



Water pollution estimation based on the 2D transport–diffusion model and the Singular Evolutive Interpolated Kalman filter

Estimation de la pollution de l'eau basée sur un modèle 2D de transport–diffusion et sur le filtre de Kalman singulier évolutif interpolé

Tran Thu Ha^a, Pham Dinh Tuan^{b,*}, Hoang Van Lai^a, Nguyen Hong Phong^a

^a Vietnamese Aca. of Sci. and Tech. – 18 Hoang Quoc Viet, Institute of Mechanics – 264 Doi Can, and University of Engineering and Tech. – 144 Xuan Thuy, Hanoi, Vietnam

^b Jean-Kuntzmann Laboratory, INPG/UJF/CNRS, BP 53, 38041 Grenoble cedex, France

ARTICLE INFO

Article history:

Received 4 May 2012

Accepted 17 October 2013

Available online 6 January 2014

Keywords:

Data assimilation

Finite volume method

2D Hydraulic model

Kalman filter

Pollution

Mots-clés:

Assimilation de données

Méthode des volumes finis

Modèle hydraulique 2D

Filtre de Kalman

Pollution

ABSTRACT

In this paper, the 2D mathematical water pollution model describing the transport–diffusion processes of some contaminant substances in Thanh Nhan Lake in Hanoi is considered. The finite-volume method is used to solve the model equations. The Singular Evolutive Interpolated Kalman filter is applied to evaluate the pollution level at arbitrary mesh point based only on a small number of measurement points.

© 2013 Académie des sciences. Published by Elsevier Masson SAS. All rights reserved.

RÉSUMÉ

Dans cet article, on étudie le processus de transport–diffusion de substances contaminantes dans le lac Thanh Nhan de Hanoi. Cela se traduit par un modèle mathématique bimensionnel d'équations aux dérivées partielles. Une approximation numérique de la solution est obtenue par une méthode de volumes finis. Des mesures de pollution sont faites en un nombre de points faible. Pour évaluer la pollution en tout point de discrétisation, on propose d'utiliser la méthode du filtre de Kalman singulier évolutif et interpolé.

© 2013 Académie des sciences. Published by Elsevier Masson SAS. All rights reserved.

1. Introduction

Water pollution is nowadays one of the most serious problems. Water quality influences directly public health. It is of interest to be able to evaluate the water pollution level at any point in the area of interest and at any time. But in practice, only a small number of measurements can be made.

In this paper, 2D hydraulic and pollution models are used to describe the transport of pollution substances. The linear 2D water pollution has been studied by the semi-group method in [1] to prove the unique existence of the problem solution. The nonlinear 2D-Imech water pollution model has been studied in [2]. This model has been developed to simulate the transport of pollution substances and thus can be used to estimate the pollution level if the initial values for the model

* Corresponding author. Tel.: +33 476514423; fax: +33 476631263.

E-mail addresses: ttha@imech.ac.vn (T.H. Tran), apham@sfr.fr (D.T. Pham).

equations are known and the model parameters are adequately specified. Since these initial values for the model are unknown, estimates are used instead, which are often taken as the average values over a long run of the model. As these estimates will not be exact and the model can be imperfect or not adequately specified, pollution estimation from the model will contain errors. Data assimilation techniques permit to combine the available limited number of measurements and the model to obtain a more precise estimate of the system state. In this paper, we shall use the Singular Evolutive Interpolated Kalman (SEIK) filter as a data assimilation method, as it has been seen to be successful in many large-scale data assimilation problems [3–5]. The SEIK filter is a kind of Kalman filter which incorporates two basic tools:

- interpolation: the Kalman filter is designed for a linear model. For nonlinear models, one often makes a linearization based on the first-order Taylor approximation, leading to the so-called extended Kalman filter. The SEIK filter uses interpolation instead, which handles better the nonlinearity of the model;
- order reduction: in the present problem, as well as in many other problems in geophysics, the system is described by several vector (or scalar) fields, which after discretization results in a big state vector of very high dimension. The filter error covariance matrix would then be huge, leading to unaffordable computation cost. The SEIK filter incorporates an order reduction, by making the filter error covariance matrix singular with low rank. This permits its application to very large systems.

The SEIK filter was first introduced in [5] and is further described, for example, in [3,6]. It supersedes the previous SEEK (singular evolutive extended Kalman) filter [4], which, similar to the extended Kalman filter, is based on the first-order linearization of the system. An advantage of the SEIK over the SEEK filter, apart from the fact that it is more robust against a nonlinearity of the system, is that it avoids the computation of the gradient (Jacobian) of the system transition operator, which is not easy for complex systems.

There is a strong similarity between the SEEK filter and the Ensemble Kalman Filter (EnKF) [7,8] and the Unscented Kalman Filter (UKF) [9–11]. The EnKF, however, used ordinary sampling, while the SEIK filter uses second-order exact sampling, which propagates more accurately the filter error covariance matrix. The UKF, on the other hand, uses deterministic sampling points and more points than necessary, while the SEIK and EnKF use the minimum number of interpolating points (ensemble members). Further, the UKF in its original form does not incorporate order reduction and thus is not applicable to very large systems.

2. Formulation of the water pollution problem

A 2D water pollution model consists of a hydraulic model and a transport–diffusion of pollution substances model. In our hydraulic model, the Saint-Venant equation is used as follows [12]:

$$\frac{\partial z}{\partial t} + \frac{\partial(uh)}{\partial x} + \frac{\partial(vh)}{\partial y} = 0, \quad \text{in } \Omega \quad (1)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + g \frac{\partial z}{\partial x} = - \frac{gu(u^2 + v^2)^{1/2}}{K_x^2 h^{4/3}}, \quad \text{in } \Omega \quad (2)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + g \frac{\partial z}{\partial y} = - \frac{gv(u^2 + v^2)^{1/2}}{K_y^2 h^{4/3}}, \quad \text{in } \Omega \quad (3)$$

Here,

- Ω is a bounded domain of R^2 with a boundary Γ ,
- z is the free surface elevation,
- h is the water depth,
- u is the average velocity in the x direction,
- v is the average velocity in the y direction,
- g is the gravity acceleration,
- K_x is the Strickler coefficient in the x direction,
- K_y is the Strickler coefficient in the y direction.

We suppose that there are m substances dissolved in water. Then our transport and diffusion processes of pollution substances are described by the following equation [13]:

$$\frac{\partial C_i}{\partial t} + u \frac{\partial C_i}{\partial x} + v \frac{\partial C_i}{\partial y} - D_i \left(\frac{\partial^2 C_i}{\partial x^2} + \frac{\partial^2 C_i}{\partial y^2} \right) = f_i, \quad \text{in } \Omega, \quad i = 1, \dots, m \quad (4)$$

Here,

- C_i is the concentration of the i -th substance,

- f_i is the pollution source of the i -th substance (see formula (22)),
- D_i is the diffusion coefficient of i -th substance.

Eq. (4) can be rewritten in the vector form as follows:

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} + v \frac{\partial C}{\partial y} - \mathbf{D} \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) = f, \quad \text{in } \Omega \tag{5}$$

where: $C = (C_1, \dots, C_m)$, $f = (f_1, \dots, f_m)$, \mathbf{D} is the diagonal matrix with diagonal elements D_1, \dots, D_m . The initial conditions are, $U = (u, v)$ denoting the water velocity vector:

$$\begin{aligned} U(x, y, 0) &= U_0(x, y) \\ z(x, y, 0) &= z_0(x, y) \\ C(x, y, 0) &= C_0(x, y) \end{aligned}$$

In the slow changing process for water pollution problem, the boundary conditions are:

- $U(x, y, t) \cdot \mathbf{n} = \bar{U}_{in}(t)$, $C(x, y, t) \cdot \mathbf{n} = \bar{C}_{in}(t)$ on inflow boundary Γ_1 ,
- $z(x, y, t) = \bar{z}(t)$, $\partial C / \partial n = 0$ on outflow boundary Γ_2 ,
- $U(x, y, t) \cdot \mathbf{n} = 0$, $C(x, y, t) \cdot \mathbf{n} = 0$ on solid wall S_w ,

where \mathbf{n} is the unit vector normal to Γ , Γ_2 and Γ_1 are respectively the outflow and inflow boundaries of the Ω domain, S_w is the solid wall of Ω and $\Gamma = \Gamma_1 \cup \Gamma_2 \cup S_w$ is the boundary of the region Ω .

3. Algorithm for calculating 2D water pollution

3.1. Equation for 2D flow system

To numerically solve the above model equations, a cell centered finite volume method is used (see [12]). The type of mesh employed in this method is an unstructured triangulation of the solution domain, enabling arbitrarily shaped geometries to be accommodated more easily than a square grid system (see [12]). The mesh points of the grid are numbered in some given but arbitrary way.

To apply the finite-volume method, Eqs. (1)–(3) are rewritten as follows:

$$\frac{\partial V}{\partial t} + \frac{\partial A}{\partial x} + \frac{\partial B}{\partial y} = F \tag{6}$$

where:

$$\begin{aligned} V &= \begin{pmatrix} z \\ u \\ v \end{pmatrix}, \quad A = \begin{pmatrix} uh \\ \frac{1}{2}u^2 + gz \\ uv \end{pmatrix}, \quad B = \begin{pmatrix} vh \\ uv \\ \frac{1}{2}v^2 + gz \end{pmatrix} \\ F &= \begin{pmatrix} 0 \\ -gu \frac{\sqrt{u^2+v^2}}{K_x^2 h^{4/3}} + u \frac{\partial v}{\partial y} \\ -gv \frac{\sqrt{u^2+v^2}}{K_y^2 h^{4/3}} + v \frac{\partial u}{\partial x} \end{pmatrix} \end{aligned}$$

We divide the domain Ω , with N_p mesh points, into N small cells Ω_j , that have boundaries γ_j , $j = 1, \dots, N$. A small cell Ω_j is described in Fig. 1 with index points (x_{jk}, y_{jk}) , center point (x_{jc}, y_{jc}) and unit vectors $\mathbf{n}_{j,ik}$ normal to the lines with lengths $l_{j,ik}$ joining two points (x_{ji}, y_{ji}) and (x_{jk}, y_{jk}) ($i, k = 1, \dots, 3$ with $i \neq k$).

In this small cell Ω_j , integrating two sides of Eq. (6) with respect to the spatial variables x and y , we get the formula:

$$\int_{\Omega_j} \frac{\partial V}{\partial t} dx dy + \int_{\Omega_j} \nabla \cdot (A, B) dx dy = \int_{\Omega_j} F dx dy$$

where $\nabla = (\partial/\partial x, \partial/\partial y)$. We assume that in the small cell Ω_j the variables $z, u, v, h = z - z_{\text{bottom}}$ and F can be approximated as constants. Therefore, using Green formula, the above formula can be rewritten as follows:

$$\frac{dV}{dt} S + \oint_{\gamma_j} (A, B) \cdot \mathbf{n} d\gamma_j = FS \tag{7}$$

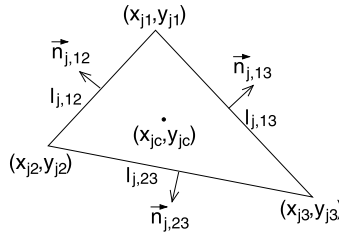


Fig. 1. A cell.

