

Document downloaded from:

<http://hdl.handle.net/10251/136956>

This paper must be cited as:

Alvarruiz Bermejo, F.; Martínez Alzamora, F.; Vidal Maciá, AM. (2018). Efficient Modeling of Active Control Valves in Water Distribution Systems Using the Loop Method. *Journal of Water Resources Planning and Management*. 144(10):1-9.
[https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000982](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000982)



The final publication is available at

[https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000982](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000982)

Copyright American Society of Civil Engineers

Additional Information

Efficient Modeling of Active Control Valves in Water Distribution Systems using the Loop Method

Fernando Alvarruiz, Ph.D.¹, Fernando Martínez Alzamora, Ph.D.², and Antonio M. Vidal, Ph.D.³

¹Dept. Sistemas Informáticos y Computación, Universitat Politècnica de València. 46022. Valencia (Spain). Email: fbermejo@dsic.upv.es (Corresponding author).

²Research Institute of Water and Environmental Engineering (IIAMA). Universitat Politècnica de València. 46022. Valencia (Spain). Email: fmartine@hma.upv.es.

³Dept. Sistemas Informáticos y Computación, Universitat Politècnica de València. 46022. Valencia (Spain). Email: a Vidal@dsic.upv.es.

Published at the *Journal of Water Resources Planning and Management*:

Fernando Alvarruiz, Fernando Martínez Alzamora, Antonio M. Vidal. (2018) “Efficient Modeling of Active Control Valves in Water Distribution Systems Using the Loop Method” *Journal of Water Resources Planning and Management*, Vol. 144(10), 04018064.

DOI: [10.1061/\(ASCE\)WR.1943-5452.0000982](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000982)

ABSTRACT

This paper presents a novel approach to model pressure and flow regulating devices in the context of the Newton-Raphson loop method for water distribution network simulation. The proposed approach uses a symmetric matrix for the underlying linear systems, which enables simpler implementation and faster solution, while producing iterations very close to the global gradient algorithm of Epanet. The structure of the matrix is kept unchanged regardless of the operational status of the valves. The paper presents results that validate its formulation, accuracy and speed in various case studies.

INTRODUCTION

Simulating the behavior of water distribution systems (WDS) by means of hydraulic models is nowadays a necessary task for decision making, both in the context of planning and daily operation of the system. Tackling issues such as leaks, water quality, asset management or resilience requires modeling the networks with an increasing level of detail. Geographic Information Systems, commonly present in most utilities in order to store all the information about the assets that make up a network, help building those models (Bartolín et al. 2008). However, manipulating and running the models, especially in the context of optimization tasks where the simulation is the computational kernel, requires the availability of increasingly faster and more efficient simulation algorithms (Guidolin et al. 2013; Mair et al. 2014; Burger et al. 2016).

Epanet (Rossman 2000) is a very efficient software package for the simulation of WDS, and can be regarded as an important benchmark to measure the results and performance of any WDS simulation software. Recently, a community effort has emerged seeking to update and extend the

open-source code independently from the USEPA. This has led to the release of EPANET 2.1 in July 2016, with a number of bug fixes and API additions.

The hydraulic algorithm used by Epanet, known as the Global Gradient Algorithm (GGA) (Todini and Pilati 1988; Todini and Rossman 2013), is one of the most popular methods for the simulation of WDS. An alternative algorithm is the looped Newton-Raphson method, initially presented by Epp and Fowler (1970), which considers a set of independent loops with an associated flow for each of them. The loop method has gained some attention lately, with different papers (Abraham and Stoianov 2016; Alvarruiz et al. 2017a; Abraham and Stoianov 2015; Alvarruiz et al. 2015; Creaco and Franchini 2014; Elhay et al. 2014) showing that it can be a competitive alternative to GGA.

A key aspect of the loop method is that the choice of the set of independent loops greatly affects the matrix sparsity of the underlying linear systems, and thus the time needed for the simulation. Maximum sparsity is obtained by selecting loops with minimum overlap among them, and methods to do so have been proposed recently (Creaco and Franchini 2015; Abraham and Stoianov 2016; Alvarruiz et al. 2015). This paper uses the loop selection algorithm *m4* from (Alvarruiz et al. 2015).

Alvarruiz et al. (2015) also consider modeling flow- and pressure-regulating valves in the context of the loop method. The work takes into account devices such as temporarily closed pipes, *Check Valves* (CVs) preventing reverse flow, *Flow Control Valves* (FCVs), that limit the maximum flow through the valve, *Pressure Reducing Valves* (PRVs), that reduce the pressure of the valve outlet to a given set value, and *Pressure Sustaining Valves* (PSVs), that keep the pressure of the valve inlet at a given value. Modeling of those devices is done without redefining the set of independent loops, using a formulation in which the linear system is expanded with additional equations/unknowns for the devices. The main part of the system is symmetric, although the expanded part is non-symmetric in the case of PRVs/PSVs. The system is solved by blocks, taking advantage of the symmetry of the main block.

This paper revisits the modelization of FCVs, PRVs and PSVs in the loop method. Unlike the formulation in (Alvarruiz et al. 2015) for PRVs/PSVs, the proposal presented here does not involve a non-symmetric expansion of the linear system, and it does not require to solve the system by blocks, either. Instead, the complete system is solved using Cholesky decomposition. This leads to a simpler implementation with faster solution of the linear systems, while also producing iterations closer to those obtained with Epanet. We also present a simpler way to treat FCVs which does not require expanding the linear system.

THE LOOP METHOD FOR WDS SIMULATION

In demand-driven simulation, the values of nodal heads h and link flows q (where a link can be either a pipe, a pump or a valve), are governed by two sets of equations, expressing the head loss through links and the mass conservation at nodes, see e.g. (Alvarruiz et al. 2017b; Todini and Rossman 2013). In particular, for each link k :

$$h_i - h_j = \phi_k(q_k) \quad (1)$$

where h_i and h_j are the head values at the end nodes of link k , and $\phi_k(q_k)$ is the head loss due to friction at link k as a function of its flow. Typically, the head loss is expressed as:

$$\phi_k(q_k) = r_k |q_k|^{\beta-1} q_k \quad (2)$$

for pipe links, where r_k is a coefficient that depends on the pipe characteristics such as length, diameter and roughness, and possibly also on the flow, while β is a constant exponent with the same value for all the network pipes. Different formulae apply for pumps and valves.

Additionally to eq. (1), a second set of equations holds, since at each unknown-head node i the sum of inflows and outflows must equal zero:

$$\sum_k \delta_k q_k - c_i = 0 \quad (3)$$

where k iterates over the set of links connected to node i , δ_k accounts for the sign of the flow (+1 if node i is the final node of link k and -1 if it is the initial node), and c_i is the demand at node i .

