ONLINE SIMPLIFICATION OF WATER DISTIRIBUTION NETWORK MODELS

Daniel Paluszczyszyn, Piotr Skworcow and Bogumil Ulanicki
Water Software Systems, De Montfort University, Leicester, UK
paluszcol@dmu.ac.uk, pskworcow@dmu.ac.uk, bul@dmu.ac.uk

Abstract

This paper presents an implementation of the simplification algorithm of water distribution network (WDN) models for the purpose of inclusion to the online optimisation strategy for the energy and leakage management in WDN, formulated within a model predictive control framework. The advantage of the online model reduction is adaptation to abnormal situations and structural changes in a network. The implementation was carried out with the utilisation of nowadays parallel programing techniques to distribute the simplification tasks across multiple CPU treads. This resulted in significant reduction of the computational time required for the simplification process of the large–scale WDN models. The authors also highlighted a problem of the energy distribution when the reduced and original models were compared.

Keywords
Implementation, optimisation, parallel programing, simplification, water distribution network.

1. INTRODUCTION

Nowadays it is common that water distribution network (WDN) models consist of thousands of elements to accurate replicate the hydraulic behaviour and topographical layout of real WDNs. This approach is appropriate for simulation purposes, however online optimisation tasks are much more complex and simplified models are required. There are different techniques of model reduction, but outcome of all of these methods is a hydraulic model with a smaller number of components than the prototype. The main aim of a reduced model is to preserve the nonlinearity of the original network and approximate its operation accurately under different conditions. The accuracy of the simplification depends on the model complexity and selected method. The common approaches to model reduction are skeletonization and variable elimination.

The skeletonization is the process of selecting for inclusion in the model only the parts of the hydraulic network that have a significant impact on the behaviour of the WDN [1] e.g. use of equivalent pipes in place of numbers of pipes connected in parallel and/or in series. In [2] authors found that that under normal demands, they could remove a large number of pipes and still not affect pressure significantly. However the skeletonization is not a single process but several different low-level element removal processes that must be applied in series. This makes difficult the utilisation of this technique for the online optimisation purposes.

In [3] authors presented an automated skeletonization methodology that can be used to achieve reduced models of WDN that accurately reproduce both, the hydraulics and non-permanent water quality parameters of the prototype. The proposed methodology was based on the resilience concept; by using the resilience index as selection criterion to remove pipes from the prototype, reduced models that simulate the hydraulics of the real network were achieved. However this method is focused on the pipes removal only.

Variable elimination is based on a mathematical formalism. A WDN mathematical model is a system of non-linear differential algebraic equations [4]. Some of variables can be eliminated from these equations using an algorithm, thus reducing the size of a model. In [5] authors proposed a mathematical method for the reduction of network models described by a large-scale system of non-linear differential algebraic equations. The algorithm involves linearisation, Gaussian elimination, and reconstruction of a reduced nonlinear model.

This paper presents an implementation of the simplification algorithm developed in [5]. The purpose of the implementation is integration the model reduction module with the online optimisation strategy developed for an energy and leakage management in WDN [6]. Additionally, the paper included the utilisation of the today's programing features and techniques to decrease the computational time required for the simplification process.

The paper is structured as follows. Section 2 contains an outline of the optimisation strategy and its requirements for the model simplification algorithm. Section 3 describes implementation of a WDN model reduction method. The results obtained to date are discussed in Section 4. Conclusions are given in Section 5.

2. PROBLEM FORMULATION

Optimisation studies of medium and large-scale WDN are typically carried out offline. This means that any changes to the water network require significant changes in the optimisation model, which leads to high cost of system maintenance. In authors [6] proposed a methodology for an online energy and leakage management in WDN, formulated within a model predictive control framework. The control strategy calculates of control actions, i.e. time schedules for pumps, valves and sources, to minimise the costs associated with energy used for water pumping and treatment and water losses due to leakage, whilst satisfying all operational constraints. The proposed control scheme is illustrated in Figure 1. The model predictive controller (MPC) computes the control actions based on the telemetry readings, provided by the SCADA systems, and constrains and boundary conditions, specified by operator, and future demands predicted by the demand forecaster. For more detailed description with a case study see [6].