In formula (7), S is the cell's area, \mathbf{n} is the unit vector normal to γ_j , with components in x and y directions denoted by n_x and n_y respectively.

To solve the above equation, we introduce the discrete scheme of integration of any function w on the boundary γ_j of cell Ω_j :

$$\oint_{\gamma_j} w \cdot n_x d\gamma_j = \frac{w_{j1} + w_{j2}}{2} \cdot n_{x,12}l_{12} + \frac{w_{j2} + w_{j3}}{2} \cdot n_{x,23}l_{23} + \frac{w_{j3} + w_{j1}}{2} \cdot n_{x,31}l_{31} \tag{8}$$

$$\oint_{\gamma_j} w \cdot n_y d\gamma_j = \frac{w_{j1} + w_{j2}}{2} \cdot n_{y,12}l_{12} + \frac{w_{j2} + w_{j3}}{2} \cdot n_{y,23}l_{23} + \frac{w_{j3} + w_{j1}}{2} \cdot n_{y,31}l_{31} \tag{9}$$

where in the j -th cell, w_{jk} is the value of w at the point number jk with coordinate (x_{jk}, y_{jk}) ($k = 1, 2, 3$).

In the small cell Ω_j , the functions $\partial w/\partial x$ and $\partial w/\partial y$ are assumed to be constants. Then, they can be calculated using the following formulas:

$$\frac{\partial w}{\partial l_{12}} = \frac{\partial w}{\partial x} \frac{\partial x}{\partial l_{12}} + \frac{\partial w}{\partial y} \frac{\partial y}{\partial l_{12}} \tag{10}$$

$$\frac{\partial w}{\partial l_{23}} = \frac{\partial w}{\partial x} \frac{\partial x}{\partial l_{23}} + \frac{\partial w}{\partial y} \frac{\partial y}{\partial l_{23}} \tag{11}$$

Therefore, the functions $\partial w/\partial x$ and $\partial w/\partial y$ can be discretized by the following formulas:

$$\frac{\partial w}{\partial x} = \frac{(w_{j2} - w_{j1})(y_{j3} - y_{j1}) - (w_{j3} - w_{j1})(y_{j2} - y_{j1})}{(x_{j2} - x_{j1})(y_{j3} - y_{j1}) - (x_{j3} - x_{j1})(y_{j2} - y_{j1})} \tag{12}$$

$$\frac{\partial w}{\partial y} = \frac{(w_{j3} - w_{j1})(x_{j2} - x_{j1}) - (w_{j2} - w_{j1})(x_{j3} - x_{j1})}{(x_{j2} - x_{j1})(y_{j3} - y_{j1}) - (x_{j3} - x_{j1})(y_{j2} - y_{j1})} \tag{13}$$

The value of function w in the center of the j -th is denoted by $w^{(j,c)}$ and is approximated by the formula:

$$w^{(j,c)} = w_{j1} + \frac{\partial w}{\partial x}(x_{j1} - x_{jc}) + \frac{\partial w}{\partial y}(y_{j1} - y_{jc}) \tag{14}$$

If the k -th mesh point is the joint point of M cells (see the left of Fig. 2), then the value of a function w at this point can be discretized by the following formula:

$$w_k = \frac{1}{M} \sum_{m=1}^M w^{(jm,c)} \tag{15}$$

where $w^{(jm,c)}$ ($m = 1, \dots, M$) are the values of function w at M cell centers (see the right-hand side of Fig. 2).

3.2. Discrete calculation scheme for (z, u, v)

For 2D flow problem the functions (z, u, v) can be calculated by the following scheme:

- (i) the starting values of (z, u, v) in N_p mesh points are given;
- (ii) the values of (z, u, v) in N_p mesh points are called the values in the previous step;

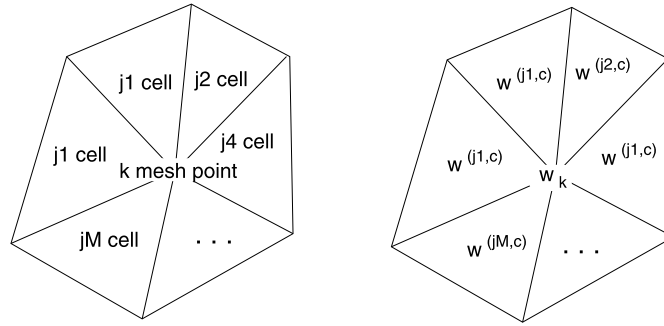


Fig. 2. Left: Joint point of M cells (the k -th mesh point). Right: w_k at this joint point of M cells.

(iii) calculate the integrals on boundary γ_j arising from Eq. (7) such as:

$$I_z = \oint_{\gamma_j} (hun_x + hvn_y) d\gamma_j \tag{16}$$

$$I_u = \oint_{\gamma_j} \left[\left(\frac{u^2}{2} + gz \right) n_x + uvn_y \right] d\gamma_j \tag{17}$$

$$I_v = \oint_{\gamma_j} \left[\left(\frac{v^2}{2} + gz \right) n_y + uvn_x \right] d\gamma_j \tag{18}$$

by a similar way as in (8), (9) with function values (z, u, v) and $h = z - z_{\text{bottom}}$ obtained from the previous step;

(iv) by the same way as in formula (14), we calculate the values of (z, u, v) at the center of the cells Ω_j ($j = 1, \dots, N$) denoted by (z_t, u_t, v_t) . These z_t, u_t, v_t values are called the values of z, u, v at the cell center of Ω_j in the previous step;

(v) calculate $h = z_t - z_{\text{bottom}}$ at the center of the cell Ω_j ;

(vi) the functions $\partial u_t / \partial x, \partial u_t / \partial y, \partial v_t / \partial x$ and $\partial v_t / \partial y$ are calculated in a similar way as in (12) and (13);

(vii) then using Eq. (7), the values of (z, u, v) at the center of the cell Ω_j in the current step can be solved using the following equations:

$$z = z_t - \frac{\Delta t}{S} I_z \tag{19}$$

$$u = u_t + \left[u_t \frac{\partial v_t}{\partial y} - \frac{g u_t \sqrt{u_t^2 + v_t^2}}{K_x^2 h^{4/3}} \right] \Delta t - \frac{\Delta t}{S} I_u \tag{20}$$

$$v = v_t + \left[v_t \frac{\partial u_t}{\partial x} - \frac{g v_t \sqrt{u_t^2 + v_t^2}}{K_y^2 h^{4/3}} \right] \Delta t - \frac{\Delta t}{S} I_v \tag{21}$$

(viii) using the values of (z, u, v) in the current step at the centers of N cells, we calculate the values of (z, u, v) at N_p mesh points in the same way as in formula (15);

(ix) for the next time step, go to item (ii) of this scheme.

3.3. Equation for transport–diffusion processes

In the pollution model, the pollution process depends on the combination of substances and on their number. This combination is represented by the source vector $f = (f_1, \dots, f_m)$ in the right-hand side of (5), where f_i ($i = 1, \dots, m$) are determined from the conversion terms and can be written as follows:

$$f_i = \sum_{j=1}^m k_{i,j} C_j \tag{22}$$

where $k_{i,j}$ ($i, j = 1, \dots, m$) are the coefficients depending on temperatures and different substances. Thus the source vector can be rewritten as:

$$f = KC \tag{23}$$

Table 1
The substances and conversion terms.

Substances	Conversion term
DO	$C(\text{DO}) = k_2(D_{os} - \text{DO}) - k_3\text{BOD} - Y_1k_4\text{NH}_3 + P_o - R_9$
BOD	$C(\text{BOD}) = -k_3\text{BOD}$
PO ₄	$C(\text{PO}_4) = Y_2k_3\text{BOD} - Y_4(P_o - R_o) + k_6\text{PO}_4$
NH ₃	$C(\text{NH}_3) = Y_3k_3\text{BOD} - k_4\text{NH}_3 - Y_5(P_o - R_o) - Y_6k_3\text{BOD}$
NO ₃	$C(\text{NO}_3) = k_4\text{NH}_3 - k_5\text{NO}_3$

where:

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & k_{1,2} & \dots & k_{1,m} \\ k_{2,1} & k_{2,2} & \dots & k_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ k_{m,1} & k_{m,2} & \dots & k_{m,m} \end{bmatrix} \tag{24}$$

In the pollution model, the following substances and the conversion terms are given in Table 1 (see [14,15], and [16]), where:

- k_2 : dissolution coefficient of oxygen through free water surface,
- k_3 : disintegrated coefficient of BOD,
- k_4 : nitrating velocity,
- k_5 : reaction velocity of nitrating,
- k_6 : disintegration coefficient of PO₄,
- Y_1 : productivity of using oxygen in nitrating process,
- Y_2 : phosphorus content in the organic mixture,
- Y_3 : nitrogen in organic mixture,
- Y_4 : phosphorus coefficient absorbed by flora,
- Y_5 : ammoniac coefficient absorbed by flora,
- Y_6 : absorption coefficient of ammoniac by organism,
- P_o : velocity of oxygen produced through photosynthesis,
- R_o : oxygen respiration velocity by microorganisms and organism,
- D_{os} : saturated oxygen content.

By the same way as in calculating (z, u, v) in a small cell Ω_j ($j = 1, \dots, N$), integrating two sides of Eq. (5) with respect to the spatial variables x and y then using Green formula, we get:

$$\int_{\Omega_j} \frac{\partial C}{\partial t} dx dy - \int_{\Omega_j} C \nabla \cdot U dx dy + \oint_{\gamma_j} (UC\mathbf{n} - D\nabla \cdot C\mathbf{n}) d\gamma_j = \int_{\Omega_j} \mathbf{K}C dx dy \tag{25}$$

Using the discrete schema described in formulas (8)–(13), the integration on cell boundary γ_j in Eq. (25) is calculated with functions (u, v) obtained by formulas (20)–(21) in the current step and $C, \partial C/\partial x, \partial C/\partial y$ obtained from the previous step:

$$I_c = \oint_{\gamma_j} \left[(u \cdot n_x + v \cdot n_y)C - D \left(\frac{\partial C}{\partial x} n_x + \frac{\partial C}{\partial y} n_y \right) \right] d\gamma_j \tag{26}$$

As in Section 3.1, h can be approximated as constant in the small cell Ω_j , then from Eq. (1), the function $\nabla \cdot U$ can be calculated approximately by the following formula:

$$\nabla \cdot U = -\frac{1}{h} \frac{\partial z}{\partial t}$$

Then Eq. (25) can be rewritten as follows:

$$\frac{\partial C}{\partial t} S + C \frac{1}{h} \frac{\partial z}{\partial t} S + I_c = \mathbf{K}CS \tag{27}$$

The values of functions $(1/h)\partial z/\partial t$ and $\partial C/\partial t$ are approximated by the formulas:

$$\frac{1}{h} \frac{\partial z}{\partial t} = \frac{z - z_t}{h\Delta t}, \quad \frac{\partial C}{\partial t} = \frac{C - C_t}{\Delta t}$$

where $z, C = (C_1, \dots, C_m)$ and $z_t, C_t = (C_{1,t}, \dots, C_{m,t})$ are the water levels and concentrations at the centers of the cell s at the current and previous steps; $(z - z_t)/h$ is obtained when z is calculated by the formula (19).

