Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

Comparison of Different Methods to Solve the Steady-state Flow Problem of District Heating Networks and Gas Networks

(Vergelijking van Verschillende Methoden voor het oplossen van het Tijdsonafhankelijke Stromingsprobleem van Warmtenetwerken en Gasnetwerken)

Verslag ten behoeve van het
Delft Institute of Applied Mathematics
als onderdeel ter verkrijging

van de graad van

BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE

door

Mare Undine Mahieu
Delft, Nederland
Februari 2021

Table 1: Summary of the important symbols

Symbols	Description
m	Number of reference nodes
n	Number of load nodes
e	Number of edges
A	Branch-nodal incidence matrix (gives the flow entering and leaving a node)
A_1	Branch-nodal incidence matrix of the reference nodes
A_2	Branch-nodal incidence matrix of the load nodes
A_k	Branch-nodal incidence matrix of the branches in a loop
A_s	Branch-nodal incidence matrix of the branches not in a loop
A_t	Branch-nodal incidence matrix of the branches in the spanning tree
A_c	Branch-nodal incidence matrix of the branches in the co-tree
k	Number of loops
B	Branch-loop incidence matrix
B_t	Branch-loop incidence matrix of the branches in the spanning tree
B_c	Branch-loop incidence matrix of the branches in the co-tree
p	Pressures in the nodes
p_1	Pressure in reference nodes
p_2	Pressure in load nodes
q	Flow rate on the edges
q_k	Flow rate in the loop branches
q_s	Flow rate in the branches that are not in a loop
q_t	Flow rate in the branches of the spanning tree
q_c	Flow rate in the branches of the co-tree
Δp	Pressure drops over each edge
L	Load at the nodes
Δq	Correction to the flows
Δq_c	Correction to the chord flows

Contents

1 Introduction

In recent years, there has been a renewed interest in district heating networks. These networks supply the heat from a source, such as waste heat from a power plant, to different consumers, such as residential houses. Using district heating networks could make the supply of heat more sustainable. Therefore, district heating networks have become an inseparable part of the infrastructure in many countries (Kuosa, 2013, p.450). In the renewed interest in district heating networks there are three major challenges: energy savings on the demand side, efficiency improvements in energy production, and the replacement of fossil fuels by various sources of renewable energy (Kuosa, 2013, p.451). The advantages of district heating networks cover practical, environmental, and safety aspects. They reduce pollutant and thermal emissions in city areas. Moreover, they can increase safety conditions and eliminate the transportation of fuel in city areas, by removing combustion systems at the final users of the thermal energy systems. Furthermore, district heating networks allow the reduction of the occupied space and maintenance at the final users of the thermal energy, by the absence of boilers (Ancona et al, 2014, p.1226).

Because of the many advantages of district heating networks it is worthwhile to the determine the flow trough the pipes in the networks in the most efficient and robust way. In this thesis, the nodal method, the loop method and the loop-node method are considered in order to make computer simulations. These computer simulations make it possible to investigate the district heating networks and gas networks. The simulations allow us to predict the behaviour of the hydraulic networks under different conditions. These predictions can then be used to make decisions about the network in the real world (Osiadacz, 1988, p.25). The ability to evaluate the performance of different hydraulic networks is important for several reasons, such as proper design, understanding the effects of dynamic changes, and to ensure meeting the demand requirements within acceptable operational limits (Altman & Boulos, 1995, p.35).

In solving the flow distribution problem for district heating networks it is important to examine the most efficient way to get the heat to the consumer. To determine this, district heating networks can be described in an abstract, mathematical way. The most efficient way to do this is graph theory, which is used in this thesis. The district heating network is modeled as a graph with nodes and branches. Then, a steady-state load flow analysis is used to determine the pressures and flows throughout the network, given a constant heat demand at the loads. The loads are the locations where the heat leaves the district heating networks, for example a house. This part of the analysis of the district heating network is called the hydraulic model. After this, the thermal equations can be added, to determine the heat transfer through the district heating network. This part of the analysis of the district heating networks is called the thermal model. The design of a district heating network is also characteristic for other technical systems such as natural gas distribution systems (Stevanovic et al, 2007, p. 1537).

