SIMPLIFICATION OF WATER SUPPLY NETWORK MODELS THROUGH LINEARISATION

Tobias Maschler and Dragan A. Savic

D.Savic@exeter.ac.uk

Report Number:99/01
1999

Centre for Water Systems,
University of Exeter,
North Park Road,
Exeter, EX4 4QF,
Devon,
United Kingdom.

Maschler, T. and D.A. Savic, (1999) Simplification of Water Supply Network Models through Linearisation, Centre for Water Systems, Report No.99/01, School of Engineering, University of Exeter, Exeter, United Kingdom, p.119.

Abstract

This work enhances a simplification algorithm for water network models with routines to identify the simplification range and to classify the importance of pipes in the reduced network model.

To aggregate a water network model, information about its control components is necessary. The variables used to optimise the operational schedules of the water supply network with the given model have also to be known. Basing on this, a simplification range is identified in the water network model. Afterwards, the model is linearised around a given working point and all redundant nodes are eliminated with Gauss-Jordan elimination. The remaining nodes are re-linked with pipes according to the structure of the simplified model. Non-important links will be deleted to keep the simplified model compact and with as less loops as possible.

The simplification algorithm is presented in detail in theory as in practise with an example network. Finally, it is applied in case studies to two water network models and the results are discussed.

The necessary information about the implementation of the developed computer program code is described alongside as well as essential information about the use of existing software.

Table of Contents

List of Figures

List of Tables

I — INTRODUCTION

The water industry in the United Kingdom spends approximately £70,000,000 per annum on electricity for pumping water supply. Similarly, almost 7% of the electricity consumed in the United States is used by the municipal water utilities. Since treated water pumping compromises the major fraction of the total energy budget, optimised operational schedules can improve the energy efficiency of a water distribution system.

System operators make these operational schedules with the aid of special decision support software. This software bases on mathematical models, which contain in some cases more than 2000 components. During an optimisation process for the operational schedules, these water network models are simulated many times for different schedules. However, only very small part of the simulation results is normally necessary, so an aggregated model is adequate. It should contain all control components of the original model and the variables, which are used to calculate the quality of the simulation. Aggregated — simplified — models will require less computation time and provide all information needed for the optimisation. Hence they will speed up the optimisation process and allow bigger models to be calculated.

This work uses a simplification algorithm Ulanicki et al (1996) — designated as "static simplification" later — to derive simplified models. Input components, as sources, tanks and reservoirs and valves, will be considered as well as those variables, which are needed for the optimisation process.

The first chapter in this report will start with an introduction into hydraulic network models and their components, mainly nodes and pipes. Afterwards, the simulation time problem will be discussed, followed by an overview of simplification approaches in literature.

Then, the simplification algorithm of Ulanicki et al (1996) will be presented. It linearises the network equations in an elegant way around a given working point and reduces the linear model with Gauss-Jordan elimination. From the reduced linearised model, a simplified non-linear model is recovered. Afterwards, an approach for the identification of the simplification area in the network model will be formulated. Next, it will be shown with brief algebraic examples, that the static simplification carries out network model component based simplification approaches like unification of nodes, parallel pipes and pipes in series and deletion of trees. In addition, the static simplification algorithm will be enhanced with the facility to delete pipes with very little influence on the simulation results, low conductance pipes.

After this theoretical part, the implementation of the static simplification algorithm will be discussed. It will be implemented as a module in the geographical information System StruMap Structural Technologies Ltd. (1996). The routines will be coded in C++. As compiler, the Borland C++-Builder Calvert (1997) will be used.

Following, a network model will be simplified and its simplification procedure discussed in detail. As case studies, the models of two water networks in Yorkshire, United Kingdom, will be simplified. One of them is very large, with around 2100 network components. The errors of the simplified models at the working point will be analysed, as well as in 24h simulations. Finally, the resulting reduction in solving time will be estimated.

II — PROBLEM FORMULATION

II. 1. Introduction to Hydraulic Network Models

This section introduces water network models. Firstly, the purpose of modelling water networks will be discussed. Then the macroscopic structure of water network models will be explained, followed by a short description of pipes and nodes, the main components of water network models. Afterwards, different simplification methods are presented. Finally, the aim of this report, to reduce the simulation time of water network models, will be formulated.

II. 1. 1. Purpose of Water Network Models

Figure 1 below shows a typical rural water network model with approximately 1100 nodes and 1300 pipes.

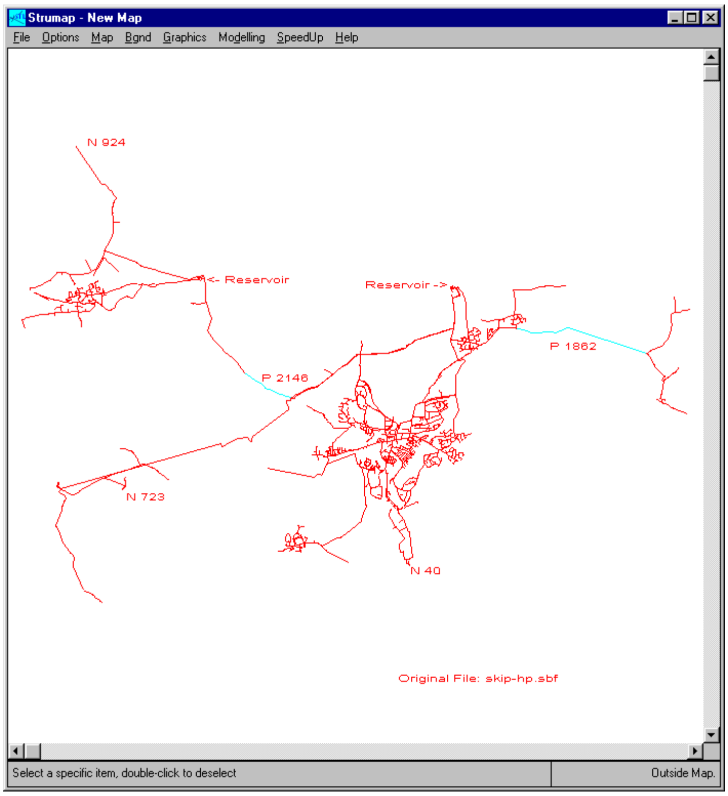


Figure 1. The Skipton Water Network Model.

Water distribution systems supply water to households, companies, fire departments and farms. The aim of water supply companies is to do this with an available head between 20 and 50m, and with as little pipe leakage as possible. Pipe leakage increases with higher pressure. Therefore, a suitable compromise must be found. This is achieved by optimising the schedules of the control devices in the network. Control devices are mainly pumps and valves.

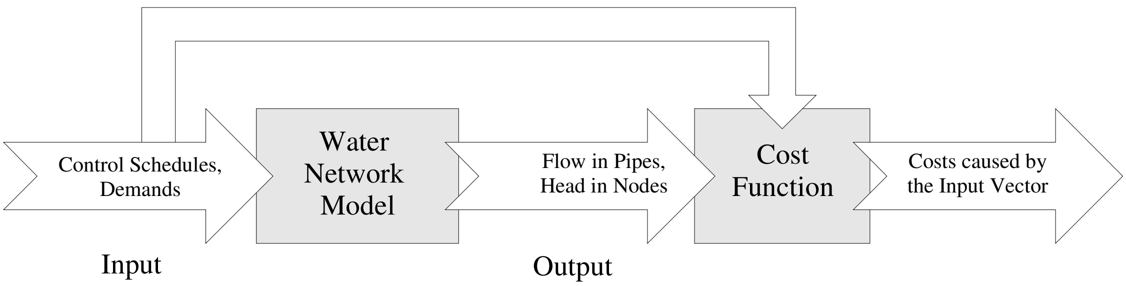


Figure 2. Input and Output for a Water Network Model.

Maschler, T. and D.A. Savic, (1999) Simplification of Water Supply Network Models through Linearisation, Centre for Water Systems, Report No.99/01, School of Engineering, University of Exeter, Exeter, United Kingdom, p.119.

Water network models are used to determine a convenient system input vector, which consists of pump, valve and source schedules as well as a demand vector for a given water network. The quality of the input vector is measured with a cost function. It weighs the input and output of the water network model and gives a scalar value (mainly in a currency unit) for it. This scalar value expresses the costs caused by the input vector.

II. 1. 2. Network Topology

Hydraulic water networks consist generally of the following components:

- Demand Nodes (nodes in network theory)
- Pipes (arcs in network theory)
- Valves
- Pumps
- Reservoirs
- Sources

Pipes connect demand nodes, valves, pumps, sources, and reservoirs. One or more pipes can be connected to a demand node. A valve is connected two pipes to two nodes. A source is usually connected to one pipe. A Reservoir has two pipes, one to refill it and one for the outflow. One pipe connects exactly two of the other network components above.

Generally, there are two types of pipe structures in a water network model:

- Trees
- Loops

A tree is a non-closed chain of connected pipes in the water network model. It can have sub-trees. A loop is a closed chain of connected pipes.

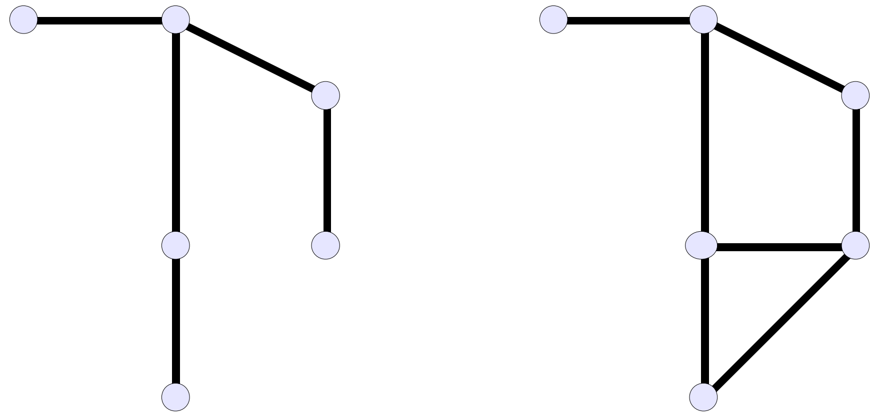


Figure 3. Example of a Tree (left) and a Loop (right) Network Structure.

The tree structure in Figure 3 has one sub-tree. The sub-tree starts at the second topmost node from the left and goes to the right hand side. The loop structure contains two independent loops, one at the top and one at the bottom.

In a network consisting of N_N nodes and N_P pipes, the number of loops N_L in a network can be determined using the following relation:

$$(1) \qquad N_L = N_P + 1 - N_N$$

The derivation of this equation is explained in the Appendix, Chapter VIII. 2.

II. 1. 2. a. Nodes

A node is a termination point for one or more pipes. It has a certain demand (withdrawal) and a unique, fixed geographic position with an elevation above ordinance datum. The available head at a demand node is the difference between the total head and the elevation.

Table 1. Attributes of a Node.

Node				
	x		m	Geographic Location
	y		m	Geographic Location
	z		m	Elevation above Ordinance Datum
	q		l/s	Demand
	H_T		m	Total Head
	H	= H_T - z	m	Available Head

The equilibrium of volumetric flow rates of all pipes connected to the node is calculated using the following equation:

$$(2) \qquad \sum_{i=1}^{N_P^j} Q_i^j = q_j$$

$Q_i^j = Q_k$:	flow in pipe k	l/s
q_j :	demand at node j	l/s
N_P^j :	Number of pipes connected to node j.	

A demand node is a node with a certain water withdrawal. Valves, sources, and reservoirs will be treated in this report as nodes because of their fixed geographic location. Sources have negative demands, as flow is put in the water network through them. The sign of the demand of reservoirs is negative, when it is being filled with water; when water is withdrawn, it is positive. Valves have no demand. The headloss in a valve is described by additional equations.

II. 1. 2. b. Pipes

A pipe connects two nodes with each other. It has a start and an end node.

Table 2. Attributes of a Pipe.

Pipe		
N_s		Start Node
N_e		End Node
d	mm	Diameter
l	m	Length
C	–	Roughness Coefficient
delta_H	m	Head Loss
Q	l/s	Flow

A pipe has a certain diameter and a length. The head loss between the start and the end node of the pipe is expressed in this report using the Hazen-Williams equation (4). Other equations, which give a relation between the flow and the head loss in a pipe, are the one of Colebrook-White and the Manning-Equation (see Chadwick and Morfett (1993)).

The head loss between the start and the end node are determined using the following relationship:

(3) $H_i - H_j = \Delta_{ij} = \Delta_k \quad k = 1,2,\ldots,N_P$

H_i, H_j: total heads at nodes *i,j* m

$\Delta_{ij} = \Delta_k$: head loss between nodes *i,j* m

N_P: Number of pipes in the hydraulic network.

The relationship between head loss Δ_k and flow Q_k is expressed with the Hazen-Williams equation (Chadwick and Morfett (1993)):

(4) $\Delta_k = r_k \cdot |Q_k|^{e_1 - 1} Q_k \quad k = 1,2,\ldots,N_P$

r_k: pipe resistance in pipe *k*

Δ_k: head loss in pipe k m

Q_k: flow in pipe k l/s

Constant: e_1:= 1.85

The Hazen-Williams pipe resistance is defined Savic and Walters (1997) as:

(5) $r_k = \dfrac{c \cdot l_k}{C_k^{e_1} \cdot d_k^{e_2}} \quad k = 1,2,\ldots,N_P$

l_k: length of pipe k m

C_k: Hazen-Williams coefficient of pipe k, dimensionless

d_k: diameter of pipe k mm

Constants: *c* := 1.215283E+10

e1 := 1.85

e2 := 4.87

The constant *c* was determined from StruMap as described in Appendix VIII. 1.

The relationships (4) and (5) can also be expressed in terms of the pipe conductance g_k instead of the pipe resistance r_k (Ulanicki et al (1996)):

(6) $$Q_k(\Delta_k) = \left|\Delta_k\right|^{\frac{1}{e_1}-1} \Delta_k \cdot g_k \quad k = 1,2,\ldots,N_P$$

where the pipe conductance g_k is

$$g_k = r_k^{-1/e_1} \quad k = 1,2,\ldots,N_P \text{ or}$$

$$g_k = \frac{C_k \cdot d_k^{\,e_2/e_1}}{c^{1/e_1} \cdot l_k^{\,1/e_1}} \quad k = 1,2,\ldots,N_P\,.$$

II. 2. Improving the Simulation Time Problem

II. 2. 1. The Simulation Time Problem

The equations described in the previous Chapter define the equation system of a water network model. The equation system is solvable explicitly only, if the network has a tree structure. Then, the head at the nodes can be directly calculated in terms of flow. This will be discussed in chapter III. 2. 4.

If a water network model has loops, it must be solved iteratively. Several solving approaches are described in Ellis and Simpson (1996). The number of equations used when solving water network model depends on the solver algorithm. HARP, the solver of StruMap (Structural Technologies Ltd. (1996)), is based on the freely available hydraulic network solver EPANET (Rossman (1994)), which uses a node-based model. A water network model is mainly represented there with the equations (2) and (4) above. Rossman (1994) writes that EPANET uses the "gradient algorithm" Todini and Pilati (1987) and sparse matrix techniques from George and Liu (1981).

Water network models may be simulated static and dynamic. As only the data at some time points is necessary for water network scheduling planning, static, steady-state solutions are used in practice: a water network model is solved using a set of hourly time steps (snapshot s) over a period of 24 hours. This allows the prognosis of changing reservoir levels and the determination of convenient valve and pump schedules. Therefore, the solution time of a water network model depends on its size, the number of snapshots and the equation solver. Further, the optimisation method being used to determine the "cheapest" input vector requires numerous simulation runs. This will be explained using Genetic Algorithms as an example.

To determine an input vector, which minimises a given cost function, different approaches are used. The "Centre for Water Systems" at the University of Exeter uses "Genetic Algorithms" (De Schaetzen and Walters (1998)). Genetic Algorithms have the ability to deal easily with non-linear, discrete, multimodal and non-differentiable functions, for example the Heavyside Step Function (CRC Press and Weisstein (1996)). The Heavyside Step Function is used to describe the on-off scheduling of pumps and valves.

Genetic Algorithms solve the network model for a range of different input vectors. Then they select the ones with the best cost function values and derive from these a new generation of input vectors which will be evaluated. This is done until the population of input vectors has converged. A further stop criterion is, for example, a fixed number of iterations or similar input vectors. Michalewicz (1992) gives a general introduction about Genetic Algorithms.

Therefore, genetic algorithms require the following run time:

$$\begin{aligned} \textit{solution time} := \quad & \textit{number of generations} \times \\ & \textit{number of individuals per generation} \times \\ & \textit{simulation time of the water network model} \quad . \end{aligned}$$

The simulation time for the Skipton water network model with approximately 1100 nodes and 1300 pipes is ~ 0.25 sec. For 1,000 generations of 100 individuals, the genetic algorithm may require ~25,000 sec, a little bit less than 7 h. So reducing the simulation time whilst maintaining the model accuracy will significantly speed up the input vector optimisation process without touching the optimisation algorithm.

II. 2. 2. Simplification Objectives

As described above, the critical item when determining convenient input vectors for water networks is the simulation time. To cut down the simulation time, the solving time of the network can be reduced. Therefore, the number of network components and especially of the loops in the network must be minimised in a way which guarantees a solution accuracy as close as possible to that of the original water network model.

Further, the simplified model must react in the same way to changing boundary conditions (i.e.: demands) (Anderson and AL-Jamal (1995)).

There are two main ways to simplify a network model:

1. *Simplification of Network Model Components.*

   The basic types of network components are maintained, but the individual network components are combined and replaced. An example for this would be to remove intermediate nodes from pipes in series.

2. *Black Box Simplification.*

   The network model is replaced with an abstract model, which provides the same functionality with less solution time. This can be done for the whole network or for parts of it. Examples are neural networks (Swiercz (1995)) or general models, whose parameters are fitted (Anderson and Al-Jamal (1995)).

The first method allows the network model to be simplified directly using the network simulation software. The second simplification technique can only be carried out in programs, which allow the integration of the black box model.

The following sections gives an overview of a number of simplification approaches described in literature. Firstly, approaches to simplifying water network models based on the model components will be presented; then approaches that replace the network model with black boxes will be described.

II. 2. 3. Simplification of Network Model Components

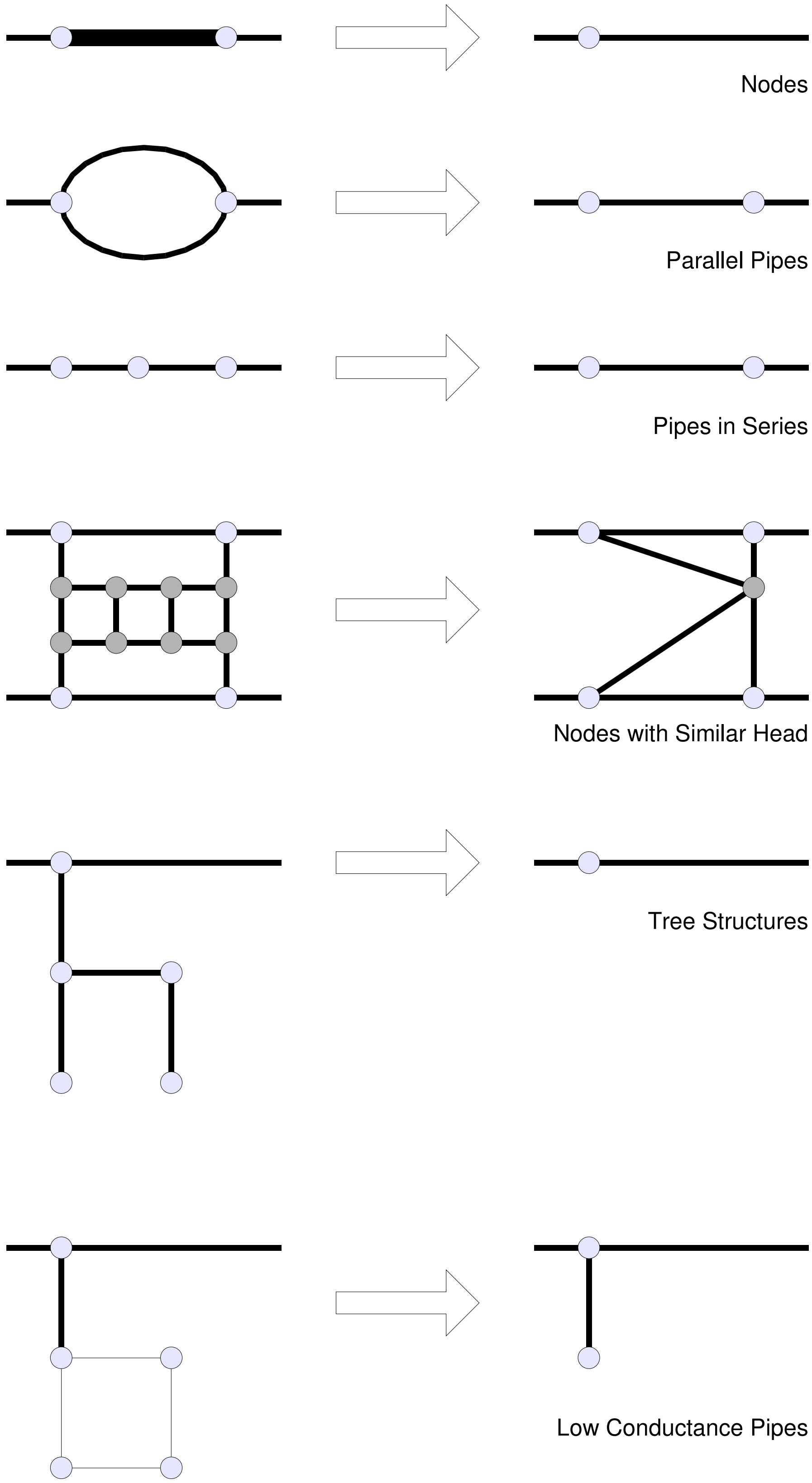


Figure 4. Simplification of Network Model Components.

II. 2. 3. a. Nodes

Nodes, that are geographically closely located can be unified to eliminate convergence problems of the solver[1] (Martínez and García-Serra (1992)). Nodes are understood to be "close", if the head loss between them is very low or in more abstract terms, the pipe resistance between them is very low. This is the same as unifying pipes in series (described below) which are of a very low pipe resistance.

II. 2. 3. b. Parallel Pipes

Pipes, which have the same start and end nodes, can be replaced by an equivalent one with an appropriate diameter and length (Martínez and García-Serra (1992)). The pipe conductance of the new pipe can be calculated by summing up the conductances of the old pipes Anderson and Al-Jamal (1995):

(7) $$g_{new} = g_{old,1} + g_{old,2}\,.$$

Afterwards, the pipe diameter can be obtained using Equation (6) with an appropriate pipe length and Hazen-Williams coefficient. This is an exact substitution as long as the head loss for the pipe does not change, but it leads to some error otherwise.

II. 2. 3. c. Series Pipes

Two pipes, connected by a single node, can be replaced by one pipe (Martínez and García-Serra (1992)). The demand of the node which is removed has to be redistributed to the start and end node of the new pipe. The resistance of the new pipe can be calculated by summing up the old pipe resistances Anderson and Al-Jamal (1995):

(8) $$r_{new} = r_{old,1} + r_{old,2}\,.$$

Now, the new diameter can be calculated using equation (5) and a suitable pipe length and Hazen-Williams coefficient. As this method concatenates two parallel pipes, this is

[1] Pipes with high resistance/ low conductance pose big problems when solving. They require far more iterations to meet the accuracy criterion of the solver than other pipes.

only an exact replacement as long as the head loss between the start and end nodes stays the same.

The redistribution of the demand of the node, which is removed, will be discussed in Chapter III. 1. 1. d.

II. 2. 3. d. Nodes with Similar Head

Anderson and Al-Jamal (1995) write that a group of nodes with similar head can be reduced to a single node with the total demand of the node group.

II. 2. 3. e. Tree Structures

Tree structures in a network can be replaced by adding the summed demand of all nodes in the tree to the node, which connects this structure Martínez and García-Serra (1992). This will be described in detail in Chapter III. 2. 4. below.

II. 2. 3. f. Low Conductance Pipes

Martínez and García-Serra (1992) emphasise that pipes below a certain diameter do not contribute significantly to the carrying capacity of a water network, so they can be neglected. They give as rule of thumb, that in small networks (with pipe diameters up to 200-250mm), pipes with less than 80mm diameter are negligible, in large networks (with pipe diameters bigger than 800mm) pipes with less than 200mm can be eliminated. Swiercz (1995) describes the same procedure as “skeletonisation”.

As the flow is the decision criterion for neglecting a pipe, it appears to be more exact to use the pipe resistance or the pipe conductance instead of the pipe diameter to decide. These terms reflect all the pipe attributes in the equations (5) and (6). Therefore, the pipe roughness with the Hazen-Williams coefficient and the pipe length will be taken into account, as well.

II. 2. 3. g. Single Input Subsystems

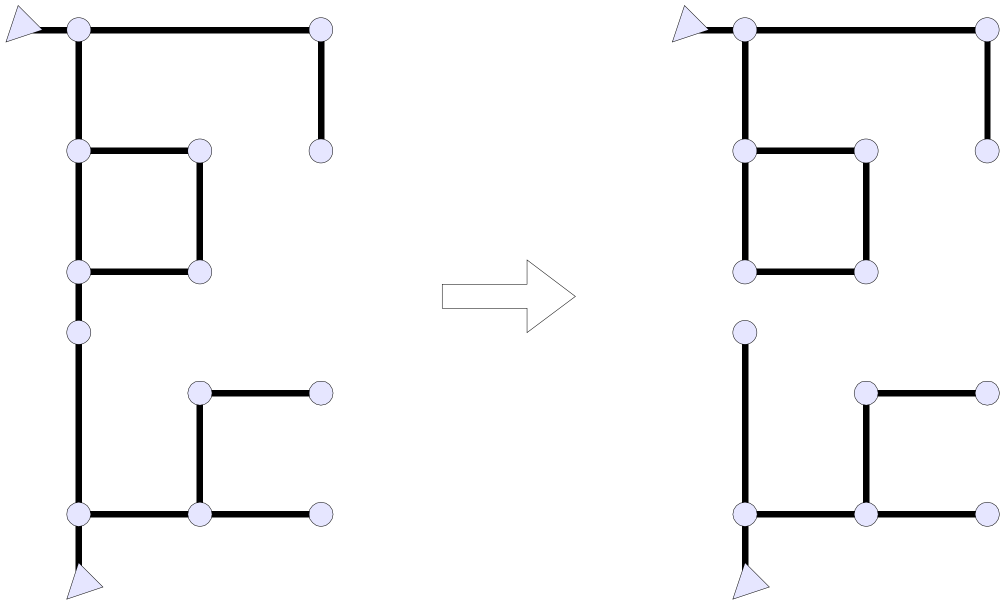


Figure 5. A Water Network Model is split into Two Single Input Subsystems.

A large water network model with multiple input points (sources and reservoirs) can be separated in multiple independent networks, when pipes, which are under the influence of two input points are being split Swarnee and Sharma (1990). This breaks the original network model down to a number of models corresponding to the number of input points. These models can be simulated with less computational effort because of their smaller size.

II. 2. 4. Black Box Simplification

The following approaches replace the network model partially or fully with a system, which provides the same function with less complexity.

II. 2. 4. a. Static Simplification

Basing on the relationships between electrical network models and water network models, Ulanicki et al (1996) present an approach to simplify latter models. The following table gives a brief overview of these similarities.

Table 3. Similarities between Electrical Network Models and Water Network Models.

