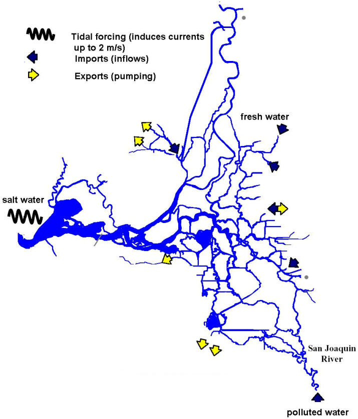
Estimation of State Noise for the Ensemble Kalman filter algorithm for 2D shallow water equations.

Aymeric Mellet and Julie Percelay

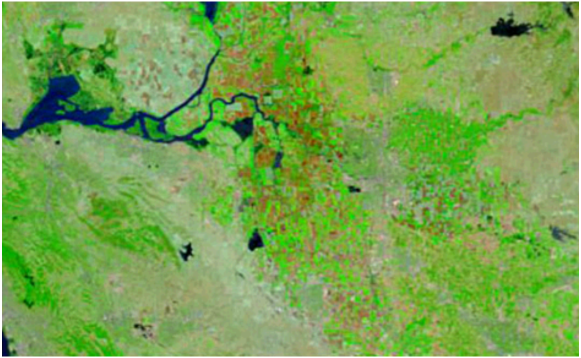
May 6, 2009

Sacramento Delta



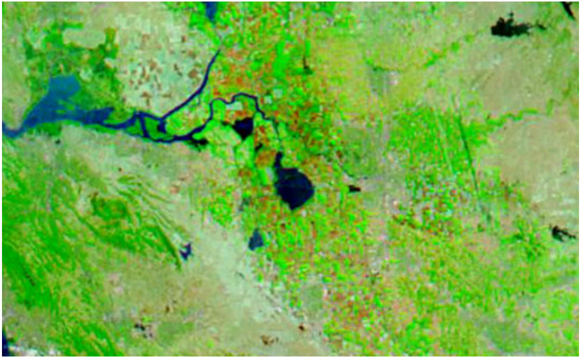
Sacramento Delta

Before the levee breach



Sacramento Delta

After the levee breach



Lagrangian sensors

Lagrangian sensors measure their positions as they drift along with the flow.
When trying to solve the two-dimensional shallow water equations, one needs accurate initial and boundary conditions which are usually not available.
Data assimilation of the Lagrangian measurements from the drifters allow to estimate the velocity field in the river even without accurate boundary conditions.

Two-dimensional shallow water equations

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

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

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

where

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

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

h: total water depth

$\overrightarrow{U} = (u, v)$: velocity field

η: free surface elevation

ν_t: turbulent diffusion coefficient

$\alpha = \alpha(x, y)$: bottom slope

n: Manning coefficient.

Two-dimensional shallow water equations

Boundary conditions:

$$u(x,y,t)\big|_{\partial\Omega_{\mathrm{land}}} = 0, \quad v(x,y,t)\big|_{\partial\Omega_{\mathrm{land}}} = 0, \tag{4}$$

$$(u(x,y,t), v(x,y,t))\big|_{\partial\Omega_{\mathrm{upstream}}} = f(x,y,t), \tag{5}$$

$$\eta(x,y,t)\big|_{\partial\Omega_{\mathrm{downstream}}} = g(x,y,t), \tag{6}$$

Initial conditions:

$$u(x,y,0) = u_0, \quad v(x,y,0) = v_0, \quad h(x,y,0) = h_0, \tag{7}$$

Drifter model

$$\frac{dx_{D_i}(t)}{dt} = u(x_{D_i}(t), y_{D_i}(t), t), \quad (8)$$

$$\frac{dy_{D_i}(t)}{dt} = v(x_{D_i}(t), y_{D_i}(t), t), \quad (9)$$

State Augmentation

We use state augmentation to simplify the observation model

$$\theta_n = \begin{pmatrix} u(t_n) \\ v(t_n) \\ h(t_n) \\ x_D(t_n) \\ y_D(t_n) \end{pmatrix} \tag{10}$$

Forward model

$$\theta_{n+1} = F_n(\theta_n) + w_n. \tag{11}$$

F_n: one time step in the discretized shallow water and drifter model.
θ_{n+1}: predicted system state at time t_{n+1}.
w_n: state noise (modeling error between the reality and the 2D model).
Q_n: covariance of the state noise.

