NETWORKS AND HETEROGENEOUS MEDIA
©American Institute of Mathematical Sciences
Volume **4**, Number **2**, June **2009**

doi:10.3934/nhm.2009.4.409
pp. **409–430**

COMPARISON OF TWO DATA ASSIMILATION ALGORITHMS FOR SHALLOW WATER FLOWS

Issam S. Strub

Systems Engineering, Civil and Environmental Engineering
604 Davis Hall
Berkeley, CA 94720-1710, USA

Julie Percelay

Environmental Engineering, Civil and Environmental Engineering
604 Davis Hall
Berkeley, CA 94720-1710, USA

Olli-Pekka Tossavainen

Systems Engineering, Civil and Environmental Engineering
Berkeley Water Center, 413 O'Brien Hall
Berkeley, CA 94720-1710, USA

Alexandre M. Bayen

Systems Engineering, Civil and Environmental Engineering
711 Davis Hall
Berkeley, CA 94720-1710, USA

Abstract. This article presents the comparison of two algorithms for data assimilation of two dimensional shallow water flows. The first algorithm is based on a linearization of the model equations and a quadratic programming (QP) formulation of the problem. The second algorithm uses Ensemble Kalman Filtering (EnKF) applied to the non-linear two dimensional shallow water equations. The two methods are implemented on a scenario in which boundary conditions and Lagrangian measurements are available. The performance of the methods is evaluated using twin experiments with experimentally measured bathymetry data and boundary conditions from a river located in the Sacramento Delta. The sensitivity of the algorithms to the number of drifters, low or high discharge and time sampling frequency is studied.

1. Introduction.

The modeling and monitoring of river hydraulics are increasingly important as they provide drinkable water for populations as well as irrigation for a variety of crops. These flows are usually modeled using the shallow water equations, whether in a one or two dimensional formulation [10]. In this approximation, the flow is assumed to be nearly horizontal and the water column well-mixed. Since these models are an approximation of the flow, they are bound to gradually drift away from the physical phenomenon. An accurate modeling requires the use of experimental measurements to keep the model accurate with respect to the actual

2000 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Data Assimilation, Shallow Water Equations, Quadratic Programming, Ensemble Kalman Filtering.

This research is funded by NSF under grant 0615299, and by the Calfed project #SCI-07-C135-42.

flow. Data assimilation, which originated several decades ago in the fields of meteorology and oceanography [6, 22], consists in incorporating experimentally measured observations into a model in order to enhance its predictive capability. Over the years, a number of different techniques have been introduced, some of which rely on variational methods [24] while others use Kalman filtering and its extensions [12, 21], optimal statistical interpolation [23], or more empirical techniques such as the nudging method [19, 26].

Flow field measurements can be obtained through various types of sensors which can be broadly divided into two categories originating from a fluid dynamics terminology: Lagrangian and Eulerian sensors. Namely, Eulerian sensors are fixed sensors which measure the properties of the flow that goes by them, whereas Lagrangian sensors are moving with the flow. Recently, Lagrangian sensors have gained in popularity in part due to the increased miniaturization of electronic devices and also due to the advantages they present over their Eulerian counterparts. Among these we will mention a noticeably reduced unit cost, portability and versatility. However, when using Lagrangian sensors, the measured variables will be expressed in Lagrangian coordinates and therefore will be different from the state variables used in the forecast model which is usually written in Eulerian coordinates. As will be shown in the following, this will require using specific methods to assimilate Lagrangian data. Until recently, data assimilation of Lagrangian measurements has been applied mainly to oceanography [13, 14, 19, 21, 23, 25, 28, 30] and meteorology [20, 24]. One of the first attempts to perform Lagrangian data assimilation for shallow water flows was made in 2006 in [16]. In the present article, we present two data assimilation methods applied to river flows, namely, a novel algorithm based on *Quadratic Programming* (QP), and an algorithm using *Ensemble Kalman Filtering* (EnKF). In particular, we analyze their respective sensitivity to measurement accuracy, number of sensors and time sampling frequency. We compare their respective performance and robustness through twin experiments, using experimentally measured bathymetry and boundary conditions from a river located in the Sacramento Delta in California. In this case, the Lagrangian sensors are drifters which float along the river recording the position at regular instants in a way which reproduces typical GPS measurements.

This article is organized as follows. In Section 2, the two-dimensional shallow water equations and the equations describing the trajectories of Lagrangian drifters are presented. In Section 3, the QP algorithm is developed. The linearization of the model equations and their discretization as an implicit numerical scheme which will be used in the form of constraints in the QP are derived. The problem statement is presented in an optimization framework. In Section 4, the EnKF algorithm is presented based on the fully non linear equations. In Section 5, the two algorithms are then evaluated and compared using twin experiments. In particular, their sensitivity to the number of drifters, low or high discharge and time sampling frequency is studied. Finally, in Section 6 conclusions and suggestions for future work are given.

2. **Hydrodynamic model.** The governing hydrodynamic equations for the modeled system are [27, 32]:

$$\frac{\partial u}{\partial t} + \overrightarrow{U} \cdot \nabla u = -g\frac{\partial \eta}{\partial x} + F_x + \frac{1}{h}\nabla \cdot (h\nu_t \nabla u) \tag{1}$$

$$\frac{\partial v}{\partial t} + \overrightarrow{U} \cdot \nabla v = -g\frac{\partial \eta}{\partial y} + F_y + \frac{1}{h}\nabla \cdot (h\nu_t \nabla v) \tag{2}$$

$$\frac{\partial h}{\partial t} + \overrightarrow{U} \cdot \nabla h + h\nabla \cdot \overrightarrow{U} = 0 \tag{3}$$

The friction forces are given by the following Manning law:

$$F_x = -\frac{1}{\cos\alpha}\frac{gm^2}{h^{4/3}}u\sqrt{u^2+v^2} \tag{4}$$

$$F_y = -\frac{1}{\cos\alpha}\frac{gm^2}{h^{4/3}}v\sqrt{u^2+v^2} \tag{5}$$

where h is the total depth of water, $\overrightarrow{U} = (u, v)$ is the velocity field, g is the gravitational acceleration, η is the free surface elevation, ν_t is the coefficient of turbulence diffusion obeying the so called k-epsilon model [27], $\alpha = \alpha(x, y)$ is the slope of the bottom, and m is the Manning coefficient. Finally, t is time and x, y are horizontal space coordinates.

The boundary and initial conditions are given by

$$u(x,y,t)\big|_{\partial\Omega_{\text{land}}} = 0, \quad v(x,y,t)\big|_{\partial\Omega_{\text{land}}} = 0, \tag{6}$$

$$(u(x,y,t), v(x,y,t))\big|_{\partial\Omega_{\text{upstream}}} = f(x,y,t), \tag{7}$$

$$\eta(x,y,t)\big|_{\partial\Omega_{\text{downstream}}} = g(x,y,t), \tag{8}$$

$$u(x,y,0) = u_0(x,y), \quad v(x,y,0) = v_0(x,y), \quad h(x,y,0) = h_0(x,y), \tag{9}$$

where $\partial\Omega$ represents the respective boundaries of our computational domain and f, g are known functions.

In addition to the equations (1)-(3) describing the flow dynamics, we model the deployed drifters as passive Lagrangian tracers. Let D_i denote the i^{th} drifter which moves with the local fluid velocity, obeying the following equations:

$$\frac{dx_{D_i}(t)}{dt} = u(x_{D_i}(t), y_{D_i}(t), t), \tag{10}$$

$$\frac{dy_{D_i}(t)}{dt} = v(x_{D_i}(t), y_{D_i}(t), t), \tag{11}$$

with the initial conditions

$$x_{D_i}(0) = x_{D_i,0}, \quad y_{D_i}(0) = y_{D_i,0}. \tag{12}$$

where x_{D_i} and y_{D_i} are the x and y-coordinates of the i^{th} drifter, respectively. In the sequel we denote the concatenated vector of drifter coordinates (x_D, y_D) by θ_D.

3. Quadratic programming based inverse modeling algorithm.

3.1. Notations.

We start this section by defining the variables which will appear in the following sections. We employ the traditional notations of variational data assimilation, set forth in [17]. Note that the variables used below are in discrete space, discrete time. The corresponding schemes used to discretize equations (1)-(3) will be outlined later.

X_n: Vector of state variables, namely the velocity components (u, v) and the water height h for each mesh point at a time instant t_n.

X_0: Initial state of the system, with respect to which the minimization is done.

X^b: Background term of the same size as X_0, which is introduced to ensure that the problem is well-posed and has a unique minimum.

Y_n: Vector of observed variables, namely the velocity components (u, v) and (potentially) the water height h for some mesh points at a time instant t_n.