Thus, gas networks have the same characteristics in analysis as the hydraulic model of district heating networks. In general, the models are even derived from the same type of equations. Namely, both networks are based on Kirchhoff's first law, Kirchhoff's second law, and a resistance law. In fact, a gas low gauge pressure network analysis also hold for a network with water as a carrier, such as a district heating network. The main difference between the models is the pipe constant and the fact that for water the pressures are absolute pressures and for gas the pressures are gauge pressures. Thus, other than these fact, is is possible to derive the models using the same steps, the same analysis (Stevanovic et al, 2007, p. 1537).

The steady-state load flow analysis of the hydraulic model of the district heating network consists of a system of nonlinear equations. The hydraulic resistance depends on the flow rate, which is dependent on the friction factor, which is a complicated function of the relative roughness and the Reynolds number. This leads to a system of nonlinear equations, which means that the problem needs to be solved iteratively (Brkic, 2011, p.2952). In this thesis, the Newton-Raphson method is used to solve the nonlinear equations. There are several methods that each have different ways of solving these nonlinear equations. Overall, there are two main groups of methods: (1) methods based on the solution of loop equations, and (2) methods based on the solution of the node equations (Brkic, 2011, p.1954). In this thesis the three most used methods will be examined, namely the nodal method, the loop method, and the loop-node method. In solving the hydraulic model of district heating networks, the loop method is mostly used. However, as stated before, gas networks have the same characteristics in design as district heating networks. Interestingly, in gas networks the method mainly used in solving the steady-state flow problem is the nodal method.

We see that both district heating networks and gas networks lead to nonlinear steady-state load flow problems that need to be solved iteratively. Since the analysis of the models uses the same steps, the description of gas networks as stated by Osiadacz (1987) is used. For this reason we will call the networks hydraulic networks in this thesis. In both gas networks and district heating networks there is no clear reason for using the nodal method, the loop method, or the loop-node method. Therefore, in this thesis, these methods are investigated. The goal is to determine when to use which of these methods. In order to determine this, a comparison will be made in terms of efficiency and robustness of the methods. This comparison will be made in order to answer the question: *For which network, with specific characteristics, is the nodal method, the loop method, or the loop-node method preferred, if any is preferred?*

In this thesis, the district heating networks and the gas networks are described. Then, a mathematical description of the networks is given and a steady-state analysis is described. Following this, the different methods, nodal, loop, and loop-node, are described. Finally, a comparison between these methods is made, based on computer storage, sensitivity to starting values, convergence properties, and computational time. This will lead to a conclusion on when to use which method.

2 Description of the networks

2.1 District heating networks

A district heating network is a network in which heat from a source, such as a power plant, is distributed through pipes to consumers, such as residential houses. Thus, a district heating network consists of supply pipes, which transport hot water and distribute the hot water from one or more heat source to several consumers, and return pipes that contain cooled water from the consumers that is transported back to the heat sources (Stevanovic et al, 2007, p. 1537). An example of such a district heating network is given in figure 1.

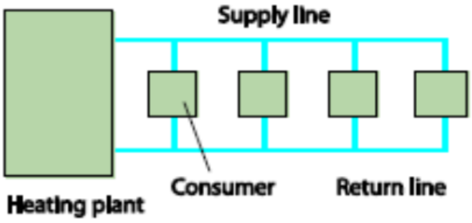


Figure 1: District heating network, Y-network (Kuosa et al, 2013, p.453))

A district heating network as given in figure 1 is also called a Y-network. These type of networks are traditionally built in Finland. It is called a Y-network, since it consists of numerous Y-connections of three pipe branches. The mass flow rate is determined according to the consumption of the different consumers, which is the demand at the loads. The supply temperature is dependent on the outdoor temperature and the temperature of the water at the source, such as a heating plant. The water flows at each of the consumers points are throttled by control valves (Kuosa et al, 2013, p.252).