Observation model

$$y_n = C_n \theta_n + \epsilon_n. \tag{12}$$

y_n: measurements from the drifters.
$C_n = (\mathbf{0}\ \mathbb{I})$: observation model.
ϵ_n: measurement noise.
R_n: covariance of the measurement noise.

EnKF Algorithm

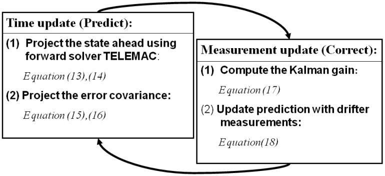
1. Initialization: An ensemble of N_{states} states $\xi_0^{(p)}$ indexed by p are generated to represent the uncertainty in θ_0.

2. Time update:

$$\xi_{n|n-1}^{(p)} = F_n(\xi_{n-1|n-1}^{(p)}) + w_{n-1}^{(p)} \quad (13)$$

$$\theta_{n|n-1} = \frac{1}{N_{\text{states}}} \sum_{p=1}^{N_{\text{states}}} \xi_{n|n-1}^{(p)} \quad (14)$$

$$E_{n|n-1} = [\xi_{n|n-1}^{(1)} - \theta_{n|n-1}, \dots, \xi_{n|n-1}^{(N_{\text{states}})} - \theta_{n|n-1}] \quad (15)$$



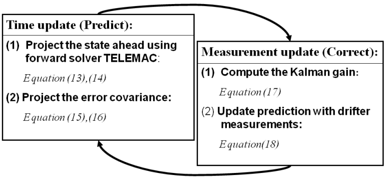
EnKF Algorithm

3. Measurement update:

$$\Gamma_{n|n-1} = \frac{1}{N_{\text{states}} - 1} E_{n|n-1} E_{n|n-1}^{\mathrm{T}} \quad (16)$$

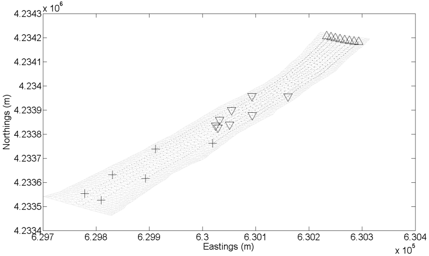
$$K_n = \Gamma_{n|n-1} C_n^{\mathrm{T}} (C_n \Gamma_{n|n-1} C_n^{\mathrm{T}} + R_n)^{-1} \quad (17)$$

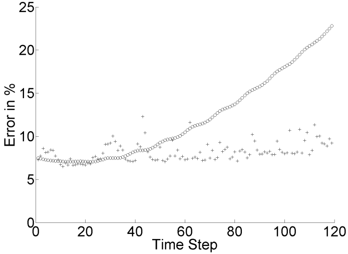
$$\xi_{n|n}^{(p)} = \xi_{n|n-1}^{(i)} + K_n (y_n - C_n \xi_{n|n-1}^{(p)} + \epsilon_n^{(p)}) \quad (18)$$



Twin experiment results

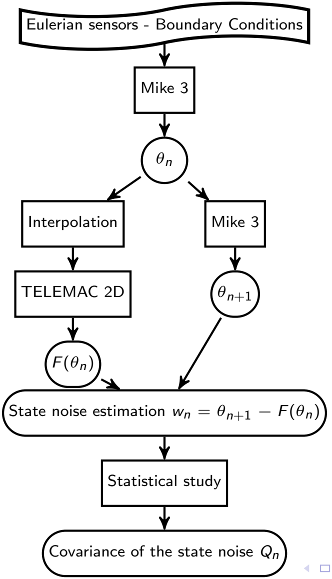
Drifters are released using a software and the EnKF algorithm runs using those measurements.





Limitations

The state noise used here is realistic for twin experiments, since we know the discrepancy between the "reality" and the model, as they are both from the same software.
We need more realistic state noise in order to have the algorithm run with the real data from November experiment.


