Modelling of a river system with multiple reaches

Maarten Breckpot * **Oscar Mauricio Agudelo** *,** **Bart De Moor** *

* *Electrical Engineering Department (SISTA-SCD), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium* {*maarten.breckpot,mauricio.agudelo,bart.demoor*}*@esat.kuleuven.be*
** *Dept. of Automation and Electronics, Universidad Autónoma de Occidente, Calle 25, No 115, 85, Cali, Colombia*

Abstract: In this paper we present a new approach to model river systems. In general the dynamics of a single reach can be described with the Saint-Venant equations. These equations can be combined with nonlinear gate equations to fully characterize the behavior of river systems with multiple reaches. Simulating the dynamics of a river system while taking all these nonlinearities into account, can take a lot of time. The complete linearization of these equations can drastically decrease the computational burden on one hand, but on the other hand it can compromise the accuracy of the results. Therefore in this paper we propose to combine the linear version of the Saint-Venant equations with the nonlinear gate equations in order to reduce the computational load while generating accurate results. In addition, we show that the use of Proper Orthogonal Decomposition (POD) and Galerkin Projection can lead to an extra computational saving.

Keywords: Modeling; Model approximation; Model reduction; Partial differential equations

1. INTRODUCTION

The dynamics of river systems are described by nonlinear partial differential equations (PDE) in combination with nonlinear algebraic equations (for the hydraulic structures). The standard way of simulating the full dynamics of these systems is by using the method of characteristics, the Preissmann scheme or a variation of the Preissmann scheme (Chaudry [2008], Strelkoff and Falvey [1993], Xu et al. [2011]). Each has its own advantages and disadvantages but all of them can take a lot of computation time. This paper proposes a new method to drastically reduce the simulation time while still achieving accurate results. The key idea is to work with a linearized model for the reach dynamics while working with the full nonlinear gate equations to connect these reaches. An additional reduction is achieved by reducing the number of states of the river system with Proper Orthogonal Decomposition (POD) and Galerking Projection.

The paper has the following structure. Section 2 describes how the dynamics of a single reach and gates can be modelled, together with a numerical scheme to simulate these dynamics with high accuracy. Section 3 briefly discusses the derivation of a linear model. Section 4 explains the underlying idea of our proposed model: the combination of linear reach models together with nonlinear gate equations. POD is used in Section 5 to reduce the states of the model derived in Section 4. Section 6 compares the performance of the different models. This paper ends with some conclusions in Section 7.

2. NONLINEAR MODEL

2.1 Single reach dynamics

Saint-Venant equations The Saint-Venant equations are often used in practice to model the evolution of water levels and discharges in river reaches. The equations are valid under certain assumptions which can be found, e.g., in Chaudry [2008]. Given these assumptions the dynamics of the water levels and discharges in a reach can be described by the following equations:

$$\frac{\partial A}{\partial h}\frac{\partial h}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\frac{Q^2}{A} + gA\frac{\partial h}{\partial x} + gA(S_\mathrm{f} - S_0) = 0 \tag{2}$$

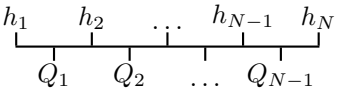
with A the cross-sectional flow area (m^2), Q the water discharge (m^3/s), h the water depth (m), g the gravity acceleration (m/s^2), S_0 the bed slope and S_f the friction slope. Equation (1) describes the conservation of mass and (2) the conservation of momentum. S_f is taken to represent any number of empirical resistance laws. The resistance law which is used in this study is the Manning relation:

$$S_\mathrm{f} = \frac{n_\mathrm{mann}^2 Q|Q|}{A^2 R^{1/3}} \tag{3}$$

where n_mann is the Manning coefficient ($\mathrm{s}/\mathrm{m}^{1/3}$), $R = A/P$ is the hydraulic radius (m) and P is the wetted perimeter of the cross section (m).

Numerical scheme Our way of simulating (1) and (2) is based on Strelkoff and Falvey [1993] and Xu et al. [2011].

In this method the time integration is based on the θ-method, e.g., $f(t_j+\theta\Delta t)=\theta f(t_j+\Delta t)+(1-\theta)f(t_j)$ with $\theta\in[0,1]$. The approximation of the partial derivatives is based on finite differences. The grid used in this paper is a staggered grid:



where $2N-1$ is the total number of grid points (N water levels and $N-1$ discharges). The derivatives in (1) are approximated by (note $h(x_i,t_j)=h_i^j$)

$$\frac{\partial h_i^j}{\partial t}\simeq\frac{h_i^{j+1}-h_i^j}{\Delta t}, \tag{4}$$

$$\frac{\partial Q_i^j}{\partial x}\simeq\frac{\theta(Q_i^{j+1}-Q_{i-1}^{j+1})+(1-\theta)(Q_i^j-Q_{i-1}^j)}{\Delta x} \tag{5}$$

$$\frac{\partial A_i^j}{\partial h}\simeq\theta\frac{\partial A_i^{j+1}}{\partial h}+(1-\theta)\frac{\partial A_i^j}{\partial h}. \tag{6}$$