Electrical network models	⇔	water network models
current I	⇔	flow Q
voltage U	⇔	head H
non-linear resistor	⇔	Pipe
$\sum I = 0$ at a node, Kirchhoff's Law I	⇔	$\sum Q = q^{nod}$ at a node Continuity Equation
$\sum U = 0$ for a loop, Kirchhoff's Law II	⇔	$\sum H = 0$ for a loop Energy Equation

The approach is based on formulating the full non-linear model, linearising it around a given working point and reducing it with the Gaussian Elimination Algorithm. A non-linear model with fewer components can then be retrieved. The case studies (Ulanicki et al (1996)) show that the largest relative errors in terms of heads are mostly less than 1% for water network models with up to 797 nodes, 1088 elements and 103 devices. The full algorithm will be described in detail in chapter III. 1. 1.

This approach is very simple and fast and allows a direct physical interpretation: the resulting simplified model also consists of pipes and nodes. It is valid for a wide range of operating conditions, but shows slight errors when the working conditions are outside the vicinity of the linearisation point.

II. 2. 4. b. Neural Network Approach

Swiercz (1995) replaces the water network model with a neural network. Neural networks are trained with given input and output vectors to mimic the behaviour of the water network model. Swiercz' example network contained 60 nodes, 124 elements and 4 reservoirs. Depending of the complexity of the neural networks, the average relative error for the heads was below 2%, while the training covered 1800-7650 epochs (snapshots). Swiercz (1995) points out that neural networks require far less calculation time than their original water network models without a loss of accuracy.

In addition to the approaches described above, Anderson and Al-Jamal (1995) use parameter-fitting to equalise the results of the simplified and the original network model. This is an additional step during the model simplification procedure to enhance the accuracy of the simplified model.

II. 3. Conclusions

In this work, the static simplification method of Ulanicki et al (1996) was chosen for network simplification. It combines the advantage of providing an abstract model, which can be expressed with pipes and nodes with the ability to reduce the water network models to a minimum without significant accuracy loss. Combining nodes with similar head was not done in this work, as the heads of nodes may vary drastically over a period although they appear to be similar at a specific time. In addition, splitting networks with multiple sources into single input systems was not carried out here.

III — NETWORK SIMPLIFICATION

This chapter describes the theory behind static simplification based on Ulanicki et al (1996) and illustrates that this method carries out a simplification based on network model components. In addition, the functioning of the method is shown by an example network model. Finally, details of the implementation, like an Activity-Diagram of the Algorithm, are discussed.

III. 1. Theoretical Development

In this section, the algorithm for static simplification will be described in detail. Afterwards, the treatment of network components will be discussed.

III. 1. 1. The Algorithm for Static Simplification

The strategy described here is based on the work by Ulanicki et al (1996). Algebraic examples are given in Section III. 2. below.

III. 1. 1. a. The Mathematical Water Network Model

It is very convenient for algebraic manipulations to express the network model with a branch incidence matrix:

(9) $\Lambda \cdot \vec{q} = \vec{q}^{\,nod}$ Kirchhoff's law I for nodes

Λ branch incidence matrix

$\vec{q}$ vector of flows in the pipes

$\vec{q}^{\,nod}$ demands in the nodes.

This matrix has one row for each node and one column for each branch. Each column has exactly one entry "-1" in the row corresponding to the node, at which the water goes into the pipe and one entry "1" at the row corresponding to the node, at which the water leaves the pipe. All other entries are "0". The matrix Λ can be also used to determine the head loss in the pipes:

(10) $\Delta\vec{h} = \Lambda^T \cdot \vec{h}$ Kirchhoff's law II

$$\Delta\vec{h} = \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{N_P} \end{bmatrix}$$ vector of head losses

$$\vec{h} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{N_N} \end{bmatrix}$$ vector of total heads in the nodes.

The relationship between the head loss and the flow in the pipes can be expressed with the pipe component law using the Hazen-Williams coefficient (6):

(11) $$\vec{q} = Q(\Delta h) = \begin{bmatrix} g_1 \cdot \left|\Delta h_1\right|^{e_3} \cdot \mathrm{sign}(\Delta h_1) \\ \vdots \\ g_{N_P} \cdot \left|\Delta h_{N_P}\right|^{e_3} \cdot \mathrm{sign}(\Delta h_{N_P}) \end{bmatrix}$$ pipe component law

g_m conductance of the pipe m; $m=1(1)N_p$

$$e_3 := \frac{1}{e_1} - 1$$

The relationship uses the following definition of the signum function:

$$sign(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ -1 & \text{for } x < 0 \end{cases}.$$

The equations (9) - (11) describe the network model completely. With these equations, the mass balance for a node n can be expressed with

$$(12) \qquad \sum_{m\in N_{N,n}} \Lambda_{n,j(n,m)} g_{j(n,m)} \left|\Delta h_{j(n,m)}\right|^{e_3} \operatorname{sign}(\Delta h_{j(m,n)}) = q_n^{nod} \qquad n,m = 1,2,\ldots,N_N$$

$j(n,m)$	branch j connecting nodes m and n;	$j \in N_P$
$N_{N,n}$	number of nodes connected to node n.	

Inserting (10) in (11) and (11) in (9) leads to a compact expression for (12):

$$(13) \qquad \Lambda Q(\underbrace{\Lambda^T \vec{h}}_{\Delta\vec{h}}) = \vec{q}^{\,nod}.$$

This expression has one redundant equation as the sum of all demands q_i^{nod} is zero in the network model:

$$(14) \qquad \sum_{n=1}^{N_N} q_n^{nod} = 0.$$

Hence, the simplification can only cover $N_N - 1$ nodes. From now on, N_N will represent the number of nodes covered by the simplification.

III. 1. 1. b. The Linear Model

Linearising (13) around the operating point $\left(\vec{h}^0, \vec{q}^{\,nod,0}\right)$ of the water network model leads to:

$$(15) \qquad \Lambda \left.\frac{dQ(\Delta\vec{h})}{d\Delta\vec{h}}\right|_{\Delta\vec{h}^0} \Lambda^T \cdot \delta\vec{h} = \delta\vec{q}^{\,nod} \text{ with the Jacobian}$$

$$A = \Lambda \left.\frac{dQ(\Delta\vec{h})}{d\Delta\vec{h}}\right|_{\Delta\vec{h}^0} \Lambda^T \quad \text{and}$$

$$\delta\vec{h} = \vec{h} - \vec{h}^0 \quad \text{and}$$

$$\delta\vec{q}^{\,nod} = \vec{q}^{\,nod} - \vec{q}^{\,nod,0} \quad \text{and}$$

$$\Delta\vec{h}^0 = \Lambda^T \cdot \vec{h}^0, \quad \text{corresponding to Kirchhoff's Law II (10).}$$

The matrix with the linearised pipe conductances is determined as follows:

$$(16) \qquad \left.\frac{dQ(\Delta\vec{h})}{d\Delta\vec{h}}\right|_{\Delta\vec{h}^0} = \mathrm{diag}\left[e_3 \cdot g_j \left|\Delta h_j^0\right|^{e_4}\right]_{j=1}^{N_P}.$$

N_P number of branches

$$e_4 := e_3 - 1 = \frac{1}{e_1} - 2.$$

This leads to the following form of the Jacobian matrix:

$$(17) \qquad A_{m,n} = \begin{cases} -e_3 \cdot g_{m,n}\left|h_n^0 - h_m^0\right|^{e_4} & \text{for } m \in N_{N,n} \\ \sum_{k \in N_n, k \neq n} e_3 \cdot g_{k,n}\left|h_n^0 - h_k^0\right|^{e_4} & \text{for } \quad m = n \\ 0 & \text{for } m \notin N_{N,n} \end{cases} \qquad m,n = 1,2,\ldots,N_N$$

The Jacobian matrix is a $N_N \times N_N$ symmetric one. A non-diagonal entry on position (*n*, *m*) indicates a pipe between the nodes *n* and *m*.

Ulanicki et al (1996) introduce the notion of the "linearised branch conductance":

$$(18) \qquad \tilde{g}_{j(n,m)} \hat{=} e_3 g_{j(n,m)}\left|h_n^0 - h_m^0\right|^{e_4} \qquad j=1,2,\ldots,N_P,\ m,n \notin N_N.$$

This notion will be called "linearised pipe conductance" in this report. It reflects the pipe attributes such as diameter, length and Hazen-Williams coefficient and, in addition, the head loss in the pipe. It will be used to transform between the non-linear and linear network model.

Ulanicki et al (1996) also introduce the notion of the "linearised node conductance":

$$(19) \qquad \tilde{g}_n \hat{=} \sum_{k \in N_n, k \neq n} \tilde{g}_{k,n} \qquad n = 1,2,\ldots,N_N.$$

The linearised node conductance is the sum of the linearised pipe conductances of the pipes which are connected to the node.

This allows the presentation of the linearised model (15) as:

$$(20)\qquad \begin{bmatrix} \tilde{g}_1 & -\tilde{g}_{1,2} & \cdots & -\tilde{g}_{1,N_N} \\ -\tilde{g}_{2,1} & \tilde{g}_2 & \cdots & -\tilde{g}_{2,N_N} \\ \vdots & \vdots & \ddots & \vdots \\ -\tilde{g}_{N_N,1} & -\tilde{g}_{N_N,2} & \cdots & \tilde{g}_{N_N} \end{bmatrix} \cdot \begin{bmatrix} \delta h_1 \\ \delta h_2 \\ \vdots \\ \delta h_{N_N} \end{bmatrix} = \begin{bmatrix} \delta q_1^{nod} \\ \delta q_2^{nod} \\ \vdots \\ \delta q_{N_N}^{nod} \end{bmatrix}$$

with the linearised branch conductances on the non-diagonal elements and the linearised node conductances on the diagonal elements.

III. 1. 1. c. Linear Model Reduction

This linearised model (20) describes the changes of heads and flows at a default working point. The model can be reduced in the following way: the rows of the nodes that shall be removed have to be put in the first *r* row positions of the matrix *A*. Afterwards, the Gauss-elimination (Hammerlin and Hoffman (1991) (referenced by Ulanicki et al (1996))) has to be applied *r* times.

This yields the reduced matrix *A(r). A(r)* has $(N_N - r)$ rows and columns. It is a sub-matrix of *A* after applying the Gauss-elimination. Its range in A goes from position $(r + 1, r + 1)$ to position (N_N, N_N).

The Gauss-elimination does the following: The pipes are restructured around the nodes, which are removed. The demand of a replaced node is redistributed between nodes connected to this node. This is done proportionally to the conductance of each branch. The conductances of the pipes are changed as well.

A non-linear model can be recovered from the reduced linear model by reading the topology from the reduced network matrix *A(r)*. Non-zero-entries indicate branches between nodes.

Ulanicki et al (1996) introduce the new node-branch incidence matrix $\Lambda^{(r)}$ with

$N_N^{(r)}$	number of nodes in the reduced model and
$N_P^{(r)}$	number of branches in the reduced model.

The new linear model can be represented in a form equivalent to (15):

(21) $\Lambda^{(r)} \dfrac{dQ^{(r)}(\Delta\vec{h}^{(r)})}{d\Delta\vec{h}^{(r)}} \Lambda^{(r)T} \cdot \delta\vec{h}^{(r)} = \delta\vec{q}^{(r)nod}$, where

$\delta\vec{q}^{(r)nod} = P^{(r)} \cdot \delta\vec{q}^{nod}$ and

$\dfrac{dQ^{(r)}(\Delta\vec{h}^{(r)})}{d\Delta\vec{h}^{(r)}} = \mathrm{diag}\left[\tilde{g}^{(r)}_{n,m}\right]_{j=1}^{N_N^{(r)}}$ is a $N_N^{(r)} \times N_N^{(r)}$ diagonal matrix.

$\tilde{g}^{(r)}_{n,m}$ is the new linear branch conductance between the nodes *n* and *m* and $\delta\vec{q}^{(r)nod}$ is the new nodal demand vector.

III. 1. 1. d. The Reduced Non-Linear Model

This yields a new non-linear model:

(22) $\Lambda^{(r)} Q^{(r)}(\Lambda^{(r)T}\vec{h}^{(r)}) = P^{(r)}\vec{q}^{nod}$

where $Q^{(r)}(\Lambda^{(r)T}\vec{h}^{(r)})$ is the function for the new nonlinear branch law

This model has the new branch conductance

(23) $g^{(r)}_{n,m} \cdot e_3 \left|h^0_n - h^0_m\right|^{e_4} = \tilde{g}^{(r)}_{n,m}$, $\quad m,n \in (N_N^{(r)} := N_N - r)$.

These steps wre done by Ulanicki et al (1996).

III. 1. 1. e. Requirements

To reduce a model, the following requirements are necessary:

1. A full hydraulic network model.
2. All values of $\vec{h}$ and $\vec{q}^{nod}$ in the working point have to be known.
3. The knowledge of which nodes shall be removed and which not.

The vector $\vec{q}^{nod}$ is known, because it is part of the input vector, while $\vec{h}$ can be obtained by solving the water network model at $\vec{q}^{nod}$. As Ulanicki et al (1996) give only the algorithm for static simplification, the third point has to be discussed in detail. This will be done in the following section.

III. 1. 2. Identification of the Simplification Range

In this section, a method will be developed to separate the part of the model which can be simplified from the rest. This model part will be denoted "simplification range" from now on. The simplification range is the part of the full model, which can be replaced during simplification without changing the behaviour of the water network model. The "simplification range" — a "black box" — has with nodes as interfaces to the in- and output components of the water network model.

Generally, it is not necessary to know all heads and flows in the system since only a subset of all system output is needed for the cost function. A simplified model with fewer details should of course have the same input vector, so valves and pumps, etc. shall be kept. To maintain their functionality, flows in selected pipes and heads at selected nodes shall not be changed by the simplification algorithm.

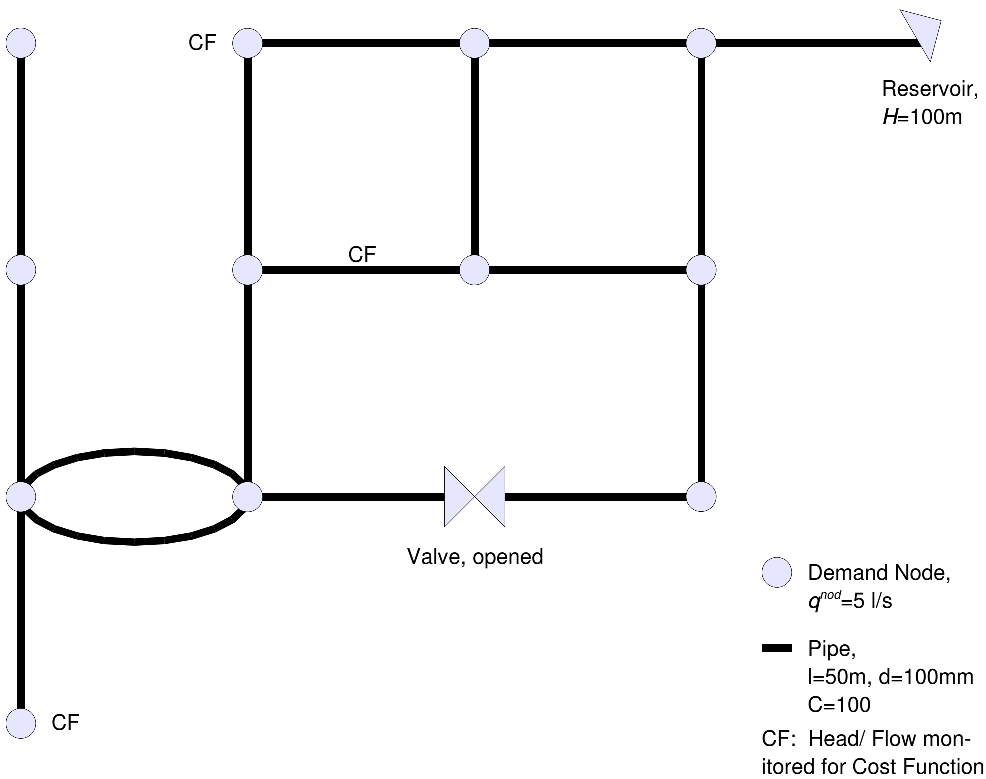


Figure 6. Example Water Network Model with Trees, Loops, Parallel Pipes, a Fixed Head Reservoir and an Open Valve.

The Water Network Model in Figure 6 will be used as an example to identify the simplification range. It contains a fixed head reservoir, an opened valve, 12 demand nodes and 17 pipes. Therefore, the number of loops is 4 (see equation (1)). All demand nodes have a demand of 5 l/s. The reservoir level is fixed at 100m. All pipes have a length of 50m, a diameter of 100mm and a Hazen-Williams coefficient of *C*=100 (see i.e. Chadwick and Morfett (1993)). The head at two nodes (those ones marked with "CF"[2] in Figure 6) and the flow in one pipe (marked with "CF", as well) are inputs for a cost function.

The aim of the identification is to find all nodes which form the boundary of the simplification range. This boundary depends on the input and output components of the water network model. Input components are all valves, reservoirs and sources. Output components are the pipes, whose flow is used for the cost function and the nodes, whose heads are input to the cost function.

The simplification range without these interface components is drawn in Figure 7.

---

[2] "CF" stands for **c**ost **f**unction.

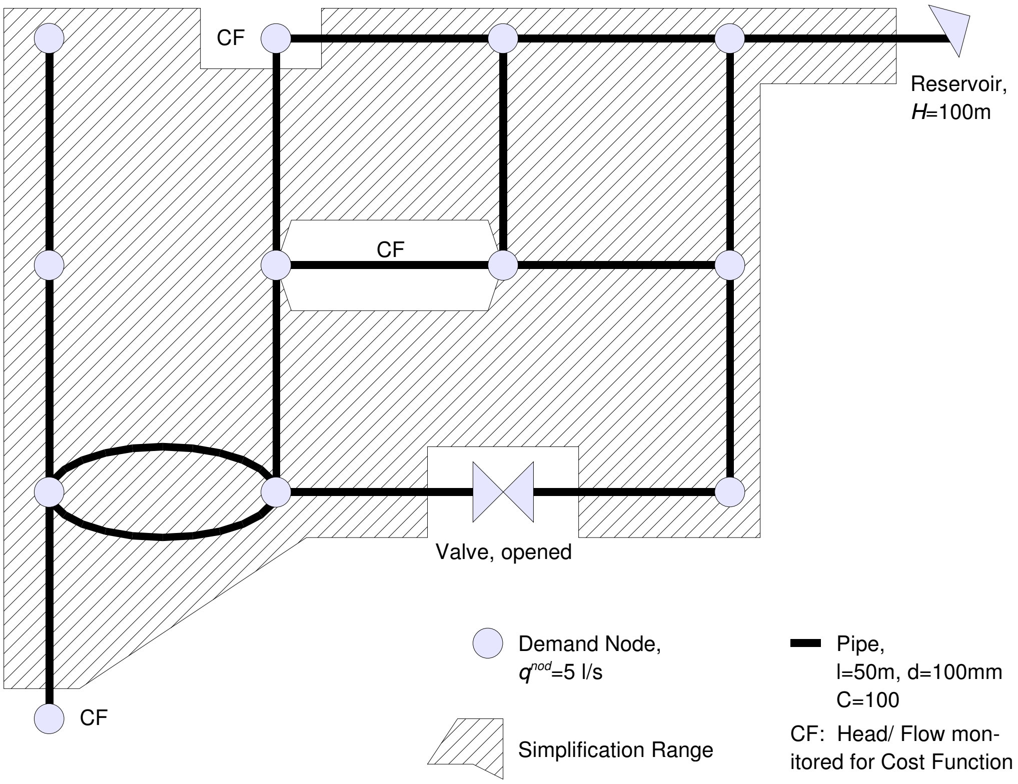


Figure 7. The Example Network with the Interface Components outside the Simplification Range.

So far, this boundary is not clearly defined. To do this, two new network component attributes will be introduced:

- *"Non-Removable"*
  Network model nodes with this attribute have to stay during simplification.
  Their attributes may be changed.
  Therefore, they form the boundary of the simplification range.
- *"Untouchable"*
  Network model components with this attribute have to stay during simplification, but their attributes are not permitted to change.
  Hence, they have to stay outside the simplification range.

Input Components have to stay unchanged during simplification. Their attributes are not allowed to change, for example to avoid a situation where a valve could get a demand associated with it. Hence, the valves have to be marked with as "untouchable". Output components are nodes, where the head is needed for the cost function. Output pipes are pipes, whose flow is needed for the cost function. Output nodes have to stay unchanged during simplification, but one of their attributes, the demand, is allowed to change because the pressure at the node should stay the same after simplification. Therefore, they have to be marked as "non-removable". The flow in output pipes would be changed, when the pipe conductance is varied, hence they have to be marked as "untouchable". The following table gives a brief overview over the assignment of attributes to network components. This has to be done manually.

Table 4. Attributes of Interface Components.

Water Network Model Component	Signal Flow Direction	Attribute
Source	Input	Untouchable
Reservoir		Untouchable
Valve		Untouchable
Pump		Untouchable
Node	Output	Non-Removable
Pipe		Untouchable

The static simplification algorithm in this work is based on nodes. Therefore, the identification algorithm must clearly identify the nodes, which shall stay and those, which can be simplified. All nodes, which are "non-removable", will stay during the simplification. However, the attribute "untouchable" can be assigned to nodes and pipes. If a node is "untouchable", its attributes have to be protected from being changed. This can be achieved by protecting the connections, i.e. the pipes connected to the node. So the pipes which are connected to an "untouchable" node have to be marked as "untouchable" as well. Similarly, preventing their nodes from being removed will protect "untouchable" pipes. Hence, the start and the end node of a pipe have to be marked as "non-removable".

The simplification range identification algorithm will have to perform the following preparation steps:

1. Mark all pipes, which are connected to input components (nodes with the attribute "untouchable") as "untouchable" as well.
2. Mark the start and end nodes of "untouchable" pipes as "non-removable".

After performing these steps, the boundary of the simplification range is defined by the non-removable nodes. This is illustrated in Figure 8.

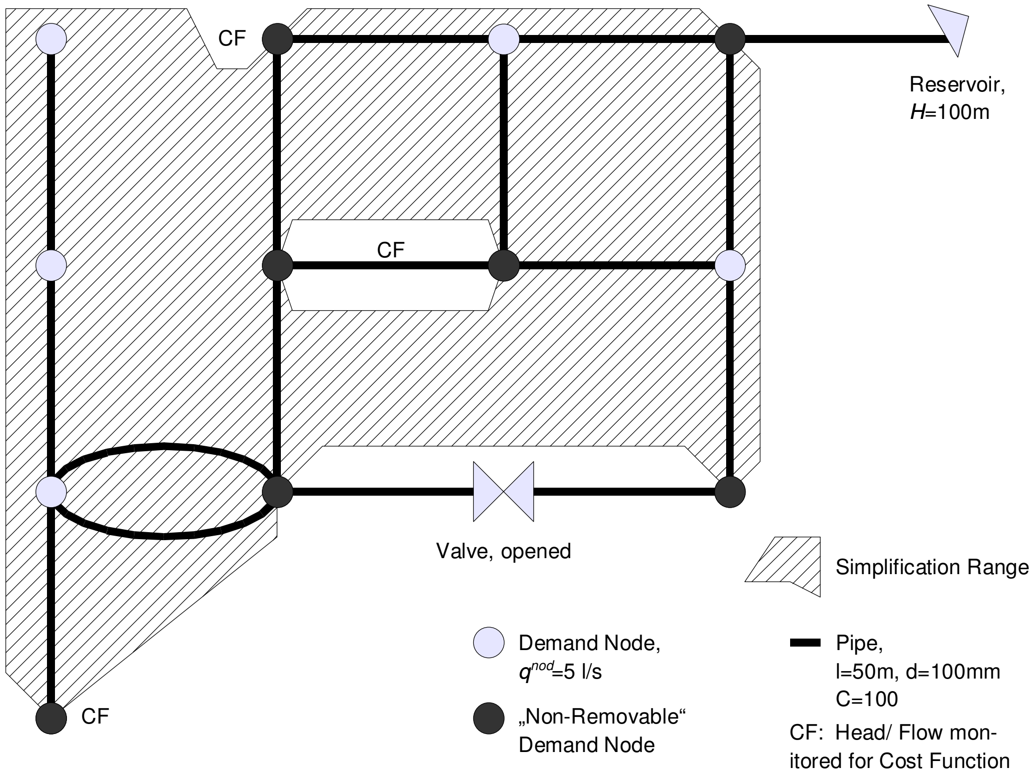


Figure 8. The Network Model with the Simplification Boundary after performing the Identification Algorithm.

Output Pipes may eventually have got assigned the linear pipe conductance of other pipes during the simplification process. This may be corrected by parallel pipes.

The steps of the identification algorithm above guarantee that no demand is assigned to valves, pumps, sources and reservoirs and that new pipes are not connected to them.

III. 1. 3. Pipe Attributes

This part explains how the pipe attribute values like the length, the diameter or the Hazen-Williams coefficient will be assigned when the simplification algorithm creates new pipes.

A pipe has to transport water from one node to another according to its conductance. When the simplified water network model is constructed from the reduced branch incidence matrix $A^{(r)}$, the pipe attributes will be specified as follows:

- The new pipe length is the geographic distance between the start and the end node.
- The Hazen-Williams coefficient is set to 100.
- The diameter of a pipe k will be calculated for the new pipes during simplification using the following relationship:

$$(24) \qquad d_k = \left(\frac{g_k^{e_1} \cdot c \cdot l_k}{C_k^{e_1}} \right)^{1/e_2} \qquad k = 1,2,\ldots,N_p^{(r)}.$$

The equation above is based on the formula for the non-linear pipe conductance (equation (6)).

In remaining pipes, the original Hazen-Williams coefficient, the diameter and the length cannot be kept since in the simplified model, these pipes may represent a conglomeration of several others.

III. 2. Algebraic Simplification of Network Model Components

This section gives some brief examples of the static simplification based on component-by-component simplification approaches.

III. 2. 1. Nodes

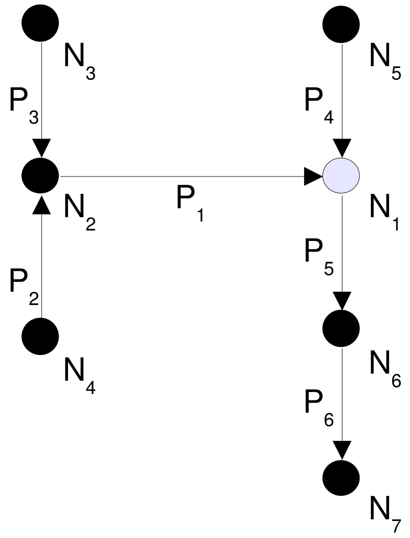


Figure 9. Node Replacement.

The water network model in Figure 9 has sources in the nodes N_3, N_4 and N_5, therefore their demand is negative. Water flows from N_3 and N_4 to N_2, from N_2 and N_5 to N_1, from N1 to N_6 and from N_6 to N_7. The network has 6 pipes and 7 nodes. Node N_1 will be removed from the water network model.