Eulerian sensors - Boundary Conditions
Mike 3
θ_n
Interpolation
Mike 3
TELEMAC 2D
θ_{n+1}
$F(\theta_n)$
State noise estimation $w_n = \theta_{n+1} - F(\theta_n)$
Statistical study
Covariance of the state noise Q_n

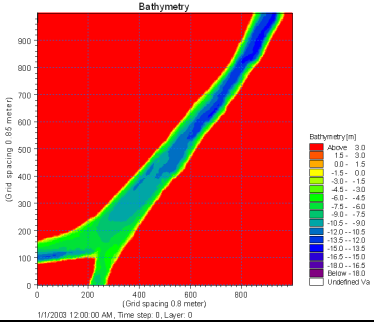
Navier-Stokes equations

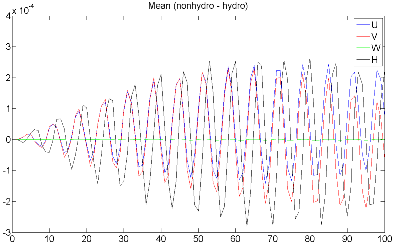
$$\frac{1}{\rho c_s^2}\frac{\partial P}{\partial t} + \frac{\partial u_j}{\partial x_j} = SS \tag{19}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} + 2\Omega_{ij} u_j = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + g_i$$
$$+\frac{\partial}{\partial x_j}(\nu_T\{\frac{\partial u_i}{\partial x_i} + \frac{\partial u_j}{\partial x_j}\} - \frac{2}{3}\delta_{ij} k) + u_i SS \tag{20}$$

Testing the hydrostatic assumption

Hydrostatic assumption: pressure in the river can be estimated using the hydrostatic pressure.
Valid for $\frac{L}{H} >> 1$, where L is the horizontal characteristic length of the river.





Navier-Stokes equations using hydrostatic assumption

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{21}$$

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial vu}{\partial y} + \frac{\partial wu}{\partial z} = -g\frac{\partial \eta}{\partial x} - \frac{1}{\rho_0}\frac{\partial p_a}{\partial x} - \frac{g}{\rho_0}\int_z^\eta \frac{\partial \rho}{\partial x} dz - \frac{1}{\rho_0 h}\left(\frac{\partial s_{xx}}{\partial x} + \frac{\partial s_{xy}}{\partial y}\right) + F_u + \frac{\partial}{\partial z}\left(\nu_t \frac{\partial u}{\partial z}\right) \tag{22}$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial wv}{\partial z} = -g\frac{\partial \eta}{\partial y} - \frac{1}{\rho_0}\frac{\partial p_a}{\partial y} - \frac{g}{\rho_0}\int_z^\eta \frac{\partial \rho}{\partial y} dz - \frac{1}{\rho_0 h}\left(\frac{\partial s_{yx}}{\partial x} + \frac{\partial s_{xy}}{\partial y}\right) + F_v + \frac{\partial}{\partial z}\left(\nu_t \frac{\partial v}{\partial z}\right) \tag{23}$$

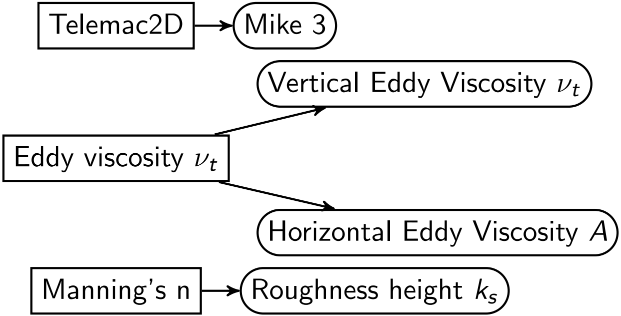
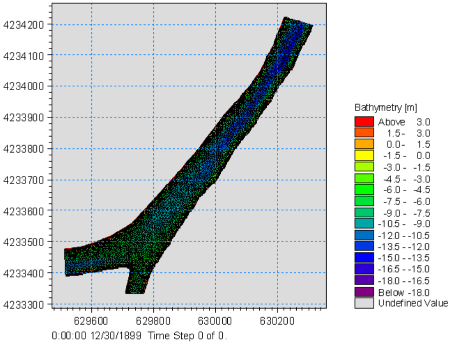


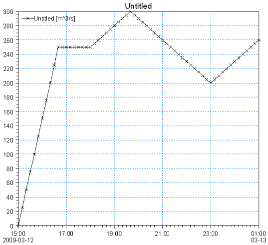
Figure: Equivalence of parameters between the two dimensional and the three dimensional models.