A similar approach is used for the terms $\partial Q/\partial t$, A, $\partial h/\partial x$ and S_f in (2). The advection term $\partial(Q^2/A)/\partial x$ is approximated with an upwinding approach:

$$\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)_i^j\simeq\begin{cases}\frac{\theta}{\Delta x}\omega\left(\frac{Q^2}{A},i+1,j+1\right)+ & Q_i^j<0\\ +\frac{1-\theta}{\Delta x}\omega\left(\frac{Q^2}{A},i+1,j\right), & \\ \frac{\theta}{\Delta x}\omega\left(\frac{Q^2}{A},i,j+1\right)+ & Q_i^j\geq 0\\ +\frac{1-\theta}{\Delta x}\omega\left(\frac{Q^2}{A},i,j\right) & \end{cases} \tag{7}$$

with

$$\omega\left(\frac{Q^2}{A},i,j\right)=\left(\frac{Q^2}{A}\right)_i^j-\left(\frac{Q^2}{A}\right)_{i-1}^j. \tag{8}$$

In this way the two PDE's are transformed into a system of nonlinear equations:

$$f\left(h_i^{j+1},h_i^j,Q_{i-1}^{j+1},Q_i^{j+1},Q_{i-1}^j,Q_i^j\right)=0,\quad\forall i=2,\ldots,N-1, \tag{9}$$

$$g\left(h_{i-1}^{j+1},\ldots,h_{i+2}^{j+1},h_{i-1}^j,\ldots,h_{i+2}^j,Q_{i-1}^{j+1},\ldots,Q_{i+1}^{j+1},Q_{i-1}^j,\ldots,Q_{i+1}^j\right)=0,\forall i=2,\ldots,N-2, \tag{10}$$

which has to be solved for $h_1^{j+1},h_2^{j+1},\ldots,h_N^{j+1}$ and $Q_1^{j+1},Q_2^{j+1},\ldots,Q_{N-1}^{j+1}$. To be able to solve this system, upstream and downstream boundary conditions are needed. These boundary conditions can be a given predefined discharge or an hydraulic structure between two connected reaches. Together with these boundary conditions, (9) and (10) can be solved with Newton's method.

A discussion about the choice of Δt and θ can be found in Clemmens et al. [2005]. However we will not elaborate on this in this paper.

2.2 Gate equations

The gates used in this paper are underflow-vertical sluices. They have been extensively described and modelled in the literature. The general equation is as follows:

$$Q=z(C_\mathrm{D},W,c,h_\mathrm{up},h_\mathrm{down}) \tag{11}$$

with Q the discharge through the gate, C_D the discharge coefficient, W the width of the gate (m), c the gate opening (m), h_up the upstream water level and h_down the downstream water level. The equation used in this study has the following form:

$$Q=C_\mathrm{D}Wc\sqrt{2gh_\mathrm{up}} \tag{12}$$

with different equations for C_D depending on the flow condition (Sepúlveda et al. [2009] and Lin et al. [2002]):

free flow: $$C_\mathrm{D}=\frac{C_\mathrm{C}}{\sqrt{1+\eta}}, \tag{13}$$

submerged flow:

$$C_\mathrm{D}=C_\mathrm{C}\frac{\left[\xi-\sqrt{\xi^2-\left(\frac{1}{\eta^2}-1\right)^2\left(1-\frac{1}{\lambda^2}\right)}\right]^{1/2}}{\frac{1}{\eta}-\eta}. \tag{14}$$

Here C_C is the contraction coefficient, $\eta=C_\mathrm{C}c/h_\mathrm{up}$, $\lambda=h_\mathrm{up}/h_\mathrm{down}$ and $\xi=(1/\eta-1)^2+2(\lambda-1)$. C_C can vary from 0.598 to 0.74 but for most engineering applications a value of 0.611 is usually taken (Henderson [1966], Liggett and Cunge [1975]). The gate is considered to be in free flow if h_down is below the limit $h_\mathrm{down,max}$ given by (Sepúlveda et al. [2009] and Lin et al. [2002]):

$$h_\mathrm{down,max}=\frac{C_\mathrm{C}c}{2}\left(\sqrt{1+\frac{16}{\eta(1+\eta)}}\right). \tag{15}$$

Otherwise the gate is in submerged flow condition.

3. LINEAR MODEL

A straightforward way to decrease the computational burden of the nonlinear model is using a linear approximation of it. In this way only a limited number of matrix-vector operations are needed to calculate the next state of the river system given its current state and the boundary conditions. When linearizing the nonlinear equations (9), (10) and (12) around a nominal operating point ($\boldsymbol{h}_\mathrm{ss}\in\mathbb{R}^N$ for the water levels, $\boldsymbol{Q}_\mathrm{ss}\in\mathbb{R}^{N-1}$ for the discharges and $\boldsymbol{u}_\mathrm{ss}\in\mathbb{R}^{n_\mathrm{u}}$ for the inputs with n_u the number of inputs), the following linear model can be found:

$$\begin{pmatrix}\Delta\boldsymbol{h}(k+1)\\ \Delta\boldsymbol{Q}(k+1)\end{pmatrix}=\bar{\boldsymbol{A}}\begin{pmatrix}\Delta\boldsymbol{h}(k)\\ \Delta\boldsymbol{Q}(k)\end{pmatrix}+\bar{\boldsymbol{B}}_1\Delta\boldsymbol{u}(k)+\bar{\boldsymbol{B}}_2\Delta\boldsymbol{u}(k+1) \tag{16}$$

