' mode (output $G(U)$ with respect to input U). As a result we get the adjoint subroutines:
$forwardG\_d\_b\,(U, U\_b, U\_d, D(U), D(U)\_b, D(U)\_d)$,
$modelG\_d\_b(U, U\_b, U\_d, Y^*, D(U), D(U)\_b, D(U)\_d)$.

Let us note that for computing $V_{\delta G} \cdot v$ by the formula (3.21) the adjoint operator is applied to the input vector first. Thus, the information transfer from the adjoint to the TLM takes places in the control space $\mathcal{U}$, which makes the transfer issue trivial. That is, $v \in \mathcal{G}$ must be supplied in $D(U)\_b$ in $forwardG\_d\_b$, the output $u \in \mathcal{U}$ is presented in U_b. Then $u = V_{\delta U} \cdot u$ must be supplied in U_d, the final result $V_{\delta G} \cdot v \in \mathcal{G}$ can be read from $D(U)\_d$.

The only modification needed is related to the fact that in $modelG\_d\_b$ the forward and the TL models are running first (first block), the adjoint model is running second (second block), whereas it must be vice versa. This can be achieved as follows. Let us introduce a logical variable *mode* as follows:
subroutine $forwardG\_d\_b\,(mode, \ldots)$
...
call $modelG\_d\_b(mode, \ldots)$
...
end $forwardG\_d\_b$

subroutine $modelG\_d\_b(mode, \ldots)$
...
$SAVE$
$if(mode = 1) then\ block1\ (forward/TLM)$
$if(mode = 2) then\ block2\ (adjoint)$
end $modelG\_d\_b$

Then, the Lanczos driver must include the call to $forwardG\_d\_b\,(mode = 1, \cdots)$ before starting iterations. This will provide the system trajectory, needed for running the adjoint model for the very first time. However, inside the main loop, the sequence of calls must be:
call $forwardG\_d\_b\,(mode = 2, \cdots)$
compute $v = V_{\delta G} \cdot v$
call $forwardG\_d\_b\,(mode = 1, \cdots)$.

Appendix III. Preissmann discretization scheme

Let us consider a function $f(x,t)$ discretized using the stensil presented in Fig.10. We denote the time increment $\Delta f_i = f_i^{j+1} - f_i^j$, and define $f(x,t)$ and its derivatives at point M as follows:

$$f|_M = (1-\theta)\frac{f_i^j + f_{i+1}^j}{2} + \theta\,\frac{f_i^{j+1} + f_{i+1}^{j+1}}{2} = \frac{f_i^j + f_{i+1}^j}{2} + \theta\,\frac{\Delta f_{i+1} + \Delta f_i}{2},$$

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

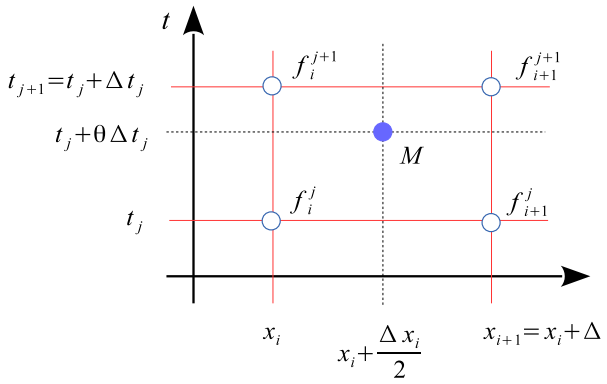

Figure 10: Preissmann discretization stencil

$$\frac{\partial f}{\partial x}\Big|_M = (1-\theta)\frac{f_{i+1}^j - f_i^j}{\Delta x_i} + \theta\,\frac{f_{i+1}^{j+1} - f_i^{j+1}}{\Delta x_i} = \frac{f_{i+1}^j - f_i^j}{\Delta x_i} + \theta\,\frac{\Delta f_{i+1} - \Delta f_i}{\Delta x_i},$$

$$\frac{\partial f}{\partial t}\Big|_M = \frac{1}{2}\left(\frac{f_{i+1}^{j+1} - f_{i+1}^j}{\Delta t_j} + \frac{f_i^{j+1} - f_i^j}{\Delta t_j}\right) = \frac{1}{2}\,\frac{\Delta f_{i+1} + \Delta f_i}{\Delta t_j},$$

where θ is the Preissmann implicitation coefficient. Applying the above formulas to equations (7.63)-(7.64) one gets a system of linear algebraic equations for Δf_i, $i = 1, \ldots n$, with the two-block-diagonal matrix, each block having dimension 2×2. This system is solved by performing the forward and backward sweeps. A few iterations at each time step are necessary to resolve the nonlinearity. This is the essence of the Preissmann method.

References

[1] A Abdolghafoorian and L Farhadi. Uncertainty quantification in land surface hydrologic modeling: toward an integrated variational data assimilation framework. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(6):2628–2637, 2016.

[2] S. Akella and I.M. Navon. Different approaches to model error formulation in 4D-Var: a study with high resolution advection schemes. *Tellus*, 61A:112–128, 2009.

[3] O.M. Alifanov, E.A. Artyukhin, and S.V. Rumyantsev. *Extreme Methods for Solving Ill-Posed Problems with Applications to Inverse Heat Transfer Problems.* Begel House Publishers, 1996.