In a network with m links, n unknown-head nodes and n_s known-head nodes, the equations given by (1) and (3) can be expressed in matrix form as:

$$\begin{aligned} \Phi(\mathbf{q}) + \mathbf{A}_{12}\tilde{\mathbf{h}} + \mathbf{A}_{10}\hat{\mathbf{h}} &= \mathbf{0} \\ \mathbf{A}_{12}^T \mathbf{q} - \mathbf{c} &= \mathbf{0} \end{aligned} \quad (4)$$

where $\tilde{\mathbf{h}} \in \mathbb{R}^n$ and $\hat{\mathbf{h}} \in \mathbb{R}^{n_s}$ are vectors of heads corresponding to the unknown-head and known-head nodes, respectively, while $\mathbf{A}_{12} \in \mathbb{R}^{m \times n}$ and $\mathbf{A}_{10} \in \mathbb{R}^{m \times n_s}$ are topology matrices describing the connections between links and nodes. Namely, \mathbf{A}_{12} indicates the connections for unknown-head nodes, with $\mathbf{A}_{12}(i, j) = 0$ if node j is not connected to link i , +1 if it is the final node of the link and -1 if it is the initial node, while \mathbf{A}_{10} does the same for known-head nodes.

The non-linear system of equations (4), in the unknowns $\tilde{\mathbf{h}}$ and \mathbf{q} , can be solved by means of the Newton-Raphson method, which leads to the following system of linear equations in each iteration:

$$\begin{bmatrix} \mathbf{D} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{q} \\ \Delta \tilde{\mathbf{h}} \end{bmatrix} = - \begin{bmatrix} \Phi^k + \mathbf{A}_{12}\tilde{\mathbf{h}}^k + \mathbf{A}_{10}\hat{\mathbf{h}} \\ \mathbf{A}_{12}^T \mathbf{q}^k - \mathbf{c} \end{bmatrix} \quad (5)$$

or, since $\Delta \tilde{\mathbf{h}} = \tilde{\mathbf{h}}^{k+1} - \tilde{\mathbf{h}}^k$:

$$\begin{bmatrix} \mathbf{D} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{q} \\ \tilde{\mathbf{h}}^{k+1} \end{bmatrix} = - \begin{bmatrix} \Phi^k + \mathbf{A}_{10}\hat{\mathbf{h}} \\ \mathbf{A}_{12}^T \mathbf{q}^k - \mathbf{c} \end{bmatrix} \quad (6)$$

where $\Delta \mathbf{q} = \mathbf{q}^{k+1} - \mathbf{q}^k$, $\Phi^k = \Phi(\mathbf{q}^k)$ and \mathbf{D} is a diagonal matrix with elements $\mathbf{D}(i, i) = \phi'_i(q_i^k) = \frac{d\phi}{dq_i}(q_i^k)$.

Assuming that \mathbf{D} is invertible, the system of equations (6) can be solved by isolating $\Delta \mathbf{q}$ in the first block row of the system and substituting in the second block row, which yields the formulation of the Global Gradient Algorithm (GGA) (Todini and Rossman 2013).

In the method known as Looped Newton-Raphson, however, the dimension of the linear system is further reduced by taking into consideration the network loops. In this context, a loop is any sequence of pipes in the network defining either a path from a node to itself, or a path between two known-head nodes. Given a network, its loops can be represented by a subset of $l = m - n$ independent loops (considering a network as a graph with m links and $n + n_s$ nodes, the number of independent loops would be $m - n - n_s + 1$, but $n_s - 1$ additional loops connecting the known-head nodes must be taken into account). The set of independent loops can be described by means of a

matrix $\mathbf{M}_{31} \in R^{l \times m}$, such that $\mathbf{M}_{31}(i, j) = 0$ if link $j \notin \text{loop } i$, $+1$ if link $j \in \text{loop } i$ and follows the direction of the loop, and -1 if link $j \in \text{loop } i$ and its direction is opposed to that of the loop.

A set of loops are independent if the corresponding rows of \mathbf{M}_{31} are linearly independent. Because the l loops have been chosen to be independent, \mathbf{M}_{31} has full row rank. It is also easy to see that $\mathbf{M}_{31}\mathbf{A}_{12} = \mathbf{0}$, which comes from the fact that if a loop has a link that goes into an unknown-head node (assuming an arbitrary direction for the loop), then it also has a corresponding link that goes out of that node. It follows that \mathbf{M}_{31}^T is a base of the kernel space of matrix \mathbf{A}_{12}^T (Abraham and Stoianov 2015).

Premultiplying the first block row of eq. (6) by \mathbf{M}_{31} yields:

$$\mathbf{M}_{31}\mathbf{D}\Delta\mathbf{q} = -\mathbf{M}_{31} \left(\Phi^k + \mathbf{A}_{10}\hat{\mathbf{h}} \right) \quad (7)$$

where each row of the equality corresponds to the energy conservation in a loop.

On the other hand, given an arbitrary flow vector \mathbf{q}^* satisfying the mass conservation equation, i.e. $\mathbf{A}_{12}^T\mathbf{q}^* - \mathbf{c} = \mathbf{0}$, it can be shown that vector \mathbf{q}^{k+1} can be expressed as $\mathbf{q}^{k+1} = \mathbf{q}^* + \mathbf{M}_{31}^T\hat{\mathbf{q}}$, for some vector $\hat{\mathbf{q}} \in R^l$ (Abraham and Stoianov 2015). In particular, from eq. (6), $\mathbf{A}_{12}^T\Delta\mathbf{q} = -\mathbf{A}_{12}^T\mathbf{q}^k + \mathbf{c}$, thus $\mathbf{A}_{12}^T\mathbf{q}^{k+1} = \mathbf{c}$. It follows that $\mathbf{A}_{12}^T(\mathbf{q}^{k+1} - \mathbf{q}^*) = \mathbf{c} - \mathbf{c} = \mathbf{0}$, which means that vector $\mathbf{q}^{k+1} - \mathbf{q}^*$ is in the kernel space of matrix \mathbf{A}_{12}^T , and because \mathbf{M}_{31}^T is a base of that space, $\mathbf{q}^{k+1} - \mathbf{q}^* = \mathbf{M}_{31}^T\hat{\mathbf{q}}$, or

$$\mathbf{q}^{k+1} = \mathbf{q}^* + \mathbf{M}_{31}^T\hat{\mathbf{q}} \quad (8)$$

for some vector $\hat{\mathbf{q}} \in R^l$.

Substituting for \mathbf{q}^{k+1} in (7) yields:

$$\mathbf{M}_{31}\mathbf{D}\mathbf{M}_{31}^T\hat{\mathbf{q}} = -\mathbf{M}_{31} \left(\Phi^k + \mathbf{A}_{10}\hat{\mathbf{h}} + \mathbf{D}(\mathbf{q}^* - \mathbf{q}^k) \right) \quad (9)$$

which is a linear system with a symmetric positive definite $l \times l$ matrix.

It should be noted that vector $\hat{\mathbf{q}}$ in eq. (8) corresponds to loop flow corrections with respect to the balanced vector \mathbf{q}^* , and not with respect to the current estimate \mathbf{q}^k . The formulation of the loop method can also be done without making use of the auxiliary balanced flow vector \mathbf{q}^* . In that case, the flow corrections $\hat{\mathbf{q}}$ are applied to \mathbf{q}^k in eq. (8) and the term $\mathbf{D}(\mathbf{q}^* - \mathbf{q}^k)$ is removed from eq. (9). However, introducing the balanced flow vector \mathbf{q}^* eliminates the need that \mathbf{q}^0 verifies the mass conservation equation (3), which makes it possible to use a more suitable starting point for the Newton-Raphson method. Successive flow vectors $\mathbf{q}^1, \mathbf{q}^2 \dots$ verify mass conservation eq. (3). However, the formulation presented in this paper for pressure-regulating devices may lead to an unbalanced flow vector \mathbf{q}^k at any given iteration k , and this can also be accommodated using the above formulation of the loop method.