III. 2. 1. a. Mathematics

The network model can be represented using the energy and the continuity equation and the pipe component law as described on page 26. The continuity equation for nodes:

$$(25) \qquad \Lambda \cdot \vec{q} = \vec{q}^{\,nod}$$

$$\Lambda = \begin{array}{c} \\ N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{array} \begin{array}{c} \begin{array}{cccccc} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \end{array} ; \ \vec{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_6 \end{bmatrix} ; \ \vec{q}^{\,nod} = \begin{bmatrix} q_1^{nod} \\ \vdots \\ q_6^{nod} \end{bmatrix} .$$

The row for N_7 was excluded because of the dependency

(26) $$q_7^{nod} = \sum_{i=1}^{N_N-1} q_i^{nod} \text{ with } N_N=7.$$

The energy equation gives the relationship between the total head at the nodes and the head loss in the pipes:

(27) $$\Delta\vec{h} = \Lambda^T \vec{h}$$

$$\Delta\vec{h} = \begin{bmatrix} \Delta h_1 \\ \vdots \\ \Delta h_6 \end{bmatrix};\ \vec{h} = \begin{bmatrix} h_1 \\ \vdots \\ h_6 \end{bmatrix}.$$

The pipe component law gives the relationship between the head loss and the flow in the pipes:

(28) $$\vec{q} = Q(\Delta\vec{h}) = \begin{bmatrix} g_1 |\Delta h_1|^{e_3} \operatorname{sign}(\Delta h_1) \\ \vdots \\ g_6 |\Delta h_6|^{e_3} \operatorname{sign}(\Delta h_6) \end{bmatrix}.$$

Linearisation of (26) with (28) around the working point $\left(\vec{h}^0, \vec{q}^{nod,0}\right)$ leads to the linearised water network model

(29) $$\underbrace{\Lambda \left.\frac{d\vec{Q}(\Delta\vec{h})}{d\Delta\vec{h}}\right|_{\vec{h}^0} \Lambda^T}_{A} \delta\vec{h} = \delta\vec{q}^{nod} \text{ with}$$

$$\delta\vec{h} = \vec{h} - \vec{h}^0 \qquad \text{and}$$

$$\delta\vec{q} = \vec{q}^{nod} - \vec{q}^{nod,0}.$$

The Jacobian$A$ of the linearised model requires the linearised pipe conductance, equation (18):

(30)
$$\left.\frac{d\vec{Q}(\Delta\vec{h})}{d\Delta\vec{h}}\right|_{\vec{h}^0} = \begin{bmatrix} e_3 g_1 |\Delta h_1|^{e_4} & 0 & \cdots & 0 \\ 0 & e_3 g_2 |\Delta h_2|^{e_4} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & e_3 g_6 |\Delta h_6|^{e_4} \end{bmatrix} = \begin{bmatrix} \tilde{g}_1 & 0 & \cdots & 0 \\ 0 & \tilde{g}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{g}_6 \end{bmatrix}.$$

This yields A from equation (25) as

(31)
$$A = \begin{bmatrix} \tilde{g}_1+\tilde{g}_4+\tilde{g}_5 & -\tilde{g}_1 & 0 & 0 & -\tilde{g}_4 & -\tilde{g}_5 \\ -\tilde{g}_1 & \tilde{g}_1+\tilde{g}_2+\tilde{g}_3 & -\tilde{g}_3 & -\tilde{g}_2 & 0 & 0 \\ 0 & -\tilde{g}_3 & \tilde{g}_3 & 0 & 0 & 0 \\ 0 & -\tilde{g}_2 & 0 & \tilde{g}_2 & 0 & 0 \\ -\tilde{g}_4 & 0 & 0 & 0 & \tilde{g}_4 & 0 \\ -\tilde{g}_5 & 0 & 0 & 0 & 0 & \tilde{g}_5+\tilde{g}_6 \end{bmatrix}.$$

The linearised network model:

(32)
$$A\delta\vec{h} = \delta\vec{q}^{\,nod}$$

$$\left[\begin{array}{c:ccccc} \tilde{g}_1+\tilde{g}_4+\tilde{g}_5 & -\tilde{g}_1 & 0 & 0 & -\tilde{g}_4 & -\tilde{g}_5 \\ \hdashline -\tilde{g}_1 & \tilde{g}_1+\tilde{g}_2+\tilde{g}_3 & -\tilde{g}_3 & -\tilde{g}_2 & 0 & 0 \\ 0 & -\tilde{g}_3 & \tilde{g}_3 & 0 & 0 & 0 \\ 0 & -\tilde{g}_2 & 0 & \tilde{g}_2 & 0 & 0 \\ -\tilde{g}_4 & 0 & 0 & 0 & \tilde{g}_4 & 0 \\ -\tilde{g}_5 & 0 & 0 & 0 & 0 & \tilde{g}_5+\tilde{g}_6 \end{array}\right] \delta\vec{h} = \delta\vec{q}^{\,nod}.$$

It can be seen that the matrix is symmetric with a dominant diagonal. The diagonal elements are the negative sum of all other row or column entries.

Now the node N_1 will be removed by Gauss-Elimination. This means, the first row and column of the equation system (32) will be eliminated. So the matrix part in the bottom right of (32) will remain. With

(33) $$f = \tilde{g}_1+\tilde{g}_4+\tilde{g}_5$$

this solving against δh_1 results to:

(34)

$$\underbrace{\begin{bmatrix} (\tilde{g}_1+\tilde{g}_2+\tilde{g}_3)-\frac{\tilde{g}_1}{f}\tilde{g}_1 & -\tilde{g}_3 & -\tilde{g}_2 & -\frac{\tilde{g}_1}{f}\tilde{g}_4 & -\frac{\tilde{g}_1}{f}\tilde{g}_5 \\ -\tilde{g}_3 & \tilde{g}_3 & 0 & 0 & 0 \\ -\tilde{g}_2 & 0 & -\tilde{g}_2 & 0 & 0 \\ -\frac{\tilde{g}_4}{f}\tilde{g}_1 & 0 & 0 & \tilde{g}_4-\frac{\tilde{g}_4}{f}\tilde{g}_5 & -\frac{\tilde{g}_4}{f}\tilde{g}_5 \\ -\frac{\tilde{g}_5}{f}\tilde{g}_1 & 0 & 0 & -\frac{\tilde{g}_5}{f}\tilde{g}_4 & \tilde{g}_5+\tilde{g}_6-\frac{\tilde{g}_5}{f}\tilde{g}_5 \end{bmatrix}}_{A^{(r)}} \delta\vec{h}^{(r)} = \underbrace{\begin{bmatrix} \delta q_2^{nod}+\frac{\tilde{g}_1}{f}\delta q_1^{nod} \\ \delta q_3^{nod} \\ \delta q_4^{nod} \\ \delta q_5^{nod}+\frac{\tilde{g}_4}{f}\delta q_1^{nod} \\ \delta q_6^{nod}+\frac{\tilde{g}_5}{f}\delta q_1^{nod} \end{bmatrix}}_{\delta\vec{q}^{(r),nod}}$$

with $A^{(r)}$ and $\delta\vec{q}^{(r),nod}$ as marked above. The properties of $A^{(r)}$ are summarised in Ulanicki et al (1996).

Now a non-linear model is recovered from the reduced model (34). Ulanicki et al (1996) writes, that a non-zero entry at a position *(i,j)* in the matrix $A^{(r)}$ indicates a pipe between the nodes N_i and N_j. The new linear pipe conductance for this pipe is $\tilde{g}_{i,j}^{(r)} = -A_{i,j}^{(r)}$. The new non-linear pipe conductance $\tilde{g}_{i,j}^{(r)}$ can be calculated using the relation

(35) $\quad g_{i,j}^{(r)} \cdot e_3 \cdot \left|h_i^0 - h_j^0\right|^{e_4} = \tilde{g}_{i,j}^{(r)}$, corresponding to (18).

The following new network topology corresponds to $A^{(r)}$ (equation (34)):

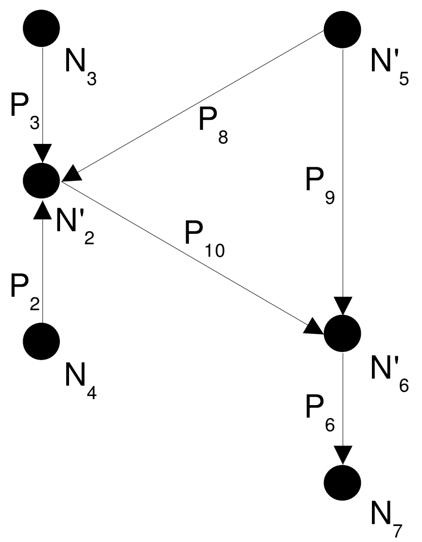


Figure 10. The new Network Structure after removing N_1.

III. 2. 1. b. Interpretation

Removing of Node N_1 resulted in the new pipes p_8 between N'_2 and N'_5, p_9 between N'_5 and N'_6 and p_{10} between N'_2 and N'_6. The nodal demands in the Nodes N'_2, N'_5 and N'_6 were changed.

So the star pipe structure around N_1 was exchanged with a triangle pipe structure between N'_2, N'_5 and N'_6. The simplified model has 4 pipes and 5 nodes.

III. 2. 2. Pipes in Series

Now, two pipes in series will be unified. In Figure 11, the source is N_5. The flow direction is from N_5 through N_3, N_1 and N_2 to N_4.

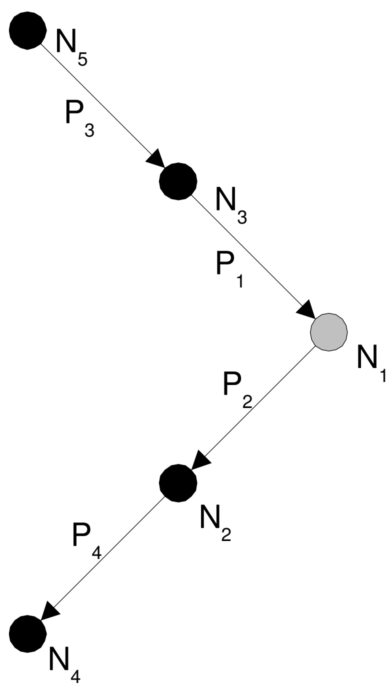


Figure 11. Network Model of Pipes in Series.

The node N_1 will be removed from the system. Therefore, the pipes P_1 and P_2 will be unified.

III. 2. 2. a. Mathematics

As the strategy of static simplification was described in detail in the last section, from now on only significant steps and results will be explained.

The branch incidence matrix Λ:

$$(36) \qquad \Lambda = \begin{array}{c} \\ N_1 \\ \vdots \\ \vdots \\ N_4 \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & \cdots & \cdots & p_4 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array}.$$

The row of node 5 was excluded because of the dependency, equation (14) above.

The Jacobian matrix A evaluates to

$$(37) \qquad A = \left[\begin{array}{c:ccc} \tilde{g}_1 + \tilde{g}_2 & -\tilde{g}_2 & -\tilde{g}_1 & 0 \\ \hdashline -\tilde{g}_2 & \tilde{g}_2 + \tilde{g}_4 & 0 & -\tilde{g}_4 \\ -\tilde{g}_1 & 0 & \tilde{g}_1 + \tilde{g}_3 & 0 \\ 0 & -\tilde{g}_4 & 0 & \tilde{g}_4 \end{array}\right].$$

Removing N_1 leads with $f = \tilde{g}_1 + \tilde{g}_2$ to

$$(38) \qquad A^{(r)} = \begin{bmatrix} \tilde{g}_2 + \tilde{g}_4 - \frac{\tilde{g}_2}{f}(\tilde{g}_2) & -\frac{\tilde{g}_1}{f}(\tilde{g}_2) & -\tilde{g}_4 \\ -\frac{\tilde{g}_2}{f}(\tilde{g}_1) & \tilde{g}_1 + \tilde{g}_3 - \frac{\tilde{g}_1}{f}(\tilde{g}_1) & 0 \\ -\tilde{g}_4 & 0 & \tilde{g}_4 \end{bmatrix} \text{ and}$$

$$\delta\vec{q}^{(r),nod} = \begin{bmatrix} \delta q_2^{nod} + \frac{\tilde{g}_1}{f}\delta q_1^{nod} \\ \delta q_3^{nod} + \frac{\tilde{g}_2}{f}\delta q_1^{nod} \\ \delta q_4^{nod} \end{bmatrix}.$$

The following new network topology can be read from $A^{(r)}$ (38):

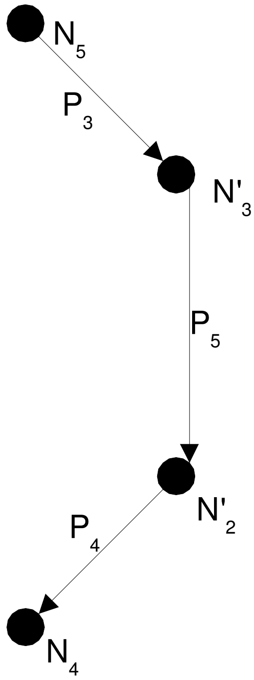


Figure 12. Network Model of Pipes in Series, with one Node removed.

III. 2. 2. b. Interpretation

After removing the node $N_{1,}$ the new pipe P_5 gets the linear conductance $\tilde{g}_5 = \frac{\tilde{g}_1 \cdot \tilde{g}_2}{f}$.

P_5 is the replaces P_1 and P_2. The demand of N_1 is redistributed to the nodes N'_2 and N'_3.

III. 2. 3. Parallel Pipes

Now, two parallel pipes will be unified. The source in the network model is N_4, from there, the water flows to N_1 and N_5. From N_1, the flow passes through two parallel pipes to N_2; from N_2, the flow goes to N_3.

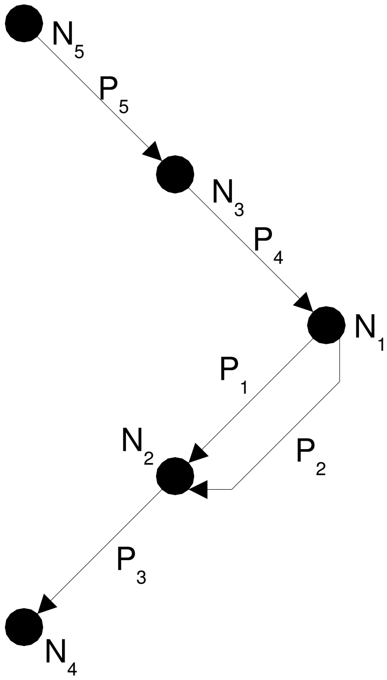


Figure 13. Network Model with Parallel Pipes.

III. 2. 3. a. Mathematics

The node branch incidence matrix Λ:

$$(39) \qquad \Lambda = \begin{array}{c} \\ N_1 \\ \vdots \\ \vdots \\ N_4 \end{array} \begin{array}{c} \begin{array}{ccccc} p_1 & \cdots & \cdots & \cdots & p_5 \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & -1 & 0 \\ 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \end{array}.$$

The demand of N_5 will be calculated by the dependcency for the demand, similar to equation (14). P_1 and P_2 have the same entries in Λ. This denotes from their identical start and end nodes.

The Jacobian matrix *A* results to:

$$(40) \qquad A = \begin{bmatrix} \tilde{g}_1 + \tilde{g}_2 + \tilde{g}_4 & -\tilde{g}_1 - \tilde{g}_2 & 0 & -\tilde{g}_4 \\ -\tilde{g}_1 - \tilde{g}_2 & \tilde{g}_1 + \tilde{g}_2 + \tilde{g}_3 & -\tilde{g}_3 & 0 \\ 0 & -\tilde{g}_3 & \tilde{g}_3 & 0 \\ -\tilde{g}_4 & 0 & 0 & \tilde{g}_4 + \tilde{g}_5 \end{bmatrix}.$$

The linear pipe conductances of P_1 and P_2 were simply added in A, so no Gaussian Elimination is needed. Therefore, the new network topology can be read directly from the Jacobian matrix A:

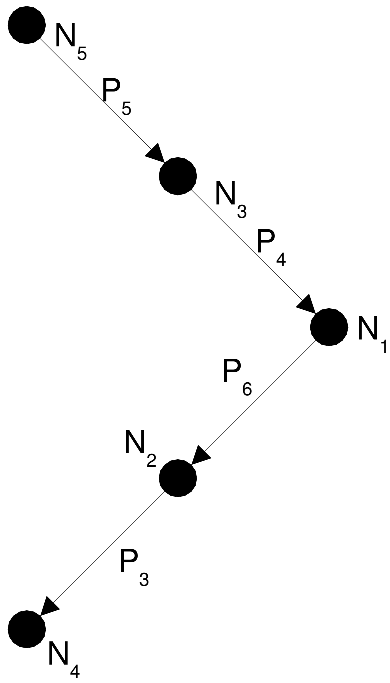


Figure 14. Network Model with Unified Parallel Pipes.

III. 2. 3. b. Interpretation

The linear pipe conductances $\tilde{g}_1$ and $\tilde{g}_2$ are added when the Jacobian matrix of the model is calculated. The single pipe P_6 linking N_1 and N_2 has the linear conductance $\tilde{g}_6 = \tilde{g}_1 + \tilde{g}_2$.

III. 2. 4. Trees

In this section, a tree in a network model will be removed with the static simplification method. The source in the network model illustrated below is the node N_4. From there, the flow goes to N_3 and from N_3 to N_5 and to N_1 via N_2. The network model has 5 nodes and 4 pipes. The pipes P_1 and P_2, which form the tree, will be removed.

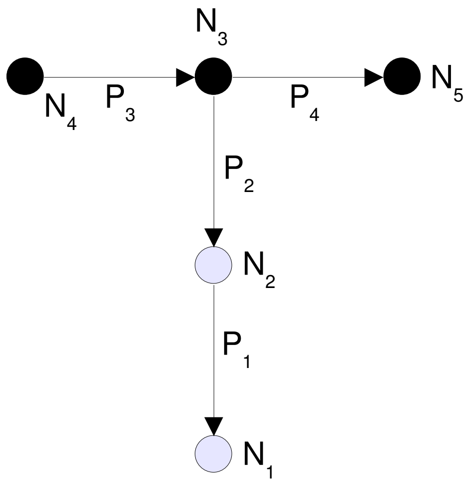


Figure 15. A Water Network Model with a Tree.

There are two different methods to do this:

- adding the demands directly
- with static simplification

Firstly, the algorithm for adding the demands directly will be explained.

III. 2. 4. a. Adding the Demands

The head in N_1 is a function of the head in N_2 and the flow in P_1, which equals to the demand in N_1 (see equation (2)):

$$(41) \qquad h_1 = h_1\left(h_2, q_1^{nod}\right) .$$

The head H_2 is a function of the head in N_3 and the flow through P_2, equal to $q_1^{nod} + q_2^{nod}$:

$$(42) \qquad h_2 = h_2\left(h_3, q_1^{nod} + q_2^{nod}\right).$$

The head h_3 in N_3 does not depend on the head of N_2. The only influence from N_2 and N_1 results from the demands q_1^{nod} and q_2^{nod}:

$$(43) \qquad h_3 = h_3\left(\ldots, q_1^{nod} + q_2^{nod} + q_3^{nod}\right).$$

So, the demands of N_1 and N_2 can be added to the demand of N_3:

(44) $$q_3^{nod,new} = q_1^{nod} + q_2^{nod} + q_3^{nod}$$

without loosing accuracy. Now, N_1 and N_2 can be deleted.

The network topology after removing the branch:

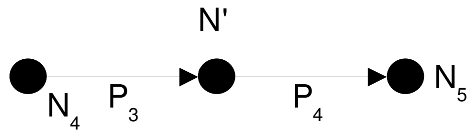


Figure 16. Water Network Model with removed Tree.

III. 2. 4. b. Static Simplification

In this section, the tree of the network model in Figure 15 will be removed with the static simplification algorithm.

The node branch incidence matrix Λ evaluates to:

(45) $$\Lambda = \begin{array}{c} \\ N_1 \\ N_2 \\ N_3 \\ N_4 \end{array} \begin{array}{c} \begin{array}{cccc} P_1 & P_2 & P_3 & P_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \end{array}.$$

The row for N_5 is excluded from Λ because of the dependcency (14).

The Jacobian matrix A results to:

(46) $$A = \left[\begin{array}{cc|cc} \tilde{g}_1 & -\tilde{g}_1 & 0 & 0 \\ -\tilde{g}_1 & \tilde{g}_1 + \tilde{g}_2 & -\tilde{g}_2 & 0 \\ \hline 0 & -\tilde{g}_2 & \tilde{g}_2 + \tilde{g}_3 + \tilde{g}_5 & -\tilde{g}_3 \\ 0 & 0 & -\tilde{g}_3 & \tilde{g}_3 \end{array} \right].$$

Now, the nodes N_1 and N_2 will be removed. Hence, the Jacobian matrix $A^{(r)}$ of the reduced model is:

(47) $$A^{(r)} = \begin{bmatrix} \tilde{g}_3 + \tilde{g}_5 & -\tilde{g}_3 \\ -\tilde{g}_3 & \tilde{g}_3 \end{bmatrix}.$$

The reduced demand vector evaluates to:

$$(48) \qquad \vec{q}^{\,nod,(r)} = \begin{bmatrix} q_1^{nod} + q_2^{nod} + q_3^{nod} \\ q_4^{nod} \end{bmatrix}.$$

As shown above, the demands are added without being weighting with the pipe conductances to the start node of the tree, here N_3. The network structure, which can be derived from the Jacobian matrix of the reduced model (47), is identical to the one in Figure 16.

III. 2. 5. Low Conductance Pipes

Elimination of low conductance pipes is not directly possible with the static simplification approach that is carried out here, because the approach is based on relinking selected nodes. So, these pipes will be simply deleted from the network model.

Low conductance pipes can be removed from the network model

- before linearising as well as
- after removing nodes with the Gaussian elimination procedure.

Static simplification splits the connections of a node to connections between the nodes, to which it is coupled. These new pipes conductances between the remaining nodes may be much lower than the lowest pipe conductance in the original network. Therefore, the conductance of a low conductance pipe might increase as well, when the connections around a node are redistributed. Hence, it makes more sense to delete low conductance pipes after the simplification, so as much as possible from the structure of the original network will remain and new low conductance pipes will be eliminated.

Low flow in a pipe is not only the consequence of a low pipe conductance, but also of a big head loss. Therefore, the head loss should be regarded as well. The linearised pipe conductance $\tilde{g}_i$ reflects the head loss, so it will be the classification variable in this work, instead of the non-linear pipe conductance $\tilde{g}_i$.

With the linearised pipe conductance of the static simplification algorithm, there are two possible decision criteria to identify low conductance pipes:

A fixed *absolute* linear pipe conductance level for the whole water network model.

A linear pipe conductance level *relative* to the smallest linearised node conductance of start and end nodes of the pipes.

Both classification criteria were implemented in this work. They were integrated in the program module, which links the remaining nodes after simplifying. The second decision criterion did sometimes not remove all pipes, which caused solver errors. These solver errors appeared unpredictable in the simulations and were sometimes half of the order of the available head in the nodes. Further, the normal errors caused by this criterion were generally much higher than the ones of the first classification criterion. So the fixed absolute linear pipe conductance level as decision criterion for low conductance pipes is much better than the relative one.

III. 3. Implementation

In this section, the modelling environment will be briefly explained. Afterwards, the activities performed in the algorithm to identify the simplification range and in the static simplification algorithm will be discussed.

III. 3. 1. Modelling Environment

The application for the static simplification is called “SpeedUp” in this report. It is integrated with the geographic information system “StruMap” (Structural Technologies LTD. (1996)). StruMap and SpeedUp communicate via the application programmer’s interface, “API”, of StruMap via an interface in C++. SpeedUp was developed with the C++ Builder (i.e. Calvert (1997)). The C++ Builder compiles the source code of SpeedUp and the interface to the API of StruMap. Further, for representing matrices in SpeedUp, a matrix library is used.

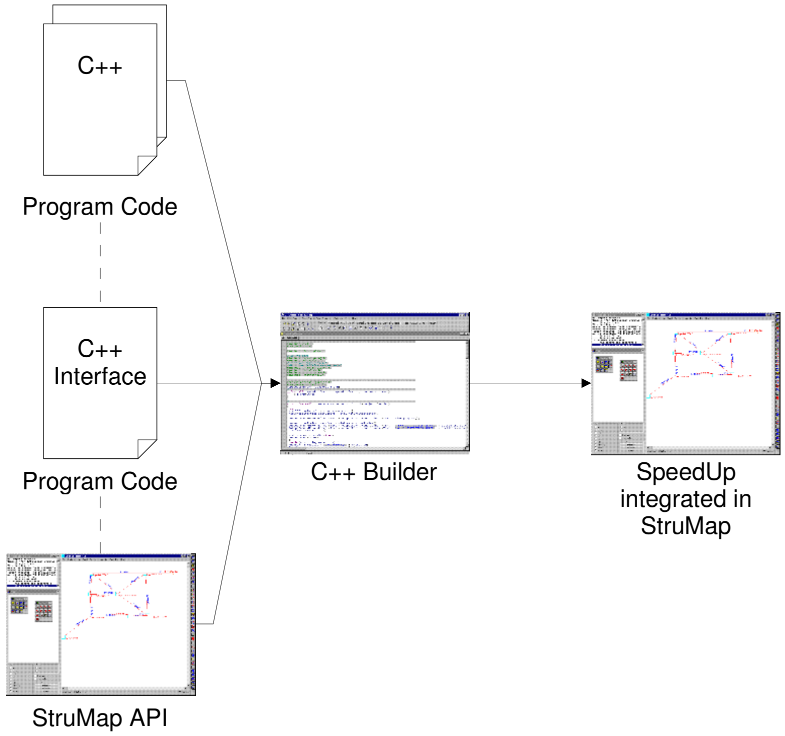


Figure 17. The Modelling Environment.

The parts of the modelling environment will be briefly explained now.

III. 3. 1. a. StruMap

The Centre for Water Systems of the School of Engineering and Computer Science uses the modelling environment “StruMap” from “Structural Technologies Limited”. It provides an environment to work with data contained in a map: it is a geographic information system. To simulate water networks, StruMap comes with a solver for water network models, HARP. Attributes of Map items can be manipulated with a built-in expression evaluator, an interpreter language programming interface. StruMap comes as a set of library files. The functions in them are accessible via an application programmer’s interface, “API”, to C. So external programmers can add custom functions to StruMap. The API is described in Structural Technologies (1996). The API itself is written in C. The StruMap version, which is used for this work, is “StruMap 2000 Release 1.3 990104” for Microsoft® Windows 95™.

III. 3. 1. b. C++ Interface to StruMap

Roger Atkinson of the Centre of Water Systems has written an excellent interface class for the StruMap API. This class encapsulates the functions of the API as methods. It was extended in this work.

III. 3. 1. c. Matrix Library

To represent the Jacobian matrix A in the program, the data structures of Press et al (1992) were used. The used functions allocate memory for the matrix and free it.

III. 3. 1. d. Programming Environment

The algorithms for this work were coded in C++ with the "Borland C++ Builder Professional Version 3.0 (Build 3.70)" from Borland International. As C++ dialect, the Borland version was used. It provides a range of functions, classes and methods specifically designed for Microsoft® Windows. For documentation about the C++ Builder, see Calvert (1997).

III. 3. 2. The Algorithm

This section illustrates the algorithms for the simplification preparation and static simplification, which were developed in this work.

III. 3. 2. a. Simplification Preparation

Figure 18 shows the activity diagram for the simplification preparation. The simplification preparation consists of three main sections: the deletion of closed valves, the protection of untouchable nodes with untouchable pipes and the following protection of untouchable pipes with non-removable nodes.

