

Proceedings

# Pressure-Driven Simulation of Water Distribution Networks: Searching for Numerical Stability <sup>†</sup>

João Muranho <sup>1,2,\*</sup>, Ana Ferreira <sup>3</sup>, Joaquim Sousa <sup>2,4</sup>, Abel Gomes <sup>1,5</sup> and Alfeu Sá Marques <sup>2,6</sup>

<sup>1</sup> Departamento de Informática, Universidade da Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal; agomes@di.ubi.pt

<sup>2</sup> INESC Coimbra, DEEC, Universidade de Coimbra, Rua Sílvio Lima, Polo II 3030-290 Coimbra, Portugal; jjoseng@isec.pt (J.S.); jasm@dec.uc.pt (A.S.M.)

<sup>3</sup> Instituto Politécnico de Castelo Branco, EST, U.T.C. Engenharia Civil, Av. do Empresário, 6000-767 Castelo Branco, Portugal; amferreira@ipcb.pt

<sup>4</sup> Departamento de Engenharia Civil, Instituto Politécnico de Coimbra, ISEC, Rua Pedro Nunes, 3030-199 Coimbra, Portugal

<sup>5</sup> Instituto de Telecomunicações, Universidade da Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal

<sup>6</sup> Departamento de Engenharia Civil, Universidade de Coimbra, Rua Luís Reis Santos, Polo II, 3030-788 Coimbra, Portugal

\* Correspondence: jmuranho@ubi.pt; Tel.: +351-275242081

† Presented at the 4th EWaS International Conference: Valuing the Water, Carbon, Ecological Footprints of Human Activities, Online, 24–27 June 2020.

Published: 4 September 2020

**Abstract:** EPANET uses a demand-driven approach to compute pressures and flows in the water distribution system. The demand-driven approach (DDA) assumes that the required demand is always fully satisfied no matter the existing pressure. In scenarios of pressure-deficient conditions the DDA results are not accurate, and a pressure-driven approach (PDA) is needed. Frequently, the PDA is accomplished by using equations that compute the available demand/leakage as a function of the current pressure. However, embedding such equations into the solver introduces convergence problems. This paper details the actions taken in WaterNetGen—an EPANET extension—to bring numerical stability to the pressure-driven solver, namely, by smoothing the pressure–demand/leakage relationship and the pump curve.

**Keywords:** demand-driven; EPANET; hydraulic solver; water leakage; pressure-driven; smooth functions; WaterNetGen

---

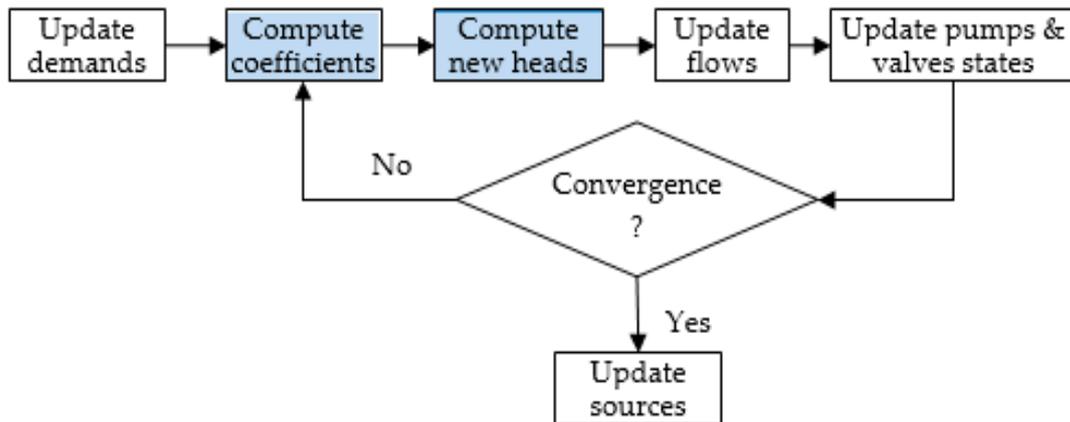
## 1. Introduction

EPANET [1] is a public domain Windows software developed by the U.S. Environmental Protection Agency (USEPA). EPANET performs extended period simulations of the hydraulic and water quality behaviour of drinking water distribution systems. EPANET uses a link-node formulation, where consumption along the pipes is assigned to their end-nodes and defined as demand. To compute the nodal heads and the link flows, EPANET iteratively solves a linear system of equations until some convergence criterion is achieved—see Figure 1 and Listing 1a.