$\mathbf{B}$: Covariance matrix of the background error. The background error is defined as the difference between the value of the state variables and background variables at each mesh point.

$\mathbf{R}_n$: Covariance matrix of the observation error. The observation error is the difference between the value of the state variables and observed variables at each mesh point.

H_n: $H_n = h_n^o \circ h^I$ is the observation operator; the operator h_n^o projects the space into the observation subspace. The operator h^I is the interpolation function. In general H_n is nonlinear although we manage to use a linear operator in our case by using the a posteriori knowledge of the position of the measurements, therefore encoded as a time varying observation matrix.

Variational data assimilation consists in obtaining the initial state that minimizes a cost function representing the L^2 norm of the difference between the state and observed variables:

$$\mathcal{J}^o(X_0) = \sum_{n=0}^{N_{\max}} (Y_n^o - H_n[X_n])^T \mathbf{R}_n^{-1} (Y_n^o - H_n[X_n]) \tag{13}$$

where the subscript n denotes the time step. The number of points in the computational domain is usually significantly higher than the number of measurements located along drifter trajectories, most of the times by a factor 10^2 or more. This can make the minimization problem ill-posed; in particular, non uniqueness of the solution may result from the ill-posedness. Thus, a term representing the L^2 norm of the difference between an estimate of the initial state X^b and the initial state (unknown), called background term $\mathcal{J}^b$ is introduced leading to a unique minimum of the cost function and providing a first guess which accelerates the convergence of the minimization algorithm. Such an estimate of the initial state of the system may be available, for example, through historical data, simulation from a previous assimilated initial state, forecast, or interpolation from a limited number of fixed Eulerian sensors located at the boundaries of the system. This leads to a new cost function, by adding the background term $\mathcal{J}^b$ to the observation term $\mathcal{J}^o$:

$$\mathcal{J}(X_0) = (X_0 - X^b)^T \mathbf{B}^{-1} (X_0 - X^b) + \sum_{n=0}^{N_{\max}} (Y_n^o - H_n[X_n])^T \mathbf{R}_n^{-1} (Y_n^o - H_n[X_n]) \tag{14}$$

$\mathbf{B}$ and $\mathbf{R}_n$ are covariance matrices of the background error and observation error respectively and will be taken equal to diagonal matrices in a simplifying assumption. Note that $\mathcal{J}^o(X_0)$ depends on X_0 implicitly through the state X_n, obtained from X_0 by the flow equations, as will be detailed in the next section.

3.2. **Linearization of the constraints.** We will formulate the variational data assimilation problem detailed above as a QP. The first step is to obtain linear constraints, which in our case will be achieved by linearizing the shallow water equations. Indeed, for numerous environmental flows, such as the ones we will consider in our experiments, the Froude number is smaller than 1, usually by several orders of magnitude, ($Fr \leqslant 0.05$ in our case), and the nonlinear effects can be neglected. We can rewrite the equations (1)-(3) under the form:

$$u_t + uu_x + vu_y + gh_x = -gb_x + F_x \tag{15}$$

$$v_t + uv_x + vv_y + gh_y = -gb_y + F_y \tag{16}$$

$$h_t + (hu)_x + (hv)_y = 0 \tag{17}$$

having neglected turbulence effects and replacing η by $h - b$ where $b(x, y)$ is the bathymetric depth.

We linearize these equations around a flow $(U_0(x, y, t), V_0(x, y, t), H_0(x, y, t))$, satisfying (15)-(17) (see [32]) and we replace the friction terms F_x and F_y from Equation (15) and Equation (16) by an empirical drag coefficient:

$$C_{x,y} = \frac{1}{\cos\alpha} \frac{gm^2}{H_0(x, y, t)^{4/3}} \sqrt{U_0(x, y, t)^2 + V_0(x, y, t)^2}.$$

In general, $U_0(x, y, t)$, $V_0(x, y, t)$ and $H_0(x, y, t)$ are not uniform, because of the geometry of the river. Based on the flow conditions, U_0, V_0 and H_0 can be considered to be static or not. The linearized equations become:

$$u_t + U_0(x, y, t)u_x + V_0(x, y, t)u_y + gh_x = -gb_x - C_{x,y}u$$

$$v_t + U_0(x, y, t)v_x + V_0(x, y, t)v_y + gh_y = -gb_y - C_{x,y}v$$

$$h_t + U_0(x, y, t)h_x + V_0(x, y, t)h_y + H_0(x, y, t)(u_x + v_y) = 0$$

To discretize these equations, we will use an implicit four point scheme. The time discretization is chosen to be implicit since an explicit time discretization would require selecting time steps small enough to satisfy the *Courant-Friedrichs-Lewy* condition, which would in turn increase unnecessarily the number of variables and the size of our QP. A number of implicit schemes for the shallow water equations have been developed recently [1, 2, 3, 7, 8, 31].

If we call $a^x_{i,j}$ and $a^y_{i,j}$ the dimensions of the rectangular cell (i, j) along the x and y axes, the spatial derivative of a given variable u can be approximated by the following finite difference scheme:

$$\frac{\partial u}{\partial x} = \frac{u^{n+1}_{i+1,j} - u^{n+1}_{i-1,j} + u^n_{i+1,j} - u^n_{i-1,j}}{4a^x_{i,j}}$$

where i, j denote the spatial indices and n the time step.

The implicit Euler scheme is used for the time discretization: $\frac{\partial u_{i,j}}{\partial t}$ is approximated by $\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}$. The previous discretization steps lead to the following numerical scheme:

$$
\begin{aligned}
&\frac{u_{i+1,j+1}^{n+1}-u_{i+1,j+1}^{n}}{\Delta t}+\frac{U_0(i,j,n)}{4a_{i,j}^{x}}(u_{i+1,j}^{n+1}-u_{i-1,j}^{n+1}+u_{i+1,j}^{n}-u_{i-1,j}^{n})\\
&\quad+\frac{V_0(i,j,n)}{4a_{i,j}^{y}}(u_{i,j+1}^{n+1}-u_{i,j-1}^{n+1}+u_{i,j+1}^{n}-u_{i,j-1}^{n})\\
&+\frac{g}{4a_{i,j}^{x}}(h_{i+1,j}^{n+1}-h_{i-1,j}^{n+1}+h_{i+1,j}^{n}-h_{i-1,j}^{n}+2(b_{i+1,j}-b_{i-1,j}))+C_{i,j}u_{i,j}^{n}=0
\end{aligned}
$$

$$
\begin{aligned}
&\frac{v_{i+1,j+1}^{n+1}-v_{i+1,j+1}^{n}}{\Delta t}+\frac{U_0(i,j,n)}{4a_{i,j}^{x}}(v_{i+1,j}^{n+1}-v_{i-1,j}^{n+1}+v_{i+1,j}^{n}-v_{i-1,j}^{n})\\
&\quad+\frac{V_0(i,j,n)}{4a_{i,j}^{y}}(v_{i,j+1}^{n+1}-v_{i,j-1}^{n+1}+v_{i,j+1}^{n}-v_{i,j-1}^{n})\\
&+\frac{g}{4a_{i,j}^{y}}(h_{i,j+1}^{n+1}-h_{i,j-1}^{n+1}+h_{i,j+1}^{n}-h_{i,j-1}^{n}+2(b_{i,j+1}-b_{i,j-1}))+C_{i,j}v_{i,j}^{n}=0
\end{aligned}
$$

$$
\begin{aligned}
&\frac{h_{i+1,j+1}^{n+1}-h_{i+1,j+1}^{n}}{\Delta t}+\frac{U_0(i,j,n)}{4a_{i,j}^{x}}(h_{i+1,j}^{n+1}-h_{i-1,j}^{n+1}+h_{i+1,j}^{n}-h_{i-1,j}^{n})\\
&\quad+\frac{V_0(i,j,n)}{4a_{i,j}^{y}}(h_{i,j+1}^{n+1}-h_{i,j-1}^{n+1}+h_{i,j+1}^{n}-h_{i,j-1}^{n})\\
&\quad+\frac{H_0(i,j,n)}{4a_{i,j}^{x}}(u_{i+1,j}^{n+1}-u_{i-1,j}^{n+1}+u_{i+1,j}^{n}-u_{i-1,j}^{n})\\
&\quad+\frac{H_0(i,j,n)}{4a_{i,j}^{y}}(v_{i,j+1}^{n+1}-v_{i,j-1}^{n+1}+v_{i,j+1}^{n}-v_{i,j-1}^{n})=0
\end{aligned}
$$