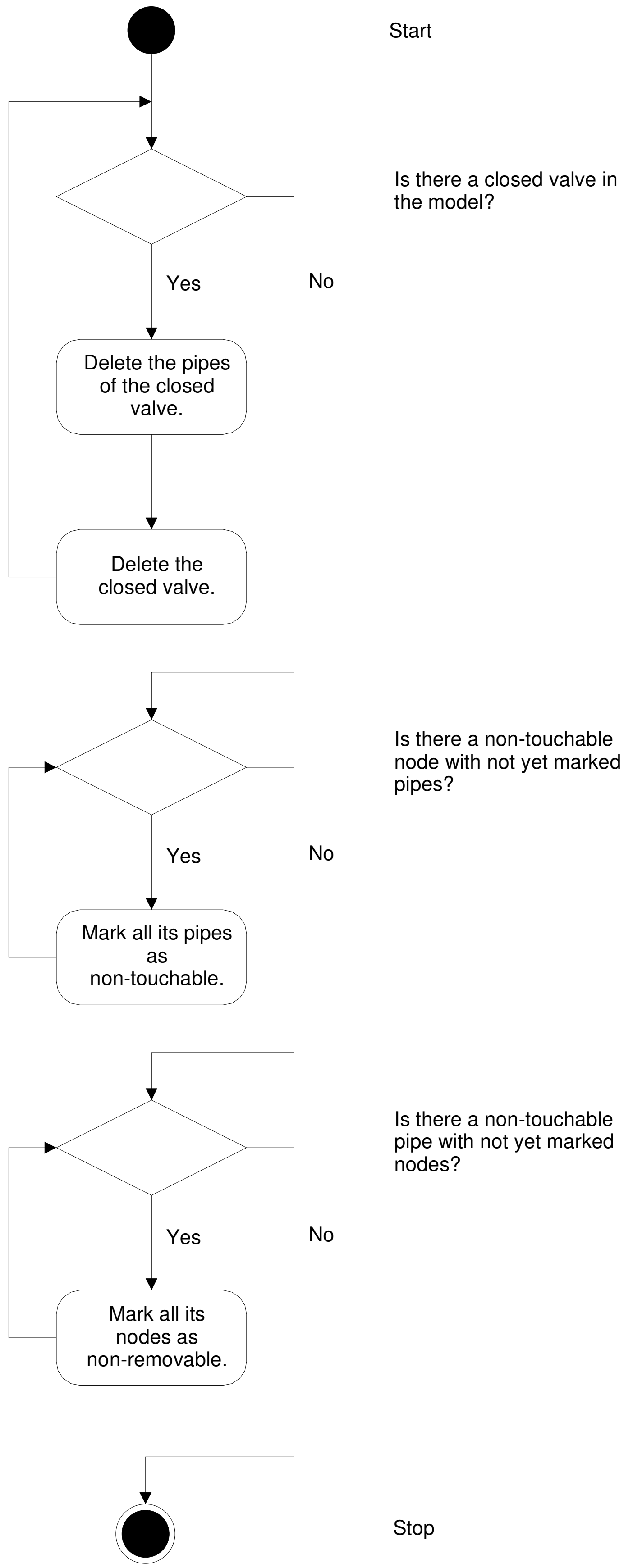


Figure 18. Activity Diagram for the Simplification Preparation.

The simplification preparation will delete closed valves, because there is no flow through them and therefore no headloss in their pipes. Pipes without headloss have a zero linear pipe conductance (see equation (18) above). Pipes with a zero linear pipe conductance cannot represent a link in the Jacobian matrix *A*, as a zero entry in *A* denotes no connection between the nodes. So network components without flow will be excluded from simplification.

The protection of untouchable nodes and afterwards the protection of untouchable pipes is carried out as described in section III. 1. 2. above.

III. 3. 2. b. Static Simplification

The activity diagram for the static simplification algorithm is plotted in Figure 20 and Figure 21 below.

Several activities are independent from each other, so they can be carried out in parallel. These activities are placed between two horizontal black bars.

Firstly, a list of all nodes of the water network model has to be made. All non-removable nodes, which will remain after simplifying, are put at the end of the list. This is advantageous, because the rows of the Jacobian matrix *A* will then not need to be sorted. Also, the node index for the rows and columns of *A* will be the same.

Likewise, a list of all pipes is required. Their start and end nodes need to be identified and their linear pipe conductance needs to be calculated as well. To bypass possible flow errors of the solver, the linear pipe conductance is not calculated in terms of flow and head loss, but in terms of the pipe attributes length, diameter, Hazen-Williams coefficient and head loss.

The Jacobian matrix needs to be initialised with a size of $(N_N - 1) \times (N_N - 1)$, because of the dependcency for the demands as explained in III. 1. 1. a. on page 25. All matrix elements have to be initialised with zero.

Now, the linear pipe conductances can be placed in the matrix *A*. This is done by substracting them from their matrix elements:

(49) $A_{n,m} := A_{n,m} - \tilde{g}_{n,m}$ and

$A_{m,n} := A_{m,n} - \tilde{g}_{n,m}$ with $n, m \in N_N$.

Thereafter, the linear node conductances, which form the diagonal band of the matrix, can be calculated. This can be done with equation (19):

$$\tilde{g}_n \mathrel{\hat{=}} \sum_{k \in N_n, k \neq n} \tilde{g}_{k,n} \qquad n = 1(1)N_N .$$

Before the Gaussian elimination can be performed, the demand vector has to be built according to the node index for the matrix *A*.

Now, the Gaussian elimination can be applied on the Jacobian matrix *A* and the demand vector. The Gaussian elimination was implemented in this work as follows:

```
// SIMPLIFY: Gauss-Elimination ========================================
for (int i=0; i< numNodesToRemove; i++) {
// iterate through all remaining rows, jacii !=0 for all rows
for (int j=i+1; j<=jac_size; j++)
     // is there something to do in the current row?
     if (jacji != 0) {
            double mult= -jacji / jacii;
            // mult is a multiplication constant, which is used at least
            // three times
            jacji = 0;
            // iterate through all columns
            for (int k=i+1; k<=jac_size; k++)
                 // is there something to do in the current column?
                 if (jacik != 0) {
                     // add (mult * jacik) to jacjk ===============
                     jacjk += mult * jacik;
                    };
                 // the following lines allocate the demand objects in memory
                 demandType demand  = (demandType)demandList->Itemsi;
                 demandType demand1 = (demandType)demandList->Itemsj;
                 // add the demands ========================================
                 demand1->addDemand(demand, mult);
                }; // end of row iteration
     }; // end of column iteration
```

The Jacobian matrix *A* is in the code above simply called `jac`. `i` is the index of the current diagonal element and `j` is the index for the actual row in `jac`. `k` is an index for the column, in which additions are carried out. The syntax of C++ is described in Calvert (1997).

$$
\begin{array}{cc}
 & \qquad\qquad\qquad k \rightarrow \\
A = \begin{array}{c} j \\ \downarrow \end{array} & \left[\begin{array}{cc:ccc}
\ddots & & & & \\
 & jac[i-1][i-1] & & & \\ \hdashline
 & \vdots & jac[i][i] & \cdots & jac[i][k] \\
\cdots & 0 & \vdots & & \vdots \\
\cdots & 0 & jac[j][i] & \cdots & jac[j][k] \\
 & \vdots & & &
\end{array}\right]
\end{array}
$$

Figure 19. Structure of the Jacobian matrix *A*.

As the matrix *A* is a sparse one, the algorithm always tests firstly, if there is something to do: when `jac[j][i]` is zero, the algorithm ignores this row, because there is nothing to eliminate. If `jac[i][k]` is zero, the algorithm ignores the column `k`. The solving algorithm was not further optimised, because the running time spent with Gaussian elimination is very small compared to the hard disk access time, which is needed to write the changed demand vectors.

After the Gaussian elimination is finished, all pipes, except of the untouchable ones are dispensable and can be deleted.

Now, the untouchable pipes need to be removed from the matrix *A*. This is done by adding their linear pipe conductance to the corresponding matrix position in *A*:

$$
(50) \qquad A_{n,m} := A_{n,m} + \tilde{g}_{n,m} \qquad \text{and}
$$

$$
A_{m,n} := A_{m,n} + \tilde{g}_{n,m} \qquad \text{with } n, m \in N_N .
$$

If the matrix element $A_{n,m}$ is still lower than zero, an additional pipe parallel to the untouchable one will be inserted automatically later on to correct the flow.

Now, all nodes, except of the non-removable ones can be deleted, as they are no longer required. The demand of the non-removable nodes can be written back.

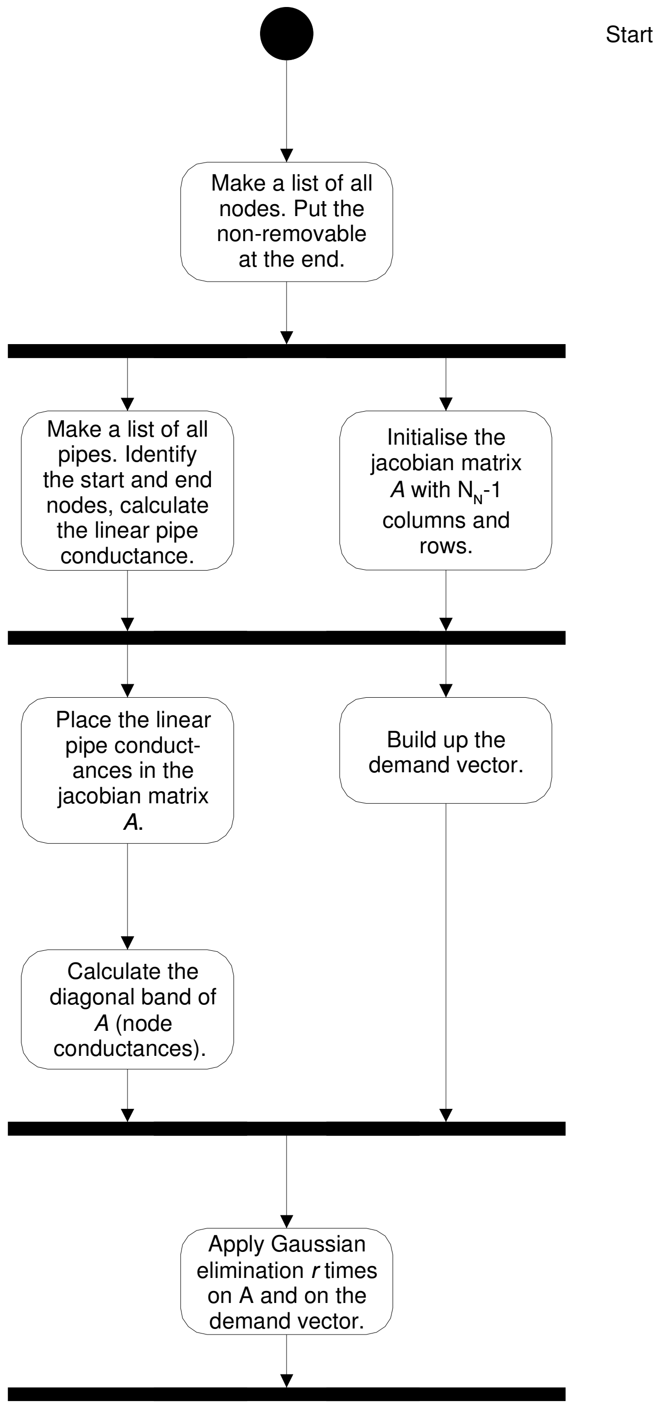


Figure 20. Part 1 of Activity Diagram for the Static Simplification.

Finally, the remaining nodes need to be reconnected with the pipe structure in $A^{(r)}$. $A^{(r)}$ is a submatrix in A from the position $(r+1)\times(r+1)$ to $(N_N-1)\times(N_N-1)$. If there is a non-zero linear pipe conductance entry in *A* at the position *(m, n)*, which meets the writeback criteria, a pipe need to be inserted in the reduced model between the node *m* and *n*. There are two writeback criteria:

Relative writeback criterion.

The linear pipe conductance $A_{n,m}$ is bigger than its lowest linear node conductance multiplied by a factor *rfact*:

$$(51) \qquad A_{n,m} > rfact \cdot \min(A_{n,n}, A_{m,m}) \qquad n, m \in (r+1) \ldots (N_N - 1) .$$

Absolute writeback criterion.

The linear pipe conductance $A_{n,m}$ is bigger than the lowest linear pipe conductance of the original network model multiplied by a factor *afact*:

$$(52) \qquad A_{n,m} > afact \cdot \min(\{A_{i,j}\}) \qquad n, m \in (r+1) \ldots (N_N - 1) \quad \text{and}$$
$$i = 1,2,\ldots,(N_N - 1) \quad \text{and}$$
$$j = 1,2,\ldots,i .$$

The lowest linear pipe conductance of the original network model was chosen as a scale. This allows a comparison relative to the linear pipe conductance of the original network model.

After simplification, some statistics are displayed to compare the simplified with the original network model.

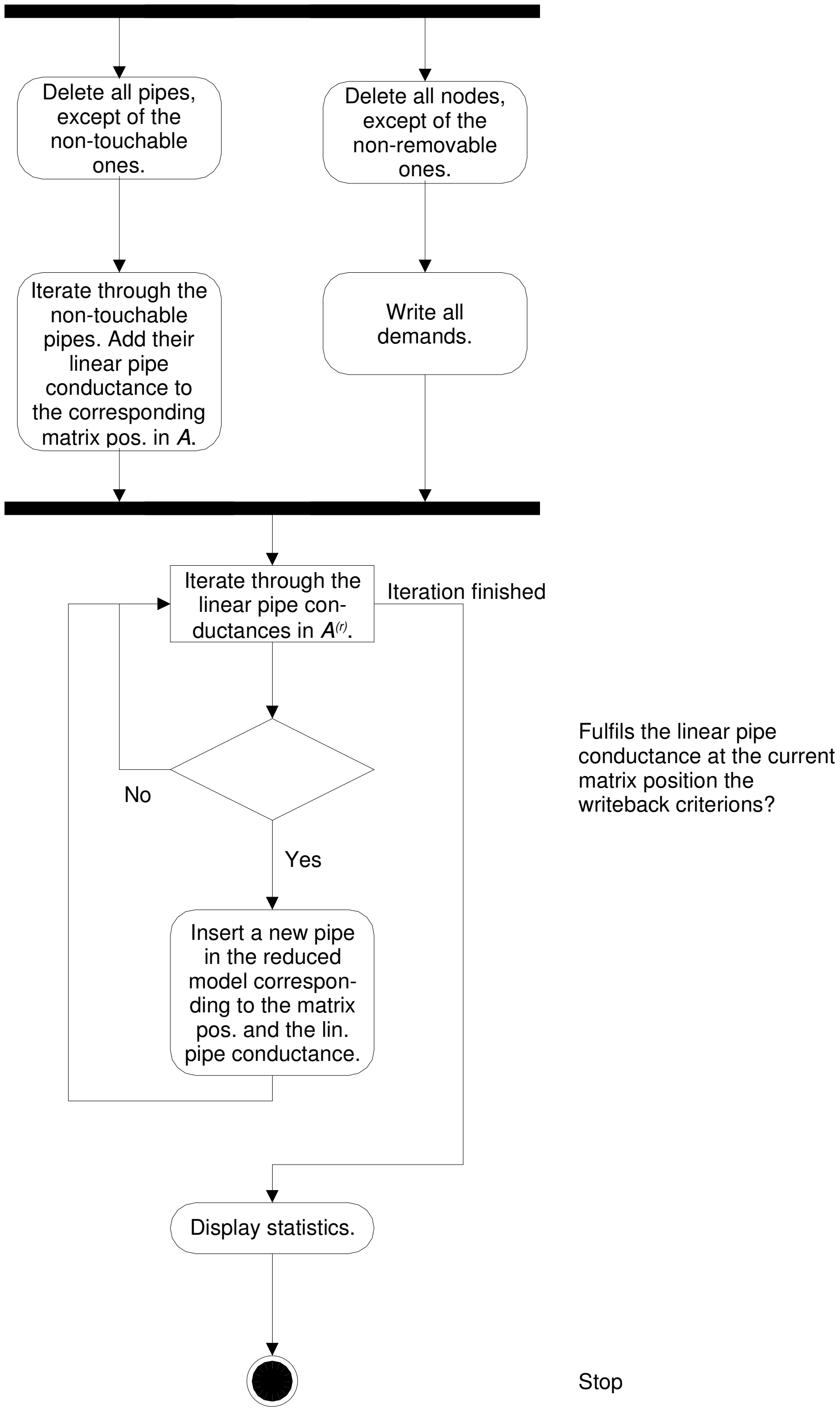


Figure 21. Part 2 of Activity Diagram for the Static Simplification.

Maschler, T. and D.A. Savic, (1999) Simplification of Water Supply Network Models through Linearisation, Centre for Water Systems, Report No.99/01, School of Engineering, University of Exeter, Exeter, United Kingdom, p.119.

III. 4. Example

In this section, the example network from above with the identified simplification boundary (Figure 8, page 35) in Chapter III. 1. 2. will be simplified with the static simplification algorithm. Low conductance pipes will not be eliminated, as the main intention of the example is to illustrate static simplification.

III. 4. 1. Procedure

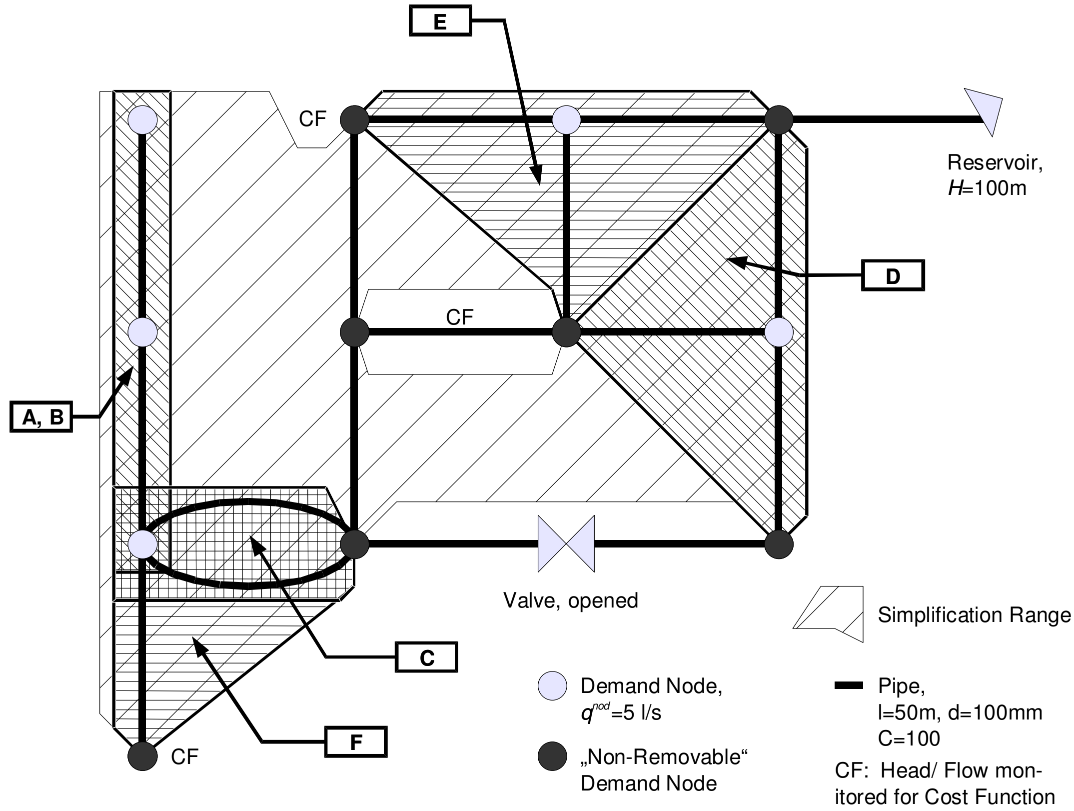


Figure 22. The Example Network with marked Simplification Areas.

In this network model, the static simplification will carry out the following simplification steps:

Table 5. Simplifications Steps in the Example Network.

Simplification Step	Marked Area in Figure 22 above
Unification of Series Pipes	A
Deletion of a Tree Structure	B
Unification of Parallel Pipes	C
Node Replacement	D, E, F

The simplification areas of these steps are illustrated in Figure 22.

The resulting network is shown in Figure 23. The static simplification replaces the network structures in the area A, B, C and F with one direct pipe. The demand of the removed nodes is added to the start and end node of this pipe. The star structures in the areas D and E are replaced with two triangle structures. The demand of the removed nodes is redistributed to the 4 corner nodes of the triangle structures.

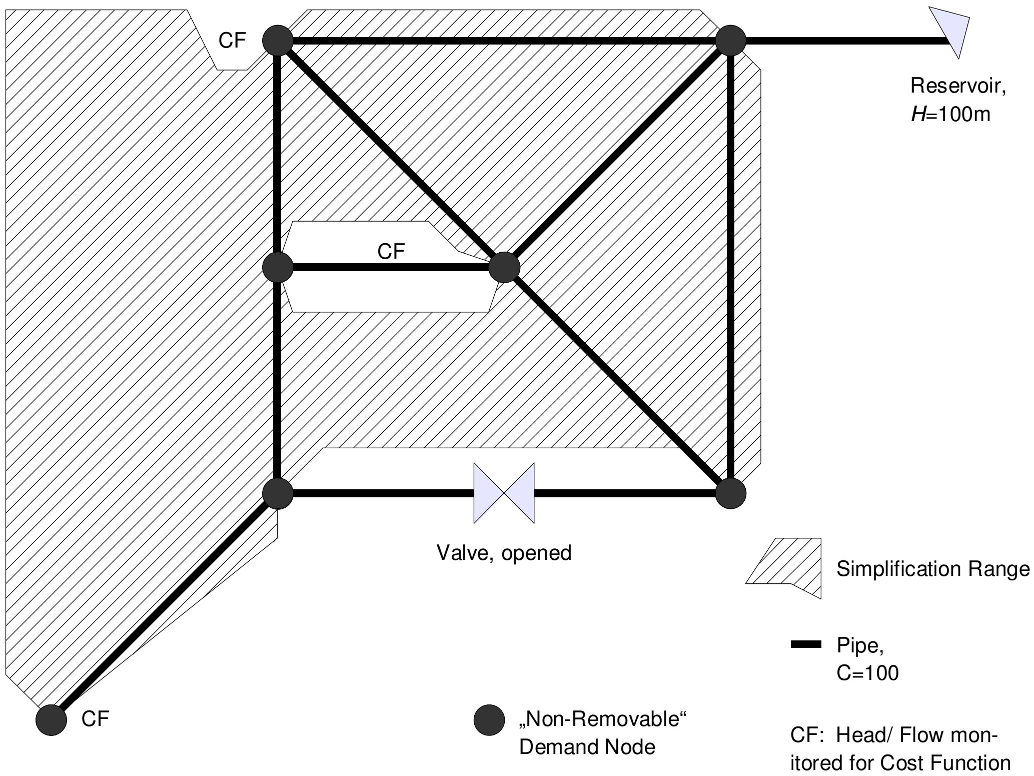


Figure 23. The simplified example network.

The simplified network model has 12 pipes and 9 nodes. Its number of loops is 4. The original network model has 17 pipes and 14 nodes. So the simplification will reduce the number of nodes to 9 and the number of pipes to 12, but the number of loops will stay the same.

III. 4. 2. Program Test

As a test, the simplification of the example network will be carried out with the static simplification algorithm in StruMap.

The screenshot below shows the heads and flows in the example water network model. A negative flow in a pipe indicates that the end node of a pipe has a higher head than its start node.

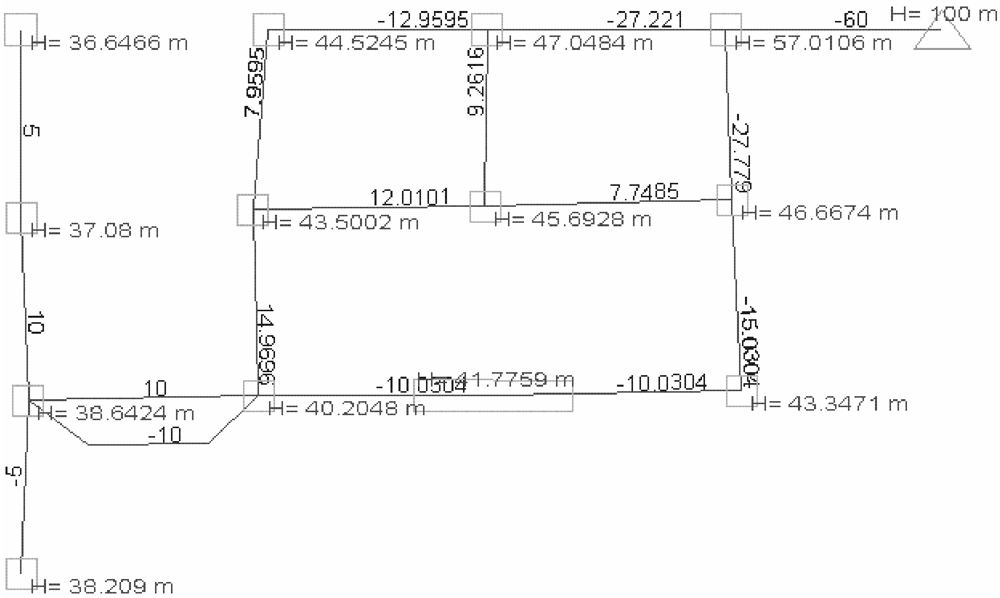


Figure 24. Screenshot of the Original Example Network[3].

III. 4. 2. a. Preparations

The simplification preparation identifies the boundary of the simplification area. Its output:

```
SIMPLIFICATION PREPARATION


      STATISTICS BEFORE
==========================================

Nodes:            14     non-removable:     2
Pipes:            17
Loops:            4
```

[3] The numbers without variable name and unit are the flows l/s.

```
N. of NW[4] Components:      31
The pipes represent 18 %         of all possible links.
The loops represent 50 o/oo      of all possible loops.

       STATISTICS AFTER
===============================================

Nodes:              14     non-removable:      9
Pipes:              17
Loops:              4
N. of NW Components:      31
The pipes represent 18 %         of all possible links.
The loops represent 50 o/oo      of all possible loops.
```

The network had before the simplification preparation only 2 non-removable nodes, whose head is output for the cost function. After the preparation, it has 9 non-removable nodes. Untouchable nodes are included in this number.

III. 4. 2. b. Simplification

Now the example network model is solved at the working point and the calculated heads and flows are assigned to the network model components. Afterwards, the network model is simplified. The output of the static simplification algorithm:

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 13x13 .

Reading pipes back.
0 had a linear branch conductance lower than
the draw-back-criteria and were not written back.


       STATISTICS BEFORE
===============================================

Nodes:              14     non-removable:      9
Pipes:              17
Loops:              4
N. of NW Components:      31
The pipes represent 18 %         of all possible links.
The loops represent 50 o/oo      of all possible loops.

       STATISTICS AFTER
===============================================

Nodes:              9      non-removable:      9
```

[4] "NW" stands for **n**et**w**ork.

Maschler, T. and D.A. Savic, (1999) Simplification of Water Supply Network Models through Linearisation, Centre for Water Systems, Report No.99/01, School of Engineering, University of Exeter, Exeter, United Kingdom, p.119.

```
Pipes:              12
Loops:              4
N. of NW Components: 21
The pipes represent 32 %         of all possible links.
The loops represent 137 o/oo     of all possible loops.