A linear system of equations results from the linearization of the mass conservation and energy equations by the Newton–Raphson method [2]. The iterative computation of Figure 1 stops when some level of accuracy is reached. Thus, the heads (and flows) are computed to satisfy the desired demand, which can lead to solutions with negative pressures. In other words, the demand is always satisfied independently of the available nodal pressure, even in the presence of negative pressures—

a physical impossibility. Due to this behaviour, it is said that EPANET performs a Demand-Driven Analysis (DDA).

A better approach is to compute the demand as a function of the current pressure—Pressure-Driven Analysis (PDA). In this case, the iterative process must include the “Update demands” block from Figure 1—see Listing 1b.



**Figure 1.** Iterative computation of heads and flows.

|                       |                            |
|-----------------------|----------------------------|
|                       | REPEAT                     |
| update_demands();     | update_demands();          |
| REPEAT                | new_coeffs();              |
| new_coeffs();         | new_heads();               |
| new_heads();          | update_flows();            |
| update_flows();       | update_pumpsvalves();      |
| update_pumpsvalves(); | apply_correctiveactions(); |
| UNTIL converges;      | UNTIL converges;           |
| Update_sources();     | Update_sources();          |
| (a)                   | (b)                        |

**Listing 1.** Pseudocode for the iterative computation of heads and flows: (a) Demand-Driven; (b) Pressure-Driven.

The computation of the demand inside the loop frequently introduces numerical instability, which requires some corrective actions. In [3], the authors used a pressure–demand/leakage relationship to compute the available demand/leakage and employ relaxation coefficients to deal with these convergence difficulties. However, the authors concluded that in some cases the relaxation coefficients are not enough to guarantee convergence. This paper proposes a smoothing technique to build a differentiable pressure–demand/leakage relationship to improve the convergence of the iterative process.

The next section briefly introduces the Global Gradient Algorithm [2], its pressure-driven version [4] and the proposed smoothing technique to be applied to demands, leakages and pump curves computation. Section 3 includes some results. Section 4 concludes the paper and discusses some future developments.

## 2. Materials and Methods

At the core of the EPANET solver is the Global Gradient Algorithm (GGA). Section 2.1 introduces the EPANET solver, Section 2.2 describes the pressure-dependent changes to the GGA and Section 2.3 describes the smoothing technique proposed.

2.1. EPANET Solver

Considering a hydraulic network composed of  $np$  pipes with unknown flow rates,  $nm$  nodes with unknown heads and  $n0$  nodes with known heads, the GGA states:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & 0 \end{bmatrix} \begin{bmatrix} Q \\ H \end{bmatrix} = \begin{bmatrix} -A_{10}H_0 \\ -q \end{bmatrix}, \tag{1}$$

where  $Q = [Q_1, Q_2, \dots, Q_{np}]^T$  is the column vector of unknown pipe flow rates;  $H = [H_1, H_2, \dots, H_{nm}]^T$  is the column vector of unknown nodal heads;  $H_0 = [H_{0_1}, H_{0_2}, \dots, H_{0_{n0}}]^T$  is the column vector of known nodal heads; and  $q = [q_1, q_2, \dots, q_{mi}]^T$  is the column vector of known demands.

In Equation (1),  $A_{11}$  is a  $np \times np$  diagonal matrix whose elements correspond to the pipe head losses;  $A_{12} = A_{21}^T$  and  $A_{10} = A_{01}^T$  are the topological incidence matrices of size  $np \times nm$  and  $np \times n0$ , respectively.

Assuming a head loss formula as follows:

$$h_i = R_i \cdot |Q_i|^{n_i-1}, \tag{2}$$

the  $A_{11}$  matrix takes the form

$$A_{11} = \begin{bmatrix} R_1|Q_1|^{n_1-1} & & & \\ & R_2|Q_2|^{n_2-1} & & \\ & & \ddots & \\ & & & R_{np}|Q_{np}|^{n_{np}-1} \end{bmatrix}, \tag{3}$$