Boundary conditions are specified upstream as:

$$u_{1,j}^{n}=u_{\text{up}}^{j}(t_n), v_{1,j}^{n}=v_{\text{up}}^{j}(t_n)$$

and downstream as:

$$h_{I,j}^{n}=h_{\text{d}}^{j}(t_n)$$

3.3. **Incorporation of Lagrangian measurements.** One issue that comes up when attempting to employ QP techniques to solve this problem is that the interpolation operator H_n is usually nonlinear. However, as mentioned before, in the present case we are able to use a linear operator. Indeed, for the example we will treat in this article, we assume that the Lagrangian sensors (drifters) have a sufficiently small time sampling period compared to the Lagrangian timescale of the body of water considered (which is realistic for the scenarios presented later); therefore, the Eulerian velocity can be approximated by the Lagrangian velocity obtained by finite differences [19, 23]:

$$u_{D_i}(\tau_n)=\frac{x_{D_i}(\tau_n)-x_{D_i}(\tau_{n-1})}{\Delta\tau} \tag{18}$$

$$v_{D_i}(\tau_n)=\frac{y_{D_i}(\tau_n)-y_{D_i}(\tau_{n-1})}{\Delta\tau} \tag{19}$$

where $x_{D_i}(\tau_n)$ and $y_{D_i}(\tau_n)$ are the positions of drifter D_i at time τ_n and $u_{D_i}(\tau_n)$, $v_{D_i}(\tau_n)$ represent the components of the velocity of the drifter at time τ_n at location $(x_{D_i}(\tau_n), y_{D_i}(\tau_n))$. $\Delta\tau$ is the sensors time sampling period.

The observed variables are equal to the state variables and $h_n^o = \mathbb{I}$ and $H_n = h^I$. For the interpolation operator, which maps measurement points to grid points, a bilinear interpolation can be used. If a finite volume scheme is used as in this article, the state variables are constant for each cell and therefore the observed variables can simply be matched to the state variables by looking in which cell the drifter is at a given time instant. Thus the bilinear interpolation operator is the identity in our case. We also assume the error covariance matrices $\mathbf{B} = \mathbb{I}$ and $\mathbf{R}_\mathrm{n} = \mathbf{R}_\mathrm{n} = \frac{1}{r}\mathbb{I}$ where r is an empirically determined scalar. The factor r is a weighting parameter to adjust the respective influences of the background and observation terms. For higher values of r, the analyzed state will be closer to the background term than the observations, whereas for lower values the regularizing effect of the background term is reduced, which in effect slows down the convergence of the algorithm. A careful choice of the parameter r is therefore essential to a good performance of the data assimilation algorithm.

3.4. **Optimization program formulation.** We can now pose the data assimilation problem as a quadratic optimization problem. $\mathbf{B}$ can be chosen so that the cost function (14) is quadratic positive definite. The discretized dynamics of the flow can be encoded in the form of linear constraints, which is one of the benefits of the quadratic method used. The numerical scheme is implicit, i.e. has to be written in the form $EX_{n+1} = FX_n + Gz_n$ where E is not necessarily invertible. F encodes a portion of the dynamics for the spatial derivatives, Gz_n encodes the inputs. In this equation, X_n encodes the vector of concatenated $(u_{i,j}^n, v_{i,j}^n, h_{i,j}^n)$ at a given n. The QP can incorporate these equations in the form of constraints at no further cost or complication. We concatenate the vectors $(u_{i,j}^n, v_{i,j}^n, h_{i,j}^n)$ for all (i, j, n) into a single vector called $\mathcal{X}$, and abbreviate the concatenated dynamics constraints by $\mathcal{A}\mathcal{X} = \mathcal{B}$, where this equation encodes the discretized flow equations. The search space for X_0, which is a portion of the concatenated vector $\mathcal{X}$ from which all other quantities depend is allowed to evolve in a set of feasible initial conditions dictated by the two-dimensional shallow water equations. Because the $(u_{i,j}^n, v_{i,j}^n, h_{i,j}^n)$ cannot take arbitrary values, the space in which $\mathcal{X}$ evolves can also be restricted to increase the speed of convergence of the method. These two constraints are encoded in the form of an inequality, $\mathcal{G}\mathcal{X} \leqslant \mathcal{H}$. The general form of the quadratic function is written using the variables $\mathcal{P}, \mathcal{Q}, \mathcal{R}$. Finally, using the previous variable definitions, and because of the background term, the minimization of equation (14) subject to the constraints listed previously can be written as a QP in the following form:

$$\begin{array}{ll} \text{minimize} & J(X_0) = \frac{1}{2}\mathcal{X}^T\mathcal{P}\mathcal{X} + \mathcal{Q}^T\mathcal{X} + \mathcal{R} \\ \text{subject to} & \qquad\quad \mathcal{G}\mathcal{X} \leqslant \mathcal{H} \\ & \qquad\quad \mathcal{A}\mathcal{X} = \mathcal{B} \end{array}$$

Note that the variables used in this definition $(\mathcal{A}, \mathcal{B}, \mathcal{G}, \mathcal{H}, \mathcal{P}, \mathcal{Q}, \mathcal{R})$ are not related to the variables used in the rest of the article designated by the same letters with other fonts. If boundary conditions measurements are available, they can be specified in the QP; if no boundary data is given, the background term boundary value will then be used in the algorithm. The twin experiments presented later in this article do not show a significant difference in relative error between the two cases.

4. **Ensemble Kalman filtering with state augmentation.** The EnKF is an alternative data assimilation method to the traditional *Extended Kalman Filter* (EKF). With nonlinear dynamics, the linearization required in EKF (for example, implicit time stepping) can be computationally too demanding. EnKF is a sequential data assimilation method which uses Monte Carlo or ensemble integrations. By integrating an ensemble of model states forward in time, it is possible to compute the mean and error covariances needed at analysis times (the time instants when measurements are used to update model output) [15, 9]. The analysis scheme in EnKF uses traditional update equations of the *Kalman Filter* (KF), except that the Kalman gain is computed using the error covariances provided by the ensemble of model states.

4.1. **Notations.** The notations for the EnKF algorithm presented in the rest of section 4 are as follows:

θ_n: Vector of state variables, namely the velocity components (u, v), the water height h for each mesh point and the positions of the drifters x_D, y_D at a time instant t_n.
F_n: Time dependent discretized forward 2D shallow water equation model.
w_n: State noise representing the error between the forward model and reality.
Q_n: Covariance of the state noise w_n.
y_n: Vector of observed variables, namely the positions of the drifters (x_D, y_D) at a time instant t_n.
ϵ_n: Measurement noise, representing the error in the GPS measurements of the drifter positions.
Γ_{ϵ_n}: Covariance matrix of the measurement noise ϵ_n.
C_n: Observation matrix.
ξ_n^p: p^{th} ensemble member at time t_n.
K_n: Kalman gain at time t_n.

4.2. **State-space equations.** We employ a state-augmentation approach that has been introduced earlier in oceanography (see [28, 21, 18] for more details). We symbolically denote by θ_F the state variables corresponding to (u, v, h) on the computational domain. θ_F is of size three times the number of nodes in the computational domain, corresponding to (u, v, h) on each node in the domain. In this approach, Lagrangian drifter positions are embedded in the state vector in discretized form and the new state vector θ_n at time t_n is written as

$$\theta_n = \begin{pmatrix} \theta_F(t_n) \\ \theta_D(t_n) \end{pmatrix}.$$

This approach makes it possible to take the Lagrangian nature of observations into account in a way which does not require deriving Eulerian quantities from Lagrangian measurements as was done in the previous section in equations (18) and (19).

By inspection, we note that there is a one-way coupled system from flow variables θ_F to drifter positions θ_D, that is, the drifter positions can be solved after the flow variables have been solved. We express this system by F_n, where as in the previous case n refers to the time step. Furthermore, due to the uncertainties in the modeling, we add a stochastic term w_n to represent these modeling errors. The noise process w_n, with covariance Q_n, is a Gaussian noise process that is used to model inaccuracies in the evolution model (discretization error, poorly known inputs etc., see for example [29]).

Thus, we can write a nonlinear discrete stochastic state space model in the following form:

$$\theta_{n+1} = F_n(\theta_n) + w_n. \tag{20}$$

Here θ_n is the predicted system state at time step n and $F_n(\theta_n)$ is one time step in the (non-linear) discretized shallow water and drifter model, which is time dependent. Note that if the equations were linear, X_n used in the previous section would be the portion of θ_n which does not contain any drifter position (labeled as θ_D).