with $\Delta\boldsymbol{h}(k)=\boldsymbol{h}(k)-\boldsymbol{h}_\mathrm{ss}$, $\Delta\boldsymbol{Q}(k)=\boldsymbol{Q}(k)-\boldsymbol{Q}_\mathrm{ss}$, $\Delta\boldsymbol{u}(k)=\boldsymbol{u}(k)-\boldsymbol{u}_\mathrm{ss}$, $\bar{\boldsymbol{A}}\in\mathbb{R}^{(2N-1)\times(2N-1)}$, $\bar{\boldsymbol{B}}_1\in\mathbb{R}^{(2N-1)\times n_\mathrm{u}}$ and $\bar{\boldsymbol{B}}_2\in\mathbb{R}^{(2N-1)\times n_u}$. The input vector $\boldsymbol{u}$ contains the upstream/downstream river discharges together with the gate positions: $\boldsymbol{u}=(Q_\mathrm{up},c_1,c_2,\ldots,c_{n_u-2},Q_\mathrm{down})^T$. Note that $\Delta\boldsymbol{u}(k+1)$ is needed to predict $\Delta\boldsymbol{h}(k+1)$ and $\Delta\boldsymbol{Q}(k+1)$ since we work with a linearized version of an implicit nonlinear scheme.

4. LINEAR-NONLINEAR MODEL

If the river system consists of just one reach without gates as boundary conditions, then the linear model provides a good approximation of the system dynamics. However if there are gates present, then the performance of the

linear model degrades if the gates do no stay close to their nominal operating point. This can be observed in the simulation results in Section 6.

A solution for this problem is to work with a combination of the linearized version of the Saint-Venant equations of each reach together with the nonlinear gate equations. This means that the effect of the gates on the water levels and the discharges is "pulled" out of the linear model. From now on this system will be referred to as the LN model. The linear part of the LN model has the following form:

$$\begin{pmatrix} \Delta \boldsymbol{h}(k+1) \\ \Delta \boldsymbol{Q}(k+1) \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} \Delta \boldsymbol{h}(k) \\ \Delta \boldsymbol{Q}(k) \end{pmatrix} + \boldsymbol{B}_1 \Delta \boldsymbol{u}(k) + \\ + \boldsymbol{B}_2 \Delta \boldsymbol{u}(k+1) \quad (17)$$

with $\boldsymbol{A} \in \mathbb{R}^{(2N-1)\times(2N-1)}$, $\boldsymbol{B}_1 \in \mathbb{R}^{(2N-1)\times n_\mathrm{u}}$ and $\boldsymbol{B}_2 \in \mathbb{R}^{2N-1\times n_\mathrm{u}}$. The main difference with (16) is that $\boldsymbol{u}$ now contains the gate discharges instead of the gate openings: $\boldsymbol{u} = \left(Q_\mathrm{up}, Q_{\mathrm{gate}_1}, \ldots, Q_{\mathrm{gate}_{n_u-2}}, Q_\mathrm{down}\right)^T$. Hence, the only extra work is to make the transformation from the gate openings $\boldsymbol{c}(k) = (c_1, \ldots, c_{n_u-2})^T$ and $\boldsymbol{c}(k+1)$ to the discharges over the gates $\boldsymbol{Q}_\mathrm{gate}(k) = \left(Q_{\mathrm{gate}_1}, \ldots, Q_{\mathrm{gate}_{n_u-2}}\right)^T$ and $\boldsymbol{Q}_\mathrm{gate}(k+1)$. At time k the current upstream and downstream water levels for each hydraulic structure are known. Given these water levels together with the current gate opening $\boldsymbol{c}(k)$, $\boldsymbol{Q}_\mathrm{gate}(k)$ can be computed from (12) - (14). The only difficulty lies in making the transformation from $\boldsymbol{c}(k+1)$ to $\boldsymbol{Q}_\mathrm{gate}(k+1)$ since the water levels $\boldsymbol{h}(k+1)$ are unknown. A solution is to work in an iterative way (Algorithm 1). The water levels $\boldsymbol{h}(k)$ are used in combination with $\boldsymbol{c}(k+1)$ to estimate $\boldsymbol{Q}_\mathrm{gate}(k+1)$. With this estimate of $\boldsymbol{u}(k+1)$ we can calculate $\boldsymbol{h}(k+1)$ and $\boldsymbol{Q}(k+1)$. $\boldsymbol{Q}_\mathrm{gate}(k+1)$ can now be refined by using $\boldsymbol{h}(k+1)$ and $\boldsymbol{c}(k+1)$, etc.

Algorithm 1 Simulation of the LN model

for k = 1,...
  $\boldsymbol{h}(k+1) = \Delta\boldsymbol{h}(k) + \boldsymbol{h}_\mathrm{ss}$;
  for i = 1,... until convergence
    for each gate j
      select h_up and h_down out of $\boldsymbol{h}(k+1)$;
      use (12) to calculate $Q_{\mathrm{gate}_j}(k+1)$ given $c_j(k+1)$, h_up and h_down and build up $\Delta\boldsymbol{u}(k+1)$;
    end
    $\begin{pmatrix} \Delta \boldsymbol{h}(k+1) \\ \Delta \boldsymbol{Q}(k+1) \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} \Delta \boldsymbol{h}(k) \\ \Delta \boldsymbol{Q}(k) \end{pmatrix} + \boldsymbol{B}_1 \Delta \boldsymbol{u}(k) + \\ + \boldsymbol{B}_2 \Delta \boldsymbol{u}(k+1)$;
  end
end

5. POD BASED LN MODEL

Notice that the LN model has the same (high) number of states as the nonlinear model. An extra gain in computation time can be obtained by reducing the number of states. One method successfully used in many applications is Proper Orthogonal Decomposition (POD). POD is a data-driven method where a suitable set of ordered orthonormal basis vectors are derived from simulation or experimental data. Reduced-order models are typically found by projecting (Galerkin projection) the full-order models on the most relevant basis vectors (Astrid [2004], Ravindran [2000]).