Considering  $D_{11}$  as the derivative of  $A_{11}$  with respect to pipe flows, the iterative formulation to find a solution of (1) can be stated as follows:

$$\begin{aligned} A^k &= A_{21}(D_{11}^k)^{-1} A_{12} \\ F^k &= (A_{21}Q^k - q) - A_{21}(D_{11}^k)^{-1} (A_{10}H_0 + A_{11}^k Q^k) \\ H^{k+1} &= (A^k)^{-1} F^k \\ Q^{k+1} &= Q^k - (D_{11}^k)^{-1} (A_{11}^k Q^k + A_{12}H^{k+1} + A_{10}H_0) \end{aligned} \tag{4}$$

The computation of the new head,  $H^{k+1}$ , corresponds to solving a system of linear equations  $Ax = b$ , with  $A = A^k$ ,  $x = H^{k+1}$  and  $b = F^k$ , where the coefficients matrix  $A$  is sparse, symmetric and positive definite—see Figure 2. EPANET solves this linear system through a direct method ( $LL^T$ —Cholesky factorization) customised to sparse matrices.

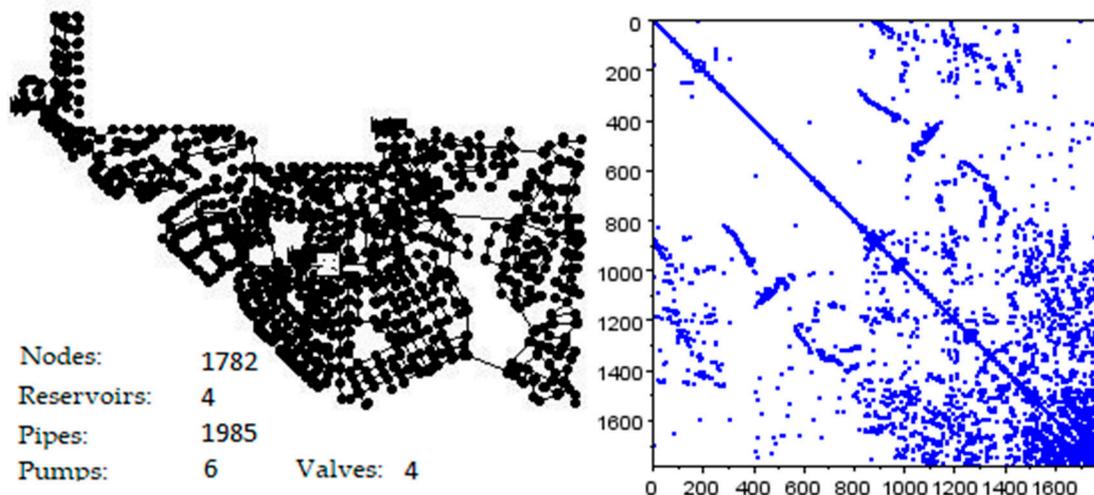


Figure 2. Wolf-Cordera Ranch network [5] and its sparse matrix.

The EPANET implementation of the linear system solver first applies an ordering (Minimum Degree) algorithm to reduce the fill-in problem linked to the factorization of sparse matrices [6]. Figure 2 shows the matrix sparsity pattern that results from the ordering phase (and before numerical factorization) of the benchmark network Wolf–Cordera Ranch.

### 2.2. Pressure-Dependent Demand

There are two types of pressure dependent demands to be considered: the demand itself and pipe leakage. The pressure-dependent demand for a node  $i$  can be computed by the following relationship [7]:

$$q_i^{avl} (P_i) = q_i^{req} \times DSR (P_i, P_i^{min}, P_i^{ref}, \alpha), \tag{5}$$

where  $P_i^{ref}$  is the reference (or service) pressure necessary to fully satisfy the required demand  $q_i^{req}$ ,  $P_i^{min}$  is the pressure below which no water can be supplied,  $P_i$  is the current pressure at node  $i$  and DSR (Demand Satisfaction Ratio) is the ratio between the available and the required demands defined by

$$DSR(P_i, P_i^{min}, P_i^{ref}, \alpha) = \begin{cases} 1 & P_i \geq P_i^{ref} \\ \left( \frac{P_i - P_i^{min}}{P_i^{ref} - P_i^{min}} \right)^\alpha & P_i^{min} < P_i < P_i^{ref} \\ 0 & P_i \leq P_i^{min} \end{cases}, \tag{6}$$

The exponent of the pressure–demand relationship,  $\alpha$ , typically assumes the value 0.5, but should be set by calibration. Figure 3 illustrates the DSR for different values of the exponent and considering  $P_i^{min} = 3$  m and  $P_i^{ref} = 14$  m.