For the observations, we use an additive noise model

$$y_n = C_n\theta_n + \epsilon_n. \tag{21}$$

The observation vector is y_n and the observation model that relates the state variables to the measurements is C_n. The noise process ϵ_n is the measurement noise with covariance Γ_{ϵ_n}. Without loss of generality, noise processes are assumed to have zero mean. Furthermore, with these choices, the observation model C_n becomes simply $C_n = (\mathbf{0}\ \mathbb{I})$ where $\mathbf{0}$ is a zero matrix of size three times the number of nodes by two times the number of drifters for the three variables (u, v and h) and $\mathbb{I}$ is the identity operator of size the number of drifters times two.

The numerical solution of the 2D shallow-water equations and drifter positions in (20) is computed using the commercial hydrodynamic software TELEMAC 2D [27]. TELEMAC uses a streamline upwind Petrov-Galerkin based finite element solver for hydrodynamic equations. Therefore, in practice, the F_n step of equation (20) is performed numerically by the software.

4.3. **Ensemble Kalman filter (EnKF).** In the filtering problem, the aim is to compute conditional expectations

$$\theta_{n|n} = E(\theta_n|y_n, \ldots, y_1).$$

In the case of linear observation and evolution equations and for Gaussian noise processes, the recursive Kalman filter algorithm can be used for determining the estimates of conditional expectation $\theta_{n|n}$ and covariance $\Gamma_{n|n}$. If the evolution and/or observation equations are nonlinear and differentiable, the extended Kalman filter can be applied [4]. However, to avoid the cumbersome linearization of a nonlinear finite difference scheme and to preserve the higher order statistics, which may be lost in the linearization, we employ the EnKF.

For the state space model (20)–(21), the EnKF algorithm can be summarized as in [15, 12]:

Algorithm

1. Initialization: An ensemble of N_{states} states $\xi_0^{(p)}$ indexed by p are generated to represent the uncertainty in θ_0.
2. Time update:

$$\xi_{n|n-1}^{(p)} = F_{n-1}(\xi_{n-1|n-1}^{(p)}) + w_{n-1}^{(p)} \tag{22}$$

$$\theta_{n|n-1} = \frac{1}{N_{\text{states}}} \sum_{p=1}^{N_{\text{states}}} \xi_{n|n-1}^{(p)} \tag{23}$$

$$E_{n|n-1} = [\xi_{n|n-1}^{(1)} - \theta_{n|n-1}, \ldots, \xi_{n|n-1}^{(N_{\text{states}})} - \theta_{n|n-1}] \tag{24}$$

3. Measurement update:

$$\Gamma_{n|n-1} = \frac{1}{N_{\text{states}} - 1} E_{n|n-1} E^{\mathrm{T}}_{n|n-1} \tag{25}$$

$$K_n = \Gamma_{n|n-1} C_n^{\mathrm{T}} (C_n \Gamma_{n|n-1} C_n^{\mathrm{T}} + \Gamma_{\epsilon_n})^{-1} \tag{26}$$

$$\xi^{(p)}_{n|n} = \xi^{(p)}_{n|n-1} + K_n (y_n - C_n \xi^{(p)}_{n|n-1} + \epsilon_n^{(p)}) \tag{27}$$

where the ensemble of state vectors are generated with the realizations $w_n^{(p)}$ and $\epsilon_n^{(p)}$ of the noise processes w_n and ϵ_n, respectively. In the previous equations, an important step is that at measurement times, each measurement is represented by an ensemble. This ensemble has the actual measurement y_n as mean and the variance of the ensemble is used to represent measurement errors. This is done by adding perturbations $\epsilon_n^{(p)}$ to measurements drawn from a distribution with zero mean and covariance equal to measurement error covariance matrix Γ_{ϵ_n}. This ensures that the updated ensemble has a variance that is not too low [9]. The graphical illustration of the EnKF algorithm is presented in Fig. 1.

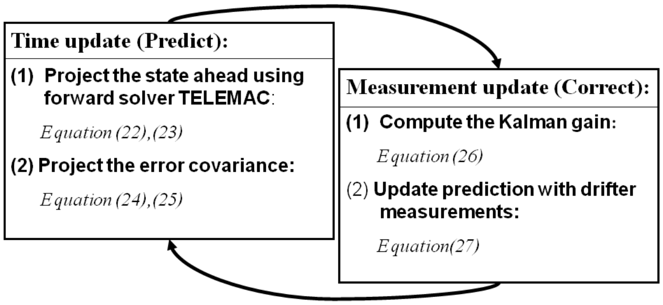

Time update (Predict):
(1) Project the state ahead using forward solver TELEMAC:
Equation (22),(23)
(2) Project the error covariance:
Equation (24),(25)
Measurement update (Correct):
(1) Compute the Kalman gain:
Equation (26)
(2) Update prediction with drifter measurements:
Equation(27)

FIGURE 1. Graphical illustration of the EnKF algorithm.

4.4. **EnKF implementation.** The initial ensemble for the EnKF is generated from historical knowledge on the boundary conditions in the river. We generate an ensemble of size $N_{\text{states}} = 100$ as follows. The idea is to produce N_{states} different initial velocity fields representing the different states of the river. In order to do this, we use data from the DSM2 model, which are available for a period of eight years. From these data, we extract statistical characteristics of the boundary conditions: a mean value of discharge upstream, a mean value of surface elevation downstream and a covariance matrix of those two quantities. From this, we generate a set of discharges and free surface elevations $(q^{(p)},\eta^{(p)})$, $p = 1, \ldots, N_{\text{states}}$ that correspond to the statistical characteristics deduced from the historical data. For each discharge $q^{(p)}$, we deduce a velocity profile

$$(u^{(p)}, v^{(p)})\big|_{\partial\Omega_{\text{upstream}}},$$

that is normal to the upstream boundary and proportional to the square root of the water depth at the upstream boundary. The downstream boundary condition is the surface elevation obtained directly from the historical data statistics:

$$\eta^{(p)}\big|_{\partial\Omega_{\text{downstream}}} = \eta^{(p)}.$$

In order to obtain realistic initial states for the EnKF algorithm, a 150 minute stabilization run is performed as follows. The forward solver TELEMAC 2D is started from zero velocity and it runs for 150 minutes for each ensemble member using the set $(u^{(p)}, v^{(p)})\big|_{\partial\Omega_{\text{upstream}}}, \eta^{(p)}\big|_{\partial\Omega_{\text{downstream}}}$ as boundary conditions. By doing this, numerically stable and physically meaningful initial states for flow variables $(u_0^{(p)}, v_0^{(p)}, h_0^{(p)})$ are obtained. Furthermore, the zero velocity initial condition that we use to initialize TELEMAC 2D is not affecting the flow variables when the state estimation starts.

When stable states are reached, state estimation can start. In the state estimation procedure we do not update the velocity at the nodes that correspond to the land boundary. This is to make sure that we do not violate the no-slip condition. The initial boundary conditions for each ensemble member at time step zero are $(u^{(p)}, v^{(p)})\big|_{\partial\Omega_{\text{upstream}}}, \eta^{(p)}\big|_{\partial\Omega_{\text{downstream}}}$; these are obtained from the last time step of the stabilizing run.

We set the standard deviation of the measurement noise ϵ_n to be 0.5 m in both x and y directions. This indicates our confidence on the measurement data, and can be seen as a realistic choice for the accuracy of GPS receivers.

The state noise models the discrepancy between the model and reality. In this work it is assumed that the state noise for the flow variables can be deduced from the properties of large number of ensemble members. The covariance matrix Q_n for the flow variables is computed as the covariance of the initial ensemble members around the mean flow. Thus, the state noise reflects the variations in the velocity and depth field due to the uncertain and changing boundary conditions.

We further assume that the mesh nodes corresponding to the land nodes are not affected by the state noise in order not to violate the no-slip condition on the land boundaries. In addition, we assume that there is a small uncertainty in the model which predicts the drifter positions. This uncertainty is modeled by a white noise, that is the covariance matrix for the drifters is a diagonal matrix. This uncertainty is indicated by setting the standard deviation of the state noise in the covariance matrix Q_n corresponding to all drifter coordinates to be 0.5 m in both x and y directions.

5. **Results.**

5.1. **Twin experiments settings.**

We compare the two methods using twin experiments. The *true* state of the river and the position of the drifters are generated using the software TELEMAC. We perform several experiments in order to test the influence of the number of drifters and their initial position in the river.

The domain is the Sacramento River, upstream of the intersection with the Georgiana Slough, in the Sacramento delta. The bathymetry is provided by the *United States Geological Survey* (USGS) as shown in Fig. 2. The 930 m long section of the Sacramento River is 85 m wide in its narrowest part and 190 m wide in its largest part. The flow in this part of the Sacramento River is driven by tidal forcing. A typical velocity field is represented in Fig. 3 for a case of low flow.