       COMPARISON
================================================

              New          Old          Compared

Nodes:        9            14           -35 %
Pipes:        12           17           -29 %
Loops:        4            4            0 %
NW Components: 21          31           -32 %
```

The Jacobian matrix A has the dimensions 13×13. The number of nodes after simplification is nine, so only the non-removable nodes have stayed. The new water network model has 5 pipes less than the original one. The number of loops has not changed — there are still 4 loops. These results are the same as predicted in the section above.

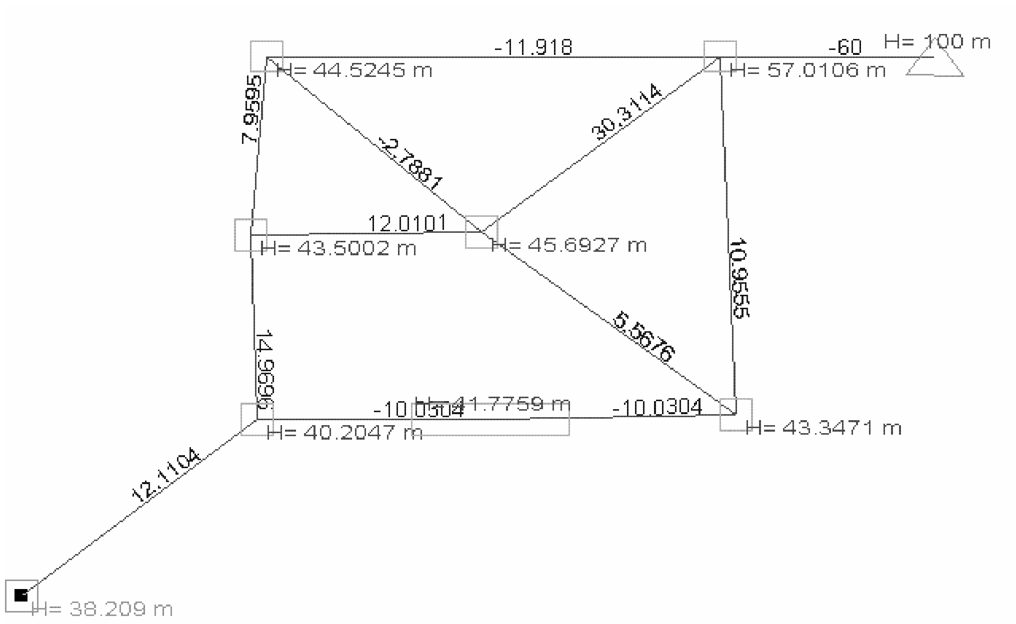


Figure 25. The Simplified Example Network[5].

The structure of the simplified network model in Figure 25 is similar to the proposed one in Figure 23. The heads in all the nodes are the same than the ones in the original

[5] The black point in the bottom left edge in the screenshot is the cursor.

network model in Figure 24 above. The flows in the untouchable pipes have stayed the same as well.

III. 5. Summary

This chapter introduced the static simplification algorithm of Ulanicki et al (1996) and showed, that the network model component based simplification approaches node unification, parallel and series pipes unification and tree elimination are carried out by the static simplification algorithm.

The static simplification algorithm was enhanced with two methods to eliminate low conductance pipes after simplifying the network model. As well, a method was presented to identify the simplification range in a network model. These are the main extensions of Ulanicki et al (1996).

Then, the algorithm for the program was presented in form of activity diagrams. Finally, a simplification example was given with the static simplification algorithm in this work.

IV — CASE STUDIES

This chapter describes the simplification procedure and the results for two large water network models. The model simplifications will be carried out for several cases with the absolute and the relative writeback criterion. Further, the solving time of the original water network models and the simplified ones will be benchmarked.

IV. 1. Skipton Water Network Model

IV. 1. 1. Brief Description

Skipton is a rural town situated in North Yorkshire, United Kingdom. Yorkshire is an English shire next to the boundary of Scotland. Skipton is situated on the south boundary of the Yorkshire Dales, approximately 35km north-west of Leeds.

Figure 26. Skipton.

The Skipton water supply network model consists of about 1000 nodes and approximately 1300 pipes. The number of loops is roughly 200. A screenshot of the Skipton water network model can be found in chapter II. 1. 1. on page 11.

The network contains the following special network components:

- ➢ 2 "Variable Head Reservoirs" (StruMap Property "R_VHR").
- ➢ 1 "Floating Valve" (StruMap Property "V_FL1").
- ➢ 42 "Motorised Throttling Valves" (StruMap Property "V_MTV").
  36 of them are totally closed and six are totally opened.

For a cost function, the flow in some pipes, as well as the heads of some nodes, will be chosen. As this report covers only the simplification algorithm to minimise the simulation time, the network components above will be chosen randomly.

IV. 1. 2. Preparations

All special network such as reservoirs, valves and sources were marked as "untouchable", as well as those pipes, whose flow will be used in the cost function: pipe 2146 and 1862. The nodes for the cost function were marked as "non-removable": node 723, 40 and 924. The number corresponds to their attribute "UID" in StruMap.

Now the simplification preparation routine was started. Its output:

```
       SIMPLIFICATION PREPARATION


       STATISTICS BEFORE
==================================================

Nodes:             1091   non-removable:      3
Pipes:             1295
Loops:             205
N. of NW Components: 2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
==================================================