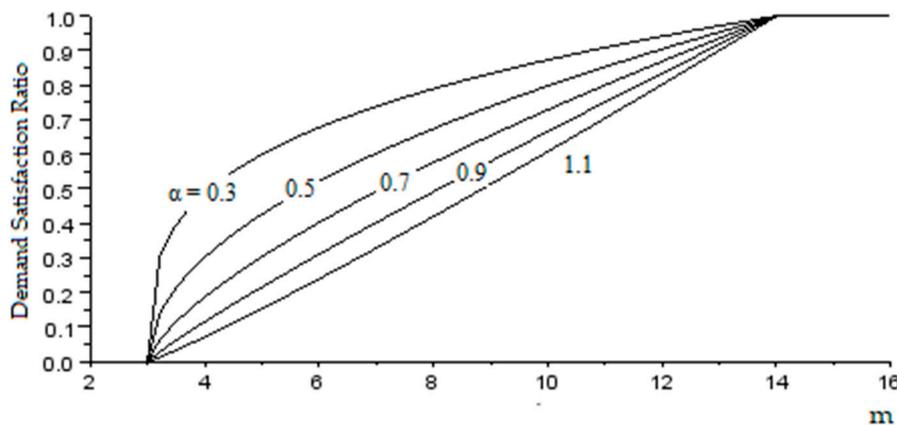


Figure 3. Demand satisfaction ratio curves for different exponent ( $\alpha$ ) values.

Leakage also depends on the pressure. In fact, leakage increases when pressure rises and decreases when pressure falls. The pipe leakage can be expressed by the sum of two components: background leakage and bursts leakage. The pressure–leakage relationship can be set as follows [8]:

$$q_k^{leak} (P_k) = \begin{cases} \beta_k l_k (P_k)^{\alpha_k} + C_k (P_k)^{\delta_k} & P_k > 0 \\ 0 & P_k \leq 0 \end{cases}, \tag{7}$$

where  $q_k^{leak}$  is the total leakage flow along pipe  $k$ ;  $l_k$  is the length of pipe  $k$ ;  $\alpha_k$  and  $\beta_k$  are parameters of the background leakage model;  $C_k$  and  $\delta_k$  are parameters of the bursts leakage model (classical orifice flow formulas); and  $P_k$  is the average pressure in pipe  $k$  computed as the arithmetic mean of the pressure values of its end nodes. According to Lambert [9], the  $\alpha_k$  parameter can take values between

0.5 and 2.5, depending on the leak type. The  $\beta_k$  parameter is related to the pipe material deterioration, and its value must be set by calibration (initial values can be set around  $10^{-7}$ ).

The iterative formulation for the PDA is as follows:

$$\begin{aligned}
 DDL_{22}^k &= D_{22}^k + DL_{22}^k \\
 A^k &= A_{21}(D_{11}^k)^{-1} A_{12} - DDL_{22}^k \\
 F^k &= [A_{21}Q^k - (q_{avl}^k - q_{leak}^k)] - A_{21}(D_{11}^k)^{-1} (A_{10}H_0 + A_{11}Q^k) - DDL_{22}^k H^k, \\
 H^{k+1} &= (A^k)^{-1} F^k \\
 Q^{k+1} &= Q^k - (D_{11}^k)^{-1} (A_{11}Q^k + A_{12}H^{k+1} + A_{10}H_0)
 \end{aligned}
 \tag{8}$$

where  $DL_{22}$  and  $D_{22}$  are the derivatives of  $q_{leak}$  and  $q_{avl}$  functions, Equations (5)–(7), with respect to pipe pressures and nodal pressures, respectively, and  $k$  is the current step.

WaterNetGen [4,10] uses relaxation coefficients to improve PDA convergence.

### 2.3. Function Smoothing

Despite the incorporation of relaxation coefficients in the pressure-dependent formulation, the PDA still evidences problems in some cases [3]. One source of such problems is the inclusion of the derivative terms of Equations (5)–(7) in the iterative process. For example, considering expression (6), when the current pressure approaches the minimum required pressure ( $P^{min}$ ), the derivative tends to infinity. These derivative terms should be smoothed.

One way of dealing with ill-conditioned behaviour derivative terms is to cut part of the pressure–demand relationship extremity. Figure 4 illustrates the idea for the DSR (6): the extremity is replaced by another well-behaved curve, chosen to be differentiable.