The software TELEMAC is used to perform a non linear forward simulation from which the drifter position measurements and the so called *true state* of the river are obtained. The true state and the drifter measurements are computed in a denser mesh (2972 nodes, 5484 triangular elements) compared to the mesh used in the

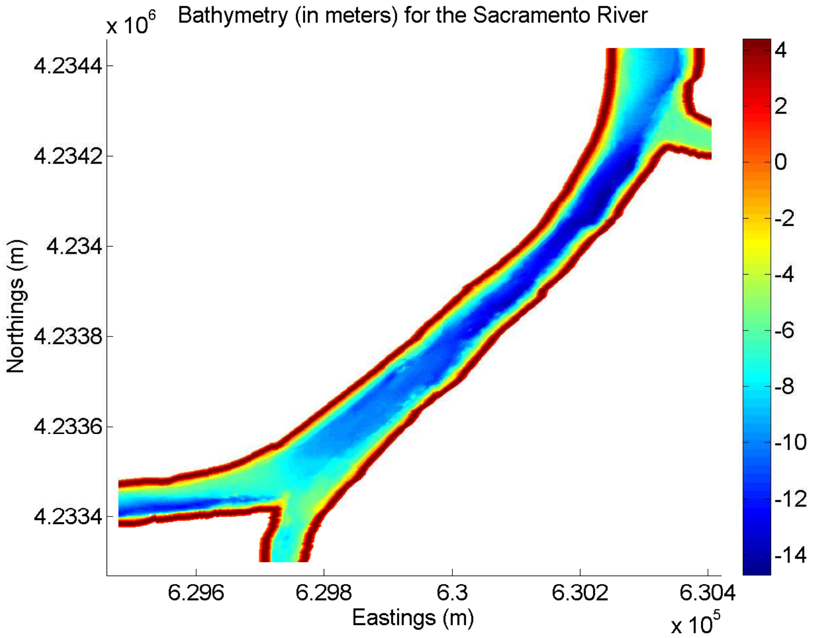


Figure 2. Bathymetry in the Sacramento River (m). The bathymetry on this 930m section of the Sacramento River goes from -14m in the deepest part to +2m on the river banks.

solution of the inverse problem for EnKF (1550 nodes, 2788 triangular elements) and the curvilinear mesh used in the QP (37×6 cells).

The boundary conditions for models are computed using the *Delta Simulation Model II* (DSM2) [5]. It is a model of the San Francisco Bay and Sacramento Delta that provides discharge and surface elevation at various locations every hour. The bottom friction is modeled using Manning's law. The Manning coefficient is chosen to be constant in time and space and equal to 0.02, corresponding to a straight gravel bottom [11]. The simulation runs for two and a half hours before the drifters' release so that a stable state is reached.

Three different cases are considered. First we model a situation with constant boundary conditions. The second and third test cases are situations with varying boundary conditions extracted from DSM2 results, corresponding to a high and a low flow situations. DSM2 values are available at a one hour sampling rate and interpolated linearly. The first time period, later denoted as Period 1, corresponds a high flow situation on March 17th 2006 from 4PM to 5PM. The second time period, later denoted as Period 2, corresponds to a low flow situation on June 29th 2006 from 1AM to 2AM. Fig. 3 represents the velocity field at the end of this period. During the one hour period, twenty-four drifters are released. The first eight drifters are released at t_0 along a line upstream. Then, at t_0 + 20 minutes, we release eight more drifters upstream, followed by eight more at t_0 + 40 minutes. Drifter positions are then recorded every 30 seconds until t_0 + 60 minutes. Fig. 4 shows snapshots of drifter positions for the low flow case.

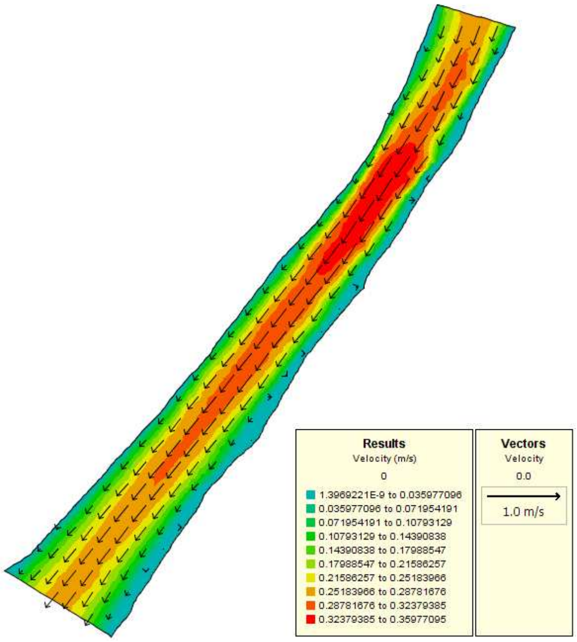
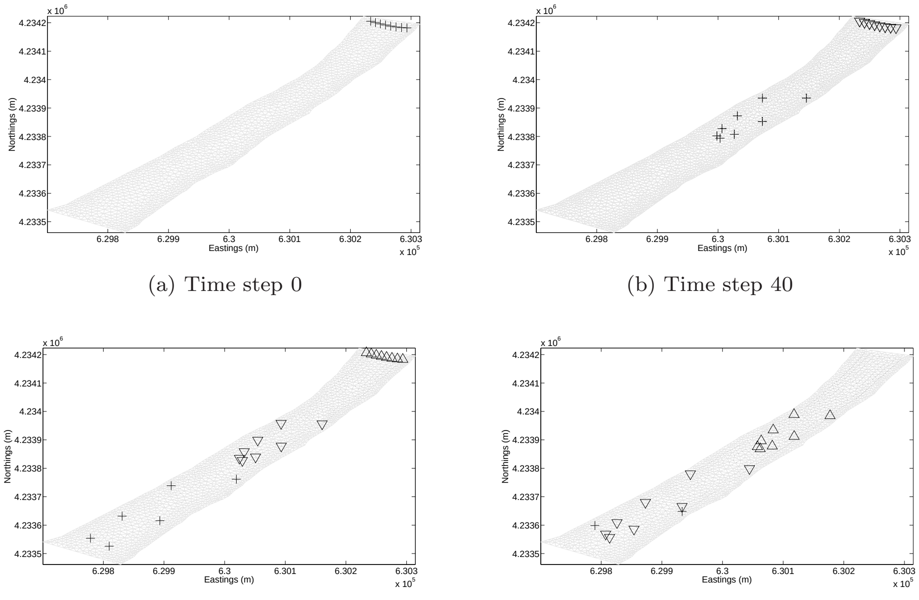


FIGURE 3. Velocity field in the Sacramento River (m/s) for low flow boundary conditions. The background color represents the magnitude of the velocity.

Different tests can be run using these simulations. First, we investigate the influence of the number of drifters, by using four, eight, twelve or twenty-four out of the total twenty-four released drifters in the assimilation algorithm. Besides, we test the influence of the deployment strategy in releasing drifters on one line at the initial time or on several rows at different time steps. Then, we study the influence of the frequency of the measurements by using only one measurement every minute instead of every 30 seconds.

The comparison between the two methods is made a posteriori: the QP runs on a period of one hour and recovers the state of the system at the beginning of this period, when the drifters were released. Additionally, the QP based algorithm generates the state of the system at every time step over the period of assimilation, as part of the state $\mathcal{X}$ reconstructed. The QP is solved using a large scale QP software (in our case CPLEX) with the algorithm itself being coded through the AMPL modeling language. For a one hour period, about 20 iterations of a barrier method are necessary for the solver to converge to a solution which takes about five seconds on a desktop computer. A background term, necessary for the QP algorithm, is generated from the available information (in our case, the upstream boundary conditions). The EnKF generates an estimate of the state vector at each time step in real time. The duration of a one-hour run is between six to eight hours on a standard desktop computer. The evolution of the *relative root mean square* (RMS) error is computed for both the algorithms (QP based and EnKF) and the respective errors are compared for each twin experiment.

5.2. **Results and comparison.** A total of 25 cases are considered, using the three different boundary condition settings as described previously, either four, eight,



(a) Time step 0

(b) Time step 40

(c) Time step 80

(d) Time step 120

FIGURE 4. Eight drifters are released at time step 0 (+), eight drifters at time step 40 ($\bigtriangledown$), eight drifters at time step 80 ($\bigtriangleup$). Drifter positions are sampled with 30-second-time-interval until time step 120.

twelve or twenty-four of drifters released on different configurations and using a 30 second sampling or a 60 second sampling for the measurements.