Eq. (27) can be discretized as follows:

$$\frac{C - C_t}{\Delta t} S + C_t \frac{1}{h} \frac{z - z_t}{\Delta t} S + I_c = \mathbf{K} C_t S \quad (28)$$

Thus in small cell Ω_j , C can be calculated by the formula:

$$C = \mathbf{K} C_t \Delta t + C_t \left(1 - \frac{z - z_t}{h} \right) - \frac{\Delta t}{S} I_c \quad (29)$$

where I_c is obtained by Eq. (26). Then using the obtained C in the centers of N cells we can calculate C in N_p mesh points in the same way as in formula (15).

3.4. Discrete calculation scheme for (z, u, v) and C

The discrete scheme calculating (z, u, v) and C can be briefly written by the following steps:

- (i) the starting values of (z, u, v) and C are given in N_p mesh points;
- (ii) the values of (z, u, v) and C in N_p mesh points are called the values in the previous step;
- (iii) on cell boundaries γ_j ($j = 1, \dots, N$), the integrals involved in formulas (19)–(21) such as I_z, I_u, I_v (see formulas (16)–(18)) are calculated in the same way as in formulas (8)–(9) with the function values z, u, v obtained from the previous step;
- (iv) calculate the values of (z, u, v) and C at cell centers for the previous step, denoted by (z_t, u_t, v_t) and C_t , and their derivatives, denoted by $\partial u_t / \partial x, \partial u_t / \partial y, \partial v_t / \partial x, \partial v_t / \partial y, \partial C_t / \partial x, \partial C_t / \partial y$, in the same way as in formulas (13)–(14);
- (v) calculate (z, u, v) and $(z - z_t) / h$ at N cell centers for the current step by formulas (19)–(21);
- (vi) on cell boundary γ_j , the integral I_c appearing in formula (29) is calculated by formula (26) with the function values (u, v) obtained in the current step and $C, \partial C / \partial x, \partial C / \partial y$ obtained in the previous step;
- (vii) using the obtained values of I_c and $(z - z_t) / h$, calculate C at N cell centers for the current step by formula (29);
- (viii) using the values of (z, u, v) and C in the current step at N cell centers obtained from the above steps, calculate (z, u, v) and C at N_p mesh points in the same way as in formula (15);
- (ix) for the next time step, go back to item (ii) of this scheme.

3.5. The technique for getting a stable concentration solution C

For the stable solution of $V = (z, u, v)$, the algorithm is working well with a small time step (see [12]). To get a stable numerical solution C , the scheme will be developed as following:

- (i) in step $k - 1$, after the values of vector functions C^{k-1} in N_p mesh points are computed, the values of vector functions C in N cell centers, denoted by C_t , are obtained in the same way as in formula (14);
- (ii) the derivatives $C'_x = \partial C^{k-1} / \partial x, C'_y = \partial C^{k-1} / \partial y$ in N cell centers can be calculated in the same way as in formulas (8)–(9);
- (iii) by the above scheme, the values of C in N cell centers are obtained by the function $C^* = f_C(U^k, C_t, C'_x, C'_y)$;
- (iv) By the new C^* in N cell centers, their values in N_p points are calculated in the same way as in formula (15);
- (v) we go back to the item (ii) of this scheme;
- (vi) the iteration will be stopped when the values C^* obtained from the last two calculations are close enough. In our experiment, the number of iterations is about 5;
- (vii) after the last iteration, we put $C^k = C^*$ in N cell centers and in N_p mesh points for the k -th step of the algorithm.

3.6. Testing the algorithm

We will test the hydraulic algorithm by two test cases and the pollution algorithm by another two test cases. In these test cases, we consider the problems of water flows in a rectangular lake with length L and width W ($0 \leq x \leq L$ in the hydraulic test cases and $-L/2 \leq x \leq L/2$ in the pollution test cases, and $0 \leq y \leq W$). The velocities by the x direction u_{in} into the lake and the water depth out h_{out} are constant in time. Then the hydraulic state will tend to a stable one. One can compute the exact solution, which is used for comparison with the simulation. The boundary conditions are:

- on the inflow boundary, $x = 0, 0 \leq y \leq W: u(0, y, t) = u_{in}(t)$ (see Figs. 3, 4, left), $v(0, y, t) = v_{in} = 0$,
- on the outflow boundary, $x = L, 0 \leq y \leq W: h(L, y, t) = h_{out}(t)$ (see Figs. 3, 4, middle),
- on the solid wall $y = 0, y = W, 0 \leq x \leq L: v(x, y, t) = 0$.

The initial conditions for the hydraulic test cases are: $u_0 = v_0 = 0$ and $h(x, y, 0) = h_0(x, y)$ (see Figs. 3, 4, right).

With these boundary conditions for the hydraulic test cases, we can obtain the exact solutions using a 1D hydraulic model (see test case 2 in [17], and test case 1 of Section 6.3 in [18, p. 118]). The difference between our hydraulic test cases

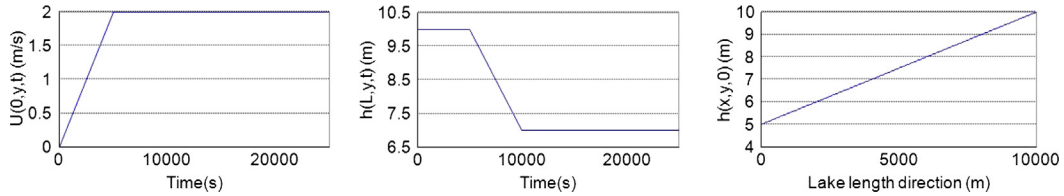


Fig. 3. Boundary conditions and initial conditions of the first hydraulic test case.

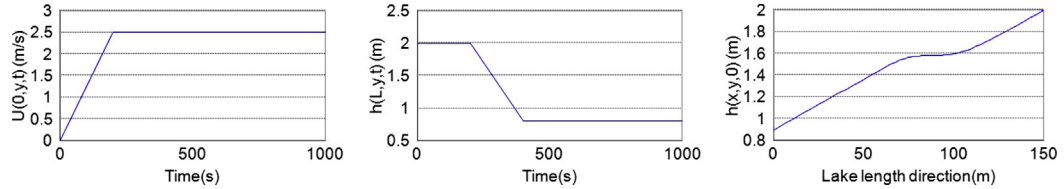


Fig. 4. Boundary conditions and initial conditions of the second hydraulic test case.

Table 2

Data of the hydraulic test cases.

Probl.	W (m)	L (m)	K	u_{in} (m/s)	v_{in} (m/s)	h_{out} (m)	$S_0(x)$	Mesh type	Time step (s)	$h_{in}(m)$ on the gate into lake
1	150	10000	30.6	2	0	7	0.0005	Triangular	3	5.139558
2	10	150	33.3	2.5	0	0.800054	formula (30)	Triangular	0.3	0.800289

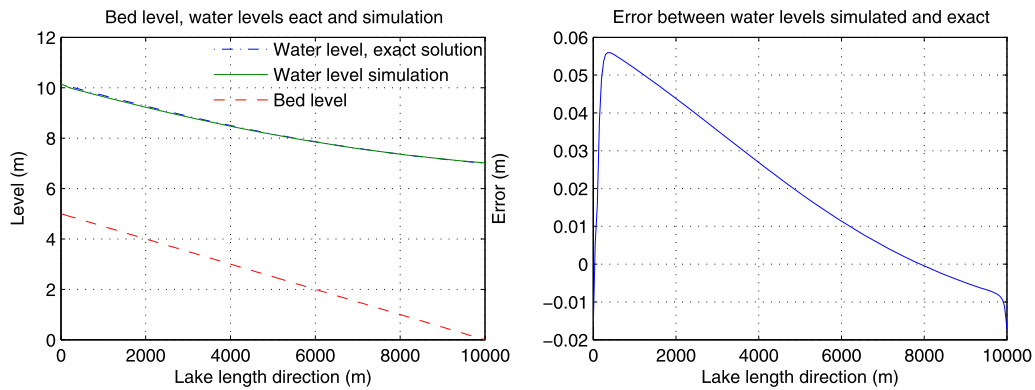


Fig. 5. (Color online.) The first hydraulic test case. Left: bed level and water levels from simulation and exact solution. Right: simulation error.

and the test cases in [17,18] is that we change the discharge conditions for the velocity conditions into the lakes at the gate, using formula $u_{in} = Q_{in}/(W \cdot h_{in})$. Here, h_{in} is obtained by the exact solution of the considered test cases in [17,18]. The data of the hydraulic test cases are presented in Table 2, in which:

- K is the Strickler friction coefficient,
- $S_0(x)$ is the bed slope of lake by x direction. For the second test case (see Section 6.3 in [18]), it is given by the following formula:

$$S_0(x) = \left(1 - \frac{(u.a)^2 W}{9.08665 \hat{h}^3(x) W^3}\right) \hat{h}'(x) + \frac{(u.a)^2 (W + 2\hat{h}(x))^{4/3}}{\hat{h}^{10/3}(x) W^{10/3} K^2} \tag{30}$$

where $a = W \cdot h$ is the lake's section area crossing x direction and the function \hat{h} is: $\hat{h}(x) = 0.8 + 0.25 \exp(-33.75(x/150 - 1/2)^2)$.

The bed levels comparisons and the errors between simulations and exact solutions of water levels for the these two test cases are presented in Figs. 5–7. From these figures, the errors between exact solutions and simulations of the first and the second hydraulic test cases are less than 4 cm and 5.5 cm, respectively. The maximum error in those test cases are actually less than 1% and 5.1% percent of the true values.

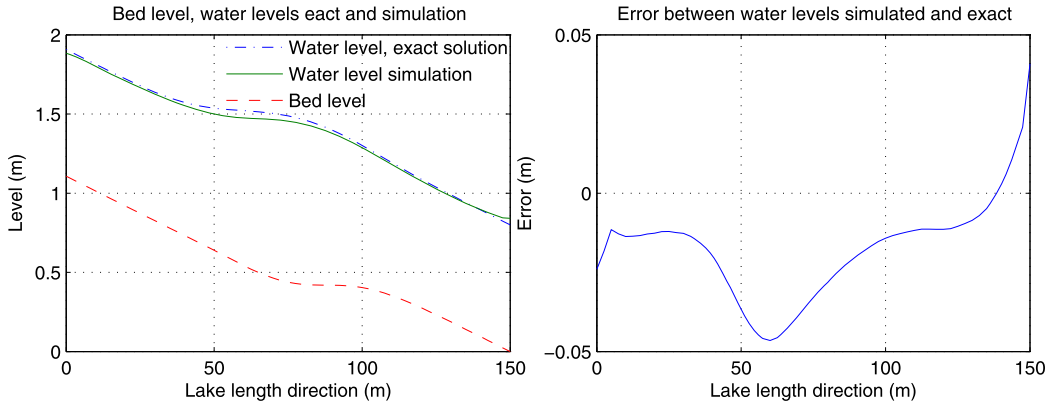


Fig. 6. (Color online.) The second hydraulic test case. Left: bed level and water levels from simulation and exact solution. Right: simulation error.