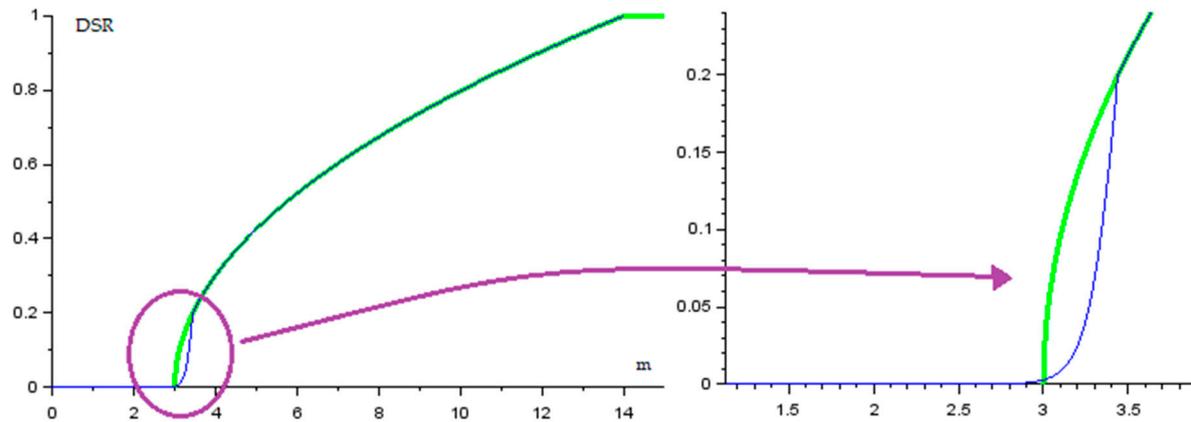


Figure 4. Smoothing the demand satisfaction ratio curve.

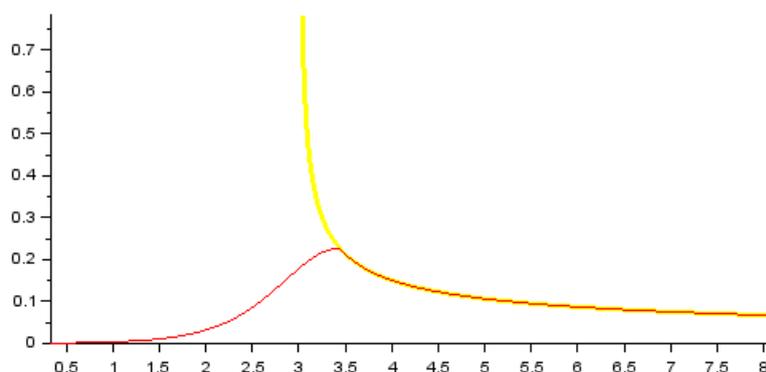
The logistic sigmoid function (Equation (9)) is a good candidate to this operation, because it is differentiable and can be parametrised. Thus, the logistic function can be implanted at an appropriate location of the original function, for example, at a point  $X_0$  corresponding to a DSR of 20%.

$$\begin{aligned}
 f(x) &= \frac{1}{1 + e^{-\frac{x - X_0}{K}}} \\
 f'(x) &= \frac{f(x)(1 - f(x))}{K}
 \end{aligned}
 \tag{9}$$

The value of  $K$  can be computed to guarantee smooth derivatives too:

$$K = \frac{(P_i^{ref} - P_i^{min})^\alpha}{2 \cdot \alpha (X_0 - P_i^{min})^{\alpha-1}}, \tag{10}$$

Figure 5 shows the graph of derivatives of the original and the smoothed functions and yellow and red lines, respectively.



**Figure 5.** Demand satisfaction ratio curve derivatives: original (yellow) and smoothed (red).

The above function smoothing technique is a little elaborate to be applied to pressure–leakage functions. Thus, following the same idea of function smoothing, a pragmatic approach results: use the cubic Hermite interpolation [11]. Cubic Hermite curves provide continuity of first-order derivatives and match the set of (chosen) slopes. They are simple to calculate and only need two control points and two control tangents. Accordingly, two points can be considered  $x_1 = P^{min}$  and  $x_2 = x_1 + \Delta$ , with function values  $f_1 = DSR(x_1, P^{min}, P^{max}, 0.5) = 0.0$  and  $f_2 = DSR(x_2, P^{min}, P^{max}, 0.5)$ . The  $\Delta$  can be selected to produce a good smooth derivative curve (Figure 5). The corresponding tangent points,  $df_1$  and  $df_2$ , can be set to a small value (for example,  $1 \times 10^{-10}$ ) and to the derivative of DSR on  $x_2$ , respectively. The curve coefficients ( $a, b, c, d$ ) can be obtained by solving the following linear system:

$$\begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ 3x_1^2 & 2x_1 & 1 & 0 \\ 3x_2^2 & 2x_2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ df_1 \\ df_2 \end{bmatrix}, \tag{11}$$

which is equivalent to computing

$$\begin{aligned} a &= \frac{2(f_2 - f_1) - (x_2 - x_1)(df_1 + df_2)}{x_1^3 - x_2^3 + 3x_1x_2(x_2 - x_1)} \\ b &= \frac{(df_2 - df_1) - 3(x_2^2 - x_1^2) \cdot a}{2(x_2 - x_1)}, \\ c &= df_1 - 3x_1^2 \cdot a - 2x_1 \cdot b \\ d &= f_1 - x_1^3 \cdot a - x_1^2 \cdot b - x_1 \cdot c \end{aligned} \tag{12}$$

In WaterNetGen, the cubic Hermite curve, with the necessary adjustments, is applied to both extremities of the DSR function, to leakage (background and burst) components and to the pump curves.

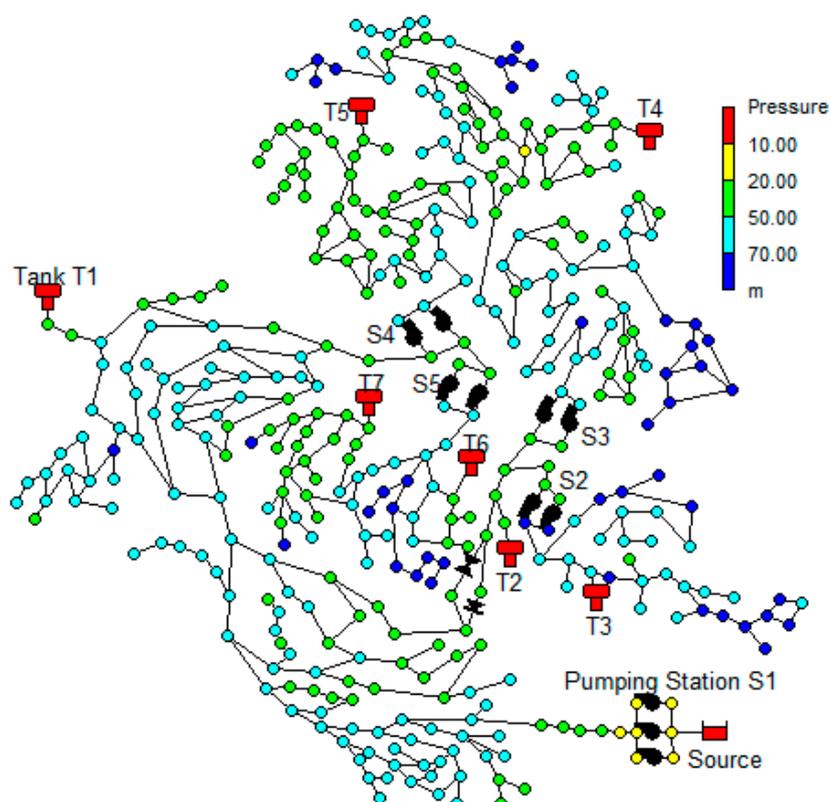
### 3. Results

WaterNetGen is an EPANET extension that allows both demand-driven and pressure-driven simulations. The PDA is achieved with pressure–demand/leakage relationships (Equations (6)–(8)) and with relaxation coefficients. However, in a previous work [3] the authors found that in certain

cases the process did not converge. These cases are related to valve interactions and to problems with the derivative terms.

Figure 6 shows the C-Town network, used in the Battle of Background Leakage Assessment for Water Networks (BBLAWN) [12], where WaterNetGen was used to define a pump scheduling in order to minimise the operational costs (energy) and background leakage. In some runs WaterNetGen revealed some difficulties and did not converge.

The tool with the proposed smoothing technique is running again and almost does a decent job. However, there remain some cases where the process does not converge. That is, although there are improvements in the number of successful runs, there are still cases that deserve more research.