The boundary conditions that are used here correspond to various configurations of the river. We use constant boundary conditions picked from DSM2 values. Discharge upstream is $Q_{\text{up}} = 235.3$ m^3/s and the surface elevation downstream is $\eta_{\text{down}} = 1.49$ m. This corresponds to normal conditions in the river. The Sacramento River is tidally forced, thus constant boundary conditions do not occur in the system. The constant boundary condition case is a benchmark for the other cases.

For the low flow varying boundary conditions, we use Period 2. The discharge upstream varies from 300.2 m^3/s at 1AM to 223.1 m^3/s at 2AM. The surface elevation downstream varies from 1.52 m at 1AM to 1.58 m at 2AM. The discharge here is smaller as the snow melt period is over. As it is at the beginning of the dry season, the river still flows towards the ocean. The variation of the discharge is still due to tidal forcing.

For the high flow varying boundary conditions, we use Period 1. The discharge upstream varies from 722.3 m^3/s at 4PM to 686.1 m^3/s at 5PM. The surface elevation downstream varies from 2.91 m at 4PM to 2.95 m at 5PM. This high flow conditions are typical of snow melt season in the Sacramento Delta. The flow varies because of tidal forcing and the discharge is high because of snow melt runoff.

For each of the scenarios described previously, we use either four, eight, twelve or twenty-four of the released drifters. For the four drifter case, four of the eight drifters released at t_0 are used for the QP. For the EnKF, two of the drifters released at t_0 and two of the drifters released at t_0 + 20 minutes are used. For the eight drifter case, the QP uses the eight drifters released at t_0, while the EnKF uses either the same eight drifters or four of the eight drifters released at t_0 and four of the eight drifters released at t_0 + 20 minutes. For the twelve drifter case, the EnKF uses four of the eight drifters released at t_0, t_0 + 20 minutes and t_0 + 40 minutes. For the twenty-four drifter case, both QP and EnKF use all of the eight drifters released at t_0, t_0 + 20 minutes and t_0 + 40 minutes. For each of these cases, we use either a 30 second time step or a one-minute time step.

For each case, the relative RMS error is computed for the velocity:

$$\varepsilon(t) = \sqrt{\frac{\sum_{i=1}^{\text{NNode}} \|\overrightarrow{U}_{\text{True}}(\mathbf{x}_i, t) - \overrightarrow{U}(\mathbf{x}_i, t)\|^2}{\sum_{i=1}^{\text{NNode}} \|\overrightarrow{U}_{\text{True}}(\mathbf{x}_i, t)\|^2}} \tag{28}$$

where $\mathbf{x}_i$ represents a generic point on the mesh generated to solve these problems numerically and NNode represents the number of nodes on this mesh.

The results for all the cases and for the two algorithms are summarized in table 1.

5.3. Result overview.

Table 1 shows that the evolution of the error in time is different for both methods. The error is relatively stable for the case of the QP while it significantly varies for the EnKF as can be seen in Fig. 5 as well. The reason for the highly variable error in the EnKF is that the right model to use for the state noise is still an open problem. Future work on its refinement should provide more accurate estimates.

Fig. 5 shows the evolution in time of the error for the EnKF method compared to a reference case (forward simulation with no update of the state using the drifter measurements). The assimilation of the drifter positions allows the algorithm to catch the variation of the flow and thus correct the velocity field.

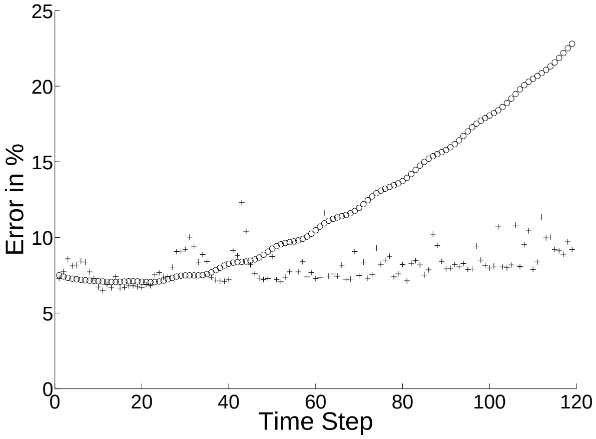


FIGURE 5. Time evolution of the relative error of the velocity for EnKF estimate (+) and the reference case (○), in the case of low flow boundary conditions using 24 drifters with a time sampling of 30 seconds. The EnKF estimate captures the variations in the velocity field and thus, has a smaller error.

Time		t_0		t_0+30min		t_0+60min		Mean	
N_d	M.S.	QP	EnKF	QP	EnKF	QP	EnKF	QP	EnKF
Constant Flow									
4	30s	19.8%	7.4%	20.0%	20.1%	20.0%	15.7%	20.0%	18.0%
	60s	19.8%	16.4%	20.0%	28.0%	20.0%	13.4%	20.0%	20.0%
8	30s	8.0%	7.0%	9.6%	9.0%	9.6%	9.8%	9.4%	8.7%
	60s	8.0%	18.1%	9.6%	15.3%	9.6%	14.7%	9.4%	12.1%
12	30s	N.A.	7.9%	N.A.	7.6%	N.A.	9.7%	N.A.	8.3%
	60s	N.A.	18.1%	N.A.	11.2%	N.A.	14.4%	N.A.	12.8%
24	30s	N.A.	7.1%	N.A.	8.1%	N.A.	9.2%	N.A.	8.5%
	60s	N.A.	16.1%	N.A.	15.7%	N.A.	15.1%	N.A.	13.2%
Low Flow									
4	30s	19.8%	6.6%	20.6%	13.4%	20.8%	20.5%	20.5%	15.3%
	60s	20.1%	N.A.	20.2%	N.A.	20.2%	N.A.	20.2%	N.A.
8	30s	4.8%	7.2%	4.9%	8.2%	5.8%	10.9%	5.1%	9.2%
	60s	5.0%	8.6%	5.1%	11.3%	5.8%	12.2%	5.2%	12.1%
8(1)	30s	4.8%	7.6%	4.9%	8.1%	5.8%	15.0%	5.1%	9.2%
12	30s	N.A.	7.1%	N.A.	9.7%	N.A.	11.5%	N.A.	10.1%
	60s	N.A.	8.2%	N.A.	9.3%	N.A.	14.5%	N.A.	10.5%
24	30s	5.2%	7.3%	5.1%	7.3%	5.0%	9.2%	5.1%	8.2%
	60s	N.A.	7.5%	N.A.	7.5%	N.A.	10.6%	N.A	9.3%
High Flow									
4	30s	20.6%	7.9%	20.4%	12.8%	20.5%	10.5%	20.5%	11.2%
	60s	25.9%	8.1%	25.7%	12.9%	25.9%	12.2%	25.8%	11.6%
8	30s	7.5%	8.1%	6.8%	8.4%	7.0%	12.2%	7.0%	9.1%
	60s	19.8%	11.1%	19.5%	8.5%	19.7%	14.6%	19.6%	9.6%
12	30s	N.A.	8.4%	N.A.	8.9%	N.A.	9.1%	N.A.	8.5%
	60s	N.A.	10.9%	N.A.	8.8%	N.A.	8.4%	N.A.	8.5%
24	30s	N.A.	8.6%	N.A.	10.4%	N.A.	7.9%	N.A.	8.2%
	60s	N.A.	11.3%	N.A.	9.3%	N.A.	7.0%	N.A.	8.4%

TABLE 1. Relative RMS error as computed in equation (28) for the different test cases. N_d is the number of drifters. M.S. is the measurement sampling time. For each case, the errors are reported for the initial time (t_0), after half of the total assimilation time (t_0 + 30 minutes) and at the end of the experiment (t_0 + 60 minutes). The mean error over the time period is also reported. For four and eight drifters, QP uses one row of drifters released at time step t_0 and the EnKF uses two rows of drifters released at t_0 and t_0 + 20 minutes. For twelve and twenty-four drifters, three rows of drifters are used for both algorithms. The case labeled (1) corresponds to a release of one row of drifters only for EnKF. N.A. means the simulation was not realized for that particular case.

Fig. 6 shows the relative error of the QP estimate and EnKF estimate at time step 0 for the case with low flow varying boundary conditions using eight drifters and a time sampling of 30 seconds.

The error is located on the land boundary, where relative error can be up to 90% on the boundary. The absolute error in that case is actually around 0.1m/s. The error is high along the land boundary because the EnKF corrects the velocity field in an area around where drifters are. There are no drifters close to the boundary and thus the error is higher. The same phenomenon can be seen for the QP estimate: the error is highest around the land boundaries because of the absence of drifters in those areas.