In POD, we start by observing that the vectors $\Delta\boldsymbol{h}(k)$ and $\Delta\boldsymbol{Q}(k)$ can be rewritten as a sum of orthonormal basis vectors

$$\Delta\boldsymbol{h}(k) = \sum_{j=1}^{N} a_j(k)\tilde{\boldsymbol{\varphi}}_j,\ \Delta\boldsymbol{Q}(k) = \sum_{j=1}^{N-1} b_j(k)\hat{\boldsymbol{\varphi}}_j \quad (18)$$

with $\tilde{\boldsymbol{\varphi}}_j \in \mathbb{R}^N, \forall j = 1, \ldots, N$ and $\hat{\boldsymbol{\varphi}}_j \in \mathbb{R}^{N-1}, \forall j = 1, \ldots, N-1$ two sets of orthonormal basis vectors with corresponding time-varying coefficients $a_j(k) \in \mathbb{R}, \forall j = 1, \ldots, N$, and $b_j(k) \in \mathbb{R}, \forall j = 1, \ldots, N-1$ respectively.

The main dynamics of the system can be described with the first n_h and n_Q most relevant basis vectors, since they contain the main spatial correlations. Hence, the n_hth order approximation of $\Delta\boldsymbol{h}(k)$ and the n_Qth order approximation of $\Delta\boldsymbol{Q}(k)$ are given by

$$\begin{aligned} \Delta\boldsymbol{h}_{n_\mathrm{h}}(k) &= \sum_{j=1}^{n_\mathrm{h}} a_j(k)\tilde{\boldsymbol{\varphi}}_j = \tilde{\boldsymbol{\Phi}}_{n_\mathrm{h}}\boldsymbol{a}(k),\ n_\mathrm{h} \ll N, \\ \Delta\boldsymbol{Q}_{n_\mathrm{Q}}(k) &= \sum_{j=1}^{n_\mathrm{Q}} b_j(k)\hat{\boldsymbol{\varphi}}_j = \hat{\boldsymbol{\Phi}}_{n_\mathrm{Q}}\boldsymbol{b}(k),\ n_\mathrm{Q} \ll N-1, \end{aligned} \quad (19)$$

where $\tilde{\boldsymbol{\Phi}}_{n_\mathrm{h}} = \left(\tilde{\boldsymbol{\varphi}}_1, \ldots, \tilde{\boldsymbol{\varphi}}_{n_\mathrm{h}}\right)$, $\boldsymbol{a}(k) = \left(a_1(k), \ldots, a_{n_\mathrm{h}}(k)\right)^T$, $\hat{\boldsymbol{\Phi}}_{n_\mathrm{Q}} = \left(\hat{\boldsymbol{\varphi}}_1, \ldots, \hat{\boldsymbol{\varphi}}_{n_\mathrm{Q}}\right)$ and $\boldsymbol{b}(k) = \left(b_1(k), \ldots, b_{n_\mathrm{Q}}(k)\right)^T$.

To derive the POD basis vectors, simulation data of the linear model is gathered in the snapshot matrices $\tilde{\boldsymbol{X}} \in \mathbb{R}^{N\times M}$ and $\hat{\boldsymbol{X}} \in \mathbb{R}^{N-1\times M}$ with M the number of time samples. The POD basis vectors are found by calculating the singular value decomposition (SVD) of the snapshot matrices:

$$\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{\Psi}}^T,\ \hat{\boldsymbol{X}} = \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Psi}}^T \quad (20)$$

with $\tilde{\boldsymbol{\Phi}} \in \mathbb{R}^{N\times N}$, $\tilde{\boldsymbol{\Psi}} \in \mathbb{R}^{M\times M}$, $\hat{\boldsymbol{\Phi}} \in \mathbb{R}^{(N-1)\times(N-1)}$ and $\hat{\boldsymbol{\Psi}} \in \mathbb{R}^{M\times M}$ unitary matrices and $\tilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{N\times M}$ and $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{(N-1)\times M}$ diagonal matrices containing the singular values of $\tilde{\boldsymbol{X}}$ and $\hat{\boldsymbol{X}}$ respectively. The POD basis vectors are the columns of $\tilde{\boldsymbol{\Phi}}$ and $\hat{\boldsymbol{\Phi}}$:

$$\tilde{\boldsymbol{\Phi}} = \left(\tilde{\boldsymbol{\varphi}}_1, \ldots, \tilde{\boldsymbol{\varphi}}_N\right),\ \hat{\boldsymbol{\Phi}} = \left(\hat{\boldsymbol{\varphi}}_1, \ldots, \hat{\boldsymbol{\varphi}}_{N-1}\right). \quad (21)$$

Each singular value indicates the importance of the corresponding POD basis vector. By using the Galerkin projection we can derive a dynamical model for $\boldsymbol{a}(k)$ and $\boldsymbol{b}(k)$, that is, a reduced-order model for (17).