Finally, the vector of heads $\tilde{\mathbf{h}}^{k+1}$ can be obtained by updating the link head losses as $\Phi^k + \mathbf{D}\Delta\mathbf{q}$, and traversing the network using the known-head nodes as starting points. Equivalently, $\tilde{\mathbf{h}}^{k+1}$ can be computed through the following linear system, which is obtained by premultiplying the first block row of system (6) by \mathbf{A}_{12}^T :

$$\mathbf{A}_{12}^T\mathbf{A}_{12}\tilde{\mathbf{h}}^{k+1} = -\mathbf{A}_{12}^T \left(\Phi^k + \mathbf{D}\Delta\mathbf{q} + \mathbf{A}_{10}\hat{\mathbf{h}} \right) \quad (10)$$

Compared to the GGA, the loop method presents the advantage that the linear system (9) solved in each iteration has a matrix of size $l \times l$, which is in most cases much smaller than the

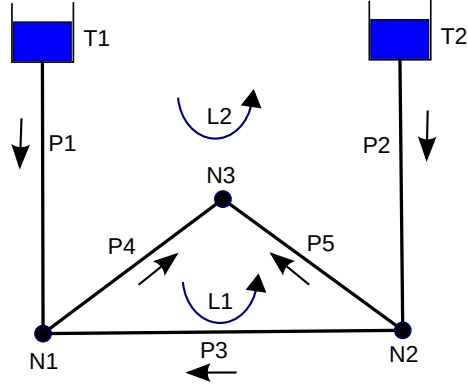


Fig. 1. A simple network

corresponding matrix for the GGA, of size $n \times n$. Computation of heads, e.g. by traversing the network, is inexpensive, because the order to follow to traverse it can be precomputed. Furthermore, computation of heads can be avoided at the first iterations, and in some cases it can even be carried out only at the end of the iterative process.

Example

Considering the simple network shown in Fig 1, the Newton-Raphson iteration of eq. (6) takes the form:

$$\begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & | & 1 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & | & 0 & 1 & 0 \\ 0 & 0 & d_3 & 0 & 0 & | & 1 & -1 & 0 \\ 0 & 0 & 0 & d_4 & 0 & | & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & d_5 & | & 0 & -1 & 1 \\ \hline 1 & 0 & 1 & -1 & 0 & | & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & | & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & | & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta q_4 \\ \Delta q_5 \\ \tilde{h}_1^{k+1} \\ \tilde{h}_2^{k+1} \\ \tilde{h}_3^{k+1} \end{bmatrix} = \begin{bmatrix} -\phi_1^k + \hat{h}_1 \\ -\phi_2^k + \hat{h}_2 \\ -\phi_3^k \\ -\phi_4^k \\ -\phi_5^k \\ \hline -q_1^k - q_3^k + q_4^k + c_1 \\ -q_2^k + q_3^k + q_5^k + c_2 \\ -q_4^k - q_5^k + c_3 \end{bmatrix} \quad (11)$$

In order to use the loop method, a set of independent loops must be chosen. Choosing the loops L1 and L2 shown in Fig. 1 produces the following loop matrix:

$$\mathbf{M}_{31} = \begin{bmatrix} 0 & 0 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 \end{bmatrix}$$

Accordingly, the linear system (9) is then:

$$\begin{bmatrix} d_3 + d_4 + d_5 & -d_4 - d_5 \\ -d_4 - d_5 & d_1 + d_2 + d_4 + d_5 \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \end{bmatrix} = \begin{bmatrix} \hat{\phi}_3^k + \hat{\phi}_4^k - \hat{\phi}_5^k \\ -\hat{\phi}_1^k + \hat{\phi}_2^k - \hat{\phi}_4^k + \hat{\phi}_5^k + \hat{h}_1 - \hat{h}_2 \end{bmatrix} \quad (12)$$

where $\hat{\phi}_i^k = \phi_i^k + d_i(q_i^* - q_i^k)$. As explained above, \mathbf{q}^* is any arbitrary flow vector satisfying the mass conservation equation (3), and it is not updated during the Newton-Raphson steps.

INCLUDING CONTROL ELEMENTS IN THE SIMULATION

The inclusion in the simulation of control elements such as CVs, FCVs, PRVs and PSVs presents an important challenge. Those elements affect the hydraulic conditions of the network (pressures and flows), which in turn affect the operating conditions and behavior of those same elements, e.g. causing a valve to be closed or fully opened.

Approaches to deal with that can be divided in two groups. Firstly, (Deuerlein et al. 2009a; Deuerlein et al. 2005; Deuerlein et al. 2009b; Piller and van Zyl 2014) propose approaches in which the problem is formulated as the minimization of the *content* or *co-content* functions subject to inequality constraints. The resulting constrained optimization problem must be solved by methods that are more demanding, from a computational point of view, than the Newton-Raphson approach used for the the basic simulation problem. As an alternative, a more heuristic approach can be used, as is done in Epanet (Rossman 1999), where an initial status of the valves is assumed at the beginning of the iterative process, checked between the iterations and, if necessary, adjusted by specific heuristics (e.g. changing the status of a CV from "open" to "closed" if, at the current iteration, the head at the inlet is lower than at the outlet, or if the flow is negative). Although there is no guarantee that this approach will find the correct valve status in all cases, see e.g. (Simpson 1999), it works well in practice and is a widely accepted method in the hydraulic modeling community.

The work described in this paper uses the second approach, adopting the same heuristics as Epanet to update the status of valves in each iteration. In that context, the presence of control valves affects the formulation of the loop method given by eqs. (8) and (9). If, e.g. a valve closes, the network topology changes, rendering the initially chosen set of independent loops inconsistent with it. Arsene et al. (2012) propose a partial redefinition of the set of loops based on the update of a spanning tree. However, redefining the set of loops implies modifying the sparsity pattern of the system matrix in eq. (9). If that system is solved by means of a direct method, as is normally the case, a change in the sparsity pattern makes it necessary to repeat the process of symbolic factorization of the matrix, increasing the computing time.

Alvarruiz et al. (2015) present a method to deal with closed valves and FCV which avoids the need to redefine the set of independent loops. This is done by expanding the linear system (9) in a symmetric way and solving the expanded system by blocks, where the main block corresponds to the solution of the basic unexpanded system (9).

This paper uses a simpler approach to deal with closed valves and FCV, showing its relationship with the formulation of (Alvarruiz et al. 2015), and validating the approach by simulation results.

In the case of PRVs/PSVs, the linear system (9) has to be modified to account for the fact that the head at the control node of the valve is known (assuming the valve is active, i.e. not fully open or closed), while the head loss function $\phi_i(q_i)$ and its derivative d_i are unknown.

Jeppson (1976) first introduced an approach to deal with PRVs in the loop method, in which a loop (or path) is formed that goes from the control node of the valve to a known-head node. That loop replaces one of the original loops containing the PRV, and the rest of the loops where the PRV is present are redefined to avoid the valve. The modified set of loops is used for the energy conservation equation (7), while the original set of loops is used for the correction flows of eq. (8), which implies that eq. (9) is transformed into a system with a non-symmetric matrix.