Nodes:             1091   non-removable:      134
Pipes:             1295
Loops:             205
N. of NW Components: 2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.
```

The number of non-removable nodes increases from 3 to 134 as the simplification area is identified. This high number denotes of the numerous valves in the network model.

After the simplification preparation, the network was solved for a snapshot at 8 o'clock and the heads and flows were imported as the linearisation point for the network. The network was saved and SpeedUp restarted.

IV. 1. 3. Simplification

The simplification procedure was run for the following cases:

1. without writeback criterion

Three times with the following parameters with the *relative writeback criterion*:

2. 0.00001
3. 0.0001
4. 0.0002

And four times with the *absolute writeback criterion*:

5. 0.956
6. 2
7. 3
8. 10

The output of the static simplification routine can be found in the appendix, chapter VIII. 4. , which starts on page 106.

Depending on the level of the writeback criterion, the simplification procedure reduces the number of network components differently:

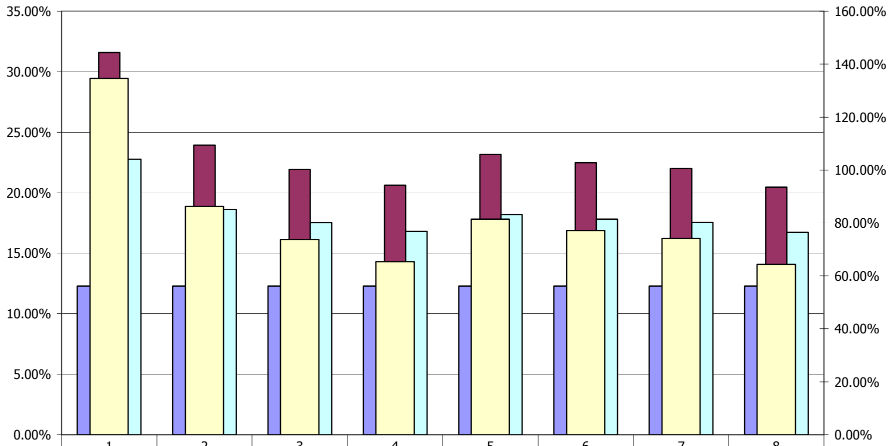
Table 6. Comparison of the Number of Network Components.

Case	Nodes	Pipes	Loops	NW Components	Pipes/ possible links	loops/ possible loops
Orig. NW	1091	1295	205	2386	0 %	0 ‰
1	134	409	276	543	9 %	68 ‰
2	134	310	177	444	7 %	44 ‰
3	134	284	151	418	6 %	37 ‰
4	134	267	134	401	6 %	33 ‰
5	134	300	167	434	7 %	41 ‰
6	134	291	158	425	7 %	39 ‰
7	134	285	152	419	6 %	37 ‰
8	134	265	132	399	6 %	32 ‰

The first column gives the case number, the next four columns contain the absolute numbers of nodes, pipes, loops and network components. The prefinal column refers the number of pipes to the possible number of network links as a percentage. The last column refers the number of loops to the possible number of loops as parts per thousand. Hence, the last two columns reflect the link density of the network model. The maximum number of links in a water network model was calculated as described in Appendix VIII. 3. on page 104.

Figure 27 shows the new numbers of network components relative to the number of network components in the original water network model:

[6] The lowest linear pipe conductance of the original network is kept in the simplified one. It is seen as a lower limit to which small changes are allowed. Therefore, if the solver has no problems with the original network, it will have no problems with the simplified network, also.



	1	2	3	4	5	6	7	8
Nodes	12.28%	12.28%	12.28%	12.28%	12.28%	12.28%	12.28%	12.28%
Pipes	31.58%	23.94%	21.93%	20.62%	23.17%	22.47%	22.01%	20.46%
NW Elements	22.76%	18.61%	17.52%	16.81%	18.19%	17.81%	17.56%	16.72%
Loops	134.63%	86.34%	73.66%	65.37%	81.46%	77.07%	74.15%	64.39%

Case

Figure 27. Numbers of Network Components after Simplification[7] compared to the ones before for the Skipton Water Network Model.

As expected, the number of nodes is reduced to the same value in each simplification. The number of pipes in the simplified network model varies, their number depends on the height of the relative and absolute writeback criteria. All simplification approaches with writeback criteria reduce the total number of pipes more than without writeback criterion. In the simplification with all pipes (case 1), the number of loops is about $^1/_3$ higher than in the original network model. With writeback criteria, the number of loops is between ~¼ and ~$^1/_3$ lower than in the original network model. The numbers of network components get reduced to ~23% for all pipes and down to ~17% for the highest relative and absolute writeback criteria.

[7] The percentage of loops refers to the secondary value axis, all others to the first one.

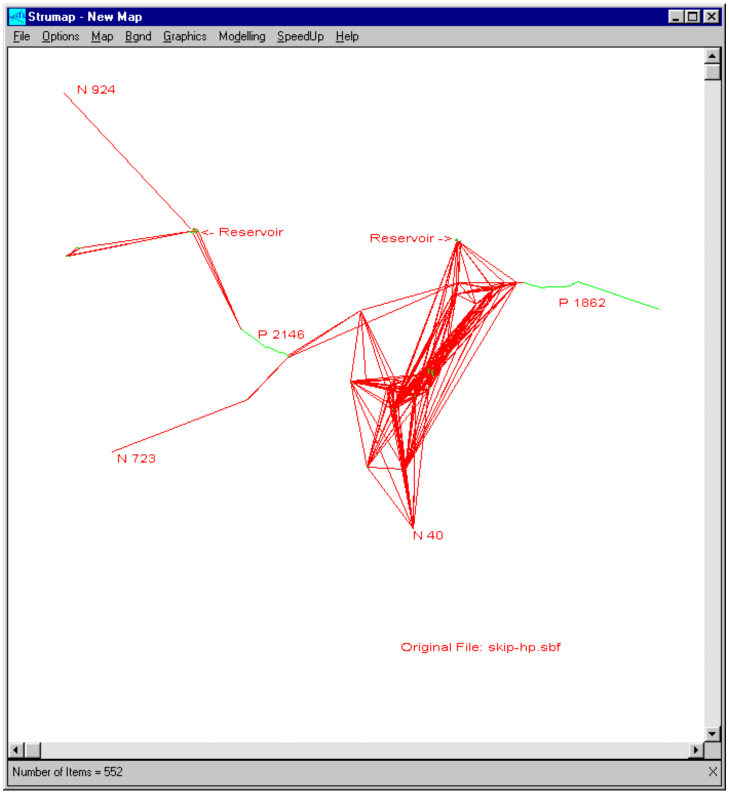


Figure 28. The network after Simplification,
all linear pipe conductances were written back.

The simplified network model with all pipes in Figure 28 has a totally different appearance than the original network model. The models, which were simplified with writeback criteria, have the same appearance, but their pipe networking is less dense.

IV. 1. 4. Errors at the 08:00 Snapshot

Now, the errors of the simplified models at the 8:00 snapshot will be discussed.

After simplification, only the non-removable nodes of a network model and its untouchable pipes will stay the same. Therefore, the head in the non-removable demand nodes and flows in the untouchable pipes will be analysed here. To facilitate the mapping between the heads and flows, the nodes and pipes are sorted in ascending order according to their label ("UID") in StruMap. This results in a grouping effect, because network components, which are geographically close, have also close numbers. Hence, the errors of flow in both pipes, which are connecting a valve, appear together in the following figures.

Firstly, the results of the simplification with all pipes will be compared. Then, the absolute errors produced by the relative writeback criterion will be viewed, followed by the ones of the absolute writeback criterion. Finally, the relative errors of all performed simplifications will be compared.

IV. 1. 4. a. Simplification with all Pipes

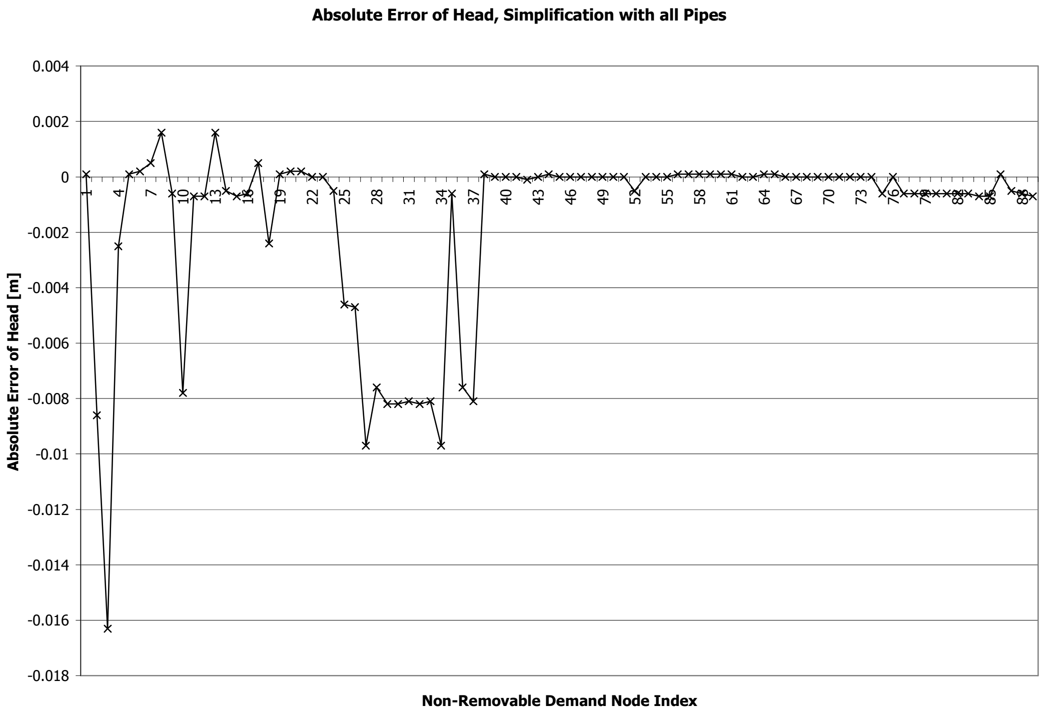


Figure 29. Absolute Error of Head, Simplification with all Pipes.

The maximum absolute error of the head is 0.0163m and the standard deviation is 0.003m. Hence, these results are very acceptable.

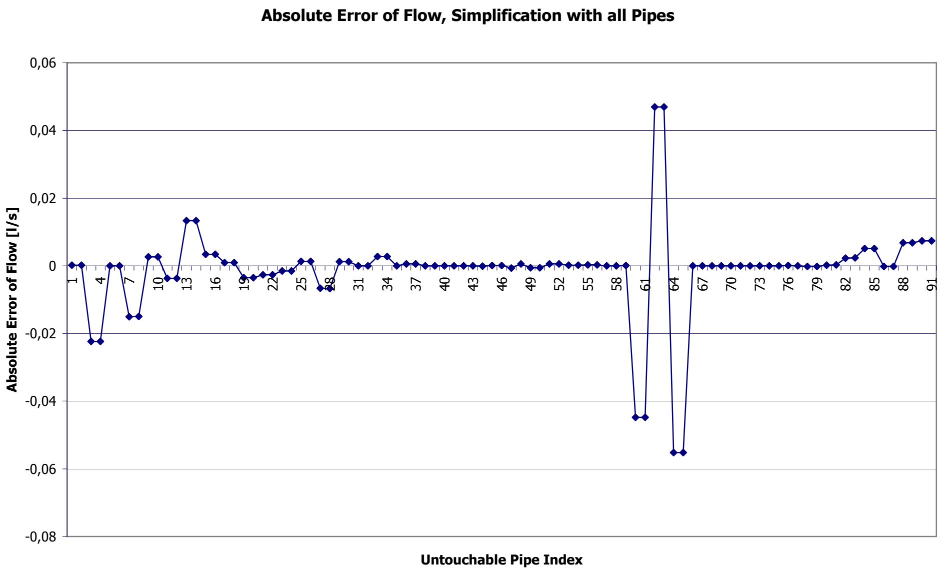


Figure 30. Absolute Error of Flow, Simplification with all Pipes.

The maximum flow error is 0.055 l/s, the standard deviation is 0.013 l/s. The values above will be further analysed relative to the original flow in the untouchable pipes to identify high relative errors.

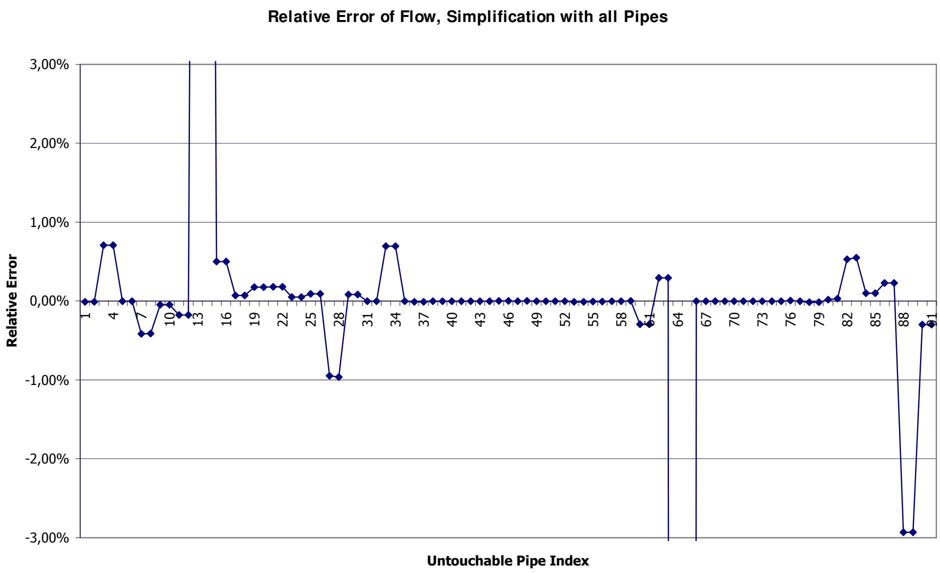


Figure 31. Relative Error of Flow, Simplification with all Pipes.

The relative errors of flow are mainly beyond ±1%, there is one peak with 23.17% in the left third of the figure above, one in the right third with –1346.34%. These extreme peaks appear in the untouchable pipes with the lowest flow of all:

Table 7. Pipes with the highest relative error.

Label (UID)	Original Flow l/s	Simplified Flow l/s	Abs. Error Flow l/s	Relative Error
1725	0.0574	0.0441	0.0133	23.17%
1727	0.0575	0.0442	0.0133	23.13%
2181	0.0041	0.0593	-0.0552	-1346.34%
2183	0.0041	0.0593	-0.0552	-1346.34%

It is very likely, that solver causes these errors. Except for these errors, the accuracy of the flows is very acceptable also.

IV. 1. 4. b. Relative Writeback Criterion

The following figures illustrate the absolute error of the simplified models with the relative writeback criterion. Firstly, the absolute error of the heads will be viewed.

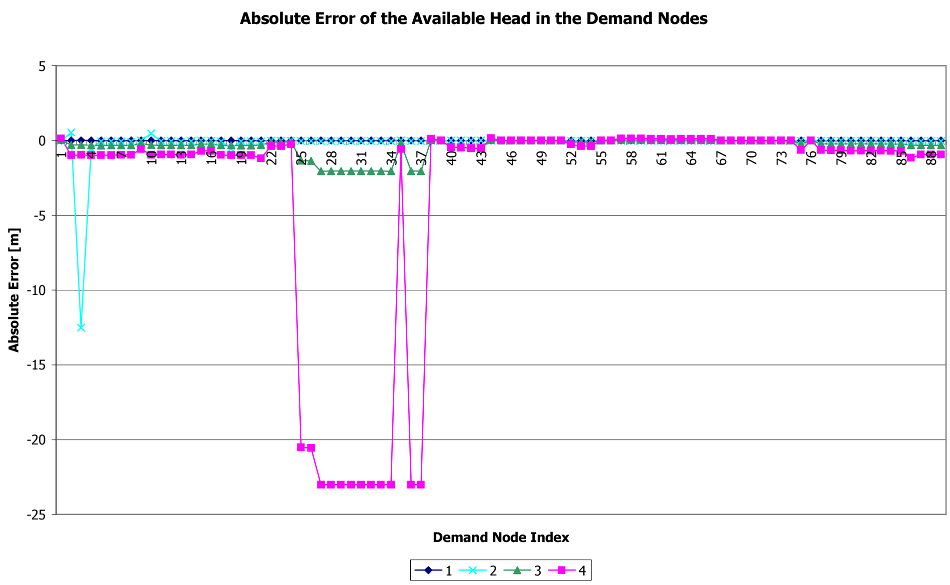


Figure 32. Absolute Error of the Available Head in the Demand Nodes for the relative cases.

As expected, the models with a higher relative writeback criterion show a higher absolute error. The error appears unpredictable, as the peak of case 2 in the left hand side and the sequence of case 4 in the left middle of the figure illustrate. These peaks are extraordinary large, compared to the all other absolute errors. Their size is about ten times bigger.

Now, the absolute errors of the flow will be regarded.

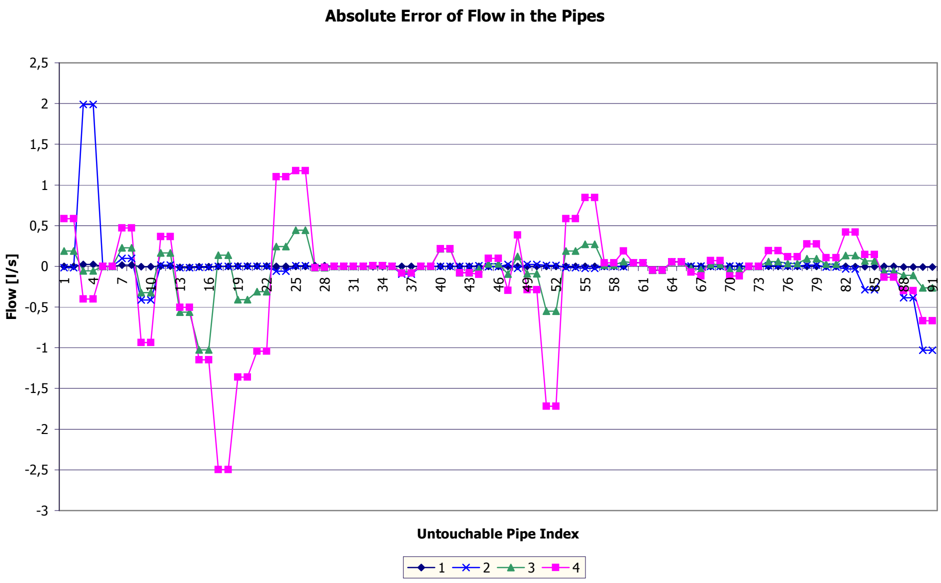


Figure 33. Absolute Error of Flow in the Pipes.

The absolute errors of flows for high relative writeback levels are partially of the same size as the flows in the pipes themselves. Again, the errors seem to appear randomly, as it can be seen when case 2 is compared to case 4.

IV. 1. 4. c. Absolute Writeback Criterion

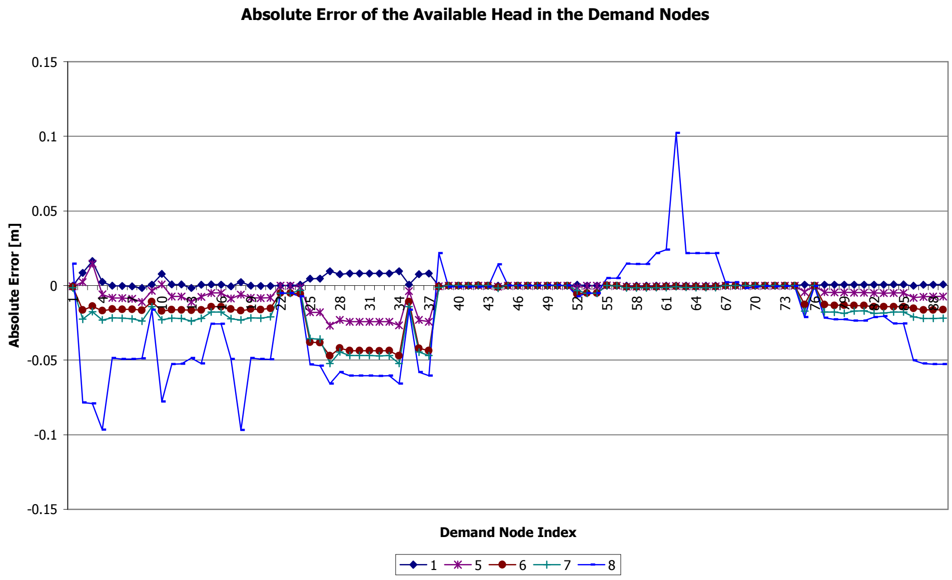


Figure 34. Absolute Error of the Available Head in the Demand Nodes.

With the absolute writeback criterion, the absolute errors of the heads are of magnitudes smaller than with the relative writeback criterion. Except for case 8, the errors are generally all within ± 5 cm. The latter writeback criterion shows errors, which are mainly below ± 10 cm. The largest ones are around ± 0.1 m, a very good value. The errors seem to increase slightly with the level of the absolute writeback criterion.

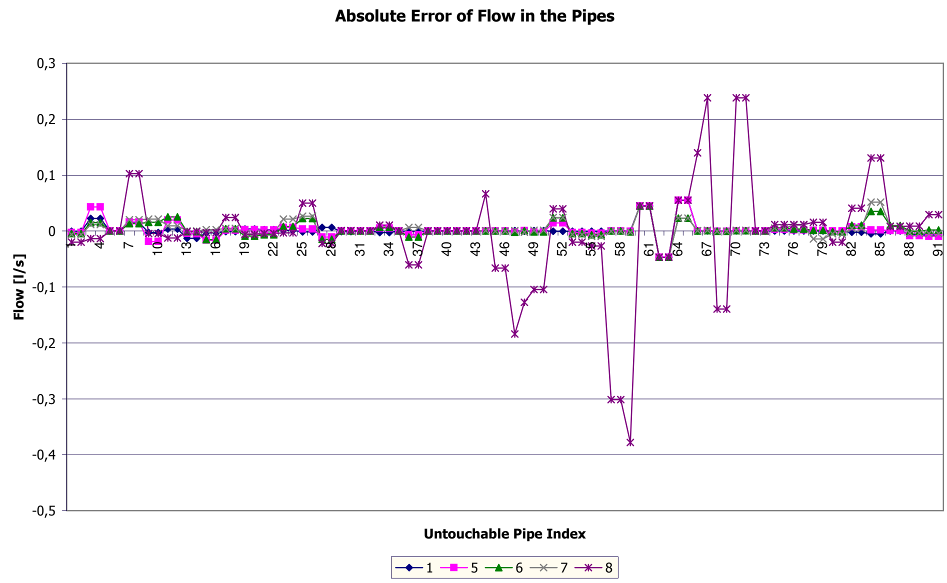


Figure 35. Absolute Error of Flow in the Pipes.

The absolute error of flow lies mainly beyond ± 50 ml/s, except for the errors of case 8, where the absolute error is mostly beyond ± 250 ml/s . As the absolute errors of the available heads, the absolute errors of flow increase with the level of the writeback criterion.

IV. 1. 4. d. Comparison of Maximum and Standard Deviation of the Absolute Value of the Absolute Errors

The following table shows the maximal values of the absolute values of the absolute errors for the simplifications.

Table 8. Maximal Errors and Std. Deviation for the Heads in the Simplifications.

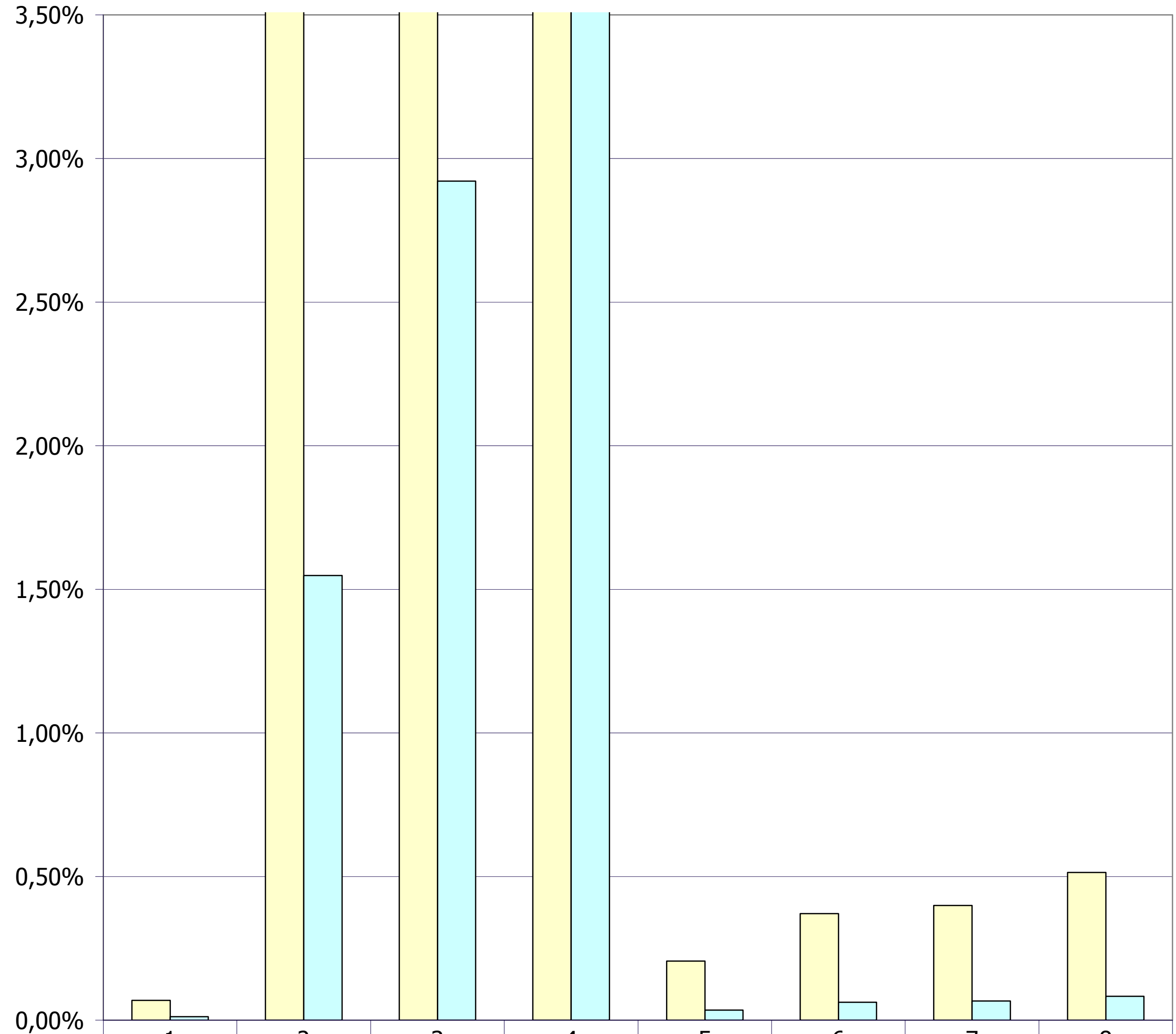
	1	2	3	4	5	6	7	8
Demand Nodes (|Absolute Error| of Available Head, m)								
Max	0.0163	12.5	2.04	23.0	0.0268	0.0471	0.0522	0.102
Std Dev	0.003	1.33	0.628	7.620	0.00781	0.0140	0.0154	0.0270
Pipes (|Absolute Error| of Flow, l/s)								
Max	0.0552	1.988	1.025	2.498	0.0552	0.0469	0.0514	0.378
Std Dev	0.0125	0.331	0.195	0.527	0.0135	0.0116	0.0127	0.0754

The maximal errors are generally much higher for the relative writeback criterion then for the absolute one. The maximal absolute error of flow is lower than the absolute error of flow (for case 5, 6 and 7) in the simplified network with all pipes. This may result from the absence of the pipes with the lowest conductances in the simplified network with the mentioned writeback criteria.

The same phenomenon is true for the standard deviation of the absolute error in the pipes of case 6.

IV. 1. 4. e. Comparison of Relative Errors

Firstly, the relative errors of the heads will be compared.



	1	2	3	4	5	6	7	8
Max	0,07%	14,61%	17,39%	195,78%	0,21%	0,37%	0,40%	0,51%
Std Dev	0,01%	1,55%	2,92%	34,01%	0,03%	0,06%	0,07%	0,08%

Simplification Case

Figure 36. Comparison of the Relative Error of the Head in the Nodes: Maximal Value, Standard Deviation.

For the relative writeback criterion, the maximal relative errors are all higher than 10%, their standard deviations are generally higher than 1%. For the absolute writeback criterion, the maximal absolute errors are except for the one of case 8 with 0.51%, lower than 0.5%, which is very good. The standard deviation for the absolute errors of the absolute writeback criterion is for all simplifications below 0.1%, an excellent result.

Now, the relative error of the flow will be compared.

Table 9. Maximal Relative Error and Maximal Relative Std. Deviation for the Flows of the simplified Network Models.

case	1	2	3	4	5	6	7	8
Max	1346.34%	1346.34%	1346.34%	1346.34%	1346.34%	565.85%	563.41%	1346.34%
Std Dev	198.41%	199.17%	242.07%	233.77%	198.40%	83.16%	82.98%	198.30%

The high maximum values and the high standard deviations result from the four pipes (see Table 7), which have a very low conductance and therefore a very low flow:

Table 10. Pipes with very low Flow in the Simplified Network Models.

case	Org. NW	1	2	3	4	5	6	7	8
Label					Flow l/s				
1725	0.0574	0.0441	0.042	-0.5066	-0.4478	0.0527	0.0566	0.0573	0.0564
1727	0.0575	0.0442	0.0421	-0.5066	-0.4478	0.0526	0.0566	0.0574	0.0564
2181	0.0041	0.0593	0.0593	0.0593	0.0593	0.0593	0.0272	0.0272	0.0593
2183	0.0041	0.0593	0.0592	0.0593	0.0592	0.0593	0.0273	0.0272	0.0593

This cause could derive from a solving error. In further analyses, these four pipes will not be included. The following chart was generated without the mentioned pipes:

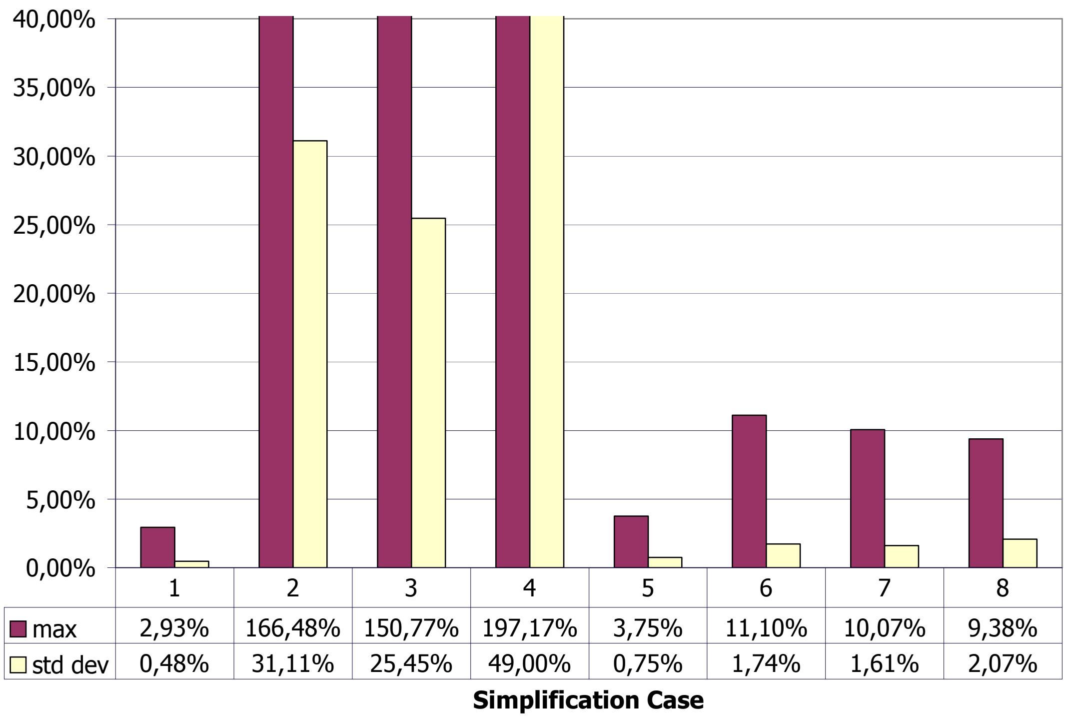


Figure 37. Comparison of Relative Error of the Flow in the Pipes: Maximal Value, Standard Deviation.

It can be seen that the maximal relative errors of flow are more than ten times as worse with the relative writeback criterion as with the absolute one. The maximum relative errors with the absolute writeback criterion are around 10%, the standard deviation is about 3 to 4 times as high as in the simplified model with all pipes.

IV. 1. 4. f. Conclusions

As the errors were much worse using the relative writeback criterion, rather than using the absolute one, the absolute writeback criterion shall be given preference for this water network model. With the absolute writeback criterion, there is a minimal growth of error in the head, the error of flow increases more.

IV. 1. 5. 24-Hour Simulation

IV. 1. 5. a. Absolute Errors

The following table gives a brief overview over the maximal errors and the standard deviation of the Skipton water network model in a 24h simulation.

Table 11. Absolute maximal Errors and Standard Deviation of the Skipton Water Network Model in a 24h Simulation.

	maximal Errors			Standard Deviation		
Case	1	5	8	1	5	8
Head in the Nodes m						
N 723	0.733	0.733	0.759	0.175	0.175	0.175
N 924	1.236	1.236	1.220	0.290	0.290	0.283
N 40	0.041	0.037	0.071	0.010	0.009	0.019
Flow in the Pipes l/s						
P 2146	0.016	0.016	0.386	0.004	0.004	0.037
P 1862	0.000	0.000	0.000	0.000	0.000	0.000

All the maximal absolute errors in terms of heads for the first node, node 723 are all below 0.8m. Node 924 has maximal absolute errors below 1.5m. Node 40 has the smallest ones: the absolute errors are smaller than 10cm. Case 5 has the same maximal absolute errors than case 1. Case 8 is against case 1 still acceptable, the errors did not get much bigger with the highest absolute writeback level.

The pipe 1862 shows no errors at all. This pipe supplies the whole area on the right hand side in Figure 1 on its own. Therefore, all demand of its supply area is summed up at its end.

Now, the absolute error of the heads 723, 924 and 40 and the absolute error of the flow in pipe 2146 will be investigated.

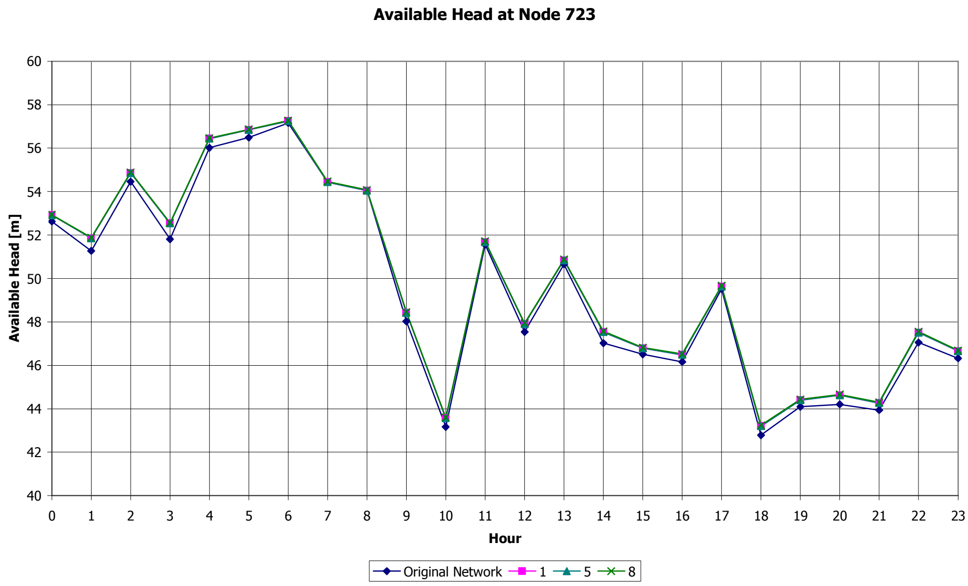


Figure 38. Available Head at Node 723 during 24 hours.

The available head is the same at 08:00 for all cases. It differs the more, the available head is away from the available head at the working point. The curves for the cases 1 and 5 are identical.

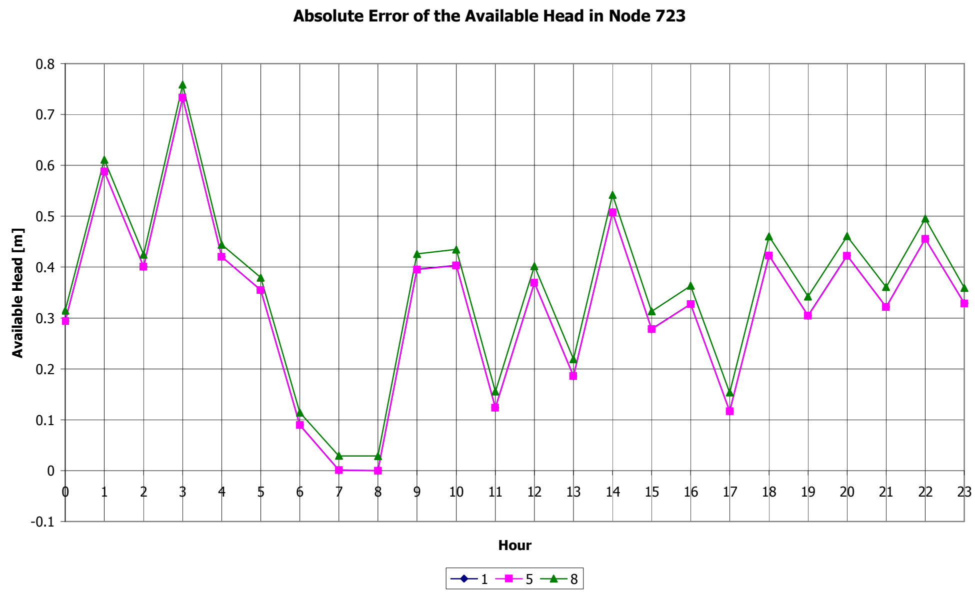


Figure 39. Absolute Error of Available Head in Node 723 during 24 Hours.

The absolute error of available head vanishes for all cases at 07:00 and 08:00. As seen above, the errors of the cases 1 and 5 have exactly the same pattern. The error of case 8 is only a little bit higher than the ones of case 1 and 5.

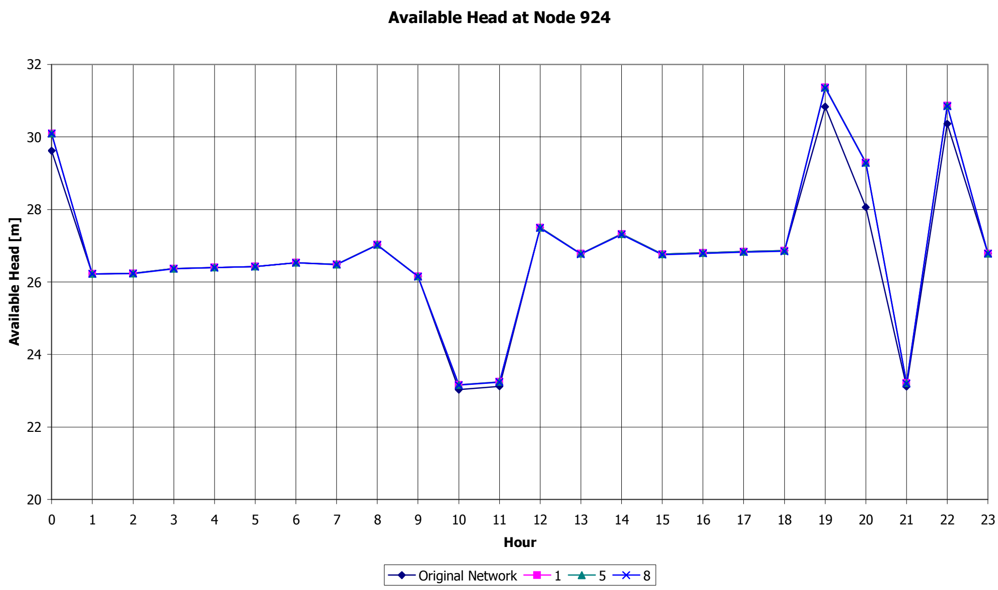


Figure 40. Available Head at Node 924 during 24 hours.

The available head at node 924 follows from 00:00 to 18:00 same curve, but at 19:00 and 20:00, the available heads of the simplified network are all high above the available head of the original network model.

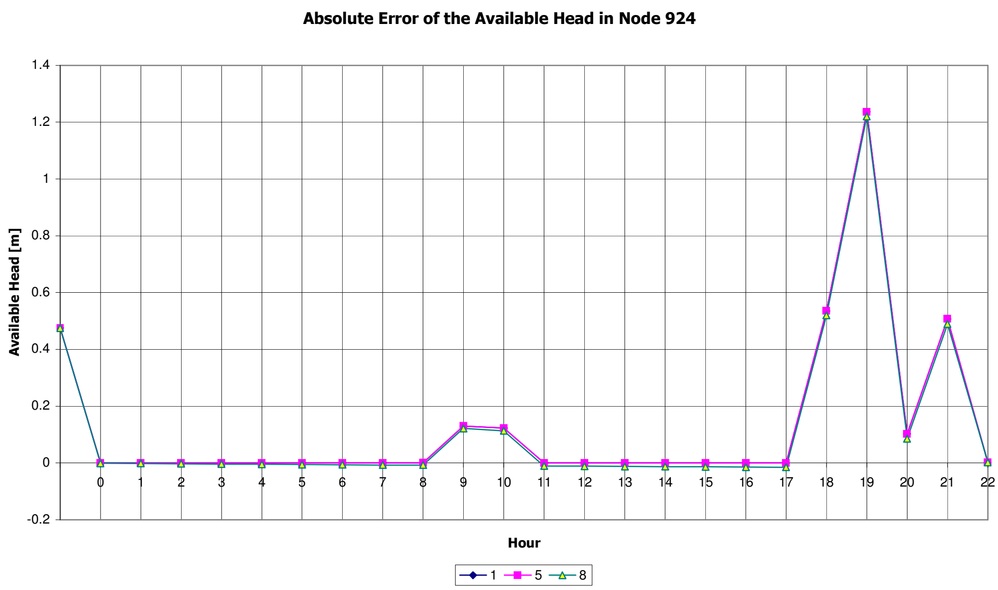


Figure 41. Absolute Error of Available Head in Node 924 during 24 Hours.

The absolute error of available head follows the same pattern for all three cases. There is a strong peak at 19:00, the absolute error peak is less than 5% of the demand at 19:00 at node 924. This peak may result from different demands in the evening in the northwest of the water network compared with its rest.

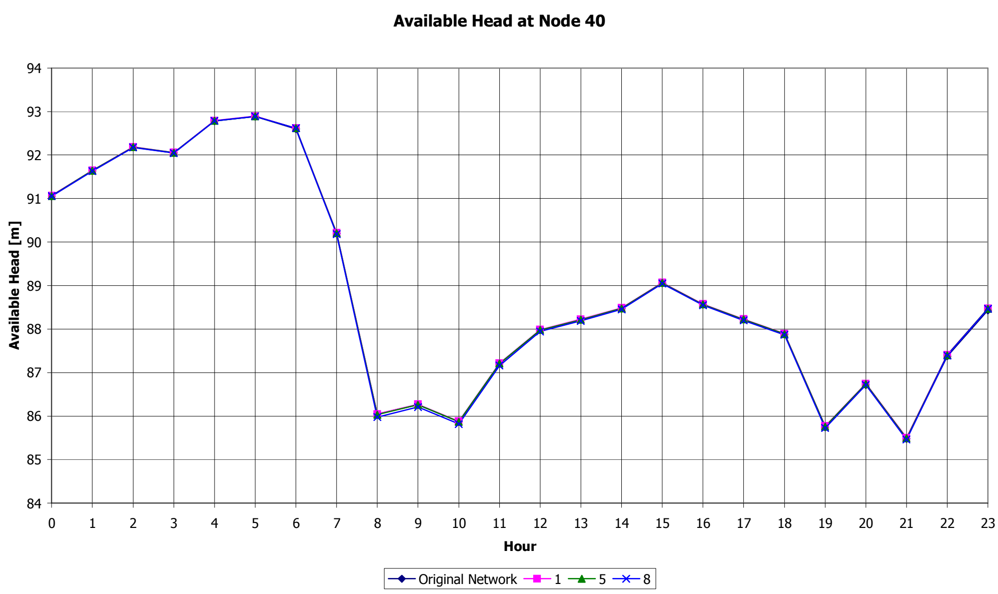


Figure 42. Available Head at Node 40 during 24 hours.

The available heads at node 40 are identical with the available heads for all three cases.

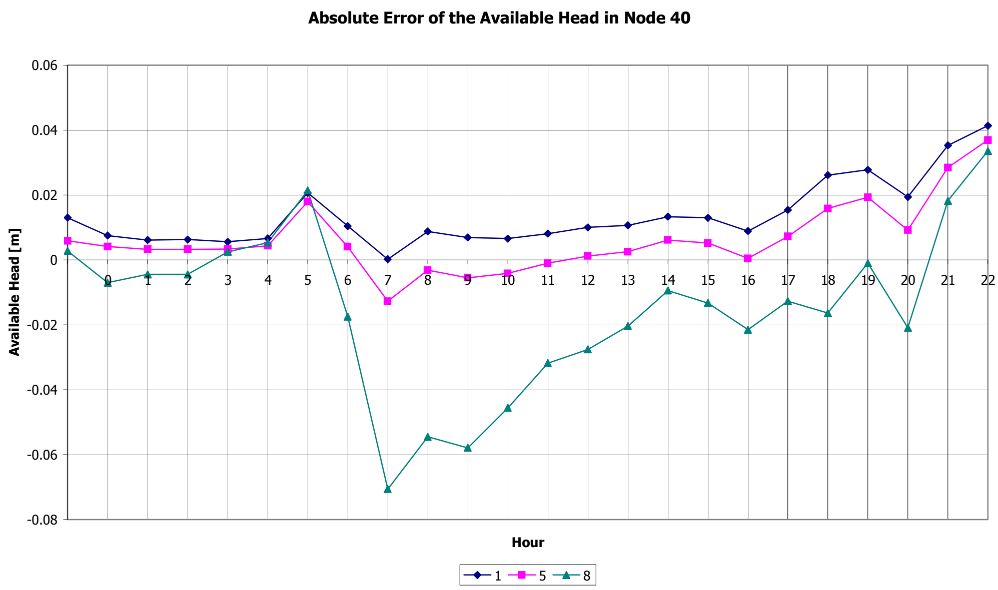


Figure 43. Absolute Error of Available Head in Node 40 during 24 Hours.

As seen in Table 11 above, the absolute error of ±7cm is the lowest one in the three observed nodes. The absolute error of case 8 is far bigger than the ones of the other two cases.

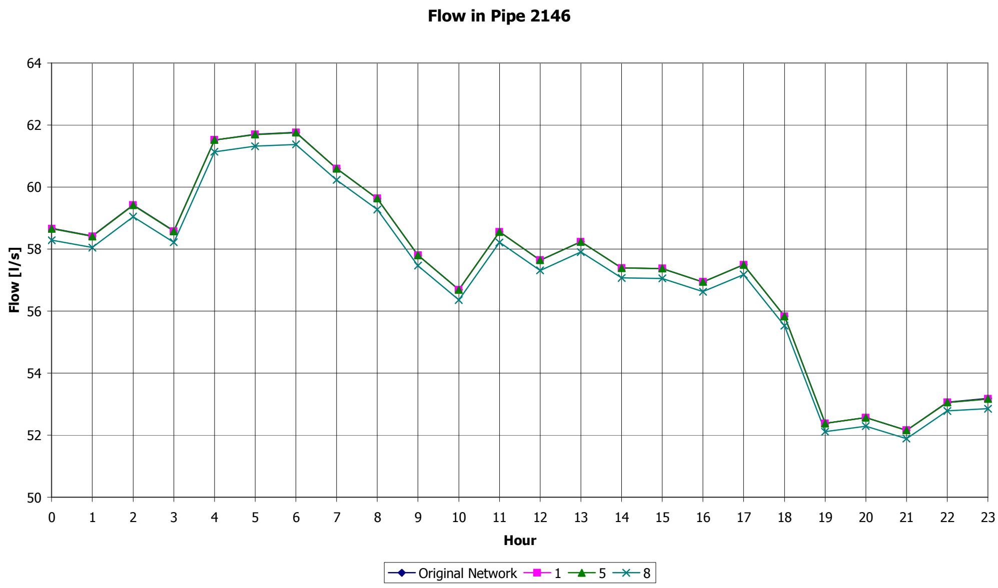


Figure 44. Flow in Pipe 2146 during 24 hours.

The flow in pipe 2146 is the same as the one of the original network model for the cases 1 and 5, the flow of case 8 lies always a little bit beyond the other flows.

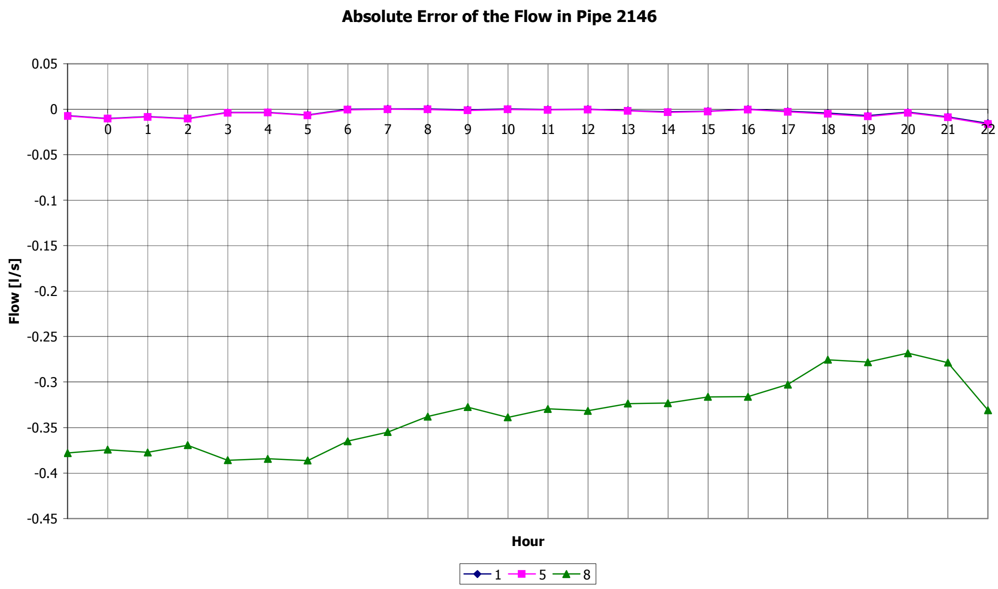


Figure 45. Absolute Error of Flow in Pipe 2146 during 24 Hours.

The absolute error in pipe 2146 is almost zero for the cases 1 and 5, but the absolute error of case 8 shows an offset between 250 and 300 ml/s.

IV. 1. 5. b. Relative Errors

In this subsection, the relative errors in 24h simulations of the simplification cases 1, 5 and 8 will be viewed.

Table 12. Maximal Relative Errors for the Simplification Cases 1, 5 and 8.

	Maximal Relative Errors			Standard Deviation		
Case	1	5	8	1	5	8
Nodes						
N 723	1.42%	1.41%	1.46%	0.36%	0.36%	0.36%
N 924	4.41%	4.41%	4.35%	1.01%	1.01%	0.99%
N 40	0.05%	0.04%	0.08%	0.01%	0.01%	0.02%
Pipes						
P 2146	0.03%	0.03%	0.64%	0.01%	0.01%	0.04%
P 1862	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Although all relative errors lie beyond 5%, only the maximal relative errors for node 924 are bigger than 1.5%. The standard deviation is mainly between 0 and 1% for the investigated network components. The relative error in pipe 1862 is zero, as seen above.

IV. 1. 5. c. Conclusions

IV. 1. 6. Benchmarking

To determine the solving time reduction when simplifying, the following models were run 500, 1000 and 2000 times at the 8 o'clock snapshot:

- The full model.
- The simplification cases 1, 5 and 8.

Due to their comparatively high errors, a simplification case with a relative writeback criterion was not analysed here.

The benchmarking was done by determining the difference between start and end time for the numbers of runs mentioned above. The solving process includes writing the results to the hard disk. So the actual solving time alone would be slightly smaller than determined here.

Table 13. Overview over the total solving time.

runs	Full 1/100 s	Case 1 1/100 s	Case 5 1/100 s	Case 8 1/100 s
500	12545	3553	2895	2801

Maschler, T. and D.A. Savic, (1999) Simplification of Water Supply Network Models through Linearisation, Centre for Water Systems, Report No.99/01, School of Engineering, University of Exeter, Exeter, United Kingdom, p.119.

1000	23981	6882	5773	5306
2000	47763	14159	11546	11171

Table 14. The solving time per run:

runs	Full 1/100 s	Case 1 1/100 s	Case 5 1/100 s	Case 81/100 s
500	25.09	7.106	5.79	5.602
1000	23.981	6.882	5.773	5.306
2000	23.8815	7.0795	5.773	5.5855

The relative solving time is illustrated in Figure 46 below.

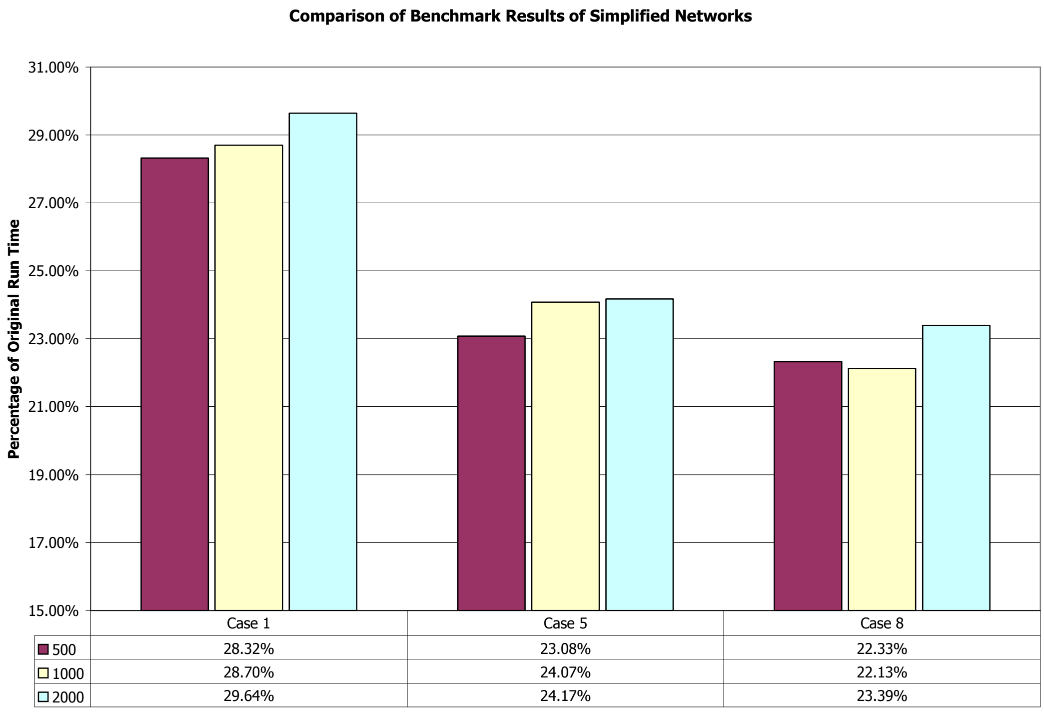


Figure 46. Comparison of Benchmark Results of Simplified Networks.

The total solving time seems to increase slightly with the number of runs. This may be due to the growing output file on the hard disk. Over all, it can be said that the solving time decreases to approximately one third of the original one, when using the simplified model with all pipes. With the writeback criterion, the solving times decreases even further, to around one fourth and slightly lower, depending on the set limit.

IV. 1. 6. a. Conclusions

A writeback criterion is a good choice to increase solving speed by further ~5%. The solving speed does not increase significantly for a higher level of the absolute writeback criterion.

IV. 1. 7. Summary

In terms of errors in the available head, a moderate absolute writeback criterion increases the relative error only very slightly, in terms of errors in the flows a little bit more. As well, the solving time decreases further with an absolute writeback criterion. The relative writeback criterion was not able to remove low conductance pipes from the simplified water network models, which caused solver errors.

IV. 2. Hawksworth Lane Water Network Model in Guiseley

IV. 2. 1. Brief Description

Guiseley is a small town in Yorkshire, 14km north-west of Leeds. The Hawksworth Lane water supply network is situated in the part of the city west of the railway.

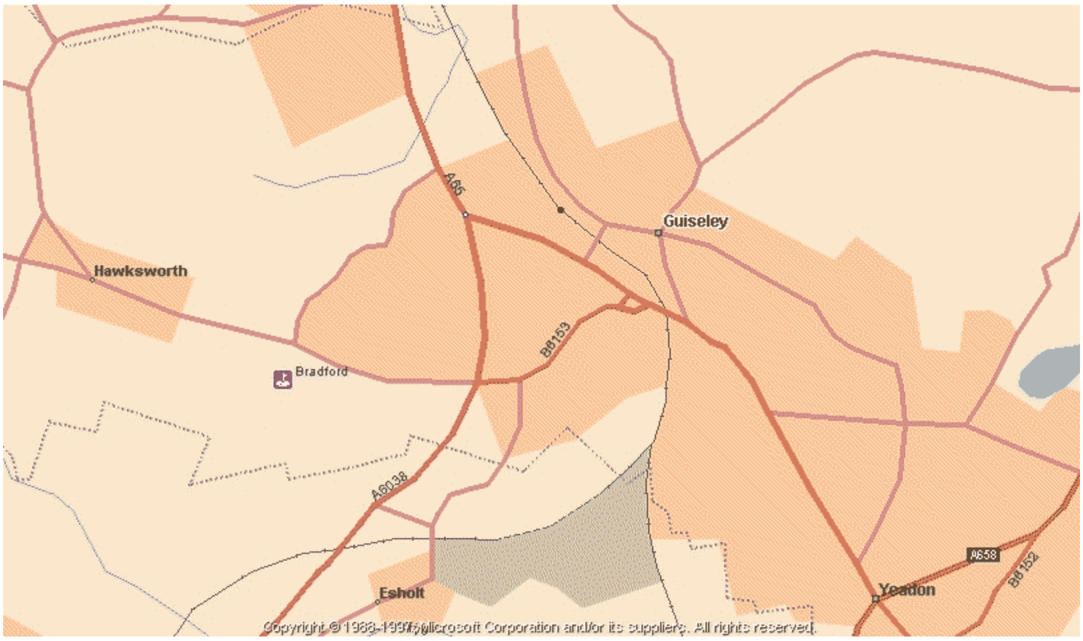


Figure 47. Guiseley.

The Hawksworth Lane water network model has approximately 500 pipes and 450 demand nodes. It contains the following special network components:

- 1 "Source" (StruMap Property "V_Source").
- 2 "Pressure Reducing Valves" (StruMap Property "V_PRV").
- 11 "Closed Valves" (StruMap Property "V_SLUICE_CLOSED").

For a cost function, the heads in some nodes and the flow in some pipes will be chosen. This will be done randomly, as in section IV. 1.

IV. 2. 2. Preparations

The nodes 1722, 1558 and 1675 were marked as non-removable and the pipes 1443 and 1236 were marked as untouchable. The number corresponds to the attribute "UID" in StruMap. The mentioned special network components from above were marked as untouchable, as well.

The simplification preparation routine returned the following output:

```
SIMPLIFICATION PREPARATION
```

```
11 closed valves have been deleted.

       STATISTICS BEFORE
================================================

Nodes:              461    non-removable:      3
Pipes:              519
Loops:              59
N. of NW Components: 980
The pipes represent 0 %         of all possible links.
The loops represent 0 o/oo      of all possible loops.

       STATISTICS AFTER
================================================

Nodes:              450    non-removable:      15
Pipes:              497
Loops:              48
N. of NW Components: 947
The pipes represent 0 %         of all possible links.
The loops represent 0 o/oo      of all possible loops.
```

The simplification preparation routine removed the 11 closed valves, because their connections cannot be represented by the static simplification. This was discussed in chapter III. 3. 2. a. above. The number of non-removable nodes increased from the 3 nodes for the cost function to 15. 4 of the new non-removable nodes are start and end nodes for the 2 untouchable pipes, 3 represent the special network components source and pressure reducing valves. The remaining 5 non-removable nodes protect the special network components from being removed. A screenshot of the Hawksworth Lane water network model can be found in Figure 48 below.

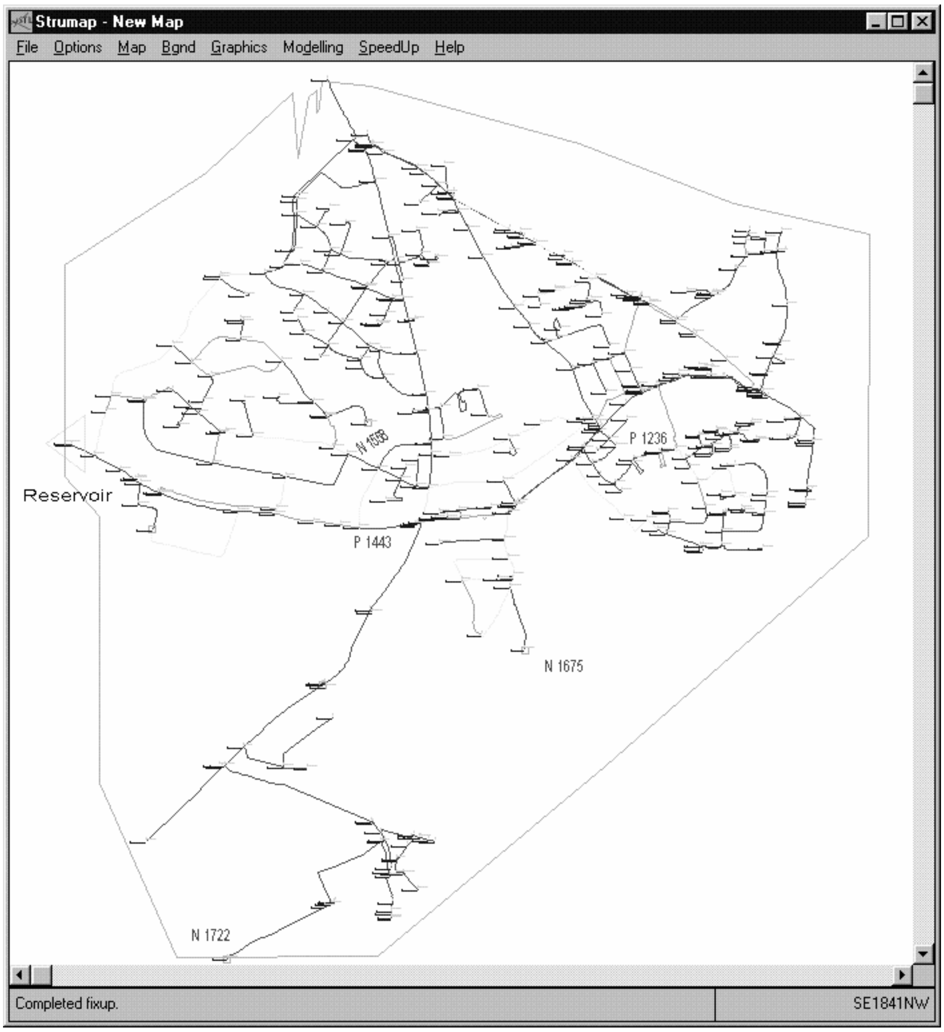


Figure 48. Screenshot of the original Hawksworth Lane water network model.

IV. 2. 3. Simplification

The simplification was done for the following cases:

1. without writeback criterion

2 times with the absolute writeback criterion:

2. level: 0.95
3. level: 10

And 2 times with the *relative writeback criterion*:

4. level: 0.0002
5. level: 0.001

A screenshot of the Hawksworth Lane water network model simplified with case 1 is shown below.

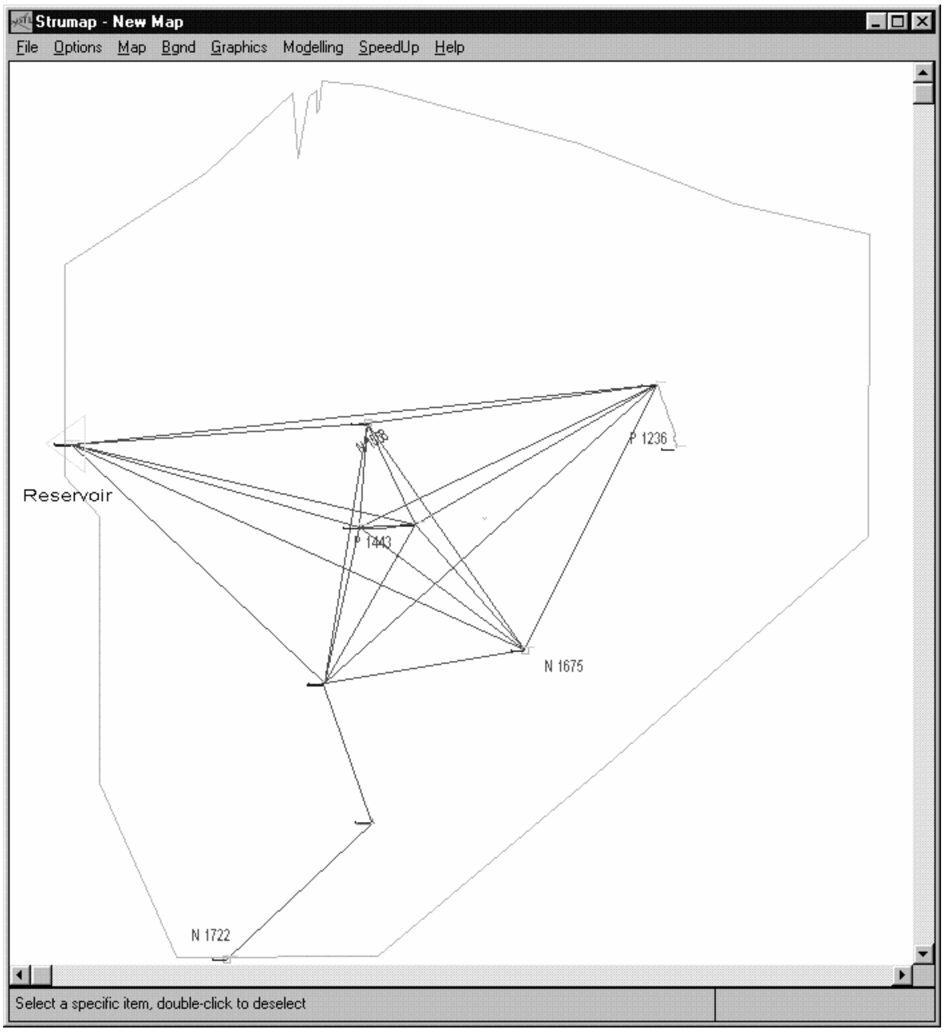


Figure 49. Screenshot of the simplified Hawksworth Lane water network model.

The output of the static simplification routine can be found in the appendix, chapter VIII. 4. 2. on page 8. The absolute writeback criterion did not delete any pipe in the Hawksworth Lane network model in the cases 2 and 3. This is due to the high reduction of network components. The relative writeback criterion removed in the 4th case 8 pipes from the 30 remaining after simplification, in the 5th case 10 pipes, $^1/_3$rd of the remaining ones. The following table gives a brief overview over the reduction of the number of network components:

Table 15. Comparison of the number of network components.

Case	Nodes	Pipes	Loops	NW Components	Pipes/	loops/ possible loops

					possible links	
Orig. NW	450	497	48	947	0 %	0 ‰
1,2,3	15	30	16	45	35 %	225 ‰
4	15	22	8	37	25 %	112 ‰
5	15	20	6	35	23 %	84 ‰

The following chart gives an overview over the new numbers of network components relative to the ones of the original water network model:

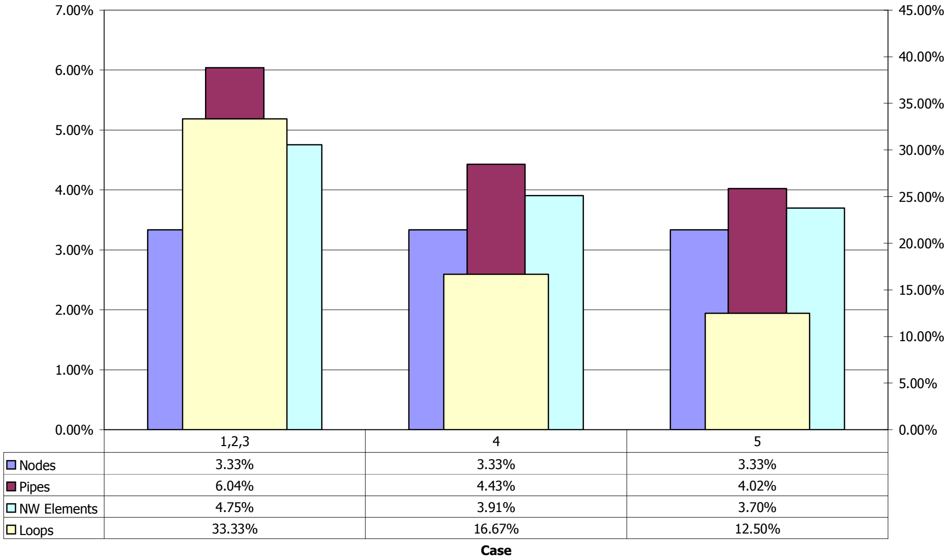


Figure 50. Numbers of Network Components after Simplification compared to the ones before for the Hawksworth Lane water network model[8].

The number of nodes gets reduced to approximately 3% of the original number of nodes for all cases. The number of pipes gets reduced to ~6% in case 1,2 and 3 and to ~4% in case 5. The number of loops gets reduced to ~33% for the cases 1,2 and 3 and by further ~17% to ~17% for case 4 and by ~21% to 12.5% in case 5.

IV. 2. 4. Errors

In this section, the errors of the cases 1, 4 and 5 at the 08:00 will be compared.

[8] The number of loops refers to the secondary value axis on the right hand side of the figure.

IV. 2. 4. a. Errors in the Non-Removable Nodes

The errors in the non-removable nodes in terms of head at the 08:00 snapshot are given below.

Table 16. Errors in the non-removable nodes.

	abs. Err.			rel. Err.		
Label	1	4	5	1	4	5
1506	0	0	0	0.00%	0.00%	0.00%
1522	-1E-04	0.0002	0.0269	0.00%	0.00%	0.05%
1523	-0.0002	0.0003	0.0291	0.00%	0.00%	0.04%
1558	0.0001	0.0001	0.0184	0.00%	0.00%	0.03%
1675	-0.0002	0.0012	0.03	0.00%	0.00%	0.04%
1678	-0.0003	0.0003	0.0291	0.00%	0.00%	0.03%
1679	0	0	0	0.00%	0.00%	0.00%
1690	0	0	0	0.00%	0.00%	0.00%
1691	0	0	0	0.00%	0.00%	0.00%
1722	0	0	0	0.00%	0.00%	0.00%
1752	-0.0004	0.0006	0.0294	0.00%	0.00%	0.03%
1812	-0.0003	0.0006	0.0294	0.00%	0.00%	0.03%

The highest absolute errors are lower than one millimetre for case one, for case 4 and 5 lower than 3cm. The relative errors are zero for case 1 and 4 and for case 5 beyond 0.5%.

IV. 2. 4. b. Errors in the Untouchable Pipes

The following table gives the flow errors at the 08:00 snapshot in the untouchable pipes.

Table 17. Errors in the Untouchable Pipes.

	abs. Err.			rel. Err.		
Label	1	4	5	1	4	5
1065	0.0006	0.0006	0.0006	0.00%	0.00%	0.00%
1236	0	0	0	0.00%	0.00%	0.00%
1443	-0.0043	-0.0007	0.1579	0.03%	0.00%	-0.98%
1955	0	0	0	0.00%	0.00%	0.00%
1956	0	0	0	0.00%	0.00%	0.00%
1958	0	0	0	0.00%	0.00%	0.00%
1959	0	0	0	0.00%	0.00%	0.00%

The relative errors are mainly zero, except of the relative error in the pipe 1443 in case 5. Its absolute value is beyond 1%, which is still very good.

IV. 2. 5. Errors in the 24-Hour-Simulations

In this section, the maximal relative errors and the standard deviation of the relative errors will be compared for 24-hour simulations of the resulting network models for the cases 1, 4 and 5.

Table 18. Maximal Relative Errors and Standard Deviation of the Relative Errors of the Hawksworth Lane Network Model.

	max. |rel. error|			Std. Dev.		
Case	1	4	5	1	4	5
Nodes						
1722	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
1558	0.00%	0.00%	0.03%	0.00%	0.00%	0.01%
1675	0.00%	0.00%	0.04%	0.00%	0.00%	0.01%
Pipes						
1443	0.05%	0.08%	1.07%	0.01%	0.01%	0.02%
1236	0.01%	0.01%	0.01%	0.00%	0.00%	0.00%

The maximal relative error is for the cases 1, 5 and 5 well beyond 0.1%, except for the pipe 1443 in case 5. The standard deviation is beyond 0.05% for all nodes and cases.

IV. 2. 6. Benchmarking

The original network and the model case studies 1, 4 and 5 were used in this section. As above in chapter IV. 1. 6. , the solving time is slightly smaller than determined here, because of the writing to the hard disk.

The following table gives an overview over the total solving time for 500, 1000 and 2000 runs.

Table 19. Solving Time for different Numbers of Runs for the Hawksworth Lane Water Network Model.

Runs	Orig. NW Model	Case 1	Case 4	Case 5
500	5009	203	187	176
1000	9826	417	379	368
2000	19614	851	752	742

From the table above, the relative solving time in table Table 20 below was calculated:

Table 20. Solving Time per Run for the Hawksworth Lane Water Network Model.

Runs	Orig. NW Model	Case 1	Case 4	Case 5

500	10.018	0.406	0.374	0.352
1000	9.826	0.417	0.379	0.368
2000	9.807	0.4255	0.376	0.371

The following figure illustrates the solving times of the simplified models compared to the one of the original model.

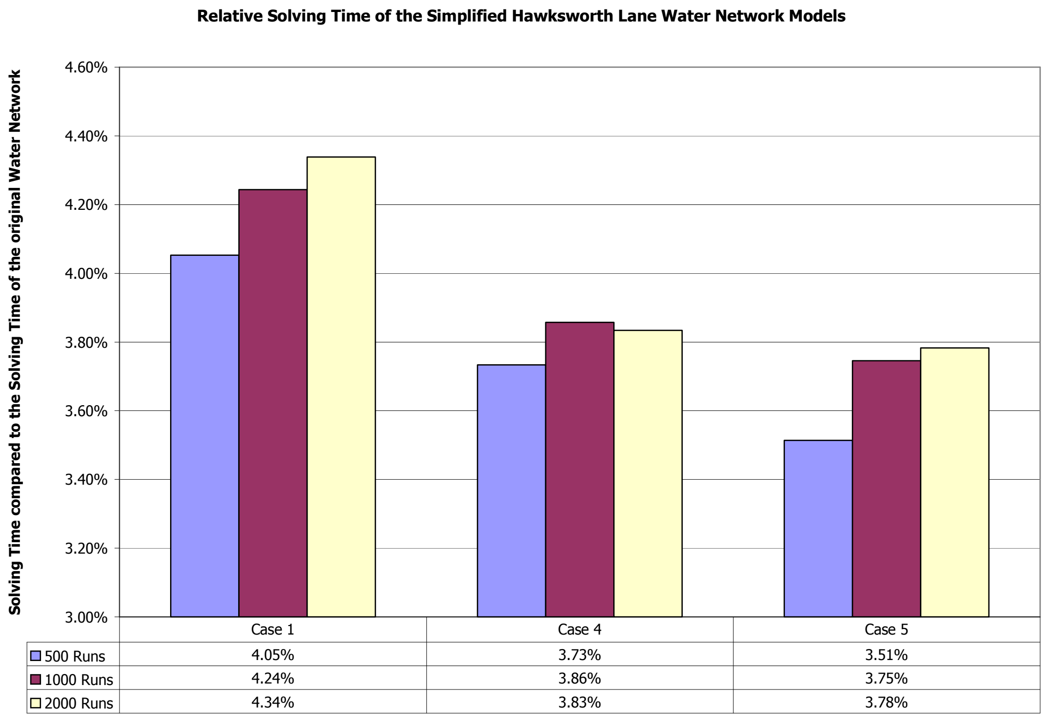


Figure 51. Relative Solving Time of the Simplified Hawksworth Lane Water Network Models.

The solving time does not decrease significantly for the cases 4 and 5 compared to case 1. It can be said, that the solving time decreases to about 4 per cent for all runs and to slightly less with the absolute writeback criterion.

IV. 2. 7. Summary

The Hawksworth Lane water network model has in the current configuration only 3 special network components. Therefore, the simplified water network models are very small. The absolute writeback criterion did not remove pipes with the given decision level. The relative writeback criterion resulted with the tested levels with about $^1/_3$rd less pipes than the simplified model with all pipes. The solving time went down beyond 5%

of the one of the original network model. It did not decrease much more with the absolute writeback criterion.

IV. 3. Conclusions

The solving time gets reduced drastically in all cases for both water network models: it sank to $^1/_3$ and less for the Skipton Water network model and to approximately 1/25 for the Hawksworth Lane water network model. The number of network components of the simplified network models — and therefore their solving time — is depends on the number of special network components and on the output vector for the cost function.

The relative writeback criterion did not remove all pipes in the simplifications of the Hawksworth Lane network model, which caused solver errors, but the absolute writeback criterion did so.

The 24h simulations were except for the worst cases all very well. To ensure that possible errors will not affect the cost function, the errors of the heads and flows for the output vector should be regarded for the worst settings of the input vector.

V — SUMMARY AND CONCLUSIONS

V. 1. Summary

This report has described and analysed methodologies to simplify water network models for simulation purposes. To do so, it was structured in three major chapters, the problem formulation, the development of a simplification algorithm and case studies.

The problem formulation introduced briefly the purpose of water network models and their network topology. It explained the equations that describe the behaviour of the main water network model components, nodes and pipes. Next, it defined the simulation time problem and discussed simplification objectives. Then, it gave a brief overview over simplification algorithms found in recent literature, component-based approaches as well as those, where the network model is substituted with an adequate, quicker model.

The second chapter, dealing with the development of a simplification algorithm, elucidates the "static simplification" algorithm of Ulanicki et al (1996) and defined requirements for it. Based on these requirements and on the interfaces for the water network scheduling optimisation process, an algorithm was developed to identify the simplification range in the water network model. As well, the treatment of pipe attributes was discussed. Following, it was shown that the static simplification algorithm carries out the network model component based simplification approaches unification of nodes, parallel pipes and pipes in series and tree elimination. In addition, the static simplification algorithm was improved with the facility to delete low conductance pipes before generating the simplified model. Then, the implementation of the algorithm was discussed. The static simplification algorithm was included as a module in the geographic information system StruMap. The coding was done in C++.

Then, the algorithm was explained in detail with activity diagrams. Finally, an example for the algorithm was given and tested with the program.

The case studies dealt with two network models, one of the water supply network in Skipton, Yorkshire, and another one with a model on the water supply network of the Hawksworth Lane Area in Guiseley, Yorkshire. Both networks were simplified with a number of different writeback criterion levels to eliminate low conductance pipes. The Skipton water network model with about 50 special network components showed extreme solving errors with relative writeback criteria, but very low errors with absolute writeback criteria. The Hawksworth Lane water network model with only 3 special network components did not lose any pipe with a moderate absolute writeback criterion, because the simplified model was very compact. The relative writeback criterion deleted up to $^{1}/_{3}{}^{rd}$ of the pipes in the simplified model with an almost neglectable error. The simplified models were benchmarked, it turned out that the simplified Skipton water network model is solved about 3-4 times faster, the simplified Hawksworth Lane water network model around 25 times faster. Both models were checked in 24h simulations, as well.

V. 2. Recommendations for Future Work

The geographical information system StruMap displays after using the simplification modules sometimes memory management error messages. To avoid the dependency on StruMap, the static simplification algorithm could be coupled to a database or be made an application on its own. Further, special network components like pumps, valves, etc. could be identified automatically. In addition, the absolute writeback criterion should be reformulated, so that it relates to the total sum of linear pipe conductances in the network model.

For various tasks of water network models, component-based simplification is very useful to maintain as much as possible of the network structure. The component-based simplifications, which are discussed in this report, could be implemented for these tasks. As s final suggestion, the static simplification algorithm could be merged with a solver. Therefore, redundant heads and flows in the output vector of the network model would be avoided when optimising schedules, for example with genetic algorithms.

VI — REFERENCES

Savic, D.A. and Walters, G.A. (1997): GENETIC ALGORITHMS FOR LEAST-COST DESIGN OF WATER DISTRIBUTION NETWORKS. Journal of Water Resources Planning and Management, March/ April 1997, p. 67ff.

Ulanicki, B., Zehnpfund, A., Martínez, F. (1996): SIMPLIFICATION OF WATER DISTRIBUTION NETWORK MODELS. Hydroinformatics 1996: Rotterdam, pp. 493-500.

Michalewicz, Z. (1992): GENETIC ALGORITHMS + DATA STRUCTURES = EVOLUTION PROGRAMS. Springer-Verlag: Berlin, Heidelberg, 1992.

CRC Press and Weisstein E.W.(1996): CRC CONCISE ENCYCLOPEDIA OF MATHEMATICS. Internet Version: http://www.treasure-troves.com/math/ (24-06-1999).

Anderson, E.J., Al-Jamal, K.H. (1995): HYDRAULIC NETWORK SIMPLIFICATION. Journal of Water Resources Planning and Management, May/ June 1995, pp. 235-240.

Swarnee, P.K., Sharma, A.K. (1990): DECOMPOSITION OF LARGE WATER DISTRIBUTION SYSTEMS. Journal of Environmental Engineering, Vol. 116, No. 2, March/April 1990, pp. 269-283.

Martínez, F., García-Serra, J. (1992): MATHEMATICAL MODELLING OF WATER DISTRIBUTION SYSTEMS IN SERVICE. Conference Paper presented at the “International Course on Water Supply Systems. State of the Art and Future Trends”. In Valencia (Spain) from 19-23 October 1992. Computational Mechanics Publications: Southampton, 1993.

Maschler, T. and D.A. Savic, (1999) Simplification of Water Supply Network Models through Linearisation, Centre for Water Systems, Report No.99/01, School of Engineering, University of Exeter, Exeter, United Kingdom, p.119.

Hammerlin, G. and Hoffman, K.H. (1991): NUMERICAL MATHEMATICS. Graduate Tests in Mathematics. Springer-Verlag: New York, Berlin, Heidelberg, London, 1991.

Structural Technologies Ltd (1996): STRUMAP: GEOGRAPHIC INFORMATION SYSTEM. Program documentation. Software version n9.

Calvert, C. (1997): BORLAND® C++BUILDER UNLEASHED. SAMS Publishing: Indianapolis, 1997.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. (1994): NUMERICAL RECIPES IN C. THE ART OF SCIENTIFIC COMPUTING. Second Edition. Cambridge University Press: 1988, 1992.

Water Software Systems (1999): FUNDAMENTAL WATER NETWORK MODEL – ONTOLOGY. On the Internet: http://www.eng.dmu.ac.uk/~wss/ontology/ontology.doc (12-07-99).

Swiercz, M. (1995): A NEURAL NETWORK APPROACH TO SIMPLIFY MATHEMATICAL MODELS OF URBAN WATER DISTRIBUTION SYSTEMS. 7th IFAC/ IFORS/ IMACS Symposium on Large Scale Systems: Theory and Applications – LSS '95, London, 11-13 July 1995.

Ellis, D.J., Simpson A.R. (1996): CONVERGENCE OF ITERATIVE SOLVERS FOR A SIMULATION OF A WATER DISTRIBUTION PIPE NETWORK. University of Adelaide, Department of Civil and Environmental Engineering, Research Report No. R138, Aug. 1996.

Rossman L.A. (1994): EPANET USERS MANUAL. VERSION 1.1. Risk Reduction Engineering Laboratory, Office of Research and Development, U.S. Environmental Protection Agency Cincinnati (Jan 1994). On the Internet: http://www.epa.gov/ORD/NRMRL/epanet/ (13-07-99).

Todini, E. and Pilati, S. (1987): A GRADIENT METHOD FOR THE ANALYSIS OF PIPE NETWORKS. International Conference on Computer Applications for Water Supply and Distribution, Leicester Polytechnic, UK, September 8-10, 1987.

George, A. and Liu, J. W-H. (1981): COMPUTER SOLUTION OF LARGE SPARSE POSITIVE DEFINITE SYSTEMS. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1981.

Maschler, T. and D.A. Savic, (1999) Simplification of Water Supply Network Models through Linearisation, Centre for Water Systems, Report No.99/01, School of Engineering, University of Exeter, Exeter, United Kingdom, p.119.

De Schaetzen W., Savic D.A., Walters G.A. (1998): A GENETIC ALGORITHM APPROACH FOR PUMP SCHEDULING IN WATER SUPPLY SYSTEMS. 3rd Hydroinformatics conference of the International Association for Hydraulic Research. Copenhagen, Denmark. August 1998.

Chadwick A. and Morfett J. (1993): HYDAULICS IN CIVIL AND ENVIRONMENTAL ENGINEERING. Second Edition 1993. London (UK): Chapman and Hall.

VII — ACKNOWLEDGEMENTS

This project was carried out at the University of Exeter, United Kingdom, as part of the ERASMUS/ SOCRATES exchange programme with the University of Stuttgart, Germany, during the academic year 1998/99. I would like to thank the following people, whose efforts made this project possible:

- Prof. D.H. Owens at the University of Exeter and
- Prof. M. Zeitz at the University of Stuttgart.

For the supervision of this interesting project and excellent support throughout all steps of this work, I would like to thank my supervisors:

- Dr. Dragan Savic and
- Dr. Godfrey Walters.

The team spirit of the "Centre for Water Systems" gave me an excellent working environment with motivation and support, for which I would like to thank its current and former members:

- Bernhard Bonfiglio
- Alberto Alvarez
- Roger Atkinson
- Florian Bessler
- Jim Davidson
- Mark Engelhardt
- Zoran Kapelan
- Eduard Keedwell
- Mark Morley
- Martin Parker
- Marc Randall-Smith
- Werner de Schaetzen
- Jenny Thorne
- Jakobus van Zyl

Finally I would like to thank all members of the staff of the University of Exeter and all British and International students, who made this year at the south cost of Devon an excellent, enriching experience for me.

VIII — APPENDIX

VIII. 1. Calculation of the *c* Constant in the Hazen-Williams Equation for StruMap

The constant c was calculated basing on the head difference, the pipe diameter, the Hazen-Williams-Coefficient, the pipe length and the flow of eight pipes in a water network model in StruMap.

Therefore, the following relationship derived from (6) on page 16:

(53) $$c = \frac{C_k^{e_1} \cdot d_k^{e_2}}{l_k} \cdot \frac{\Delta_k}{\left|Q_k\right|^{e_1 - 1} Q_k}; \qquad k=1,2,\ldots,N_p$$

Table 21. Calculation of the c Constant for the Hazen-Williams-Formula.

Difference in Total Head m	Diameter mm	H.W. coeff.	Length mm	Flow l/s	calculated Constant c
0.0960	76	20	112.3400	0.1388	1.215581E+10
0.1066	76	20	11.9300	0.4938	1.214828E+10
0.5302	76	20	55.4400	0.5120	1.216003E+10
0.4567	254	90	238.82	23.1289	1.215521E+10
0.1756	254	100	111.5	23.1368	1.215707E+10
0.9534	76	40	374.26	0.501	1.215556E+10
0.0171	254	90	8.79	23.3623	1.213786E+10

In the program code the average of the calculated c values above is used:

c :=1.215283E+10 .

VIII. 2. The Number of Loops in a Water Network

Let N_P be the number of pipes in a network and N_N the number of nodes. A network formed as a tree has one starting node, every other node is linked to this node or another one with exactly one pipe. So, for a tree network the following relation can be found:

(54) $N_P = N_N - 1$.

When there are loops in a network, the number of pipes is greater than the number of nodes excluding the starting node:

(55) $N_P > N_N - 1$.

The supernumerary pipes form the loops of the network. Hence, the number of loops N_L can be introduced as:

(56) $N_L = N_P - (N_N - 1) = N_P + 1 - N_N$.

VIII. 3. Determination of the Maximum Number of Pipes in a Water Network

The maximum number of Pipes is used to express the pipe density and the loop density of a water network in SpeedUp. The output of the simplification modules of SpeedUp relates the number of pipes in the network model to the possible number of pipes. In addition, it relates the number of loops to the possible number of loops in the network model. Both will be derived via the maximum number of links in a general network.

VIII. 3. 1. The Maximum Number of Links in a General Network

The maximum number of links M_L in a network is the number of pipes in a network, where every node is connected through exactly one pipe with every other node[9].

[9] A link can be expressed by one or more parallel pipes.

Every network begins with a starting node. When adding a node to the network, it has to be connected to all other nodes. Therefore, the number of links increases by the number of already existing nodes:

(57) $M_L(N_N) = M_L(N_N - 1) + (N_N - 1)$ with $M_L(1) = 1$ and $M_L(0) = 0$.

This recursive relationship is equal to the following sum:

(58) $$M_L(N_N) = \sum_{i=1}^{N_N - 1} i \quad ; N_N > 0 \; .$$

To prove this, (57) will be converted:

(59) $M_L(N_N) - M_L(N_N - 1) = (N_N - 1)$.

The same will be done with equation (58):

(60) $$M_L(N_N) - M_L(N_N - 1) = \sum_{i=1}^{N_N - 1} i - \sum_{i=1}^{N_N - 2} i = (N_N - 1) \; .$$

VIII. 3. 2. The Maximum Number of Pipes in a Water Network

Water networks possess a number of special components, which influence the maximum number of links in the network.

- It shall be assumed that **reservoirs and sources**, treated as nodes, are connected with exactly one pipe to the rest of the network.
- **Valves and other special network components** are treated as nodes as well. Characteristically they operate between exactly two nodes. Therefore, they require exactly two pipes.

Therefore, the formula for the maximum number of links, equation (58), will be upgraded:

- The maximum number of links in a general network can only be found between all normal nodes, their number is

(61) $N_{Norm} = N_N - N_R - N_{Spec}$ with

N_{Norm} number of normal nodes,

N_R: number of reservoirs

and N_{Spec}: number of valves, etc. .

- Each reservoir requires one pipe, so it increases the maximum number of links by one.
- Each special network component increases the maximum number of links by two.

This results in the following upgraded equation:

(62) $$M_L(N_N, N_R, N_{Spec}) = \sum_{i=1}^{N_N-1-N_R-N_{Spec}} i + N_R + 2 \cdot N_{Spec}$$

with $N_R > 0;\ N_{Spec} \geq 0;\ N_N > N_R + N_{Spec}$.

VIII. 4. Output of the Simplification Routine for the Skipton Water Network

VIII. 4. 1. Skipton Water Network

The static simplification algorithm generated the following output. The static simplification algorithm used the heads and the flows at the 8 o'clock snapshot. The snapshot was directly calculated, not as part of a 24h simulation.

VIII. 4. 1. a. Case 1

All new linear pipe conductances of the Jacobian matrix of the model were written back into StruMap.