The state vector (17) can be written down as follows:

$$\Delta\boldsymbol{x}(k) = \begin{pmatrix} \Delta \boldsymbol{h}(k) \\ \Delta \boldsymbol{Q}(k) \end{pmatrix}. \quad (22)$$

Its nth order approximation ($n = n_\mathrm{h} + n_\mathrm{Q}$) will be given by

$$\Delta\boldsymbol{x}_n(k) = \begin{pmatrix} \Delta \boldsymbol{h}_{n_\mathrm{h}}(k) \\ \Delta \boldsymbol{Q}_{n_\mathrm{Q}}(k) \end{pmatrix} = \boldsymbol{\Phi}_n \begin{pmatrix} \boldsymbol{a}(k) \\ \boldsymbol{b}(k) \end{pmatrix} \quad (23)$$

with

$$\boldsymbol{\Phi}_n = \begin{pmatrix} \tilde{\boldsymbol{\Phi}}_{n_\mathrm{h}} & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{\Phi}}_{n_\mathrm{Q}} \end{pmatrix}. \quad (24)$$

If we define a residual function for (17) as follows:

$$R\left(\Delta \boldsymbol{x}(k+1), \Delta \boldsymbol{x}(k)\right) = \Delta \boldsymbol{x}(k+1) - \boldsymbol{A}\Delta \boldsymbol{x}(k) - \boldsymbol{B}_1 \Delta \boldsymbol{u}(k) - \boldsymbol{B}_2 \Delta \boldsymbol{u}(k+1) \quad (25)$$

and we replace $\Delta \boldsymbol{x}(k)$ and $\Delta \boldsymbol{x}(k+1)$ by their nth order approximations, the Galerkin projection states that the projection of $R(\Delta \boldsymbol{x}(k+1), \Delta \boldsymbol{x}(k))$ on the space spanned by the basis vectors $\boldsymbol{\Phi}_n$ vanishes:

$$\langle R(\Delta \boldsymbol{x}_n(k+1), \Delta \boldsymbol{x}_n(k)), \boldsymbol{\varphi}_j \rangle = 0, \quad \forall j = 1, \ldots, n_\mathrm{h} + n_\mathrm{Q} \quad (26)$$

with $\langle ., . \rangle$ the inner product and $\boldsymbol{\varphi}_j$ the jth column of $\boldsymbol{\Phi}_n$. Equation (26) can be rewritten as

$$\left\langle \boldsymbol{\Phi}_n \begin{pmatrix} \boldsymbol{a}(k+1) \\ \boldsymbol{b}(k+1) \end{pmatrix}, \boldsymbol{\Phi}_n \right\rangle = \left\langle \boldsymbol{A}\boldsymbol{\Phi}_n \begin{pmatrix} \boldsymbol{a}(k) \\ \boldsymbol{b}(k) \end{pmatrix} + \boldsymbol{B}_1 \Delta \boldsymbol{u}(k) + \boldsymbol{B}_2 \Delta \boldsymbol{u}(k+1), \boldsymbol{\Phi}_n \right\rangle. \quad (27)$$

By evaluating the inner product in (27), we get the reduced-order model of the river system,

$$\begin{pmatrix} \boldsymbol{a}(k+1) \\ \boldsymbol{b}(k+1) \end{pmatrix} = \boldsymbol{A}_\mathrm{r} \begin{pmatrix} \boldsymbol{a}(k) \\ \boldsymbol{b}(k) \end{pmatrix} + \boldsymbol{B}_{1,\mathrm{r}} \Delta \boldsymbol{u}(k) + \boldsymbol{B}_{2,\mathrm{r}} \Delta \boldsymbol{u}(k+1) \quad (28)$$

with $\boldsymbol{A}_\mathrm{r} = \boldsymbol{\Phi}_n^T \boldsymbol{A} \boldsymbol{\Phi}_n \in \mathbb{R}^{(n_\mathrm{h}+n_\mathrm{Q})\times(n_\mathrm{h}+n_\mathrm{Q})}$, $\boldsymbol{B}_{1,\mathrm{r}} = \boldsymbol{\Phi}_n^T \boldsymbol{B}_1 \in \mathbb{R}^{(n_\mathrm{h}+n_\mathrm{Q})\times n_\mathrm{u}}$ and $\boldsymbol{B}_{2,\mathrm{r}} = \boldsymbol{\Phi}_n^T \boldsymbol{B}_2 \in \mathbb{R}^{(n_\mathrm{h}+n_\mathrm{Q})\times n_\mathrm{u}}$.

The steps for simulating the POD model are described in Algorithm 2.