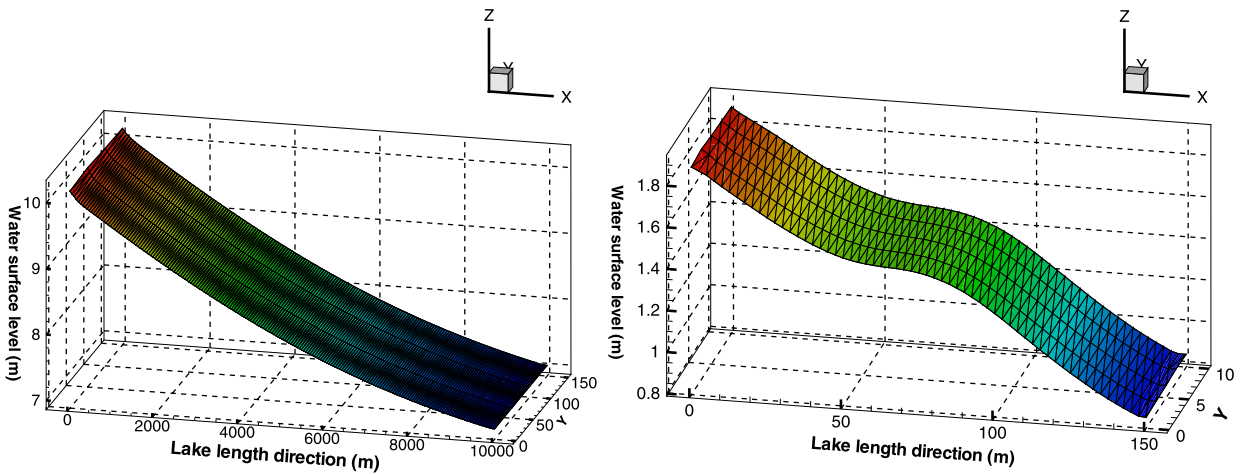


Fig. 7. (Color online.) Lake’s water surface level in the first (left) and second (right) hydraulic test cases. (The unlabelled axis correspond to the lake’s width direction.)

Table 3
Data of pollution test cases.

Prob.	W (m)	L (m)	K	u_{in} (m/s)	v_{in} (m/s)	$h_{out}(x)$	$S_0(x)$	Mesh type	Time step (s)	D	C_{in}	$\partial C_{out}/\partial n$
1	200	10 000	30.6	0.3	0	0.8	0.00025	Triangular	1	30	2	0
2	200	10 000	30.6	0	0	7	0.00025	Triangular	1	30	2	0

To test the numerical algorithm for the 2D pollution problem, we consider the pollution equation (5) for only one substance ($m = 1$). The source function f in the right-hand side of this equation is then obtained by the formula:

$$f = R_a - R_s \tag{31}$$

where R_a is the generative velocity of the substance and R_s is the losing velocity of the substance.

The data for the two pollution test cases are given in Table 3. In these pollution test cases, we suppose that there is a dissolvable substance distributed in this lake only along the x direction (the concentration is not changing along the y direction) with the initial condition having discontinuity at $x = 0$ (see [19]), defined by $C(x, 0) = C_1$ when $x < 0$ and $C(x, 0) = C_2$ when $x > 0$ and the boundary condition at infinity $C(x, t) = C_1$ when $x \rightarrow \infty$ and $C_2(x, 0) = C_2$ when $x \rightarrow -\infty$. In our test cases, we let $C_1 = 2$, $C_2 = 0$. The substance’s boundary conditions on in- and out-gates of lake are: $C_{in} = 2$, $\partial C_{out}/\partial n = 0$. In the calculation process, we suppose that the substance is not generative and not lost. Then we have $R_a = 0$ and $R_s = 0$. Therefore, Eq. (5) becomes:

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} = D \frac{\partial^2 C}{\partial x^2} \tag{32}$$

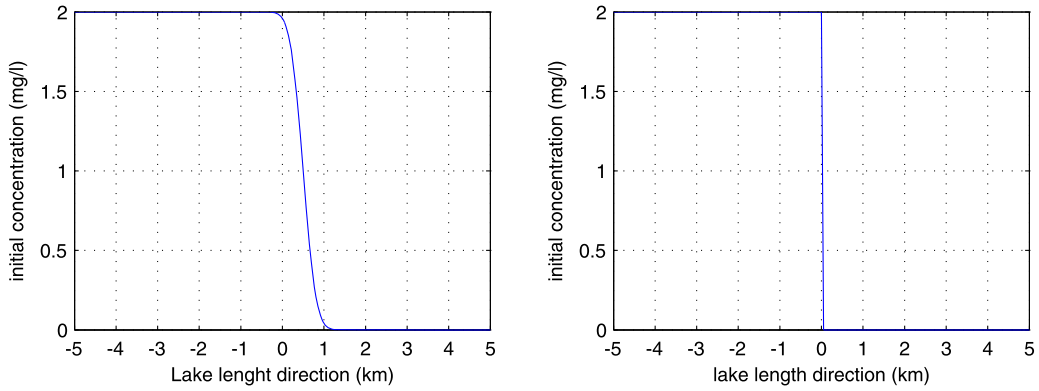


Fig. 8. The initial concentration conditions of the first (left) and second (right) pollution test cases.

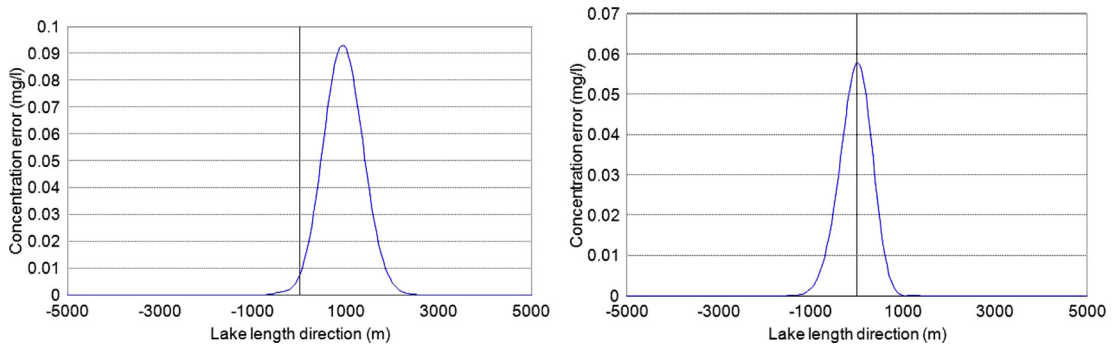


Fig. 9. The concentration errors between the exact solution and the simulation result (at time 2000 s) for the first (left) and second (right) test cases.

– The first test case for the pollution algorithm

The exact solution of Eq. (32) can be obtained for all times t , expressed in terms of the complementary error function $\text{erfc}(x)$ (see [19]):

$$C(x, t) = A_1 - A_2 \cdot \text{erfc}\left(\frac{x - ut}{\sqrt{Dt}}\right) \tag{33}$$

where $A_1 = (C_2 + C_1)/2$, $A_2 = (C_2 - C_1)/2$.

The large downward jump (at $x = 0$) of the initial concentration condition in this test case will cause difficulty in the numerical algorithm solving the problem. To overcome this difficulty, the initial concentration condition needs to be smoothed out. Therefore, we take as initial condition for the computational algorithm the exact concentration solution at time $t = 1000$ s (see Fig. 8).

– The second test case for the pollution algorithm

In this test case, u and v are equal to zero. Hence, Eq. (32) becomes:

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} \tag{34}$$

The exact solution of this equation is:

$$C(x, t) = A_1 - A_2 \cdot \text{erfc}\left(\frac{x}{\sqrt{Dt}}\right) \tag{35}$$

The large downward jump (at $x = 0$) of the initial concentration condition in this test case does not cause difficulty in the numerical algorithm solving the problem. Therefore, the initial concentration condition for the computation algorithm is left the same as in the theoretical problem (see Fig. 8).

The results of the above test cases are presented in Figs. 9–11. In these figures, the concentration results at time 2000 s after the start of the simulation are presented. It is easy to see in Fig. 11 that the errors are quite small: the mean absolute error is less than 0.04% and 0.54% of the mean concentration level, for the first and second test cases, respectively.

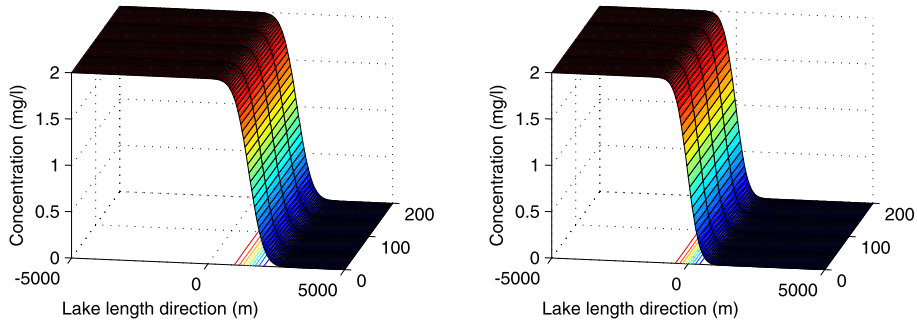


Fig. 10. (Color online.) The simulation results of concentration (at time 2000 s) for the first (left) and second (right) test cases. (The unlabelled axis corresponds to the lake width direction.)

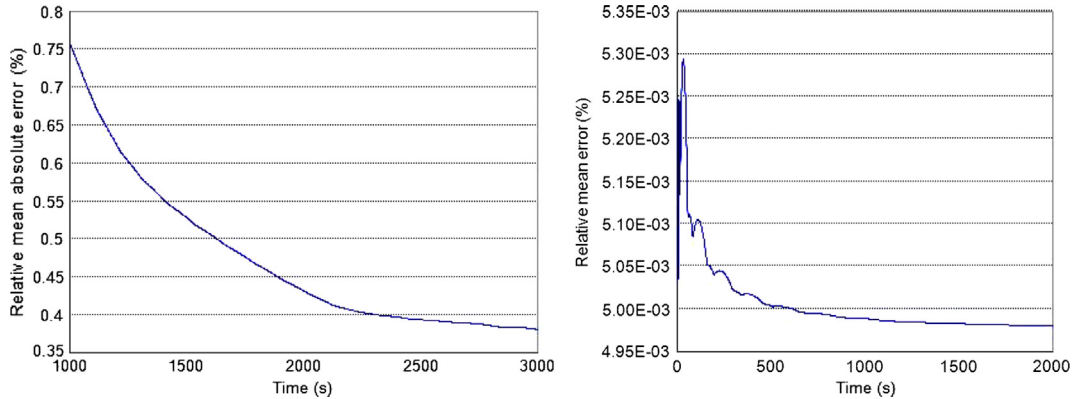


Fig. 11. Mean absolute errors of concentration in percentage of mean concentration for the first (left) and second (right) pollution test cases.

4. The SEIK filter

4.1. Model and notations

We consider a physical system described by:

$$X_k = \mathcal{F}_{k-1}(X_{k-1}) + V_k \quad (36)$$

where X_k and V_k are the state vector and the error vector at the time point t_k ; $\mathcal{F}_{k-1}(\cdot)$ is generally a nonlinear function modeling the transition of the system from time t_{k-1} to time t_k . The function $\mathcal{F}_{k-1}(\cdot)$ can involve other external (dynamical) variables which are known (or observable). Such variables will not be indicated. In the present application, X_k will be a very large vector composed of the values of pollutant concentrations at every mesh points. Other hydrological variables such as water velocity, temperature... are considered external, and assumed known for simplicity. The state vector X_k is only partially observable through the measurement equation:

$$Y_k = \mathcal{H}_k(X_k) + W_k \quad (37)$$

where \mathcal{H}_k is some possibly nonlinear function and W_k is the measurement error vector. In the present application, the observation is made only at certain points of the area. Thus, Y_k is composed of the pollutant concentrations measures at certain mesh points only. In fact, we allow that observation is made also only at certain time points, so Eq. (37) does not exist for all k .