Ateş (2017) tackles PRV modeling in the loop method by considering three different topological positions of the valve in the network, and using a different approach for each case. The set of independent loops, and thus the structure of the linear system matrix, changes depending on the operational status of the valve.

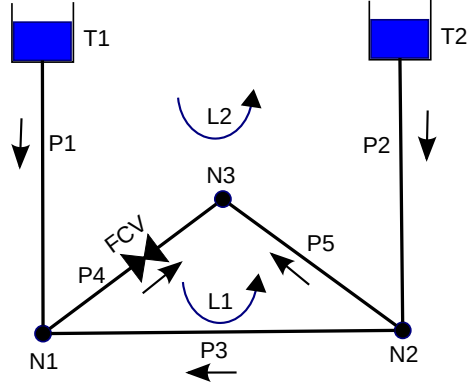


Fig. 2. A simple network with a FCV

Alvarruiz et al. (2015) present a different approach for PRVs which avoids redefining the set of loops. Similarly to the method for closed valves and FCVs of the same authors, this is done by expanding the linear system (9), in this case in a non-symmetric way, and solving the resulting system by blocks. Advantage is taken of the fact that the main block, corresponding to the basic unexpanded system, is symmetric, although other blocks require solution of non-symmetric systems.

This paper presents an alternative way to model PRVs/PSVs in the context of the loop method, which avoids redefining the set of loops, as in (Alvarruiz et al. 2015), but also uses a completely symmetric matrix, which enables the use of more efficient methods for the solution of the system, and also simplifies the solution process. The method does not need to use different formulations depending on the topological position of the PRV, and the matrix structure is unchanged regardless of the operational status of the valves.

MODELING FLOW REGULATING DEVICES

In (Alvarruiz et al. 2015), the authors present a formulation to cope with flow regulating devices in the context of the loop method. A FCV behaves as a resistive element when open, but if it is active, the head loss between its two ends is not related to the flow through it. Thus, that head loss is introduced as an unknown, and a new equation is added to express the fact that the flow through the device is known.

This can be illustrated by means of the network in Fig. 2, where line P4 is a FCV with a flow setting of s_q . In that case, system (12) is transformed into:

$$\begin{bmatrix} d_3 + d_5 & -d_5 & -1 \\ -d_5 & d_1 + d_2 + d_5 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \phi_4^{k+1} \end{bmatrix} = \begin{bmatrix} \hat{\phi}_3^k - \hat{\phi}_5^k \\ -\hat{\phi}_1^k + \hat{\phi}_2^k + \hat{\phi}_5^k + \hat{h}_1 - \hat{h}_2 \\ s_q - q_4^* \end{bmatrix} \quad (13)$$

As pointed out in (Alvarruiz et al. 2015), a simpler way to model a closed pipe, or a flow regulating device in general, would be to use a pipe that opposes a great resistance to any deviation of its flow with respect to the setting, i.e. a pipe with a head-loss given by the expression $\phi_i^k = \alpha(q_i^k - s_q)$, where α is a large scalar ($\alpha = 10^8$ in the simulations of this paper). In the example, system (11)

becomes:

$$\left[\begin{array}{ccccc|ccc} d_1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & d_3 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & \alpha & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & d_5 & 0 & -1 & 1 \\ \hline 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{array} \right] \left[\begin{array}{c} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta q_4 \\ \Delta q_5 \\ \tilde{h}_1^{k+1} \\ \tilde{h}_2^{k+1} \\ \tilde{h}_3^{k+1} \end{array} \right] = \left[\begin{array}{c} -\phi_1^k + \hat{h}_1 \\ -\phi_2^k + \hat{h}_2 \\ -\phi_3^k \\ -\alpha(q_4^k - s_q) \\ -\phi_5^k \\ \hline -q_1^k - q_3^k + q_4^k + c_1 \\ -q_2^k + q_3^k + q_5^k + c_2 \\ -q_4^k - q_5^k + c_3 \end{array} \right] \quad (14)$$

and accordingly, the linear system for the loop method would be given by eq. (12), with $d_4 = \alpha$ and $\hat{\phi}_4^k = \phi_4^k + \alpha(q_4^* - q_4^k) = \alpha(q_4^* - s_q)$.

It can be argued that this formulation introduces large numbers in the linear system matrix, which could lead to a poorly conditioned system, producing large round-off errors. However, this paper shows, by means of simulation results, that the approach works well in practice. Moreover, the problem of large numbers in the linear system matrix is also encountered in Epanet's GGA method, when dealing with pipes with a very small resistance and/or flow (and consequently, a very small value of d_i).

It is worth noting that there is a relationship between the two formulations for flow-regulating devices, in such a way that the second formulation can be obtained by introducing a perturbation on the linear system corresponding to the first formulation. In particular, introducing a small value in the element at (3, 3) of the matrix in system (13) yields:

$$\left[\begin{array}{ccc|c} d_3 + d_5 & -d_5 & -1 & \hat{q}_1 \\ -d_5 & d_1 + d_2 + d_5 & 1 & \hat{q}_2 \\ -1 & 1 & -1/\alpha & \phi_4^{k+1} \end{array} \right] \left[\begin{array}{c} \hat{q}_1 \\ \hat{q}_2 \\ \phi_4^{k+1} \end{array} \right] = \left[\begin{array}{c} \hat{\phi}_3^k - \hat{\phi}_5^k \\ -\hat{\phi}_1^k + \hat{\phi}_2^k + \hat{\phi}_5^k + \hat{h}_1 - \hat{h}_2 \\ s_q - q_4^* \end{array} \right] \quad (15)$$

From the last row:

$$\phi_4^{k+1} = \alpha(-\hat{q}_1 + \hat{q}_2 - s_q + q_4^*) = -\alpha\hat{q}_1 + \alpha\hat{q}_2 + \alpha(q_4^* - s_q) \quad (16)$$

and substituting in the first row results in eq. (12) with $d_4 = \alpha$ and $\hat{\phi}_4^k = \alpha(q_4^* - s_q)$.

MODELING PRESSURE REGULATING DEVICES

This section considers the inclusion of PRVs and PSVs in the loop method. Given the analogy between both types of valves, it focuses on PRVs, briefly commenting later the difference in the case of PSVs.

A disadvantage of the formulation presented in (Alvarruiz et al. 2015) for PRVs/PSVs is the introduction of non-symmetry in the linear system matrix, even though the main matrix block preserves its symmetry. This non-symmetry, however, is avoided in Epanet's formulation for that type of devices. This section shows a formulation for PRVs/PSVs which can be considered the translation to the loop method of Epanet's formulation for those devices.

The price to pay in order to preserve the symmetry of the linear system matrix, either in GGA or the loop method, is the introduction of an approximation in the linear system (6), which could increase the number of Newton-Raphson iterations. However, the use of a symmetric matrix reduces the time needed to solve the system in each iteration, and also the memory requirements.