Algorithm 2 Simulation of the POD based LN model

build the snapshot matrices $\tilde{\boldsymbol{X}}$ and $\hat{\boldsymbol{X}}$;
use SVD to find $\tilde{\boldsymbol{\Phi}}_{n_\mathrm{h}}$ and $\hat{\boldsymbol{\Phi}}_{n_\mathrm{Q}}$ and construct $\boldsymbol{\Phi}_n$;
$\begin{pmatrix} \boldsymbol{a}(0) \\ \boldsymbol{b}(0) \end{pmatrix} = \boldsymbol{\Phi}_n \left(\begin{pmatrix} \boldsymbol{h}(0) \\ \boldsymbol{Q}(0) \end{pmatrix} - \begin{pmatrix} \boldsymbol{h}_\mathrm{ss} \\ \boldsymbol{Q}_\mathrm{ss} \end{pmatrix} \right)$;
for k = 1,...
  $\boldsymbol{h}_{n_\mathrm{h}}(k+1) = \tilde{\boldsymbol{\Phi}}_{n_\mathrm{h}}^T \boldsymbol{a}(k) + \boldsymbol{h}_\mathrm{ss}$;
  for i = 1,... until convergence
    for each gate j
      select h_up and h_down out of $\boldsymbol{h}_n(k+1)$;
      use (12) to calculate $Q_{\mathrm{gate}_j}(k+1)$ given $c_j(k+1)$, h_up and h_down and build up $\Delta \boldsymbol{u}(k+1)$;
    end
    $\begin{pmatrix} \boldsymbol{a}(k+1) \\ \boldsymbol{b}(k+1) \end{pmatrix} = \boldsymbol{A}_\mathrm{r} \begin{pmatrix} \boldsymbol{a}(k) \\ \boldsymbol{b}(k) \end{pmatrix} + \boldsymbol{B}_{1,\mathrm{r}} \Delta \boldsymbol{u}(k) + \boldsymbol{B}_{2,\mathrm{r}} \Delta \boldsymbol{u}(k+1)$;
    $\boldsymbol{h}_{n_\mathrm{h}}(k+1) = \hat{\boldsymbol{\Phi}}_{n_\mathrm{h}}^T \boldsymbol{a}(k+1) + \boldsymbol{h}_\mathrm{ss}$;
  end
end

6. SIMULATION RESULTS

The different models presented in this paper are tested on a river system consisting of two equal rectangular reaches connected with one gate. Fig. 1 shows the river system together with the meaning of some parameters. Table 1 contains the values of the parameters of the reaches and the gate. The upstream and downstream discharges of the river system are considered given, as well as the evolution of the gate positions (see Fig. 2). The POD basis vectors of the reduced-order LN model are derived from the system response of the linearized reach dynamics when applying step changes to the boundary conditions of each reach (166 time samples were gathered). The first 12 basis vectors of $\tilde{\boldsymbol{\Phi}}$ and the first 20 basis vectors of $\hat{\boldsymbol{\Phi}}$ were selected to build the matrix $\boldsymbol{\Phi}_n$. This means a reduction from 1602 states to 64 states. A sampling time Δt of 60 s was used for the linear and the LN models.

Fig. 3 shows the water levels obtained with the models described in the previous sections at different time instants. The results for the nonlinear model can be taken as the "truth". We can see that in general the performance of the linear model is worse than the performance of the LN model and the POD based LN model. The right figure at the bottom shows that there is a significant steady state error in the water levels for the linear model. This is not the case for the LN models. The figure also shows that the model reduction with POD has almost no influence on

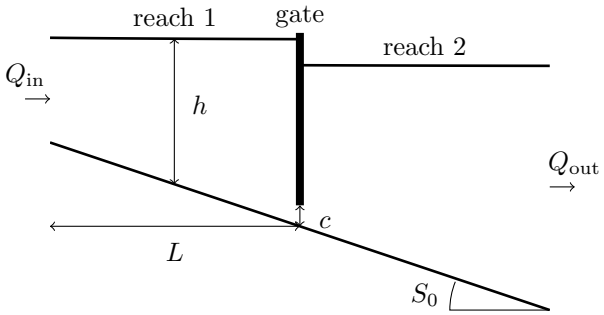


Fig. 1. Schematic structure of the river system with Q_in and Q_out the discharges at the boundaries, c the gate position, h the water levels measured from the bottom of the reaches, L the length of each reach and S_0 the channel slope.

parameters	values
N	401
S_0	0.001
reach length L	4000 m
n_mann	0.015 s/m$^{1/3}$
bottom width B	6.1 m
Q_ss	20 m^3/s
$h_\mathrm{ss,up}$	5 m
gate width W	6.1 m

Table 1. Parameter values of the river reaches and the gate.

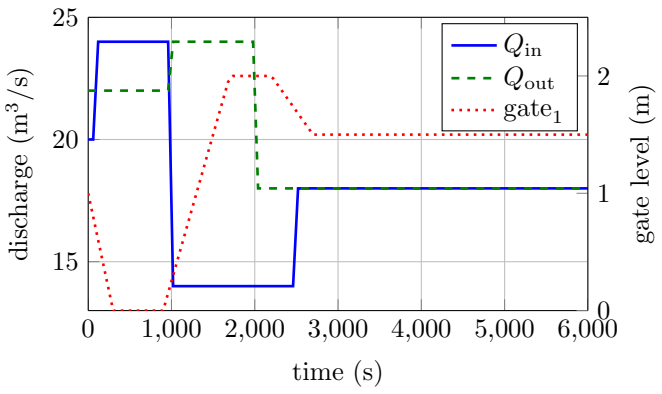


Fig. 2. Evolution of the upstream and downstream discharges together with the gate position for the simulation test case.