It is assumed that the vectors V_k , W_k , $k = 1, 2, \dots$, are independent random vectors of zero mean and covariance matrix \mathbf{Q}_k and \mathbf{R}_k , respectively.

The aim is to estimate the state X_k based on the measurements Y_1, \dots, Y_k up to time t_k .

The SEIK filter works in a sequence of two stages: forecast and correction, preceded by an initialization stage.

- *Initialization stage.* An initial estimate \hat{X}_0 of the state vector and its error covariance matrix \mathbf{P}_0 must be provided. This can be obtained from historic data about the steady state of the system or more practically from a preliminary simulation study. A long sequence state vectors $\tilde{X}_1, \dots, \tilde{X}_N$ is generated from the model equation (by solving the direct problem). This provides an estimated distribution of the steady state of the state vectors, more precisely estimates of

its means, by $\hat{X}_0 = (1/N) \sum_{j=1}^N \tilde{X}_j$, and of its covariance matrix, by $\tilde{\mathbf{P}}_0 = (1/N) \sum_{j=1}^N (\tilde{X}_j - \hat{X}_0)(\tilde{X}_j - \hat{X}_0)^T$. (Here and in the sequel T denotes the transpose.) We take \hat{X}_0 as the estimate of the initial state. The initial error covariance matrix could be $\tilde{\mathbf{P}}_0$. However, the SEIK filter requires an initial error covariance matrix \mathbf{P}_0 of low rank r ; therefore, we shall take it as a rank- r approximation to $\tilde{\mathbf{P}}_0$. In practice, r can be quite low (relative to the state vector dimension). Let $\tilde{\mathbf{P}}_0 = \sum_{j=1}^{N-1} \lambda_j Z_j Z_j^T$ be an eigen-decomposition of $\tilde{\mathbf{P}}_0$ where $\lambda_1 \geq \dots \geq \lambda_{N-1}$ are its eigenvalues, arranged in decreasing order, and Z_1, \dots, Z_{N-1} are the corresponding (normalized) eigenvectors. The sequence of eigenvalues usually decreases to zero very fast so that one may retain only the first r terms in the above sum, yielding a rank- r approximation to $\tilde{\mathbf{P}}_0$: $\mathbf{P}_0 = \sum_{j=1}^r \lambda_j Z_j Z_j^T$. The quality of the approximation can be measured by the ratio $(\lambda_1 + \dots + \lambda_r)/(\lambda_1 + \dots + \lambda_{N-1})$. A ratio 0.9 or greater is often judged sufficient. Note that \mathbf{P}_0 can be expressed in the factorized form $\mathbf{L}_0 \mathbf{U}_0 \mathbf{L}_0^T$ where $\mathbf{L}_0 = [Z_1 \dots Z_r]$ and \mathbf{U}_0 is the diagonal matrix with diagonal elements $\lambda_1, \dots, \lambda_r$. Such a form is needed in subsequent stages.

– *Forecast (or prediction) stage.* This stage aims at forecasting the state vector and providing the forecast error, based on the model equation and the knowledge of the previous system state. Thus, at time t_{k-1} , let us assume that an estimate of the state vector \hat{X}_{k-1} and its error covariance matrix \mathbf{P}_{k-1} are available.

It is of interest to express \hat{X}_{k-1} and \mathbf{P}_{k-1} as a sample mean and covariance matrix. We assume that, as mentioned above, \mathbf{P}_{k-1} is given in the factorized form: $\mathbf{L}_{k-1} \mathbf{U}_{k-1} \mathbf{L}_{k-1}^T$. We first write it as $(\mathbf{L}_{k-1} \mathbf{U}_{k-1}^{1/2} \mathbf{\Omega}_{k-1})(\mathbf{L}_{k-1} \mathbf{U}_{k-1}^{1/2} \mathbf{\Omega}_{k-1})^T$, where $\mathbf{U}_{k-1}^{1/2}$ is a square root of \mathbf{U}_{k-1} defined by $\mathbf{U}_{k-1}^{1/2} \mathbf{U}_{k-1}^{1/2 T} = \mathbf{U}_{k-1}$ and $\mathbf{\Omega}_{k-1}$ is an $r \times (r+1)$ matrix (randomly chosen) such that its row vectors are orthonormal, with components summing to zero. (An algorithm for constructing such a matrix is available, see [6].) Then define $X_{k-1}^i - \hat{X}_{k-1}$ to be the i -th column of $\sqrt{r} \mathbf{L}_{k-1} \mathbf{U}_{k-1}^{-1/2} \mathbf{\Omega}_{k-1}$, one has:

$$\mathbf{P}_{k-1} = \frac{1}{r} \sum_{i=1}^{r+1} (X_{k-1}^i - \hat{X}_{k-1})(X_{k-1}^i - \hat{X}_{k-1})^T$$

Explicitly, $X_{k-1}^i = \hat{X}_{k-1} + \sqrt{r} \mathbf{L}_{k-1} \mathbf{U}_{k-1}^{-1/2} \mathbf{\Omega}_{k-1}^{(i)}$, where $\mathbf{\Omega}_{k-1}^{(i)}$ denotes the i -th column of $\mathbf{\Omega}_{k-1}$. Since $\sum_{i=1}^{r+1} \mathbf{\Omega}_{k-1}^{(i)} = \mathbf{0}$, \hat{X}_{k-1} is indeed the sample mean of $X_{k-1}^1, \dots, X_{k-1}^{r+1}$. Of course, their sample covariance matrix is \mathbf{P}_{k-1} .

The set $\{X_{k-1}^i, i = 1, \dots, r+1\}$ may be viewed as a representation of the distribution of the state vector at time t_{k-1} , as far as second-order statistics are concerned. Thus it is natural to forecast X_k by the sample means of $X_{k-}^i = \mathcal{F}_{k-1}(X_{k-1}^i)$:

$$\hat{X}_{k-} = \frac{1}{r+1} \sum_{i=1}^{r+1} X_{k-}^i \tag{38}$$

The forecast error covariance matrix would be:

$$\mathbf{P}_{k-} = \frac{1}{r} \sum_{i=1}^{r+1} (X_{k-}^i - \hat{X}_{k-})(X_{k-}^i - \hat{X}_{k-})^T + \bar{\mathbf{Q}}_k \tag{39}$$

$\bar{\mathbf{Q}}_k = \bar{\mathbf{Q}}_{k-1} + \mathbf{Q}_k$ denotes the cumulative dynamical (model) error covariance matrix, at time t_k .

– *Correction stage.* This stage takes into account the information provided by the observation to obtain a new more accurate estimate of the state vector and update the error covariance matrix accordingly.

If no observation has been made at time t_k , there will be no correction, one just put $X_k^i = X_{k-}^i$ and continue at the next forecasting stage. Otherwise, to perform the correction without excessive computational cost, the SEIK filter assumes that the forecast error covariance matrix (39) is of low rank r and given in a factorized form $\mathbf{P}_{k-} = \mathbf{L}_k \mathbf{U}_k \mathbf{L}_k^T$. Note that the first term in (39) can already be written in this form. Indeed, let \mathbf{T} be an $(r+1) \times r$ matrix with columns forming a basis of the linear subspace of all $(r+1)$ -vectors with components summing to zero,¹ then:

$$[X_{k-}^1 - \hat{X}_{k-} \quad \dots \quad X_{k-}^{r+1} - \hat{X}_{k-}] \mathbf{T} (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T = [X_{k-}^1 - \hat{X}_{k-} \quad \dots \quad X_{k-}^{r+1} - \hat{X}_{k-}]$$

since any row vector of the matrix in the above right-hand side can be written as $(\mathbf{T}\mathbf{v})^T$ and $(\mathbf{T}\mathbf{v})^T [\mathbf{T} (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T] = (\mathbf{T}\mathbf{v})^T$. From this result and the fact that $[1 \dots 1] \mathbf{T} = \mathbf{0}$, it can be seen that the first term in (39) also equals:

¹ An example of \mathbf{T} is:

$$\mathbf{T} = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ 0 & & 1 & \\ 0 & \dots & 0 & \end{bmatrix} - \frac{1}{r+1} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{bmatrix} \tag{40}$$

The product $\mathbf{T}^T \mathbf{T}$ (which will be needed later) has $r/(r+1)$ on the diagonal and $-1/(r+1)$ outside this diagonal.

Table 4

Data of the SEIK filter test case.

Prob.	W (m)	L (m)	K	u_{in} (m/s)	v_{in} (m/s)	h_{out} (m)	$S_0(x)$	Mesh type	Time step (s)	h_{in} (m)	D	$k_{1,1}$	C_{in} (mg/l)	∂C_{out} / ∂n
1	200	10 000	30.6	2	0	7	0.0005	Triangular	1	5.139558	$1.7E^{-6}$	$-4.05E^{-6}$	24	0
2	800	3000	30.6	0.35	0	7	0.0	Triangular	1	7	$1.7E^{-6}$	$-4.05E^{-6}$	24	0

$$[X_{k-}^1 \quad \dots \quad X_{k-}^{r+1}] \mathbf{T} (\mathbf{r} \mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T [X_{k-}^1 \quad \dots \quad X_{k-}^{r+1}]^T = \mathbf{L}_k [(r+1) \mathbf{T}^T \mathbf{T}]^{-1} \mathbf{L}_k^T$$

with

$$\mathbf{L}_k = [X_{k-}^1 \quad \dots \quad X_{k-}^{r+1}] \mathbf{T} \quad (41)$$

To obtain a factorized form for \mathbf{P}_{k-} , we will approximate $\bar{\mathbf{Q}}_k$ by $\mathbf{L}_k (\mathbf{L}_k^T \mathbf{L}_k)^{-1} (\mathbf{L}_k^T \bar{\mathbf{Q}}_k \mathbf{L}_k) (\mathbf{L}_k^T \mathbf{L}_k)^{-1} \mathbf{L}_k^T$. (No approximation is needed if $\bar{\mathbf{Q}}_k = \mathbf{0}$.) Then $\mathbf{P}_{k-} = \mathbf{L}_k \mathbf{U}_{k-} \mathbf{L}_k^T$ where:

$$\mathbf{U}_{k-} = (\mathbf{r} \mathbf{T}^T \mathbf{T})^{-1} + (\mathbf{L}_k^T \mathbf{L}_k)^{-1} (\mathbf{L}_k^T \bar{\mathbf{Q}}_k \mathbf{L}_k) (\mathbf{L}_k^T \mathbf{L}_k)^{-1} \quad (42)$$

For simplicity, consider the case that \mathcal{H}_k is linear: $\mathcal{H}_k(X_k) = \mathbf{H}_k X_k$. The formula for the correction stage is then the same as for the Kalman filter:

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_{k-} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k-} \mathbf{H}_k^T \mathbf{R}_k)^{-1} \\ \hat{X}_k &= \hat{X}_{k-} + \mathbf{G}_k (Y_k - \mathbf{H}_k \hat{X}_{k-}) \end{aligned} \quad (43)$$

The error covariance matrix (after correction) is:

$$\mathbf{P}_k = \mathbf{P}_{k-} - \mathbf{P}_{k-} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k-} \mathbf{H}_k^T \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_{k-}$$

From the factorization of \mathbf{P}_{k-} , the matrix \mathbf{P}_k also admits a factorized form $\mathbf{L}_k \mathbf{U}_k \mathbf{L}_k^T$ where:

$$\mathbf{U}_k = \mathbf{U}_{k-} - \mathbf{U}_{k-} \mathbf{L}_k^T \mathbf{H}_k^T (\mathbf{H}_k \mathbf{L}_k \mathbf{U}_{k-} \mathbf{L}_k^T \mathbf{H}_k^T \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_{k-}$$

By a matrix identity, the above right-hand side is invertible if \mathbf{U}_{k-} is and admits as its inverse:

$$\mathbf{U}_k^{-1} = \mathbf{U}_{k-}^{-1} + \mathbf{L}_k^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{L}_k \quad (44)$$

The matrix \mathbf{G}_k can also be computed alternatively as:

$$\mathbf{G}_k = \mathbf{P}_k \mathbf{H}_k^T \mathbf{R}_k^{-1} = \mathbf{L}_k \mathbf{U}_k \mathbf{L}_k^T \mathbf{H}_k^T \mathbf{R}_k^{-1} \quad (45)$$

More details of calculations can be found in [4,5].