The drawback of Y-networks is that they need a large number of valves. For this reason, another type of district heating network is designed, called a ring network, see figure 2. This type of district heating network is suitable in case every consumer has their own centrifugal pump, instead of a valve. In ring networks, every consumer in the district heating network has equal pipe lengths, and therefore equal pressure losses (Kuosa et al, 2013, p.452).

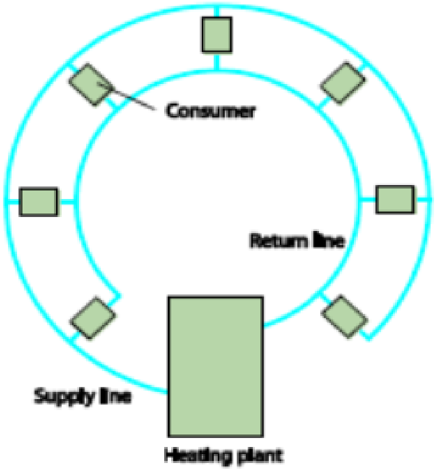


Figure 2: District heating network, ring network (Kuosa et al, 2013, p.453)

District heating networks can also contain non-pipe elements. During the flow of hydraulics through pipes, this flow loses part of its initial energy due to frictional resistance. This results in a loss of pressure over each pipe. To control this loss of pressure over large networks, non-pipe elements can be added to the network. One of these non-pipe elements is valves. In district heating networks, water flows at local consumers are throttled by valves. The water flow has a tendency to flow through the shortest routes, where the pipes have the lowest resistance. This is why the valves of the closest consumers are throttled most in the network when compared to those of the other consumers. This causes large local pressure differences and losses, which can complicate the use of the network. District heating networks can also contain pumps. These pumps are used to control the pressure in the network. There are two types of pumps, namely a main pump and a pump at the consumers (Kuosa, 2013, p.452).

2.2 Gas networks

A gas network is a network in which gas is distributed from a source to several different consumers, also called loads. The gas network consists of pipes, through which the gas flows, and loads, at which gas is consumed. There are different types of gas networks. The first type is low-pressure networks. These are networks that operate between 0-75 mbar gauge (Osiadacz, 1987, p.76). The second type of networks is medium-pressure networks. These are networks that operate between 0.75-7.0 bar gauge (Osiadacz, 1987, p.77). The third type of networks is high-pressure networks. these are networks that operate above 7.0 bar gauge (Osiadacz, 1987, p.78).

Just like district heating networks, gas networks can contain non-pipe elements. One of these elements is valves, which control the flow through the pipes of the network (Osiadacz, 1987, p.33). Another non-pipe element of gas networks is a pressure regulator, which automatically vary the rate of gas flow through a pipe to maintain a (preset) outlet pressure (Osiadacz, 1987, p.26). Gas networks can also include compressors, which compensate the loss of pressure in a network (Osiadacz, 1987, p.17).

3 Mathematical description

To solve a problem involving a hydraulic network, graph theory is used. Any network can be described by a set of matrices based on the topology of the network (Osiadacz, 1987, p.40). All different hydraulic networks, whether gas or water, can be represented as a directed graph (Abeysekera et al, 2016, p.993). Graph theory is used, since it enables the calculations to be performed in the most simple way. Graph theory represents a network by means of incidence properties of network components. An advantage of this is that the network representation is explicit (Osiadacz, 1987, p.35). In a graph, the pipelines are represented as edges (or branches) and the interconnections of the pipelines, loads, and sources are represented as nodes (Abeysekera et al, 2016, p.993). Here, a definite direction is assigned to each branch. This means that a graph of a hydraulic network is a directed graph, where each branch has a certain ordered pair of nodes (Osiadacz, 1987, p.36).