A relationship between the roughness height and the Manning's n can be found as:

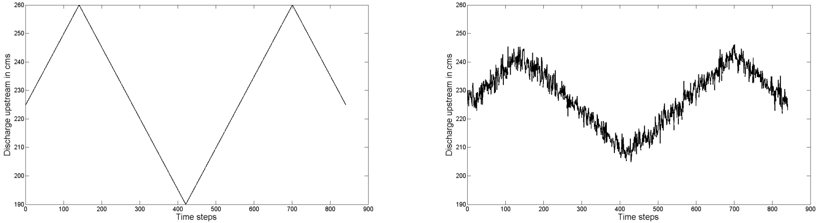
$$n = \frac{k_s^{\frac{1}{6}}}{25.4} \tag{24}$$

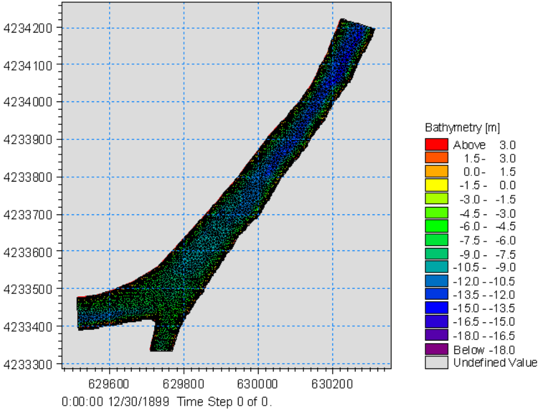
Three dimensional simulations



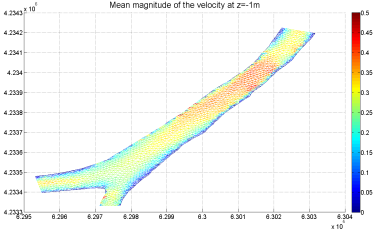


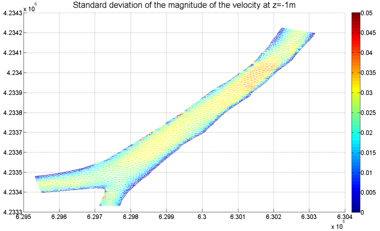
Boundary conditions



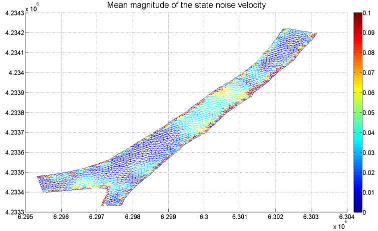

4234200
4234100
4234000
4233900
4233800
4233700
4233600
4233500
4233400
4233300
629600
629800
630000
630200
0:00:00 12/30/1899 Time Step 0 of 0.
Bathymetry [m]
Above 3.0
1.5 - 3.0
0.0 - 1.5
-1.5 - 0.0
-3.0 - -1.5
-4.5 - -3.0
-6.0 - -4.5
-7.5 - -6.0
-9.0 - -7.5
-10.5 - -9.0
-12.0 - -10.5
-13.5 - -12.0
-15.0 - -13.5
-16.5 - -15.0
-18.0 - -16.5
Below -18.0
Undefined Value

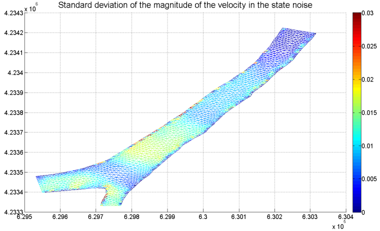
Result of the 3D simulation





State noise characteristics





Real data

Cross validation principle

No data are available to validate.

- Remove 1 or 2 drifters from the data set.
- Run the EnKF on the remaining drifters.
- Compute the trajectory of the extracted drifters using the estimated velocity field.
- Compare it with the real trajectory.

Future work

- Run the EnKF using the real data set and the generated state noise.
- Generate a state noise model for the drifters.

Questions?