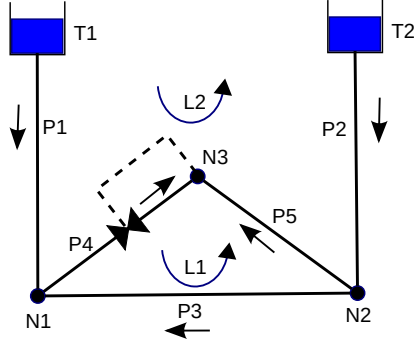


Fig. 3. A simple network with a PRV

Fig. 3 shows a modification of the network of Fig. 1, in which the link P4 is a PRV. If the valve is active, the value of d_4 in the corresponding linear system (11) is unknown, while h_3 is known, so that the system would be modified in the following way (elements that change are shown in brackets):

$$\begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & | & 1 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & | & 0 & 1 & 0 \\ 0 & 0 & d_3 & 0 & 0 & | & 1 & -1 & 0 \\ 0 & 0 & 0 & [0] & 0 & | & [0] & 0 & 1 \\ 0 & 0 & 0 & 0 & d_5 & | & 0 & -1 & 1 \\ \hline 1 & 0 & 1 & -1 & 0 & | & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & | & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & | & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta q_4 \\ \Delta q_5 \\ \hline \tilde{h}_1^{k+1} \\ \tilde{h}_2^{k+1} \\ \tilde{h}_3^{k+1} \end{bmatrix} = \begin{bmatrix} -\phi_1^k + \hat{h}_1 \\ -\phi_2^k + \hat{h}_2 \\ -\phi_3^k \\ [s_h] \\ -\phi_5^k \\ \hline -q_1^k - q_3^k + q_4^k + c_1 \\ -q_2^k + q_3^k + q_5^k + c_2 \\ -q_4^k - q_5^k + c_3 \end{bmatrix} \quad (17)$$

where s_h is the head fixed by the PRV for its downstream node N3 (i.e. the sum of the node elevation plus the valve pressure setting). Thus, the fourth equation of the above system merely states the fact that the head in that node is determined by the valve setting.

Note that the system is no longer symmetric. In order to keep the symmetry, the formulation used by Epanet introduces an approximation in the equation corresponding to the mass balance of the upstream node of the PRV, whereby the flow increment of the PRV itself is neglected, i.e.

$$\begin{bmatrix} d_1 & 0 & 0 & 0 & 0 & | & 1 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 & | & 0 & 1 & 0 \\ 0 & 0 & d_3 & 0 & 0 & | & 1 & -1 & 0 \\ 0 & 0 & 0 & [0] & 0 & | & [0] & 0 & 1 \\ 0 & 0 & 0 & 0 & d_5 & | & 0 & -1 & 1 \\ \hline 1 & 0 & 1 & [0] & 0 & | & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & | & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & | & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta q_4 \\ \Delta q_5 \\ \hline \tilde{h}_1^{k+1} \\ \tilde{h}_2^{k+1} \\ \tilde{h}_3^{k+1} \end{bmatrix} = \begin{bmatrix} -\phi_1^k + \hat{h}_1 \\ -\phi_2^k + \hat{h}_2 \\ -\phi_3^k \\ [s_h] \\ -\phi_5^k \\ \hline -q_1^k - q_3^k + q_4^k + c_1 \\ -q_2^k + q_3^k + q_5^k + c_2 \\ -q_4^k - q_5^k + c_3 \end{bmatrix} \quad (18)$$

Additionally, the GGA method needs to compute \mathbf{D}^{-1} to solve eq. (18), which is not possible now since the matrix element (4,4) is zero. For that reason, Epanet replaces that element with a small value ($1/\alpha$).

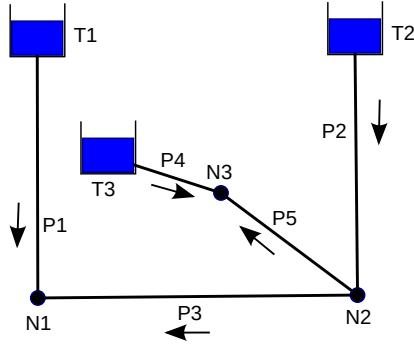


Fig. 4. Modeling of the PRV in the sample network

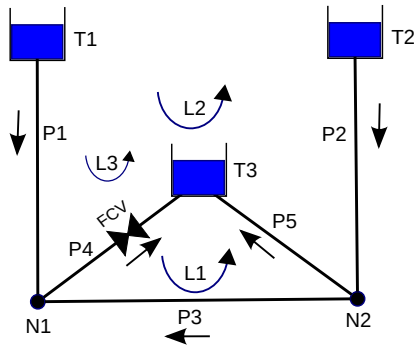


Fig. 5. Alternative modelling of the PRV in the sample network

System (18) can be interpreted as corresponding to the network of Fig. 4, where a fictitious tank (T3) has been added, and the PRV has turned into a zero-resistance pipe linking the valve's downstream node with the fictitious tank. Additionally, the demand of the valve's upstream node (N1) is modified in each iteration k to get mass balance if the flow through the PRV is not negative, i.e. $c_1^k = c_1 + \max(q_4^k, 0)$.

The system of eqs. (18) could be transformed to the corresponding loop-based system, but in doing so two difficulties are encountered. Firstly, the initial set of loops (L1 and L2) is no longer valid, as is clear from Fig. 4. Secondly, the balanced vector \mathbf{q}^* should be recomputed at each iteration, since the fictitious demand for node N1 changes.

Those difficulties can be avoided by considering an equivalent of the network in Fig. 4, as shown in Fig. 5, where the fictitious tank T3 and the node N3 have been merged, and the PRV has been replaced by a FCV. The setting of the FCV is fixed in each iteration, analogously to the adjustment of the demand of N1 in Fig. 4. In particular, the flow through the FCV is fixed to get mass balance at node N3, provided that the flow is not negative, i.e. $s_q = \max(0, c_3 - q_5^k)$.

Using the formulation described previously for flow-regulating devices, the Newton-Raphson iteration for the network in Fig. 5 can be obtained by modifying eq. (14), converting N3 into a tank

with head equal to s_h :

$$\left[\begin{array}{ccccc|cc} d_1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & d_2 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & d_3 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & \alpha & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & d_5 & 0 & -1 \\ \hline 1 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 \end{array} \right] \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta q_4 \\ \Delta q_5 \\ \tilde{h}_1^{k+1} \\ \tilde{h}_2^{k+1} \end{bmatrix} = \begin{bmatrix} -\phi_1^k + \hat{h}_1 \\ -\phi_2^k + \hat{h}_2 \\ -\phi_3^k \\ -\alpha(q_4^k - s_q) - s_h \\ -\phi_5^k - s_h \\ \hline -q_1^k - q_3^k + q_4^k + c_1 \\ -q_2^k + q_3^k + q_5^k + c_2 \end{bmatrix} \quad (19)$$

The previous system can be transformed to the corresponding loop-based system, taking into account that, since an unknown-head node has been replaced by a tank, a new loop has to be added, corresponding to a path linking the fictitious tank to any other tank. For example, using the path through P1 and P4:

$$\bar{\mathbf{M}}_{31} = \begin{bmatrix} 0 & 0 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (20)$$

Consequently, the linear system of the loop method is:

$$\begin{bmatrix} d_3 + d_4 + d_5 & -d_4 - d_5 & -d_4 \\ -d_4 - d_5 & d_1 + d_2 + d_4 + d_5 & d_1 + d_4 \\ -d_4 & d_1 + d_4 & d_1 + d_4 \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{bmatrix} = \begin{bmatrix} \hat{\phi}_3^k + \hat{\phi}_4^k - \hat{\phi}_5^k \\ -\hat{\phi}_1^k + \hat{\phi}_2^k - \hat{\phi}_4^k + \hat{\phi}_5^k + \hat{h}_1 - \hat{h}_2 \\ -\hat{\phi}_1^k - \hat{\phi}_4^k + \hat{h}_1 - s_h \end{bmatrix} \quad (21)$$

where $d_4 = \alpha$ and $\hat{\phi}_4^k = \alpha(q_4^* - s_q)$.