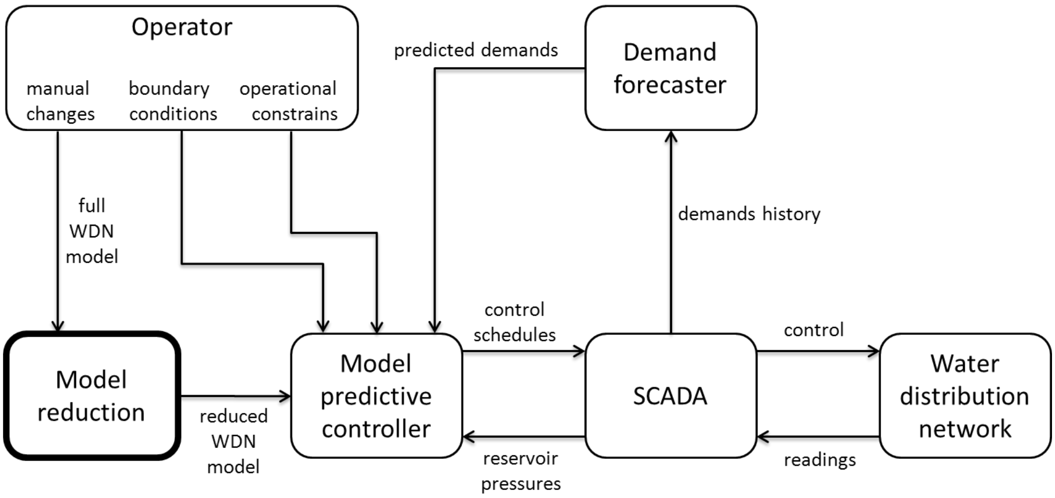


Figure 1. The control scheme.

The approach proposed by authors in [6] is a model-based. Due to fact that WDN models can consist of thousands of elements, each with a hydraulic equation, along with the MPC requirements for a computational power forced an inclusion a WDN model reduction algorithm to the proposed scheme. It was essential that reduced model preserves the WDN nonlinearities and approximates its operations accurately under different conditions. Moreover the reduction algorithm should be suitable for an online calculation. The simplification method chosen, presented in [5], is a mathematical method for the reduction of network models described by a large-scale system of non-linear differential algebraic equations. The approach proceed by the following steps: full nonlinear model formulation, model linearisation at specified time, linear model reduction using Gaussian elimination and nonlinear reduced model reconstruction. The method was successfully implemented and tested on many WDNs. However in this paper the reduction algorithm was implemented to meet the requirements of the online optimisation strategy.

3.1 Requirements

The main module of control strategy illustrated in Figure 1 was implemented in C# but it also employed the open-source hydraulic simulator called EPANET [7], mathematical modelling language called GAMS [8] and nonlinear programming solver CONOPT [9]. To ensure a compatibility and future integration the implementation of the model reduction algorithm needed to consider communication and data structure standards used by the mentioned tools.

The idea of an online optimisation required the simplification process to be completed within specified time to allow the controller to compute the control schedules.

During the simplification process, nodes are removed and associated demands are weighted distributed based on pipes conductance. For the control purposes it was necessary to log the demands allocation.

3.2 Tools

In order to satisfy the mentioned requirements for the implementation several approaches were considered. After further review of potential approaches it was decided to use the following tools.

To develop algorithm in C# programming language the Microsoft development environment Visual Studio 2010 was used. There are other ways to build C# programs, but Visual Studio is the most widely used and it simplifies the creation, debugging, and deployment of applications on a variety of platforms. Visual Studio 2010 comes with integrated support for .NET 4.0 Framework, which enhanced the parallel programming by providing a new runtime, new class library types, and new diagnostic tools. These features allowed writing the scalable parallel code without having to work directly with threads or the thread pool and therefore increase the performance of any computationally-intensive algorithms [10].

EPANET is an open-source software that allowed to perform extended period simulation of hydraulic and water quality behaviour within pressured pipe networks [7]. Initially designed to be a research tool quickly became widely used standard for WDN modelling, simulation and analysis. EPANET provided compatibility with INP format as it is a commonly recognized file format to store WDS models.

3.3 Implementation

The implementation was carried out based on the process illustrated in Figure 2. It involved initial simulation, preparation of the nonlinear model, indication elements to retain, application the Gaussian elimination procedure and generation a reduced nonlinear model.