5.4. **Influence of the number of drifters.** Looking at Table 1 and Fig. 7, the difference in the error when using eight, twelve or twenty-four drifters is quite small for the EnKF. For the three different boundary conditions cases and the two different time sampling, when going from twenty-four to eight drifters, the error increases of 0.2% for the constant boundary condition case with 30 second measurement

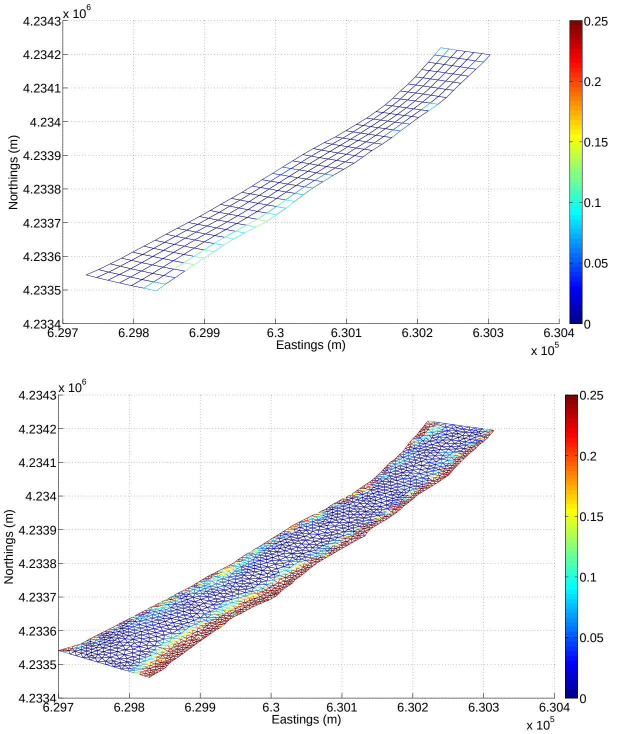


FIGURE 6. Relative error in the velocity field with varying boundary conditions, low discharge with eight drifters in a row for QP (top) and eight drifters in two rows of four for EnKF (bottom) and a time sampling of 30 seconds.

sampling (8.5% to 8.7%) to 2.8% for the low flow case with 60 second measurement sampling (9.3% to 12.1%). There is a higher difference when using four drifters: when the number of drifters goes from eight to four, the error increases of 2.1% for the high flow case with 30 second measurement sampling (9.1% to 11.2%) to 9.3% for the constant boundary condition case with 30 second measurement sampling (8.7% to 18%). The number of drifters have less influence in the cases of low and high flow varying boundary conditions while it has a remarkable influence in the case of constant boundary conditions. This is related to the fact that the EnKF method uses flow measurements to correct the prediction of the forward model.

The QP shows a significant improvement when the number of drifters increases. Dropping the number of drifters from eight to four, the RMS error in the velocity increases of 6.2% for the case of high flow with 60 second measurement sampling (19.6% to 25.8%) to 15.4% for the low flow with 30 second measurement sampling (5.1% to 20.5%). Increasing the number of drifters to twenty-four does not decrease further the error because of the deployment strategy in three rows that is used. The QP method is performing a minimization based on the initial state, thus only adding drifters on the initial row will improve the method performance.

5.5. **Influence of the deployment strategy.** It is to be noted that using several rows of drifters during the experiment for the QP does not yield any improvement to

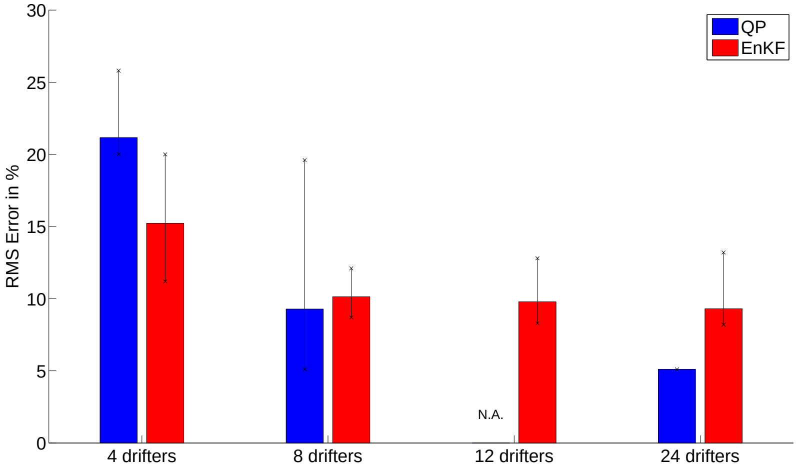


FIGURE 7. Mean RMS error for the different cases (different boundary conditions and time sampling) for four, eight, twelve and twenty four drifters. The error bar represents the variation of the error among the cases.

the relative RMS error. This can be seen in the result of the simulation using twenty-four drifters that gives results close to the result of the eight drifter simulation. Given that the QP based algorithm performs the minimization with respect to the initial state, this result is not too surprising.

In order to assess the influence of deployment strategy for EnKF, we realized a twin experiment using one row of eight drifters for the EnKF instead of two successive rows of four drifters. The chosen case was with low flow boundary conditions and a measurement sampling of 30 seconds. The mean value of the error is 9.2% which is the same as for the case with two releases of four drifters. The error is comparable during the first third of the experiment and it then increases when the drifters start exiting the domain and no new drifters are released. The noise in the EnKF results that is due to the modeling of the state noise, as stated before, explains the fact that the mean value in time for the two strategies is the same. At the end of the time period, the error is higher by 4.1% when only one row of drifters is used. The estimate after all the drifters exited the domain is thus useless. As the EnKF is doing an online inversion, the method shows better performances for constant releases of drifters rather than one release of drifters.

The two methods, QP and EnKF, show very different performances regarding the deployment strategy. Because they operate differently, each algorithm has a specific optimal drifter release strategy. The EnKF is a sequential data assimilation method which improves the model prediction as new observations are gathered whereas the QP based algorithm is a variational data assimilation method which uses all the observations available at the end of the period considered and minimizes a cost function containing both modeled and observed variables. Thus given a total number of available drifters, the optimal release strategy for the EnKF consists in splitting the drifters into a number of rows that are released at regular intervals during the experiment while for the QP based algorithm, all the drifters should be released at the beginning of the observation period.

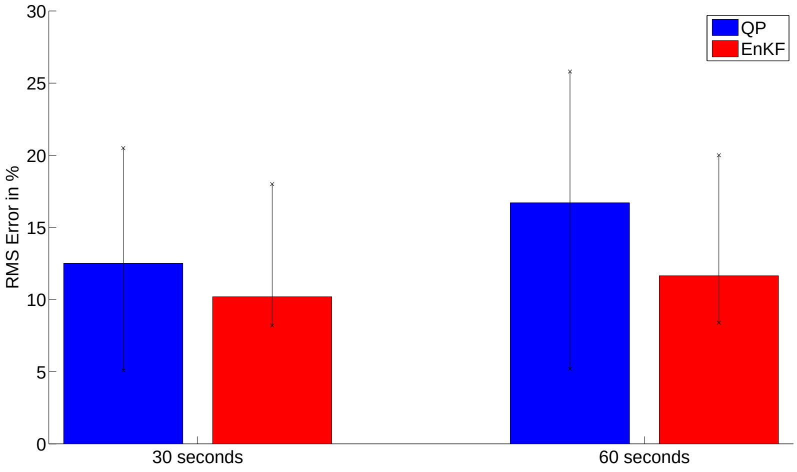


FIGURE 8. Mean RMS error for the different cases (different boundary conditions and number of drifters) for 30 second and 60 second measurement sampling. The error bar represents the variation of the error among the cases.

5.6. **Influence of time sampling.** Fig. 8 shows that time sampling has a greater influence on the QP estimate than on the EnKF estimate. This can be explained by several factors as described below.

The difference in time sampling is not relevant for the EnKF for the cases with varying boundary conditions. When the time sampling increases from 30 to 60 seconds, the relative RMS error remains constant for the high flow case with twelve drifters (8.5%) or increases of up to 2.9% for the low flow case with eight drifters (9.2% to 12.1%). The relative RMS error however increases by 2.0% for four drifters (18.0% to 20.0%) to 4.7% for twenty-four drifters (8.5% to 13.2%) in the case of constant boundary conditions. This can be explained by the fact that the time step of the direct simulation is much smaller than the sampling time for the measurements (by a factor of 30 to 60 respectively), therefore more frequent updates yield more accurate results. For the QP, the measurement sampling has no influence on the relative RMS error for the constant boundary conditions and varying boundary conditions with low flow (constant error for constant boundary conditions with four or eight drifters and increase of up to 0.3% for low flow with four drifters) while it makes a significant difference in the case of high flow (5.3% increase in RMS error with four drifters to 12.6% with eight drifters). The difference in the estimation error is explained by the fact that at velocities encountered in this experiment (0.8 m/s) a drifter will have floated two or three grid cells of the QP grid in between two measurements resulting in the absence of observations for about half of the grid cells. The same phenomenon does not occur for the EnKF. This is explained by the fact that the EnKF catches the variation of the velocity using the ensemble statistics and the drifter measurements. The magnitude of the velocity does not affect the EnKF estimate. In the case of low flow, velocities are lower (0.4 m/s) and drifters move at most from one cell to next one in between two measurements, and provide observations for almost every cell on their trajectory. Thus the error for the QP is not affected by the time sampling for low flows.