After correction, the matrix $\bar{\mathbf{Q}}_k$ is reset to $\mathbf{0}$.

In practice, the matrix \mathbf{U}_k is not computed, only its inverse is, via (44). The (big) matrix \mathbf{P}_k is not computed either, since one only needs its factorized form for the next forecasting stage. To compute the matrix $\mathbf{U}_k^{1/2}$ needed at this stage, one can make a Cholesky decomposition $\mathbf{U}_k^{-1} = \mathbf{C}_k \mathbf{C}_k^T$, where \mathbf{C}_k is lower triangular. Then $\mathbf{U}_k^{1/2} = (\mathbf{C}_k^T)^{-1}$.

4.2. Test cases of the SEIK filter algorithm

To test the SEIK algorithm, we consider two problems with one pollutant indicator BOD₅ (see Section 6.2) and with data from Table 4. In the first test case, the gate into the lake is at $x = 0$ and the gate out of the lake is at $x = 10000$. The initial conditions for u and v in this test case are the same as in the case of the first hydraulic test. In the second test case, the gate into the lake is at $x = 0$, $y \in [0, 200]$ and the gate out of the lake is at $x = 3000$, $y \in [600, 800]$. The initial conditions are: $u = 0$, $v = 0$. In two test cases, the concentration boundary condition at the gate into the lake $C_{in} = 24$ mg/l, and that at the gate out of the lake is $\partial C_{out} / \partial n = 0$. The initial concentration is $C(x, y, 0) = 24$ mg/l.

In these two test cases, the lakes with unstructured nets and the concentration fields with measurement positions P1–P6, test position P7, are shown in Figs. 12 and 15.

We will test the SEIK filter by considering a “twin experiment”. A run of the model (with arbitrary initial values) simulating the true pollutant concentration levels, is used as a reference. The initial value for the SEIK filter is taken as the average of another long model run (2000 s), with a different initial value. (Note that the initial value for this run does not matter much, since we have taken the average over a very long run.) The initial filter error covariance matrix is also constructed from this run, as described in Section 4. The reference run is used to extract the “pseudo” observations, at certain points of the lake. In this experiment, these observation points are six points P1–P6, indicated in Fig. 12, right panel. Measurements

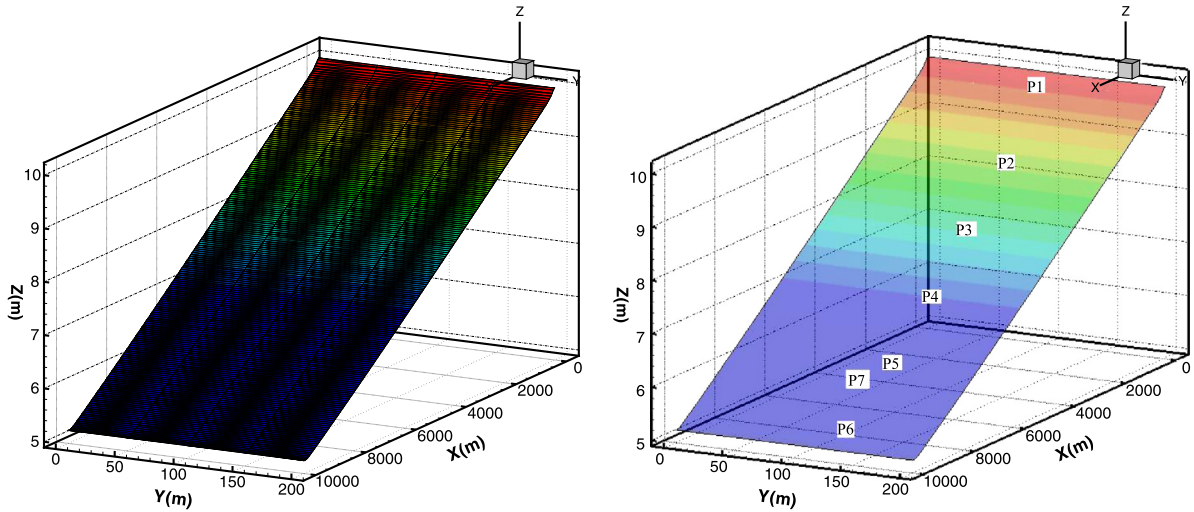


Fig. 12. (Color online.) The first Kalman filter test case: unstructured net with triangular cells (left); concentration field with measurement positions P1–P6 and test position P7 (right).

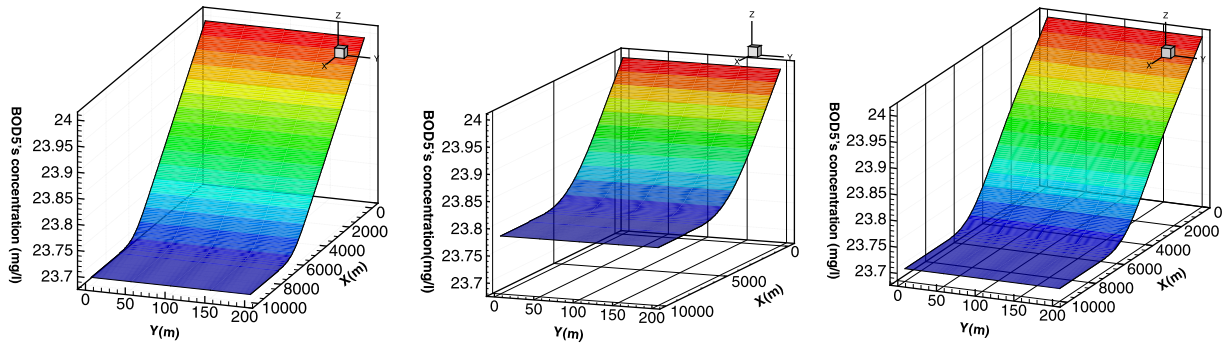


Fig. 13. (Color online.) The first Kalman filter test case: BOD₅ concentration fields of the reference (left), of the model without Kalman correction (middle), and of the model with Kalman correction (right).

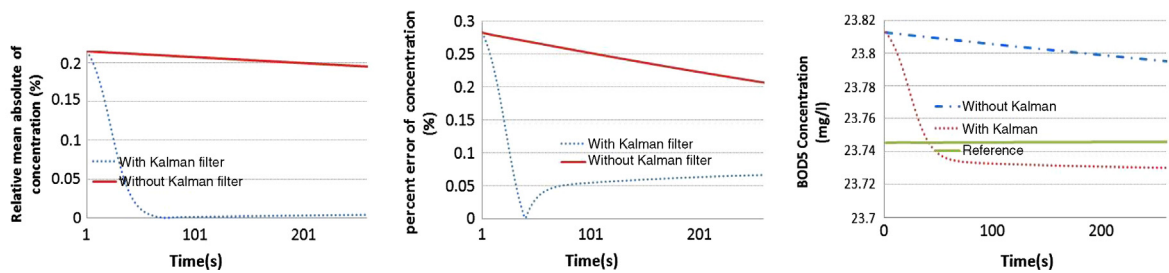


Fig. 14. (Color online.) The first Kalman filter test case: mean absolute errors of BOD₅ concentration in percentage of its mean value on the lake (left); error of BOD₅ concentration in percentage of its value at the P7 point (middle); BOD₅ concentration at the P7 point (right).

are at 1 s, 4 s, 9 s ... (one per 3 s) and the algorithm is run for 260 s. The performance of the SEIK filter can be assessed by comparing the pollutant concentration levels that it provides and that of the reference, at other non-measurement points.

The comparison results of two test cases at the time 260 s and in the period from the starting time up to time 260 s are respectively shown in Figs. 13, 14 and 16, 17. It is easy to see that the model with Kalman filter is closer to the reference than the one without Kalman filter.

5. Simulation experiment for the Thanh Nhan Lake

The Thanh Nhan Lake is located in Hanoi. Its area and water capacity are about 8.1 ha and 162 000 m³, respectively.

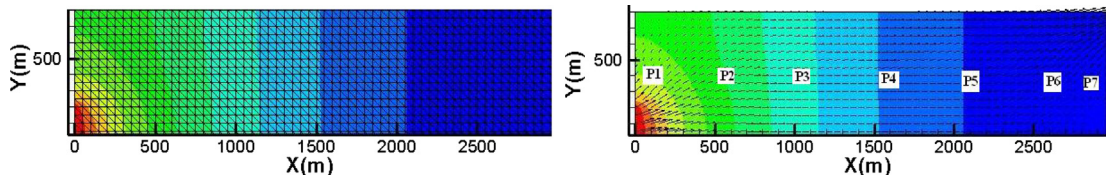


Fig. 15. (Color online.) The second Kalman filter test case: unstructured net with triangular cells (left); concentration field with measurement positions P1–P6 and test position P7 (right).

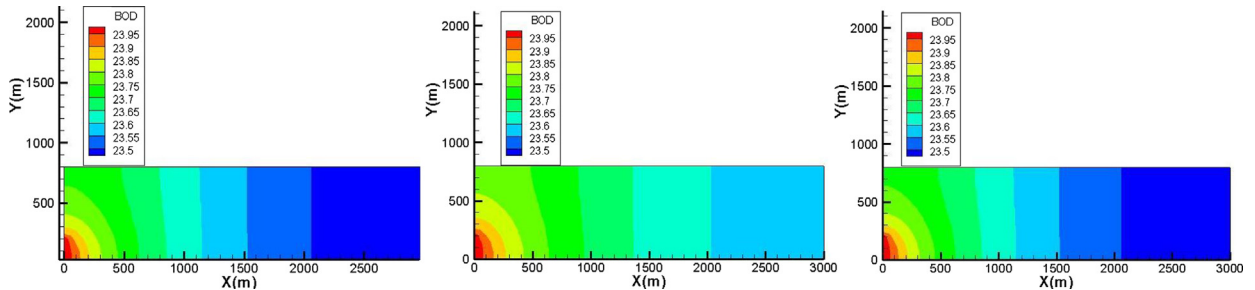


Fig. 16. (Color online.) The second Kalman filter test case: BOD₅ concentration fields of the reference (left), of the model without Kalman correction (middle) and of the model with Kalman correction (right).