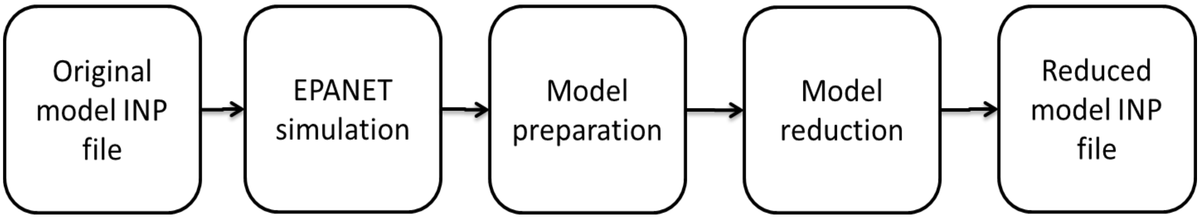


Figure 2. The model reduction procedure.

The WDN model simplification is not a straightforward process. It requires knowledge about the WDN to retain network elements with a significant importance in order to preserve hydraulic characteristics for wide range of operating conditions. A typical hydraulic simulation model contains thousands of pipes but only few reservoirs, pumps or control elements. Therefore, it is a common simplification strategy to reduce the number of pipes and nodes only and retain all the important elements. Thus the model was split up into two sub-models. One sub-model, containing pipes and nodes, is simplified, and afterwards, reunited with other part to form the reduced WDN model. This process is illustrated in Figure 3.

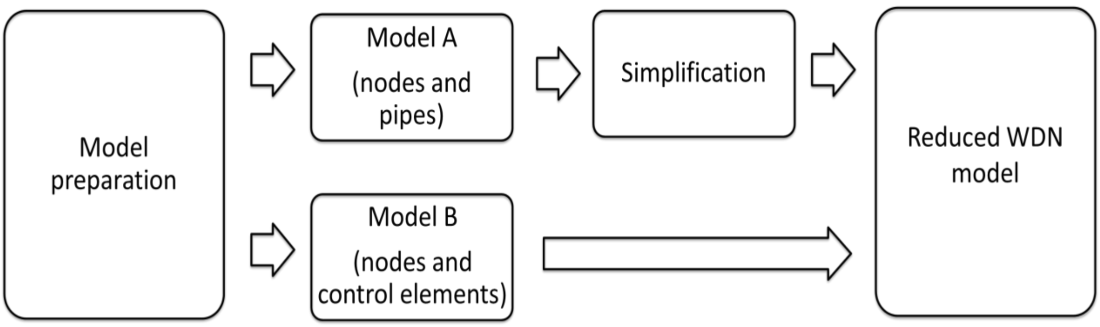


Figure 3. Model structure during simplification.

The initial application performed the simplification process as described, however the reduction process time was long i.e. for the large-scale networks could take up to few hours. Such time was not satisfactory therefore it was decided to investigate the parallel programing to exploit the potential of multicore CPUs.

Nowadays the most of computers and workstations have two or more CPU cores that allow multiple threads to be executed simultaneously. Moreover computers in the near future are expected to have significantly more cores. To take advantage of this hardware feature it was decided to parallelize the simplification algorithm code to distribute work across multiple processors. For this purpose the Visual Studio 2010 and the .NET Framework 4 libraries were employed. The results of inclusion of the parallel programing techniques drastically reduced the simplification process time. Table 1 contains simplification times performed on workstation powered by Intel® Core^{TM} i7 980X processor. The tests were performed on the large-scale network consists of 3535 nodes, 3279 pipes, 12 tanks, 5 reservoirs, 19 pumps and 418 valves.

Table 1. Number of used CPU thread versus time taken to complete the simplification process.

CPU threads	Simplification process time
1	1h 36min 01s
2	1h 13min 37s
4	0h 36min 57s
12	0h 12min 38s

4. RESULTS

The final simplification module was tested on five WDNs models representing real networks with different sizes and topographic characteristics. The details of the networks are summarised in Table 2.

The simplification algorithm performed as requested; i.e. all the reduced models adequately replicate hydraulic behaviour of the original model with an average accuracy less than 3%. However, complex and large WDNs modelled in EPANET often contains rules and controls that deteriorate the accuracy of the simplification. It is highly suggested to eliminate controls and rules and use the patterns resulting from the simulation and associate them with WDN elements. Such approach provided with hydraulic benchmark when original and simplified models were to be compared.