**Figure 6.** The C-Town network has one reservoir, seven tanks, eleven pumps, one control valve, one check valve, 432 pipes and 388 junctions.

One final note about the research. The EPANET 2.2 [13], released on December 2019, comes with PDA capability based on Equations (5) and (6). This new version was used to simulate the behaviour of the BBLAWN network, but the result is always “WARNING: System unbalanced at...”. Therefore, this new version of EPANET still needs some reworking to deal with more complex networks.

#### 4. Discussion and Conclusions

EPANET is the outstanding tool to study water distribution networks. However, EPANET is built around the demand-driven approach, which is not suitable to simulate pressure-deficient scenarios or pipe leakage. For these cases, the demand cannot be considered as constant known value, but instead must be considered as pressure dependent. A common way of specifying pressure-dependent demand is to consider a pressure–demand relationship and use it in the (iterative) simulation process.

The embedding of the pressure–demand relationship into the iterative process introduces instabilities in the solver. In the past, the authors tried to circumvent these instabilities with the inclusion of relaxation coefficients, which revealed to be essential to the convergence of the solver. Nevertheless, there persist some cases in which the solver does not converge.

The smoothing technique proposed in this work allowed one to overcome some more cases of lack of convergence. However, it is not a finished subject; there remain some cases not solved, so more research is needed.

In the future, the authors plan to add a backtracking algorithm, to assure global convergence for the Newton–Raphson method, and exploit it in combination with the techniques here introduced.

**Author Contributions:** The authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by Portuguese Foundation for Science and Technology under project UIDB/00308/2020.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Rossman, L. *EPANET 2 User's Manual*; EPA/600/R-00/057; U.S. Environmental Protection Agency: Cincinnati, OH, USA, 2000.
2. Todini, E.; Pilati, S. A gradient algorithm for the analysis of pipe networks. In *Computer Applications in Water Supply, Volume I—Systems anaLysis and Simulation*; Coulbeck, B., Orr, C.H., Eds.; John Wiley & Sons Research Studies Press: London, UK, 1988; pp. 1–20.
3. Muranho, J.; Ferreira, A.; Sousa, J.; Gomes, A.; Sá Marques, A. Convergence issues in the EPANET solver. *Procedia Eng.* **2015**, *119*, 700–709. doi:10.1016/j.proeng.2015.08.923.
4. Muranho, J.; Ferreira, A.; Sousa, J.; Gomes, A.; Sá Marques, A. Pressure-dependent Demand and Leakage Modelling with an EPANET Extension—WaterNetGen. *Procedia Eng.* **2014**, *89*, 632–639. doi:10.1016/j.proeng.2014.11.488.
5. Wolf-Cordera-Ranch Benchmark Example. 2020. Available online <https://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/expansion/wolf-cordera-ranch.php> (accessed on 12 February 2020).
6. George, A.; Liu, J.W.-H. *Computer Solution of Large Positive Definite Systems*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1981.
7. Wagner, J.M.; Shamir, U.; Marks, D.H. Water distribution reliability - Simulation methods. *J. Water Resour. Plan. Manag.* **1988**, *114*, 276–294.
8. Germanopoulos, G. A technical note on the inclusion of pressure dependent demand and leakage terms in water supply network models. *Civ. Eng. Syst.* **1985**, *2*, 171–179.
9. Lambert, A.O. What do we know about pressure: Leakage relationships in distribution systems? In *Proceedings of IWA Specialised Conference: System Approach to Leakage Control and Water Distribution Systems Management*, Brno, Czech Republic, 16–18 May 2001.
10. Muranho, J.; Ferreira, A.; Sousa, J.; Gomes, A.; Sá Marques, A. WaterNetGen: An EPANET extension for automatic water distribution network models generation and pipe sizing. *Water Supply* **2012**, *12*, 117–123.
11. Schultz, M.H. *Spline Analysis*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1973.
12. Sousa, J.; Muranho, J.; Sá Marques, A.; Gomes, R. Optimal management of water distribution networks with simulated annealing: The C-Town problem. *J. Water Res. Plan. Manag.* **2016**, *142*, C4015010.
13. Muranho, J.; Ferreira, A.; Sousa, J.; Gomes, A.; Sá Marques, A. An interface extension for the new EPANET 2x. In *Proceedings of the 4th EWaS—Efficient Water Systems International Conference*, Corfu Island, Greece, 24–27 June 2020.