At any iteration, the PRV may become inactive (closed or fully open), according to the algorithm used to adjust the status of the device. In that case, the valve behaves as a pipe, and the extra loop and its associated flow in eq. (21) should not be taken into account. That can be done by simply zeroing out the corresponding row and column, except for the diagonal element, which can be set to any non-zero value so that $\hat{q}_3 = 0$.

In general, for a network with multiple PRVs, system (21) can be written as:

$$\bar{\mathbf{M}}_{31} \bar{\mathbf{D}} \bar{\mathbf{M}}_{31}^T \hat{\mathbf{q}} = -\bar{\mathbf{M}}_{31} \left(\bar{\Phi}^k + \bar{\mathbf{A}}_{10} \bar{\mathbf{h}} + \bar{\mathbf{D}} \left(\mathbf{q}^* - \mathbf{q}^k \right) \right) \quad (22)$$

where $\bar{\mathbf{D}}$, $\bar{\Phi}^k$ are obtained from \mathbf{D} , Φ^k , with $\bar{d}_i = \alpha$, $\bar{\phi}_i^k = \alpha(q_i^k - s_q)$, for each active PRV i ; $\bar{\mathbf{h}}$ and $\bar{\mathbf{A}}_{10}$ are obtained from $\hat{\mathbf{h}}$ and \mathbf{A}_{10} , respectively, by converting PRV control nodes to tanks, and $\bar{\mathbf{M}}_{31}$ is formed by adding to \mathbf{M}_{31} the paths associated to the PRVs, as the example above shows.

For the case of PSVs, the procedure is analogous, taking into account that the fictitious tank replaces the valve's upstream node, and that the fictitious FCV is adjusted at each iteration to get mass balance also at that node.

RESULTS

The objective of this section is to evaluate the proposed formulation for control valves, both from the point of view of accuracy and simulation time. For that purpose, simulations of different networks have been carried out with three different solvers: (i) Epanet 2.1, (ii) an implementation of the loop method incorporating the formulation for control valves proposed here and (iii) an

TABLE 1. Networks considered

Network	n	n_s	m	<i>looped</i>	<i>PRV</i>	<i>PSV</i>	<i>FCV</i>	<i>CV</i>	<i>Time steps</i>
exnet	1891	2	2467	80.2%	1	0	0	3	25
urb1	4236	4	4649	70.0%	5	0	1	0	1153
urb2	7720	67	8801	89.6%	158	0	17	5	24
bwsn2	12523	4	14831	74.2%	0	1	4	2	596
urb3	25813	3	29345	75.4%	10	0	0	0	26

TABLE 2. Residuals and number iterations for the loop solver (and Epanet)

Network	ϵ_h	ϵ_q	Iters
exnet	$1.38 \cdot 10^{-3}$	$9.29 \cdot 10^{-5}$	108
urb1	$1.48 \cdot 10^{-3}$	$9.24 \cdot 10^{-5}$ ($9.27 \cdot 10^{-5}$)	1396
urb2	$1.99 \cdot 10^{-2}$ ($1.98 \cdot 10^{-2}$)	$4.07 \cdot 10^{-3}$ ($1.99 \cdot 10^{-3}$)	237 (243)
bwsn2	$5.86 \cdot 10^{-2}$	$2.90 \cdot 10^{-3}$	812
urb3	$6.67 \cdot 10^{-2}$	$6.77 \cdot 10^{-3}$	186 (185)

implementation of the loop method using the non-symmetric formulation for PRVs described in (Alvarruiz et al. 2015). While in (Alvarruiz et al. 2015) the formulation for control valves was evaluated using a MATLAB implementation, all the solvers considered here are implemented in C language. We compare the accuracy of the solvers by computing the residuals of the non-linear equation systems, and also compare the computing time needed by the solvers.

The set of networks considered in this section is presented in Table 1. For each of them, the table shows the number of unknown-head and known-head nodes (n, n_s), links (m), the percentage of links belonging to at least one loop (*looped*, note that this is independent from the particular choice of loops used), the number of different types of valves and the number of time steps in the extended period simulation. These networks have been chosen because, among those available to the authors, they were the largest networks having pressure- and flow-regulating devices.

Two of the networks (exnet and bwsn2) have been taken from the literature in the field. In particular, exnet is a benchmark water system set up by the Centre for Water Systems of Exeter University (Farmani et al. 2005), modified in order to consider a 24-hour extended period simulation, with the head at reservoirs varying according to a pattern. Network bwsn2 corresponds to network 2 proposed in (Ostfeld et al. 2008).

The other three networks correspond to real distribution systems in two cities in Spain. Urb1 supplies water to a small city of around 50,000 inhabitants, and contains a reduced number of regulating devices. Urb2 and urb3 are different models of the same city, with a population of 1.5 million inhabitants. The former is a strategic calibrated model containing details about the regulating elements of the network, including a large number of valves, tanks and pumps. Urb3 contains more details about the links and nodes that make up the network, but simplifies the regulating elements.

In order to evaluate the accuracy of the simulation results, the maximum residuals, over all time

TABLE 3. Epanet computing time (in seconds)

network	reps	total	open	coeffs	linsolve	new q, h	other
exnet	200	0.0860	0.0040	0.0565	0.0148	0.0062	0.0046
urb1	15	1.7945	0.0140	0.9118	0.4117	0.1455	0.3115
urb2	20	0.7012	0.0462	0.3939	0.1808	0.0540	0.0262
bwsn2	10	3.9965	0.1008	2.0090	0.8084	0.4026	0.6757
urb3	15	2.4135	0.4254	1.2015	0.4704	0.2064	0.1097

steps, of the non-linear equation system given by eqs. (1) and (3) have been computed, using the solutions obtained from the different solvers considered. We differentiate head loss residuals and flow residuals, as described next.

The head loss residual ϵ_h corresponds to the maximum head loss imbalance for any link, where the head loss imbalance for a link k , going from node i to node j , is derived from eq. (1) as:

$$|h_i - h_j - \phi_k(q_k)| \tag{23}$$

Closed links and active FCVs are not taken into account for the head loss residual, because its head loss is not directly evaluable, and they will contribute to the flow residual instead, as described next. As for active PRVs, if s_h is the target head of an active PRV going from node i to node j , then its head loss is $\phi_k(q_k) = h_i - s_h$, and its head loss imbalance is then $|s_h - h_j|$. Similarly, the head loss imbalance of an active PSV is $|s_h - h_i|$.

On the other hand, the flow residual ϵ_q corresponds to the maximum flow imbalance for any unknown-head node, closed link or active FCV. According to eq. (3), the flow imbalance of an unknown-head node i is

$$\left| \sum_k \delta_k q_k - c_i \right| \tag{24}$$

Flow imbalance for closed links is the deviation from a zero-flow, i.e. $|q_k|$ for a link k . As for active FCVs, their flow imbalance is the deviation from their flow setting s_q , i.e. $|q_k - s_q|$.