[4] H. Auvinen, J.M. Bardsley, H. Haario, and T. Kauranne. The variational kalman filter and an efficient implementation using limited memory bfgs. *Int. J. Num. Meth. Fluids*, 64(3):314–335, 2010.

[5] C. Burstedde and O. Ghattas. Algorithmic strategies for full waveform inversion: 1d experiments. *Geophysics*, 74:WCC37–WCC46, 2009.

[6] P. Courtier, J.-N. Thepaut, and A. Hollingsworth. A strategy for operational implementation of 4d-var, using an incremental approach. *Quart. J. Roy. Meteor. Soc.*, 120:1367–1387, 1994.

[7] M. Dashti, K.J.H. Law, and J. Stuart, A.M.and Voss. Map estimators and posterior consistency in bayesian nonparametric inverse problems. *Inverse Problems*, 29:095017, 2013.

[8] E.G.R. Davies and S.P. Simonovic. Global water resources modeling with an integrated model of the social-economic-environmental system. *Advances in Water Resources*, 34(6):684–700, 2011.

[9] J.N. Fernandes and A.H. Cardoso. Flow structure in a compound channel with smooth and rough floodplains. *EWRA European Water Publications*, 38:3–12, 2012.

[10] I. Gejadze, F.-X. Le Dimet, and V. Shutyaev. On optimal solution error covariances in variational data assimilation problems. *Journal of Computational Physics*, 229(6):2159–2178, 2010.

[11] I. Gejadze and P.-O. Malaterre. Discharge estimation under uncertainty using variational methods with application to the full saint-venant hydraulic network model. *Int. J. Num. Meth. Fluids*, 34:127–147, 2016.

[12] L. Hascoët and V. Pascual. Tapenade 2.1 user's guide. *INRIA Technical Report*, 0300, 2004.

[13] H. Hoang and R. Baraille. Stochastic simultaneous perturbation as powerful method for state and parameter estimation in high dimensional systems. In *Advances in Mathematical Research*, volume 20, pages 117–148. Nova Science Publishers, 2015.

[14] T. Isaac, N. Petra, G. Stadler, and O. Ghattas. Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the antarctic ice sheet. *J. Comput. Physics*, 296:348–368, 2015.

[15] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[16] A.G. Kalmikov and P. Heimbach. A hessian-based method for uncertainty quantification in global ocean statez estimation. *SIAM J. Sci. Comput.*, 36(5):S267–S295, 2014.

[17] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38A(2):97–110, 1986.

[18] C. Lieberman and K. Wilcox. Goal-oriented inference: Approach, linear theory, and application to advection diffusion. *SIAM J. Sci. Comput.*, 34(4):A1880–A1904, 2012.

[19] J.-L. Lions. *Contrôle Optimal des Systèmes Gouvernés par des Équations aux Dérivées Partielles*. Dunod, Paris, 1968.

Author-produced version of the article published in Journal of Computational Physics, 2016, N°325, p.358-379
The original publication is available at http://www.sciencedirect.com
http://dx.doi.org/10.1016/j.jcp.2016.08.029

[20] C. Liu, Q. Xiao, and B. Wang. An ensemble-based four-dimensional variational data assimilation scheme. part i: Technical formulation and preliminary test. *Mon. Weather Rev.*, 136:3363–3373, 2008.

[21] Honnorat M., Monnier J., and Le Dimet F.-X. Lagrangian data assimilation for river hydraulics simulations. *Comput. Visu. Sc.*, 12(5):235–246, 2009.

[22] P.-O. Malaterre, J.-P. Baume, and D. Dorchies. Simulation and integration of control for canals software (sic^2), for the design and verification of manual or automatic controllers for irrigation canals. In *USCID Conference on Planning, Operation and Automation of Irrigation Delivery Systems*, pages 377–382, Phoenix, Arizona, December 2-5 2014.

[23] P.-O. Malaterre and M. Khammash. ℓ_1 controller design for a high-order 5-pool irrigation canal system. *ASME Journal of Dynamic Systems, Measurement, and Control*, 125:639–645, 2003.

[24] G.I. Marchuk, V.I. Agoshkov, and V.P. Shutyaev. *Adjoint Equations and Perturbation Algorithms in Nonlinear Problems*. CRC Press Inc., New York, 1996.

[25] N. Martin and J. Monnier. Adjoint accuracy for the full-stokes ice flow model: limits to the transmission of basal friction variability to the surface. *The Cryosphere*, 8:721–741, 2014.

[26] P. Moireau and D. Chapelle. Reduced-order unscented Kalman filtering with application to parameter identification in large-dimensional systems. *ESAIM: Control, Optimization and Calculus of Variations*, 17(2):380–405, 2011.

[27] 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):55–79, 1998.

[28] P. Novak, V. Guinot, A. Jeffrey, and D.E. Reeve. *Hydraulic Modelling - An Introduction: Principles, Methods and Applications*. CRC Press, 2010.

[29] P. Sarma, L.J. Durlofsky, H. Aziz, and W.H. Chen. Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Comp. Geosciences*, 10:3–36, 2005.

[30] C. Sart, J.-P. Baume, P.-O. Malaterre, and V. Guinot. Adaptation of preissmann's scheme for transcritical open channel flows. *J. of Hydraulic Research*, 48(4):428–440, 2010.