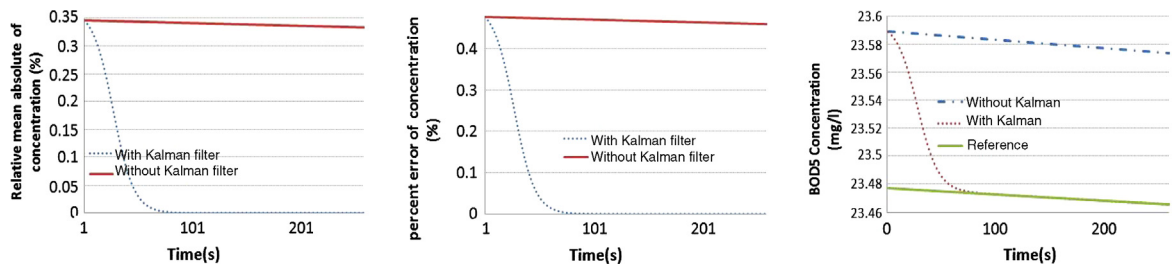


Fig. 17. (Color online.) The second Kalman filter test case: mean absolute errors of BOD₅ concentration in percentage of its mean value on the lake (left); BOD₅ concentration error in percentage of its value at the P7 point (middle); BOD₅ concentration at the P7 point (right).

5.1. Physical hydro properties

- The temperature of the lake oscillates from 23.4 °C to 23.8 °C in the dry season.
- The muddy level oscillates from 9 mg/l to 34 mg/l.
- The concentration of water-dissolved oxygen oscillates from 7.9 mg/l to 11.6 mg/l.

5.2. Chemical hydro properties

- Organic element group:
 - The content of NH₃ oscillates from 0.51 mg to 0.62 mg/l.
 - The need for chemical biology oxygen BOD₅ oscillates from 15.5 mg/l to 24.0 mg/l.
 - The need of chemical oxygen COD oscillates from 345 mg/l to 400 mg/l.
- The group of chemical toxin: it consists of some elements that have heavy metals such as mercury, lead, and manganese.

In this paper, we only study BOD₅'s, COD's and NH₃'s indicators. The BOD₅'s indicator shows the oxygen quantity needed for bacterium in reactions with oxygen of organic substances in water. The COD's indicator shows the chemical oxygen's quantity needed for reactions of organic substances with oxygen leading to disintegration processes into CO₂ and H₂O. The indicators BOD₅, COD, NH₃ have exchange relations. The chemical processes leading to the disintegration of the pollution indicators and substances are described as follows:

$$S_i(\text{BOD}_5) = k_{11}\text{BOD}_5$$

$$S_i(\text{NH}_3) = k_{21}\text{BOD}_5 + k_{22}\text{NH}_3$$

$$S_i(\text{COD}) = k_{33}\text{COD}$$

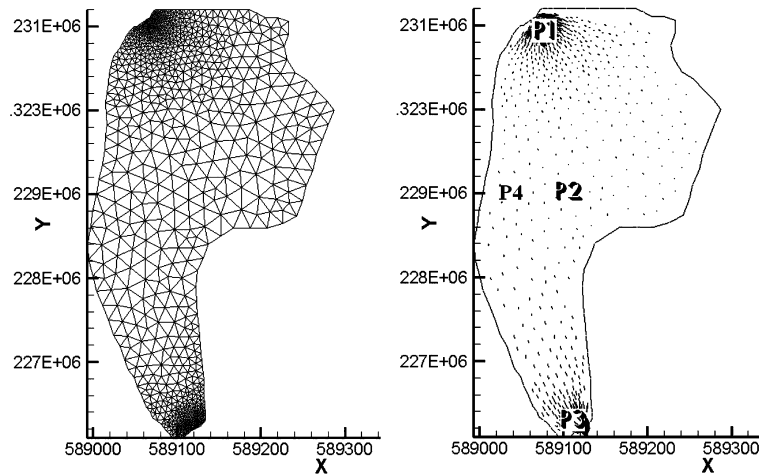


Fig. 18. Unstructured net with triangular cells (left) and velocity field with measurement positions P1, P2, P3 and test position P4 shown (right).

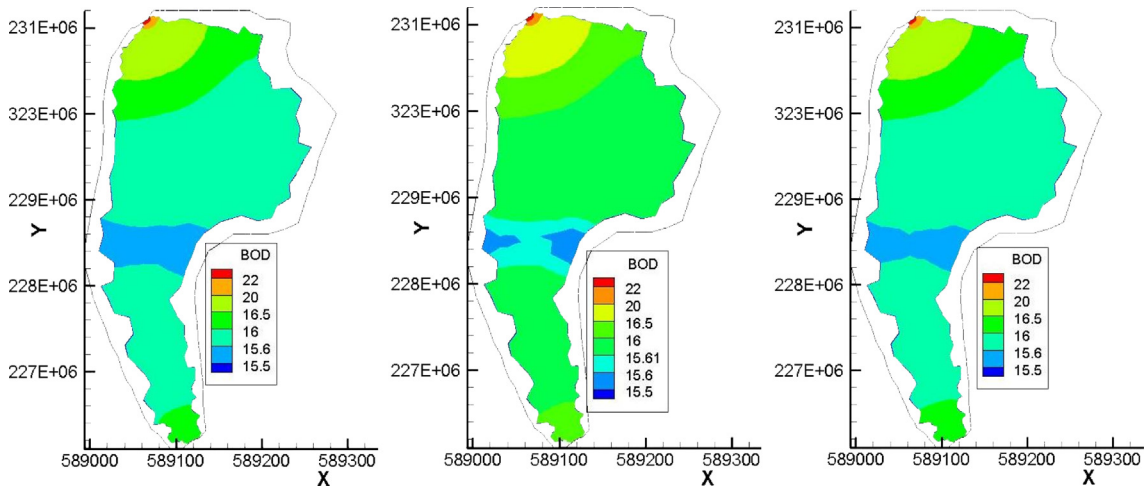


Fig. 19. (Color online.) BOD₅ concentration fields of the reference (left), of the model without Kalman correction (middle), and of the model with Kalman correction (right).

On the collected data, we establish input data for the model. The geographical data are divided into the boundary and the inside area of the lake. These data are divided by the unstructured net with 1964 triangular cells and 1058 nodes. This unstructured net is described in Fig. 18, left panel.

The data of discharges and substance contents at the gate into the lake are as follows.

- The discharges in and out of the lake are 2100 m³ per day–night.
- The content of BOD₅ at the gate into the lake is the same as the measurement value 24 mg/l.
- The content of COD at the gate into the lake is the same as the measurement value 400 mg/l.
- The content of NH₃ at the gate into the lake is the same as the measurement value 0.52 mg/l.

On the gate out there are the conditions of concentration: $\partial C_i / \partial n = 0$ ($i = 1 \dots m$).

5.3. Simulation setup and results

As in the SEIK filter test case (see Section 4.2), we consider a “twin experiment”. A run of the model (with arbitrary initial values) simulating the true pollutant concentration levels is used as a reference. The initial value for the SEIK filter is taken as the average of another long model run (7180 h), with a different initial value. The reference run is used to extract the “pseudo” observations, at certain points of the lake. In this experiment, the observation points are three points P1, P2 and P3, indicated in Fig. 18, right panel. Measurements are also only made at 1.2 h, 2.6 h and 4 h, and the algorithm is run for 6.8 h. The performance of the SEIK filter can be assessed by comparing the pollutant concentration levels that it provides and that of the reference, at other non-measurement points.

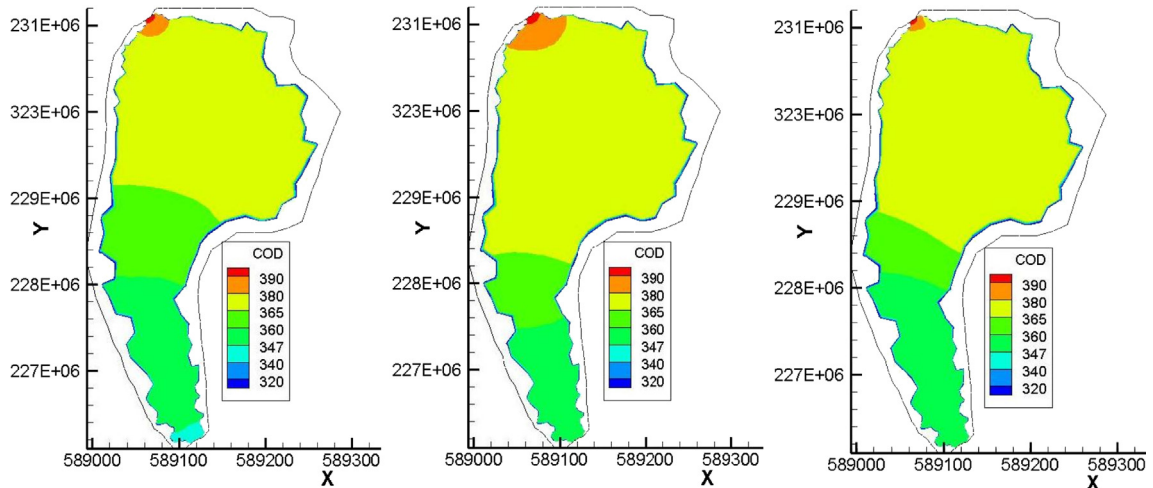


Fig. 20. (Color online.) COD concentration fields of the reference (left), of the model without Kalman correction (middle), and of the model with Kalman correction (right).

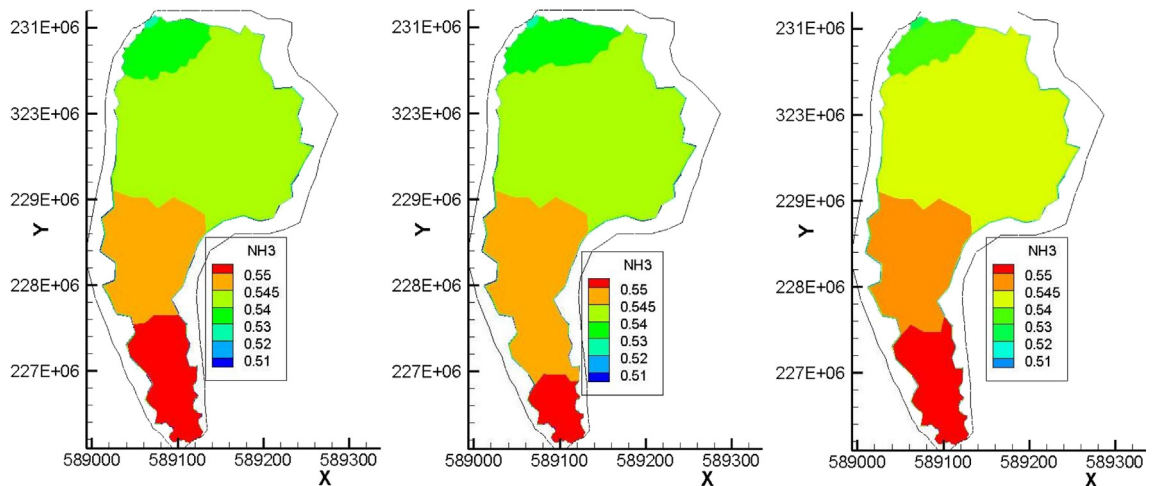


Fig. 21. (Color online.) NH_3 concentration fields of the reference (left), of the model without Kalman correction (middle), and of the model with Kalman correction (right).