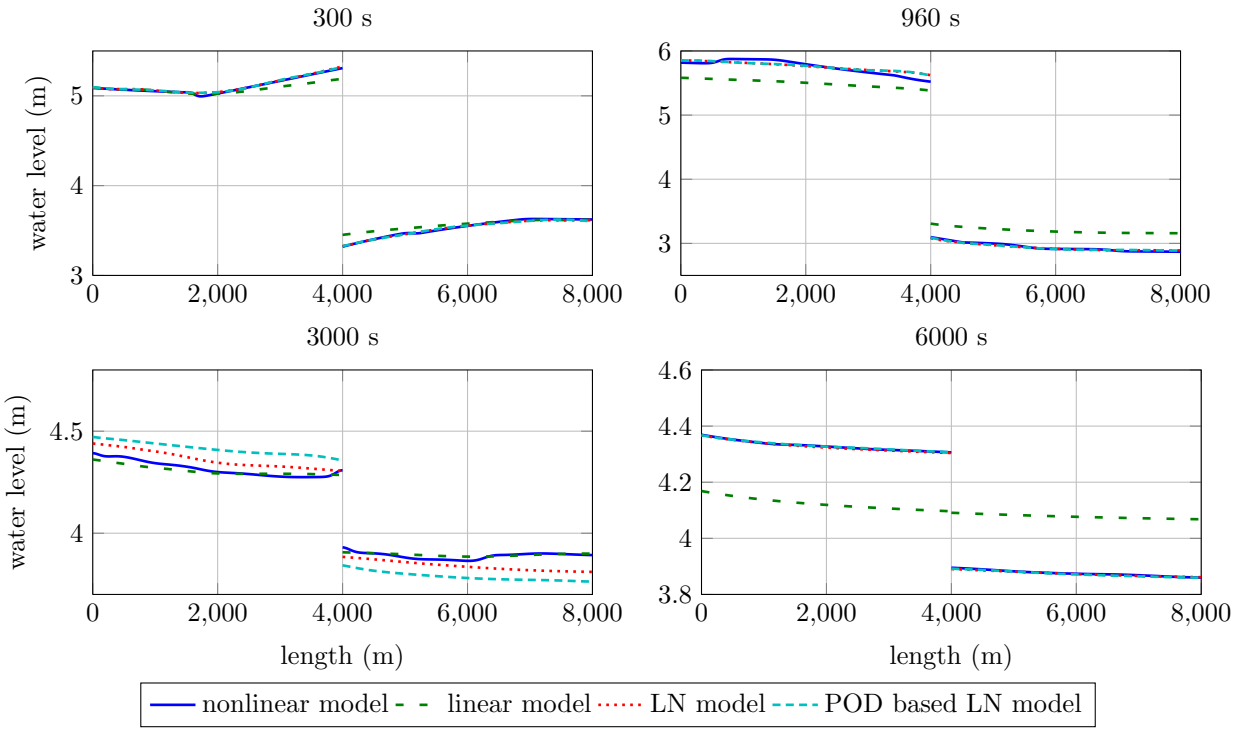


Fig. 3. Water levels at different time instants when the boundary conditions visualized in Fig. 2 are used. The water levels are shown relative to the river bottom.

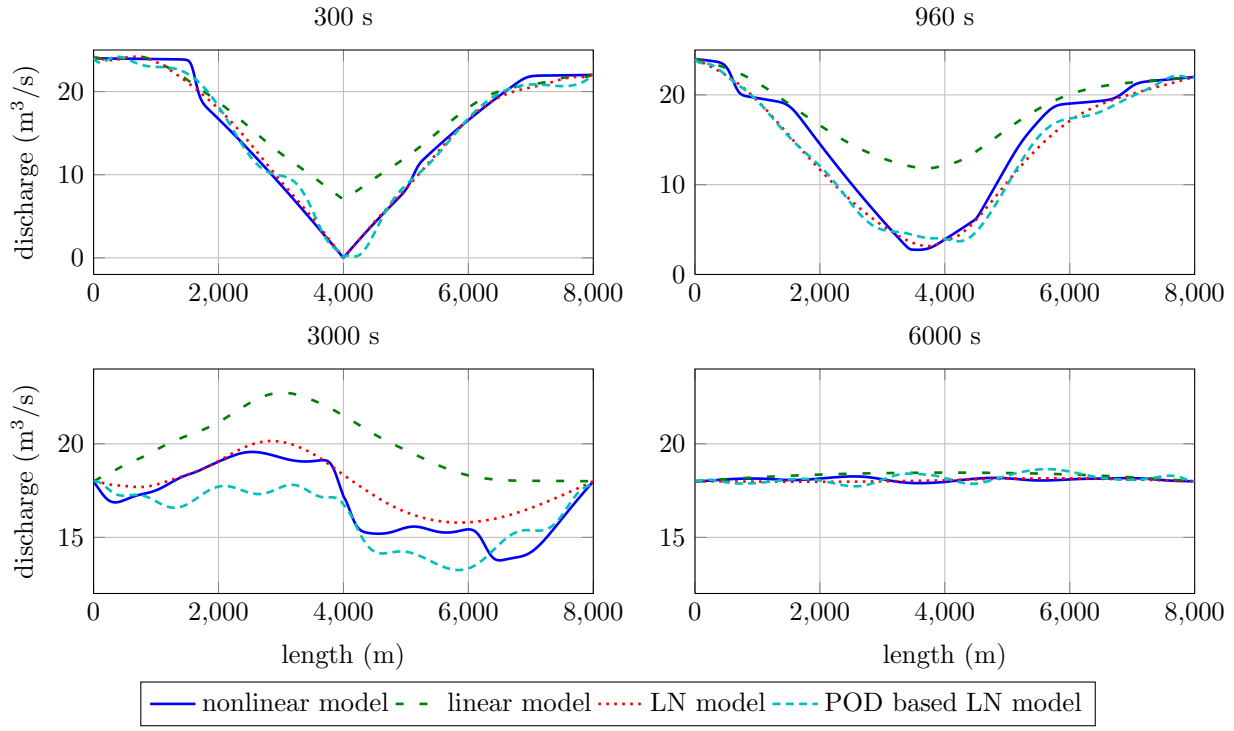


Fig. 4. Discharges at different time instants when the boundary conditions visualized in Fig. 2 are used.