```
        LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .

Reading pipes back.
0 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.
```

```
       STATISTICS BEFORE
================================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
================================================

Nodes:              134    non-removable:      134
Pipes:              409
Loops:              276
N. of NW Components:       543
The pipes represent 9 %          of all possible links.
The loops represent 68 o/oo             of all possible loops.

       COMPARISON
================================================

              New           Old           Compared

Nodes:        134           1091          -87 %
Pipes:        409           1295          -68 %
Loops:        276           205           34 %
NW Components:      543           2386          -77 %
```

VIII. 4. 1. b. Case 4

The relative writeback criterion was used with a writeback level of 0.0002.

```
              LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .

Reading pipes back whose linear conductance is bigger than
19/100 000 of the summed conductances of its nodes.
142 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.

       STATISTICS BEFORE
================================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.
```

```
       STATISTICS AFTER
================================================

Nodes:              134    non-removable:      134
Pipes:              267
Loops:              134
N. of NW Components:       401
The pipes represent 6 %          of all possible links.
The loops represent 33 o/oo              of all possible loops.

       COMPARISON
================================================

             New           Old           Compared

Nodes:       134           1091          -87 %
Pipes:       267           1295          -79 %
Loops:       134           205           -34 %
NW Components:      401           2386          -83 %
```

VIII. 4. 1. c. Case 3

The relative writeback criterion was used with a writeback level of 0.0001.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .

Reading pipes back whose linear conductance is bigger than
10/100 000 of the summed conductances of its nodes.
125 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.

       STATISTICS BEFORE
================================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
================================================

Nodes:              134    non-removable:      134
Pipes:              284
Loops:              151
N. of NW Components:       418
The pipes represent 6 %          of all possible links.
The loops represent 37 o/oo              of all possible loops.
```

```
       COMPARISON
================================================

              New           Old           Compared

Nodes:        134           1091          -87 %
Pipes:        284           1295          -78 %
Loops:        151           205           -26 %
NW Components:      418           2386          -82 %
```

VIII. 4. 1. d. Case 2

The relative writeback criterion was used with a writeback level of 0.00001.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .

Reading pipes back whose linear conductance is bigger than
1/100 000 of the summed conductances of its nodes.
97 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.

       STATISTICS BEFORE
================================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
================================================

Nodes:              134    non-removable:      134
Pipes:              310
Loops:              177
N. of NW Components:       444
The pipes represent 7 %          of all possible links.
The loops represent 44 o/oo            of all possible loops.

       COMPARISON
================================================

              New           Old           Compared

Nodes:        134           1091          -87 %
Pipes:        310           1295          -76 %
Loops:        177           205           -13 %
NW Components:      444           2386          -81 %
```

VIII. 4. 1. e. Case 5

The absolute writeback criterion was used with a writeback level of 0.95.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .

Reading pipes back
whose absolute linear conductance is bigger than
0.95 times the lowest linear conductance in the network.
109 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.

       STATISTICS BEFORE
================================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %         of all possible links.
The loops represent 0 o/oo      of all possible loops.

       STATISTICS AFTER
================================================

Nodes:              134    non-removable:      134
Pipes:              300
Loops:              167
N. of NW Components:       434
The pipes represent 7 %         of all possible links.
The loops represent 41 o/oo            of all possible loops.

       COMPARISON
================================================

             New          Old          Compared

Nodes:       134          1091         -87 %
Pipes:       300          1295         -76 %
Loops:       167          205          -18 %
NW Components:      434          2386         -81 %
```

VIII. 4. 1. f. Case 6

The absolute writeback criterion was used with a writeback level of 2.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .
```

```
Reading pipes back
whose absolute linear conductance is bigger than
2 times the lowest linear conductance in the network.
118 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.

       STATISTICS BEFORE
===============================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %           of all possible links.
The loops represent 0 o/oo        of all possible loops.

       STATISTICS AFTER
===============================================

Nodes:              134    non-removable:      134
Pipes:              291
Loops:              158
N. of NW Components:       425
The pipes represent 7 %           of all possible links.
The loops represent 39 o/oo              of all possible loops.

       COMPARISON
===============================================

             New          Old          Compared

Nodes:       134          1091         -87 %
Pipes:       291          1295         -77 %
Loops:       158          205          -22 %
NW Components:      425          2386         -82 %
```

VIII. 4. 1. g. Case 7

The absolute writeback criterion was used with a writeback level of 3.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .

Reading pipes back
whose absolute linear conductance is bigger than
3 times the lowest linear conductance in the network.
124 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.

       STATISTICS BEFORE
```

```
================================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
================================================

Nodes:              134    non-removable:      134
Pipes:              285
Loops:              152
N. of NW Components:       419
The pipes represent 6 %          of all possible links.
The loops represent 37 o/oo             of all possible loops.

       COMPARISON
================================================

             New          Old          Compared

Nodes:       134          1091         -87 %
Pipes:       285          1295         -77 %
Loops:       152          205          -25 %
NW Components:      419          2386         -82 %
```

VIII. 4. 1. h. Case 8

```
The absolute writeback criterion was used with a writeback level of 10.


       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 1090x1090 .

Reading pipes back whose
whose absolute linear conductance is bigger than
10 times the lowest linear conductance in the network.
144 had a linear branch conductance lower than
the draw-back-criteria and were not written back.

Corrected hDiff.

       STATISTICS BEFORE
================================================

Nodes:              1091   non-removable:      134
Pipes:              1295
Loops:              205
N. of NW Components:       2386
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
================================================
```

```
Nodes:              134    non-removable:      134
Pipes:              265
Loops:              132
N. of NW Components:       399
The pipes represent 6 %          of all possible links.
The loops represent 32 o/oo            of all possible loops.

       COMPARISON
==================================================

              New          Old          Compared

Nodes:        134          1091         -87 %
Pipes:        265          1295         -79 %
Loops:        132          205          -35 %
NW Components:      399          2386         -83 %
```

VIII. 4. 2. Hawksworth Lane, Guiseley Water Network Model

The static simplification algorithm generated the following output. The static simplification algorithm used the heads and the flows at the 8 o'clock snapshot. The snapshot was directly calculated, not as part of a 24h simulation.

VIII. 4. 2. a. Case 1

All pipes were written back to StruMap.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 449x449 .

Reading pipes back.
0 had a linear branch conductance lower than
the draw-back-criteria and were not written back.


       STATISTICS BEFORE
==================================================

Nodes:              450    non-removable:      15
Pipes:              497
Loops:              48
N. of NW Components:       947
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
==================================================

Nodes:              15     non-removable:      15
```

```
Pipes:              30
Loops:              16
N. of NW Components:      45
The pipes represent 35 %        of all possible links.
The loops represent 225 o/oo           of all possible loops.

      COMPARISON
===============================================

             New          Old          Compared

Nodes:       15           450          -96 %
Pipes:       30           497          -93 %
Loops:       16           48           -66 %
NW Components:      45           947          -95 %
```

VIII. 4. 2. b. Case 2 and 3

The output of the cases 2 and 3 is identical with case 1; no pipes were deleted. Case 2 uses the absolute writeback criterion with a level of 0.95. Case 3 uses the absolute writeback criterion as well, but with a level of 10.

```
      LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 449x449 .

Reading pipes back
whose absolute linear conductance is bigger than
0.95 times the lowest linear conductance in the network.
0 had a linear branch conductance lower than
the draw-back-criteria and were not written back.


      STATISTICS BEFORE
===============================================

Nodes:              450    non-removable:      15
Pipes:              497
Loops:              48
N. of NW Components:      947
The pipes represent 0 %         of all possible links.
The loops represent 0 o/oo      of all possible loops.

      STATISTICS AFTER
===============================================

Nodes:              15     non-removable:      15
Pipes:              30
Loops:              16
N. of NW Components:      45
The pipes represent 35 %        of all possible links.
The loops represent 225 o/oo           of all possible loops.

      COMPARISON
```

```
===============================================

             New          Old          Compared

Nodes:       15           450          -96 %
Pipes:       30           497          -93 %
Loops:       16           48           -66 %
NW Components:      45           947          -95 %
```

VIII. 4. 2. c. Case 4

This case uses the relative writeback criterion with a level of 0.0002.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 449x449 .

Reading pipes backwhose linear conductance is bigger than
20/100 000 of the summed conductances of its nodes.
8 had a linear branch conductance lower than
the draw-back-criteria and were not written back.


       STATISTICS BEFORE
===============================================

Nodes:              450    non-removable:      15
Pipes:              497
Loops:              48
N. of NW Components:       947
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
===============================================

Nodes:              15     non-removable:      15
Pipes:              22
Loops:              8
N. of NW Components:       37
The pipes represent 25 %         of all possible links.
The loops represent 112 o/oo           of all possible loops.

       COMPARISON
===============================================

             New          Old          Compared

Nodes:       15           450          -96 %
Pipes:       22           497          -95 %
Loops:       8            48           -83 %
NW Components:      37           947          -96 %
```

VIII. 4. 2. d. Case 5

This last case study uses the relative writeback criterion with a level of 0.001.

```
       LINEARISATION & SIMPLIFICATION

Dimension of the Jacobian Matrix: 449x449 .

Reading pipes back, whose linear conductance is bigger than
100/100 000 of the summed conductances of its nodes.
10 had a linear branch conductance lower than
the draw-back-criteria and were not written back.


       STATISTICS BEFORE
===============================================

Nodes:              450    non-removable:      15
Pipes:              497
Loops:              48
N. of NW Components:       947
The pipes represent 0 %          of all possible links.
The loops represent 0 o/oo       of all possible loops.

       STATISTICS AFTER
===============================================

Nodes:              15     non-removable:      15
Pipes:              20
Loops:              6
N. of NW Components:       35
The pipes represent 23 %         of all possible links.
The loops represent 84 o/oo             of all possible loops.

       COMPARISON
===============================================

              New           Old           Compared

Nodes:        15            450           -96 %
Pipes:        20            497           -95 %
Loops:        6             48            -87 %
NW Components:      35            947           -96 %
```

VIII. 5. User Manual

This section guides through the simplification procedure with SpeedUp.

- The attribute "non-removable" has to be set "true" for all nodes, whose head is necessary for the cost function, and "false" otherwise.
- The attribute "untouchable" has to be set "true" for all pipes, whose flow is needed for the cost function and for all input components, otherwise "false".

Now, SpeedUp can be called. This can be done by calling the main menu entry "SpeedUp→Network Simplification…". Then, the following dialog box appears:

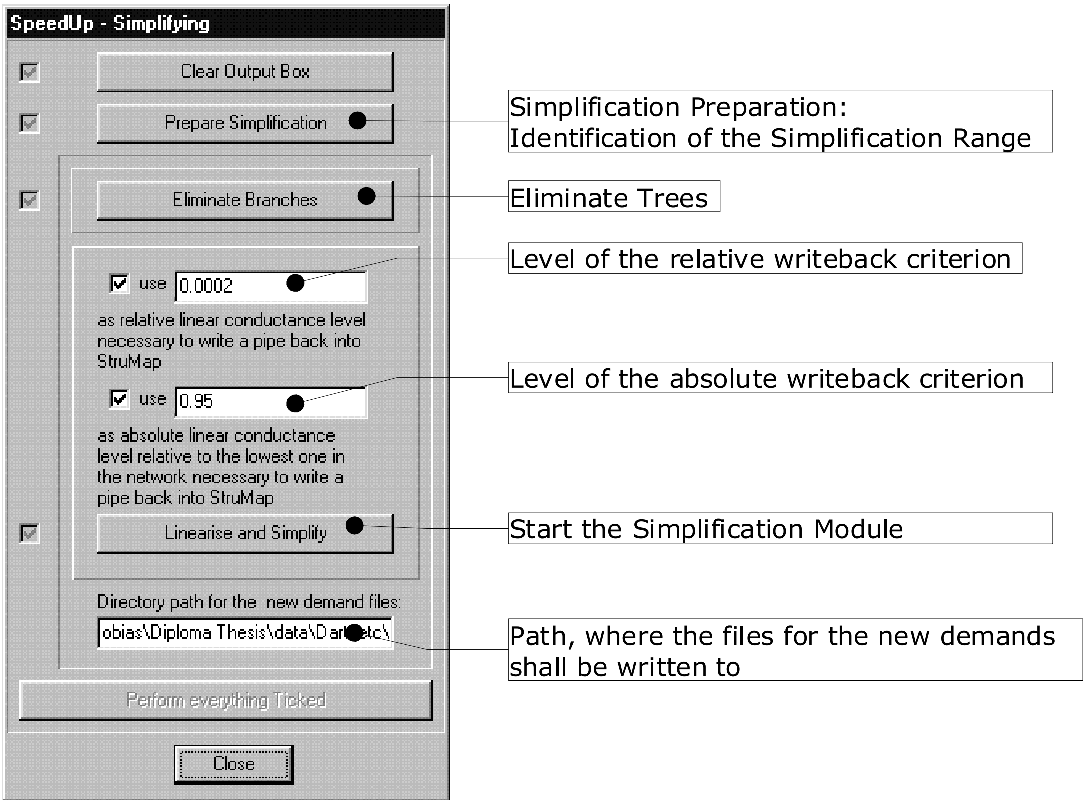


Figure 52. SpeedUp Dialog Box.

It is very important to save the network model after every step and to restart SpeedUp.

The path, where the current model is located on the hard disk, has to be inserted in the corresponding text-edit-box. It needs to finish with a backslash, "\".

Before any simplifications can be done, the simplification range needs to be identified with the button "Prepare Simplification". Afterwards, the either tree structures can be eliminated ("Eliminate Branches") or the static simplification can be applied ("Linearise and Simplify"). If a writeback criterion is necessary, it should be ticked and an appropriate level entered. Both criterions can be combined.

To simulate a network model multiple times, the HARP solver has to be called as usual. The pop-down-menu beyond "Hazen-Williams"/ "Colebrook-White" has to be set to "Genetic". A dialog box will appear and ask, how many times the model shall be run. It is very useful, to avoid saving the text output file.

[1] Pipes with high resistance/ low conductance pose big problems when solving. They require far more iterations to meet the accuracy criterion of the solver than other pipes.

[1] The term "Static Simplification" appears in Swiercz (1995) and is used here for the approach of Ulanicki et al (1996), as there is no term specified.

[1] The numbers without variable name and unit are the flows l/s.

[1] The black point in the bottom left edge in the screenshot is the cursor.

[1] The lowest linear pipe conductance of the original network is kept in the simplified one. It is seen as a lower limit to which small changes are allowed. Therefore, if the solver has no problems with the original network, it will have no problems with the simplified network, also.

[1] The percentage of loops refers to the secondary value axis, all others to the first one.

[1] The number of loops refers to the secondary value axis on the right hand side of the figure.

[1] A link can be expressed by one or more parallel pipes.