In Figs. 19–21, the vertical and horizontal axes provide the coordinates along x and y directions (with the unit: m), and the concentration indicators (with the unit: mg/l) are described in the small color boxes, which are the same for the three panels (left, middle, and right), for ease of comparison.

It is also of interest to compare the performance of the SEIK filter with “doing nothing”. To this end, we run the same model with the same initial values as in the SEIK algorithm, *but without making any correction*. The pollutant concentrations obtained from the “model only” can then be compared with that of the reference and/or that provided by the SEIK algorithm.

Figs. 19–21, right panel, show the results of the application of our algorithm (after a 6.8-h run) concerning the concentrations of BOD_5 , COD, and NH_3 respectively. For comparison, the reference fields of these pollutant concentrations (which serve to provide the measurements) and the results of the use of the model alone (without Kalman correction) are shown in the left and middle panels, respectively.

The right panels in Figs. 22–24 plot the mean absolute errors of BOD_5 , COD, and NH_3 concentrations, in percentage of the mean concentrations, obtained by our algorithm and by the “model only” approach, respectively. The left panels of these figures plot the values of these pollutant concentrations given by the reference, the “model only” approach and the filter algorithm, at some point in the middle of the lake (point P4 in Fig. 18).

The results of the experiment show the good performance of the algorithm. The values of the pollutant concentrations are very close to the reference in the middle area of the lake, as can be seen in Figs. 19–21 and 22–24. The results are less good toward the south of the lake, but on a point P4 in the middle of the lake, the errors are very small, as can be seen from Figs. 22–24, left panel. Note that these figures correspond to the last time point 6.8 h, and the last measurement is at 4 h.

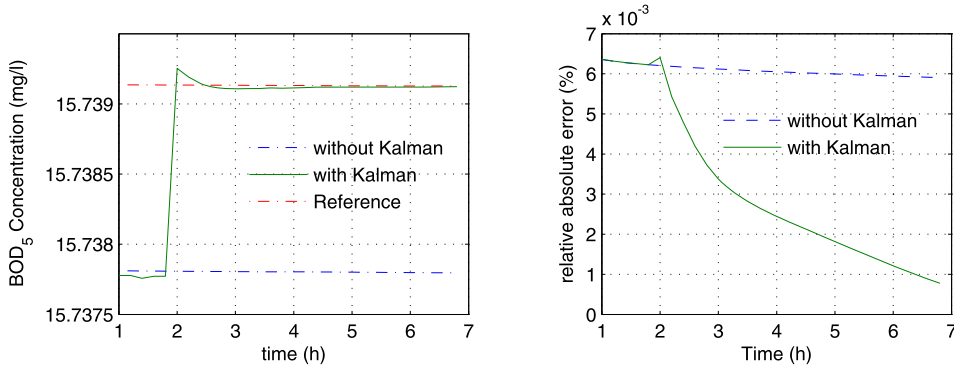


Fig. 22. (Color online.) Left: BOD₅ concentration at a point P4 in the middle of the lake, given by the reference (long dashes with stars) and by the model with (dashes) and without (solid) Kalman correction. Right: relative mean absolute error (in percentage) of BOD₅ concentration in the models with (solid) and without (dash) Kalman correction.

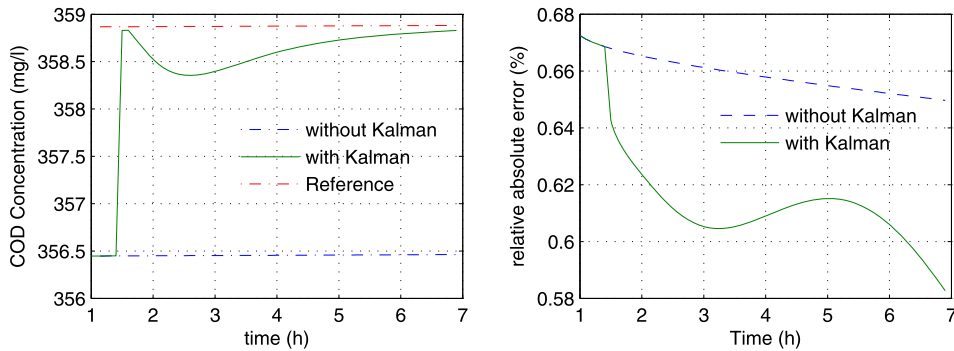


Fig. 23. (Color online.) Left: COD concentration at a point P4 in the middle of the lake, given by the reference (dashes and dots) and by the model with (solid) and without (dashes) Kalman correction. Right: relative mean absolute error (in percentage) of COD concentration in the models with (solid) and without (dashes) Kalman correction.

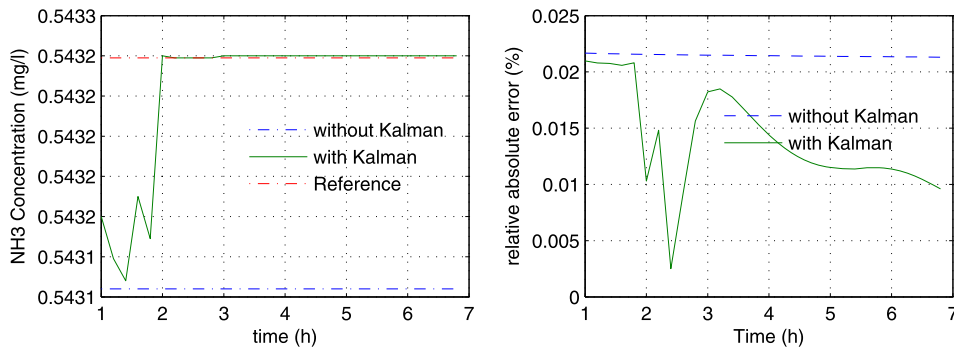


Fig. 24. (Color online.) Left: NH₃ concentration at a point P4 in the middle of the lake, given by the reference (dashes and dots) and by the model with (solid) and without (dashes) Kalman correction. Right: relative mean absolute error (in percentage) of NH₃ concentration in the models with (solid) and without (dash) Kalman correction.

In the popular PC (Intel core I5 2.6 GHz), it takes about 15 min and 2 h to run the Kalman filter algorithm with one substance and with three substances, respectively.

6. Conclusion

Our simulation results suggest the good performance of the method. We have considered only three pollutants, but the model can be used to simulate the transport of combinations of substances in water, with an unlimited number of them if we have enough data. In order to obtain coefficients $k_{i,j}$ ($i, j = 1, \dots, m$) in the combinations of pollution sources f_i ($i = 1, \dots, m$) needed in the right-hand side of Eq. (4), we use the experimental results from the Institute of Chemistry.

To apply the Kalman filter method, the source and diffusion coefficients D_i , k_{ij} have been obtained by the optimal variation method. The application of this method in this pollution problem will be described in another paper. The correction by the SEIK filter makes the simulation closer to the truth, and thus more useful for the simulation of the model. With the aid of the SEIK filter, the model can be used as a support tool to estimate substance contents and study water quality for the pollution problem.

Acknowledgements

The authors express deep gratitude to Dr. Trinh Anh Duc for his helps in realizing this paper. They are grateful to the VAST's pollution project, NAFOSTED's funds and KHCN-TB/13-18 program for their supports for this research.

References

- [1] C. Licht, T.H. Tran, Q.P. Vu, On some linearized problems of shallow water flows, *Differ. Integral Equ.* 22 (3–4) (2009) 275–283.
- [2] T.H. Tran, H.P. Nguyen, 2D-Model of contaminant water transmission processes and numerical simulation on a natural lake, *J. Mech.* 32 (3) (2010) 157–166.
- [3] I. Hoteit, D.-T. Pham, J. Blum, A simplified reduced order Kalman filtering and application to altimetric data assimilation in the Tropical Pacific, *J. Mar. Syst.* 36 (1) (2002) 101–127.
- [4] D.-T. Pham, J. Verron, M.-C. Roubaud, A singular evolutive extended Kalman filter for data assimilation in oceanography, *J. Mar. Syst.* 16 (3–4) (1998) 323–340.
- [5] D.-T. Pham, J. Verron, L. Gourdeau, Filtrés de Kalman singuliers évolutif pour l'assimilation de données en océanographie, *C. R. Acad. Sci. Paris, Ser. Ila* 326 (1998) 255–260.
- [6] D.-T. Pham, Stochastic methods for sequential data assimilation in strongly nonlinear systems, *Mon. Weather Rev.* 129 (5) (2001) 1194–1207.
- [7] G. Evensen, Sequential data assimilation with a non-linear quasi-geostrophic model using Monte-Carlo methods to forecast error statistics, *J. Geophys. Res.* 99 (C5) (1994) 10143–10162.
- [8] G. Evensen, Advanced data assimilation in strongly nonlinear dynamics, *Mon. Weather Rev.* 125 (1997) 1342–1354.
- [9] S.J. Julier, J.K. Uhlmann, A new extension of the Kalman filter to nonlinear systems, in: *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, 1997.
- [10] S. Julier, J. Uhlmann, H.F. Durrant-White, A new method for nonlinear transformation of means and covariances in filters and estimators, *IEEE Trans. Autom. Control* 45 (2000) 477–482.
- [11] E.A. Wan, R. van der Merwe, The unscented Kalman filter for nonlinear estimation, in: *Proc. IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*, Lake Louise, Alberta, Canada, 2000, pp. 153–158.
- [12] P.A. Sleight, P.H. Gaskell, M. Berzins, N.G. Wright, An unstructured finite volume algorithm for predicting flow in rivers and estuaries, *Comput. Fluids* 27 (1998) 479–508.
- [13] W. Rauch, M. Henze, L. Koncsos, P. Shanahan, L. Somlyódy, Vanrolleghem, River water quality modeling: I. State of the art, *Water Sci. Technol.* 38 (11) (1998) 237–244.
- [14] Van Manh Dinh, Minh Hanh Pham Thi, Water quality model for Quang Ninh-Haiphong area, in: *Proceedings of the 8th National Congress on Mechanics*, 2004, pp. 143–153.
- [15] Danish Hydraulic Institute, 2003, MIKE 21/3 WQ, Water quality module, Reference Manual.
- [16] WASP4, A Hydrodynamic and Water Quality Model: Model Theory, User's Manual and Programmer's Guide, January 1988, pp. 78–83.
- [17] A. Lebosse, Codes de calcul d'écoulement à surface libre filaire "LIDO", "SARA" et "REZO" (Version 2), Note de validation HE-43/92-65, EDF, 1992.
- [18] Ian Macdonald, Analysis and computation of steady open Channel flow, PhD thesis, University of Reading Dept. of Mathematics, 1996.
- [19] C. Hirsch, Numerical Computation of Internal and External Flows. Vol. 2: Computation Methods for Inviscid and Viscous Flows, Wiley-Interscience, New York, 1988.