Table 2. Water distribution networks details

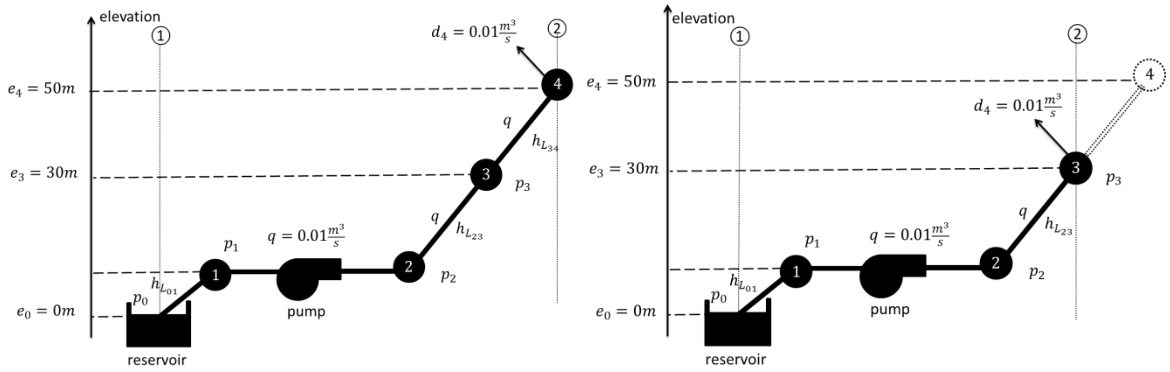
Network elements	WDN 1	WDN 2	WDN 3	WDN 4	WDN 5
	Before	*simplification*			
Nodes	166	922	1009	1527	3535
Pipes	200	690	1102	1611	3279
Tanks	1	0	3	36	12
Reservoirs	1	2	2	1	5
Pumps	1	0	1	5	19
Valves	0	289	11	107	418
	After	*simplification*			
Nodes	5	589	78	359	1023
Pipes	2	618	167	214	1340

Initially only the hydraulics analysis was performed in order to validate exactness of reduced models to reproduce the prototypes characteristics. However it was observed that despite of their hydraulics equivalence the energy distribution was not consistent when the principle of conservation of energy was employed. For convenience within a hydraulic analysis, the equation is written in terms of head as follows:

$$e_1 + \frac{p_1}{\gamma} + \frac{v_1^2}{2g} + \sum h_p = e_2 + \frac{p_2}{\gamma} + \frac{v_2^2}{2g} + \sum h_L \quad (1)$$

where e is a node elevation, p is a pressure at node, γ is a fluid specific weight, v is a fluid's velocity, g is the gravitational acceleration constant, h_p is a head added at pumps and h_L is a head loss in pipes. Thus the difference in energy at any two points connected in a network is equal to the energy gained from pumps and energy losses in pipes and fittings that occur in the path between them. Equation 1 can be written for any open path between any two points. Therefore it was used to compare the energy distribution between the vital points in original and simplified WDN models. It was observed that the energy between two points in the original and reduced models was different. Further investigation lead to conclusion that node elevation was not considered when removing it from the system. This can cause a situation where the pump speed required to satisfy minimum pressure constrains is different for the reduced model and the prototype. This is not desired for the control strategy described in Section 2.

To demonstrate the cause of this model mismatch an example is illustrated in Figure 4.



a) before simplification

b) after simplification.

Figure 4. Illustration of the elevation problem when reallocating demands in the simplification algorithm.