The residuals for the loop solver with the proposed formulation for flow- and pressure-regulating devices are shown in Table 2, together with the total number of Newton-Raphson iterations. In many cases, the values of residuals and iterations are the same for the loop solver and for Epanet (version 2.1). In the cases where there is a difference, the value for Epanet is given in brackets. The results indicate that the loop solver is as accurate as Epanet.

Tables 3 and 4 focus on the computing time. The executions have been performed on a laptop PC with an Intel Core™ i5 M450 CPU at 2.40 GHz and 4 GB RAM, running Linux Mint 18.2. All the codes have been compiled with gcc 5.4.0 compiler and the same optimization options in all cases. The times are given in seconds and correspond to elapsed real time (not CPU time).

The computing times shown in this section are very small, and can thus be affected by uncontrolled random factors, making it difficult to obtain reliable measurements. For this reason, each simulation is repeated several times and the average computing time is obtained. The number of repetitions is shown in column *reps* of table 3 and depends on the time of a single simulation, being larger for networks that take less time. The process is repeated three times, taking the minimum of those averages as the final measurement.

TABLE 4. Time reduction, loop solver vs Epanet, in % of total simulation time

network	total	open	coeffs	linsolve	new q, h	blc	other
exnet	14.4%	-0.8%	11.7%	11.5%	-7.0%	-1.0%	0.1%
urb1	15.0%	0.4%	7.8%	19.1%	-7.1%	-5.2%	-0.1%
urb2	9.1%	0.0%	6.4%	13.0%	-9.3%	-1.0%	0.0%
bwsn2	11.0%	1.3%	11.4%	16.3%	-9.2%	-9.3%	0.5%
urb3	30.1%	11.7%	14.0%	14.3%	-8.9%	-1.5%	0.5%

Table 3 shows the computing time for Epanet 2.1, with the time for the whole simulation (column *total*) and the time for different tasks, namely: creating and initializing the data structures, including the sparse matrix representation (*open*), computing the linear system coefficients (*coeffs*), solving the linear system (*linsolve*), computing the new values of q and h (*new q, h*), and other tasks (*other*). The values for *coeffs*, *linsolve* and *new q, h* , correspond to the sum of the time spent in the task in all Newton-Raphson iterations of all time steps.

Table 4 presents the computing time reduction of the loop solver with respect to Epanet. The reduction is shown for each of the above mentioned tasks, with an additional task (*blc*) corresponding to computing the balanced flow vector \mathbf{q}^* at each time step, which is not needed in Epanet. The reduction is given in all cases as a percentage expressing the time decrement in the task with respect to Epanet's total computing time, with negative values indicating that the task was slower in the loop solver than in Epanet.

It can be seen that the identification of the independent loops, included in the task *open*, does not negatively affect performance and in some cases the task is faster in the loop solver. This is especially true for urb3, where the gain (11.7%) seems to be strangely high. Further analysis has revealed that in Epanet this time is dominated by node reordering (and symbolic factorization). For our networks, reordering consumes between 70% to 95% of the time in *open*, and presents a computing time proportional to nm . In the loop solver, the dominating part of *open* is the selection of independent loops, which takes from 50% to 75% of its time, with running time roughly proportional to the number of links belonging to at least one loop. The large size of network urb3 explains the high gain in this task for the network.

There is an important time reduction for the linear system solution, something to be expected because the sizes of the systems reduce considerably. The computation of the linear system coefficients also benefits from the use of the loop solver, due to the fact that some coefficients (those corresponding to links not included in any loop) only have to be computed once at the beginning of each time step.

The rest of this section compares the results of the symmetric approach for pressure-regulating devices presented in this paper with the non-symmetric approach presented in (Alvarruiz et al. 2015), in the context of the loop method. First, table 5 considers the residuals and number of iterations. The non-symmetric approach produces lower number of iterations for the networks shown in the table, and it also obtains lower flow residuals. This was to be expected, because it uses a more exact formulation. However, the non-symmetric solver failed to converge for network urb2, and it was necessary to use a non-zero value for the parameter "Damplimit" (see Epanet documentation) in network bwsn2 to achieve convergence with the same solver. The problems

TABLE 5. Residuals and number iterations of symmetric (s) and non-symmetric (ns) approaches.

Network	ϵ_h s	ϵ_h ns	ϵ_q s	ϵ_q ns	it s	it ns
exnet	$1.38 \cdot 10^{-3}$	$2.01 \cdot 10^{-3}$	$9.29 \cdot 10^{-5}$	$1.02 \cdot 10^{-6}$	108	93
urb1	$1.48 \cdot 10^{-3}$	$9.01 \cdot 10^{-3}$	$9.24 \cdot 10^{-5}$	$2.48 \cdot 10^{-6}$	1396	1275
bwsn2	$5.86 \cdot 10^{-2}$	$6.82 \cdot 10^{-2}$	$2.90 \cdot 10^{-3}$	$2.14 \cdot 10^{-6}$	812	772
urb3	$6.67 \cdot 10^{-2}$	$8.10 \cdot 10^{-2}$	$6.77 \cdot 10^{-3}$	$1.58 \cdot 10^{-6}$	186	139

TABLE 6. Time reduction, symmetric vs non-symmetric approach

network	total	open	coeffs	linsolve	new q, h	blc	other
exnet	-10.5%	0.1%	-9.1%	0.7%	-2.3%	0.1%	0.0%
urb1	-0.1%	0.0%	-3.2%	4.7%	-1.6%	0.0%	-0.1%
bwsn2	-0.9%	-0.2%	-1.2%	0.9%	-0.7%	0.2%	0.0%
urb3	-5.0%	0.7%	-12.5%	14.1%	-6.7%	-0.1%	-0.5%

seem to be that, due to interactions between the status of different control elements, the heuristic method to find the correct status of those elements is not working properly, causing some valves to oscillate constantly between statuses. This is especially an issue for network urb2, because it contains a large number of valves. It is not clear why the non-symmetric solver is more affected by this problem. A possible solution for this issue could be to use a relaxation parameter in the descent direction as shown in (Elhay et al. 2016).

Tables 6 and 7 focus on the computing time. The first table presents the computing time reduction of the symmetric approach with respect to the non-symmetric approach, considering the complete simulation, and Table 7 shows the time reduction in each Newton-Raphson iteration. In both tables, the reduction is expressed as a percentage of the computing time of the non-symmetric solver (the total computing time in table 6, and the computing time for an iteration in table 7), with negative values indicating that the symmetric solver was slower. The lower number of iterations of the non-symmetric approach explains the fact that this solver is faster in terms of total simulation time, although the difference is below 1% for networks urb1 and bwsn2. However, the symmetric approach is faster in each iteration, mainly because of the simpler and faster linear system solution. The gain per iteration depends mainly on the number of PRVs/PSVs in the network, as these

TABLE 7. Time reduction per iteration, symmetric vs non-symmetric approach

network	iter	coeffs	linsolve	new q, h	other
exnet	2.7%	0.1%	2.2%	0.3%	0.1%
urb1	8.3%	0.2%	8.0%	-0.1%	0.2%
bwsn2	3.1%	0.5%	1.8%	0.6%	0.2%
urb3	20.1%	0.9%	19.3%	-0.1%	0.0%

elements add the non-symmetric part to the linear system.