	linear model	LN model	POD based LN model
RMSE water levels	0.16 m	0.029 m	0.043 m
RMSE discharges	3.46 m^3/s	1.25 m^3/s	1.54 m^3/s

Table 2. RMSE for the three models.

model	simulation time (s)
nonlinear model	86.99
linear model	0.703
LN model	0.793
POD based LN model	0.067

Table 3. Average simuation times for the models after 20 runs for a PC with a 2.83GHz Intel Core2 Quad CPU and a RAM of 8 GB.

the simulated water levels: there is almost no distinction between the water levels of the LN model and the POD based LN model. Similar conclusions can be made from Fig. 4 for the discharges. Again the results for the linear model deviates much more from the nonlinear model than for the two LN models. In contrast to the water levels, the model reduction has a larger influence on the discharges: the reduced order model shows more oscillations. If necessary, a higher accuracy for the water levels or the discharges can always be achieved by taking more basis vectors into account for modelling the variables.

Table 2 shows the root mean squared error (RMSE) for the water levels and the discharges. The RMSE is definded as

$$\text{RMSE} = \sqrt{\frac{1}{N_\text{s} M} \sum_{i=1}^{N_\text{s}} \sum_{j=1}^{M} \left(\hat{\boldsymbol{y}}(x_i, t_j) - \boldsymbol{y}(x_i, t_j)\right)^2} \quad (29)$$

with $\boldsymbol{y}$ the output of the nonlinear model, $\hat{\boldsymbol{y}}$ the output of the linear/LN model, N_s the number of variables and M the number of samples. The errors for both LN models are much smaller than the error of the linear model.

Table 3 shows the average simulation time for each model. As expected the linear model and the LN models require much less time than the full nonlinear model. The effect of including the nonlinear gate equations into the LN model has almost no effect on the simulation time compared with the linear model. The model reduction gives an extra speed-up. However the speed-up is not as big as the reduction of states because of the loss of structure in the system matrices. The matrices $\boldsymbol{A}$, $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are very sparse. However, their sparsity is destroyed by the multiplications with $\boldsymbol{\Phi}_n^T$ and $\boldsymbol{\Phi}_n$.

7. CONCLUSIONS

In this paper we have shown that the dynamics of a river system with hydraulic structures cannot be accurately approximated by a linear model. However if a linear model is used only for describing the dynamics of each single reach together with the nonlinear gate equations, then we can get a much better approximation, especially for the water levels, while reducing the simulation time drastically. A further reduction in computation burden is possible by using POD together with Galerkin projection. Future work will focus on the use of the LN and POD based LN models for control purposes.

ACKNOWLEDGEMENTS

M. Breckpot is a Ph. D. fellowship of the Research Foundation - Flanders (FWO) at the Katholieke Universiteit Leuven, M. Agudelo is a post-doc at the Katholieke Universiteit Leuven, B. De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium. Research supported by Research Council KUL: GOA/11/05 Ambiorics, GOA/10/09 MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC) en PFV/10/002 (OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC). IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare. Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011) ; IBBT. EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC HIGHWIND (259 166). Contract Research: AMINAL. Other: Helmholtz: viCERP. ACCM.

REFERENCES

P. Astrid. *Reduction of process simulation models: a proper orthogonal decomposition approach.* PhD thesis, Technische Universiteit Eindhoven (Netherlands), 2004.

M.H. Chaudry. *Open-Channel Flow.* Springer, 2008.

A.J. Clemmens, E. Bautista, B.T. Wahlin, and Strand R.J. Simulation of automatic canal control systems. *Journal of Irrigation and Drainage Engineering*, 131(4):324–335, August 2005.

F.M. Henderson. *Open channel flow.* Macmillan, 1966.

J.A. Liggett and J.A. Cunge. *Numerical methods of solution of the unsteady flow equations, Unsteady Flow in Open Channels*, chapter 4. Water Resources Publications, Fort Collins, 1975.

C.H. Lin, J.F. Yen, and C.T. Tsai. Influence of sluice gate contraction coefficient on distinguishing condition. *Journal of Irrigation and Drainage Engineering*, 128(4): 249–252, August 2002.

S. S. Ravindran. A reduced–order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 34(5):425–448, November 2000.

C. Sepúlveda, M. Gómez, and J. Rodellar. Benchmark of discharge calibration methods for submerged sluice gates. *Journal of Irrigation and Drainage Engineering*, 135(5):676–682, October 2009.

T.S. Strelkoff and H.T. Falvey. Numerical methods used to model unsteady canal flow. *Journal of Irrigation and Drainage Engineering*, 119(4):637–655, July/August 1993.

M. Xu, P.J. van Overloop, and N.C. van de Giesen. On the study of control effectiveness and computational efficiency of reduced saint-venant model in model predictive control of open channel flow. *Advances in Water Resources*, 34:282–290, 2011.