When the simplification algorithm removes a node from the network it distributes the node's demand to the neighbouring nodes based on the connected pipes conductance. In Figure 4a *pipe 3-4* and the *node 4* is removed thus demand d_4 is transferred to the *node 3*. The energy equation between section 1 along the reservoir surface and section 2 at the discharge node is

$$e_0 + \frac{p_0}{\gamma} + \frac{v_0^2}{2g} + h_p = e_4 + \frac{p_4}{\gamma} + \frac{v_4^2}{2g} + h_{L_{01}} + h_{L_{23}} + h_{L_{34}} \tag{2}$$

Now for the problem demonstration the following assumption were made:
$h_{L_{01}} + h_{L_{23}} + h_{L_{34}} = 0.01m$, $p_0 = p_1 = p_2 = p_3 = p_4 = 0\frac{N}{m^2}$, $v_0 = 0\frac{m}{s}, v_3 = {}_4 = 0.05\frac{m}{s}$. The energy equation simplifies to

$$h_{p_a} = e_4 + \frac{v_4^2}{2g} + h_{L_{01}} + h_{L_{23}} + h_{L_{34}} = 50.0301m \tag{3}$$

$$h_{p_b} = e_3 + \frac{v_3^2}{2g} + h_{L_{01}} + h_{L_{23}} = 30.0201m \tag{4}$$

When we calculate the power required for the pump to elevate water for both cases

$$P_{p_a} = \gamma q h_{p_a} = 4.9065kW \tag{5}$$

$$P_{p_b} = \gamma q h_{p_b} = 2.9441kW \tag{6}$$

it can be clearly seen how significant discrepancy can be introduced to the reduced model.

This affects especially tree-shaped zones in models, which after simplification are usually represented by one node with all the demands transferred to it. At the current stage of the work potential solutions to this issue are under investigation.

5. CONCLUSIONS & FURTHER WORK

The implementation of the online simplification algorithm with utilisation of nowadays parallel programing techniques was carried out. The resulted application could be integrated with the online control strategy applied to the WDN or it can be used as a standalone application for the purpose of the model simplification only. The advantage of the online model reduction is adaptation to abnormal situations and structural changes in a network, e.g. isolation of part of a network due to pipe burst. In such case an operator can change the full hydraulic model and run model reduction module to automatically produce updated simplified model.

The utilisation of parallel programing techniques increased the speed of simplification process based on the provided computational power. However the computational time can be reduced further by introducing a general-purpose parallel computing architecture (CUDA) with a new parallel programming model and instruction set that leverages the parallel compute engine in NVIDIA graphics processing units (GPU) to solve many complex computational problems in a more efficient way than on a CPU [11].

The main objective for the future work is to extend the simplification algorithm to distribute energy in the reduced model in similar manner as in the original model.

References

[1] D.V. Chase, T.M. Walski, D.A. Savic, W.M. Grayman, S. Beckwith and E. Koelle, *Advanced Water Distribution Modeling and Management*: Haestead Press, 2003.

[2] C.L. Eggener, and L. Polkowski, "Network Modeling and the Impact of Modeling Assumptions", *Journal of the American Water Works Association*, vol. 68(4), 1976, p. 189.

[3] J.G. Saldarriaga, S. Ochoa, D. Rodríguez, and J. Arbeláez, "Water distribution network skeletonization using the resilience concept " in *10th Annual Water Distribution Systems Analysis Conference WDSA2008*, Ruger National Park, South Africa, 2008.

[4] M. A. Brdys, B. Ulanicki, *Operational control of water systems: structures, algorithms, and applications*. New York: Prentice Hall, 1994.

[5] B. Ulanicki, A. Zehnpfund, and F. Martinez, "Simplification of water distribution network models," in *Proc. 2nd Intl. Conf. on Hydroinformatics*, Zurich, 1996.

[6] P. Skworcow, B. Ulanicki, H. AbdelMeguid, and D. Paluszczyszyn, "Model Predictive Control for Energy and Leakage Management in Water Distribution Systems," *in Proc. UKACC International Conference on Control*, Coventry, UK, 2010.

[7] L. A. Rossman, "EPANET Users manual," ed. Cincinnati, Ohio, US: Risk Reduction Engineering Laboratory, Office of Research & Development, US Enviromental Protection Agency, 2000.

[8] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, *GAMS: A user's guide*: GAMS Development Corporation, Washington, USA, 1998.

[9] A. S. Drud, "CONOPT: A GRG Code for large sparse dynamic non-linear optimisation problems," *Mathematical Programming*, vol. 31, 1985, pp. 153-191.

[10] Microsoft. "Parallel Programming in the .NET Framework". Available: http://msdn.microsoft.com/en-us/library/dd460693.aspx,(25th May 2011).

[11] NVIDIA, *NVIDIA CUDA Programming Guide version 3.0*: NVIDIA, 2010.