The QP is thus very sensitive to time sampling depending on the conditions in the river and the grid used in the inversion. On the other side, EnKF shows reasonable performances with fewer measurements independently of flow conditions.

6. **Conclusion.** In this article, two data assimilation methods were applied to a river in the Sacramento Delta and their respective performances were compared. The comparison was made using twin experiments, in which drifter trajectories are simulated, implemented with experimentally measured bottom topography and boundary conditions. The sensitivity of the different data assimilation techniques to the number of drifters, low or high discharge and time sampling frequency was analyzed and the respective computational costs of each method compared. The main conclusions of this study are that the QP based algorithm introduced by the authors presents a good balance of accuracy and low computational cost. The EnKF algorithm which uses the nonlinear two-dimensional shallow equations is slightly more accurate in some cases, but requires longer computational time (up to 8 hours compared to the few seconds needed for the QP algorithm). Its benefit is to capture nonlinear features in the flows, which the QP cannot. This might be investigated in the future. The QP method returns higher relative RMS errors when the number of drifters is reduced from eight to four or when the time sampling frequency is changed from 30 to 60 seconds. The EnKF is less sensitive to time sampling and number of drifters. A first attempt at estimating the optimal deployment strategy was made, showing that the QP requires one row of drifters released at the beginning of the assimilation period while the EnKF gives better estimate when several rows of drifters are released over the assimilation period. Future research directions include adding a salinity or sediment model to the shallow water equations and applying the data assimilation algorithms in this new setting. Also, field experiments will be conducted in the rivers of the Sacramento Delta and the experimental data collected used for data assimilation purposes through the methods presented in this article. Another issue is the optimal placement of the drifters and the optimal release points in order to maximize the accuracy of the assimilated state.

Acknowledgments. The authors gratefully acknowledge Prof. Mark Stacey from UC Berkeley for giving valuable information about the river hydraulics during this work, Aaron Blake from USGS for the bathymetry data used in this work and Tara Smith from the California Department of Water Resources for the DMS2 data used in this work.

REFERENCES

[1] K. Adamy and D. Pham, *A finite volume implicit Euler scheme for the linearized shallow water equations: Stability and convergence*, Numerical Functional Analysis and Optimization, **27** (2006), 757–783.

[2] F. Alcrudo and P. Garcia-Navarro, *A high-resolution Godunov-type scheme in finite volumes for the 2d shallow-water equations*, Int. J. Numer. Methods Fluids, **16** (1993), 489–505.

[3] K. Anastasiou and C. T. Chan, *Solution of the 2d shallow water equations using the finite volume method on unstructured triangular meshes*, Int. J. Numer. Methods Fluids, **24** (1997), 1225–1245.

[4] B. D. O. Anderson and J. B. Moore, “Optimal Filtering,” Prentice-Hall, inc, JH, 1979.

[5] J. Anderson and M. Mierzwa, *DSM2 Tutorial, an introduction to the Delta Simulation Model II (DSM2)*, Available online at http://wwwdwr.water.ca.gov/, State of California, Department of Water Resources (2002).

[6] R. A. Anthes, *Data assimilation and initialization of hurricane prediction models*, J. Atmos. Sci., **31** (1974), 702–719.

[7] A. Bermudez, A. Dervieux, J. A. Desideri and M. E. Vasquez, *Upwind schemes for the two-dimensional shallow water equations with variable depth using unstructured meshes*, Comput. Methods Appl. Mech. Engrg., **155** (1998), 49–72.

[8] A. Bermudez, and M. E. Vasquez, *Upwind methods for hyperbolic conservation laws with source terms*, Comput. Fluids, **23** (1994), 1049–1071.

[9] G. Burgers, P. van Leeuwen and G. Evensen, *Analysis scheme in the ensemble Kalman filter*, Monthly Weather Review, **126** (1998), 1719–1724.

[10] A. Chadwick, J. Morfett and M. Borthwick, "Hydraulics in Civil and Environmental Engineering," Spon Press, London, 4th edition, 2004.

[11] C. T. Crowe, D. F. Elger and J. A. Roberson, "Engineering Fluid Mechanics," John Wiley & Sons, Inc, US, 2001.

[12] G. Evensen, "Data Assimilation: The Ensemble Kalman Filter," Springer-Verlag, Berlin Heidelberg, 2007.

[13] J. R. Gunson and P. Malanotte-Rizzoli, *Assimilation studies of open-ocean flows 1. Estimation of initial and boundary conditions*, J. Geophys. Res., **101** (1996), 28457–28472.

[14] J. R. Gunson and P. Malanotte-Rizzoli, *Assimilation studies of open-ocean flows 2. Error measures with strongly nonlinear dynamics*, J. Geophys. Res., **101** (1996), 28473–28488.

[15] A. W. Heemink, M. Verlaan and A. J. Segers, *Variance reduced ensemble Kalman filtering*, Monthly Weather Review, **129** (2001), 1718–1728.

[16] M. Honnorat, J. Monnier and X. Le Dimet, *Lagrangian Data Assimilation for River Hydraulics Simulation*, in "European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006" (eds. E.H. Zarantonello and Author 2), TU Delft, The Netherlands, (2006).

[17] K. Ide, P. Courtier, M. Ghil and A. Lorenc, *Unified notation for data assimilation: Operational, sequential and variational*, J. Met. Soc. Japan, **75** (1997), 181–189.

[18] K. Ide, L. Kuznetsov and C. K. R. T. Jones, *Lagrangian data assimilation for point vortex systems*, Journal of Turbulence, **3** (2002), 53–59.

[19] Y. Ishikawa, T. Awaji, K. Akitomo and B. Qiu, *Successive correction of the mean sea surface height by the simultaneous assimilation of drifting buoy and altimetric data*, J. Phys. Oceanogr., **26** (1996), 2381–2397.

[20] E. Kalnay, "Atmospheric Modeling, Data Assimilation and Predictability," Cambridge University Press, Cambridge, UK, 2003.

[21] L. Kuznetsov, K. Ide and C. K. R. T. Jones, *A method for assimilation of Lagrangian data*, Mon. Wea. Rev., **131** (2003), 2247–2260.

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

[23] A. Molcard, L. I. Piterbarg, A. Griffa, T. Özgökmen and A. Mariano, *Assimilation of drifter observations for the reconstruction of the Eulerian circulation field*, J. Geophys. Res., **108** (2003), 3056.

[24] I. M. Navon, *Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography*, Dyn. Atmos. Oceans, **27** (1997), 55–79.

[25] M. Nodet, *Variational assimilation of Lagrangian Data in oceanography*, Inverse Problems, **22** (2006), 245–263.

[26] C. Paniconi, M. Marrocu, M. Putti and M. Verbunt, *Newtonian nudging for a Richards equation-based distributed hydrological model*, Adv. Water. Resour., **26** (2003), 161–178.

[27] Report EDF, *TELEMAC 2D. Version 5.2 – Principle note*, 2003.

[28] H. Salman, L. Kuznetsov and C. K. R. T. Jones, *A method for assimilating Lagrangian data into a Shallow-water equation ocean model*, Mon. Wea. Rev., **134** (2006), 1081–1101.

[29] A. Seppänen, M. Vauhkonen, E. Somersalo and J. Kaipio, *State space models in process tomography – approximation of state noise covariance*, Inverse Problems in Engineering, **9** (2001), 561–585.

[30] I. Shulman, J. K. Lewis and J. G. Mayer, *Local data assimilation in the estimation of barotropic and baroclinic open boundary conditions*, J. Geophys. Res., **104** (1999), 667–680.

[31] M. E. Vazquez-Cendon, *Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry*, J. Comput. Phys., **148** (1999), 497–526.

[32] C. B. Vreugdenhil, “Numerical Methods for Shallow Water Flow,” Kluwer Academic Publishers, Dordrecht, 1994.

Received November 2008; revised February 2009.

E-mail address: `strub@ce.berkeley.edu`
E-mail address: `julie.percelay@berkeley.edu`
E-mail address: `optossav@berkeley.edu`
E-mail address: `bayen@ce.berkeley.edu`