CONCLUSIONS AND FUTURE WORK

This paper shows that the presence of flow- and pressure-controlling valves is compatible with the loop method, keeping the matrix structure and symmetry, independently of their status. The formulation presented here for PRVs differs from (Alvarruiz et al. 2015) in that the whole system matrix is symmetric, which leads to a simpler implementation with faster solution of the linear systems, while also producing iterations closer to those obtained with Epanet.

The paper presents results for the approach, comparing it in terms of speed and accuracy with Epanet and with the non-symmetric approach of (Alvarruiz et al. 2015). For that purpose the non-symmetric formulation, previously written in Matlab, was implemented in C.

The authors show that some tasks such as the initial identification of loops, the computation of an auxiliary balanced flow and the update of flows and heads at each iteration, do not excessively penalize the total computing time, which is in fact up to 30% less than that of Epanet in some cases, due to the time reduction in the update and solution of the linear systems, and in the initial system reordering. The reduction in computing time depends on the size of the looped portion of the network, relative to its total size. In terms of accuracy, the symmetric formulation is as accurate as Epanet.

With respect to the non-symmetric approach of (Alvarruiz et al. 2015), the formulation presented here is simpler in its implementation and presents a lower computing time for each Newton-Raphson iteration. The non-symmetric approach required less iterations, but it also presented convergence problems for some of the networks tested.

Taking into account the above considerations, the symmetric approach presented here will be the choice for a new WDS simulation toolkit based on the loop method, currently under development by the authors.

REFERENCES

- Abraham, E. and Stoianov, I. (2015). "Efficient preconditioned iterative methods for hydraulic simulation of large scale water distribution networks." *Procedia Engineering*, 119, 623 – 632 Computing and Control for the Water Industry (CCWI2015).
- Abraham, E. and Stoianov, I. (2016). "Sparse null space algorithms for hydraulic analysis of large-scale water supply networks." *Journal of Hydraulic Engineering*, 142(3), 04015058.
- Alvarruiz, F., Martínez Alzamora, F., and Vidal, A. M. (2015). "Improving the efficiency of the loop method for the simulation of water distribution systems." *Journal of Water Resources Planning and Management*, 141(10), 04015019.
- Alvarruiz, F., Martínez Alzamora, F., and Vidal, A. M. (2017a). "Improving the performance of water distribution systems simulation on multicore systems." *The Journal of Supercomputing*, 73(1), 44–56.
- Alvarruiz, F., Martínez Alzamora, F., and Vidal, A. M. (2017b). "A toolkit for water distribution systems' simulation using the loop method and high performance computing." *Procedia Engineering*, 186(Supplement C), 303 – 310 XVIII Int. Conf. Water Dist. Systems, WDSA2016.
- Arsene, C., Al-Dabass, D., and Hartley, J. (2012). "A study on modeling and simulation of water distribution systems based on loop corrective flows and containing controlling hydraulics elements." *3rd Int. Conf. Intelligent Systems, Modelling and Simulation (ISMS), 2012*, 423–430.

- Ateş, S. (2017). “Hydraulic modelling of control devices in loop equations of water distribution networks.” *Flow Measurement and Instrumentation*, 53, 243 – 260.
- Bartolín, H., Martínez Alzamora, F., and Cortés, J. A. (2008). “Bringing up to date WDS models by querying. An EPANET-based GIS geodatabase.” *8th Annual Water Distribution Systems Analysis Symposium 2006*.
- Burger, G., Sitzenfrei, R., Kleidorfer, M., and Rauch, W. (2016). “Quest for a new solver for EPANET 2.” *Journal of Water Resources Planning and Management*, 142(3), 04015065.
- Creaco, E. and Franchini, M. (2014). “Comparison of Newton-Raphson global and loop algorithms for water distribution network resolution.” *Journal of Hydraulic Engineering*, 140(3), 313–321.
- Creaco, E. and Franchini, M. (2015). “The identification of loops in water distribution networks.” *Procedia Engineering*, 119, 506 – 515 Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management.
- Deuerlein, J., Cembrowicz, R., and Dempe, S. (2005). “Hydraulic simulation of water supply networks under control.” *World Water and Environmental Resources Congress 2005*.
- Deuerlein, J., Simpson, A., and Dempe, S. (2009a). “Modeling the behavior of flow regulating devices in water distribution systems using constrained nonlinear programming.” *Journal of Hydraulic Engineering*, 135(11), 970–982.
- Deuerlein, J., Simpson, A., and Gross, E. (2009b). “The never ending story of modeling control-devices in hydraulic systems analysis.” *Water Distribution Systems Analysis 2008*.
- Elhay, S., Piller, O., Deuerlein, J., and Simpson, A. R. (2016). “A robust, rapidly convergent method that solves the water distribution equations for pressure-dependent models.” *Journal of Water Resources Planning and Management*, 142(2), 04015047.
- Elhay, S., Simpson, A., Deuerlein, J., Alexander, B., and Schilders, W. (2014). “Reformulated co-tree flows method competitive with the global gradient algorithm for solving water distribution system equations.” *Journal of Water Resources Planning and Management*, 140(12).
- Epp, R. and Fowler, A. G. (1970). “Efficient code for steady-state flows in networks.” *Journal of the Hydraulics Division*, 96(1), 43–56.
- Farmani, R., Savic, D. A., and Walters, G. A. (2005). “Evolutionary multi-objective optimization in water distribution network design.” *Engineering Optimization*, 37(2), 167–183.
- Guidolin, M., Kapelan, Z., and Savic, D. (2013). “Using high performance techniques to accelerate demand-driven hydraulic solvers.” *Journal of Hydroinformatics*, 15(1), 38–54.
- Jeppson, R. W. (1976). *Analysis of flow in pipe networks*. Ann Arbor Science.
- Mair, M., Sitzenfrei, R., Kleidorfer, M., and Rauch, W. (2014). “Performance improvement with parallel numerical model simulations in the field of urban water management.” *Journal of Hydroinformatics*, 16(2), 477–486.
- Ostfeld, A., Uber, J., Salomons, E., et al. (2008). “The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms.” *Journal of Water Resources Planning and Management*, 134(6), 556–568.
- Piller, O. and van Zyl, J. (2014). “Modeling control valves in water distribution systems using a continuous state formulation.” *Journal of Hydraulic Engineering*, 140(11).
- Rossman, A. L. (2000). *Epanet 2 Users manual*. Water Supply and Water Resources Division, US Environment Protection Agency.
- Rossman, L. A. (1999). “Computer models/EPANET.” *Water Distribution Systems Handbook*, L. W. Mays, ed., McGraw-Hill Companies, Chapter 12.
- Simpson, A. (1999). “Modeling of pressure regulating devices: The last major problem to be solved

- in hydraulic simulation.” *WRPMD’99: Preparing for the 21st century*.
- Todini, E. and Pilati, S. (1988). “A gradient algorithm for the analysis of pipe networks.” *Computer Applications in Water Supply: Vol. 1—Systems Analysis and Simulation*, B. Coulbeck and C.-H. Orr, eds., Research Studies Press Ltd., 1–20.
- Todini, E. and Rossman, L. A. (2013). “Unified framework for deriving simultaneous equation algorithms for water distribution networks.” *Journal of Hydraulic Engineering*, 139(5), 511–526.