For the analysis of hydraulic networks it is necessary to make a distinction between two types of nodes, because in each network there is at least one source, at which the pressure is known, and the other nodes, the load nodes, at which the pressure is unknown. The first type of node is a reference node (or slack node). Mathematically, this node is referred to as the independent node, on which all nodal and branch quantities are dependent. In hydraulic networks there is a source node which is often used as the reference node since here, usually, the pressure is known. Let the amount of reference nodes be denoted by m. The other type of node is the load node, which are the points in the network where loads (positive or zero) are demanded on the network. Here, a positive load a demand and a zero load represents for example a junction. Let the amount of these nodes be denoted by n. For a steady-state condition the total load on the network is balanced by the inflow into the network at the source node (Osiadacz, 1987, p.41).

In the case of a district heating network, or a gas network, we have a directed graph. Let the total number of edges be denoted by e. Furthermore, the graph is connected, which means that there is a path between any pair of nodes (Altman & Boulos, 1995, p.36). We can describe which nodes are connected by which pipes, including the orientation of the edge, by a branch-nodal incidence matrix. Let A denote the branch-nodal incidence matrix that has a row for each node and a column for each edge (Arsene et al, 2004, p.63):

$$A(i,j) = \begin{cases} 1, & \text{branch j enters node i} \\ 0, & \text{branch j is not connected with node i} \\ -1, & \text{branch j leaves node i} \end{cases}$$

The branch-nodal incidence matrix of the full network has $m+n$ rows and e columns. This matrix can be divided into other matrices. First, A can be divided into A_1 and A_2. Here, A_1 denotes the branch-nodal incidence matrix with m rows and e columns. This denotes the incidence matrix of the reference nodes. A_2 denotes the reduced branch-nodal incidence matrix, which has n rows and e columns. This represents the network without the reference nodes.

Another way to divide A uses a spanning tree. A tree is a connected graph without loops. In a tree there is one, and only one, simple path between each pair of nodes in a tree (Osiadacz, 1987, p.37). A simple path is a sequence of alternate nodes and branches which begins and ends with a node, in which no branch occurs more than one time (Osiadacz, 1987, p.36). A tree of a connected graph is called a spanning tree. This is a subgraph of the graph, which includes all its nodes. The branches of a spanning tree are called tree branches and the branches of the graph that do not belong to the spanning tree are called chords, which form a co-tree (Osiadacz, 1987, p.37). Therefore, a spanning tree has its own branch-nodal matrix A_t and a co-tree incidence matrix A_c, such that $A = [A_t\ A_c]$ (Arsene et al., 2004, p.63).

To obtain a spanning tree, different search strategies can be employed, such as the depth-first search or breadth-first search. In this thesis, the breadth-first search is used to obtain the spanning tree and the co-tree. This method selects a node v, often the reference node or source, and then examines all the nodes that are incident to v. After this all the nodes u, that are incident to each of the new examined nodes are examined. If for the node u it holds that it has not been examined yet, the edge from v to u is added to the spanning tree, otherwise the edge from v to u is added to the co-tree. Next, all the nodes incident to each of the nodes examined before are examined. This is continued until all nodes have been examined (Osiadacz, 1987, p.52). In this thesis, the function *bfsearch* of Matlab is used.

In a directed graph, loops can occur. A loop is a closed path which does not use the same node more than once, except for the beginning and end nodes (Osiadacz, 1987, p.36). Let the amount of loops be denoted by k, where $k = e - (n + m)$ (Hamam & Brameller, 1971, p.1608) The loops in a graph can be described by a branch-loop incidence matrix. Let B denote the branch-loop incidence matrix that has a row for each loop and a column for each edge (Osiadacz, 1987, p.42):

$$B(v,w) = \begin{cases} 1, & \text{if edge } w \text{ has the same direction as loop } v \\ 0, & \text{if edge } w \text{ in not in loop } v \\ -1, & \text{if edge } w \text{ has opposite direction to loop } v \end{cases}$$

This matrix can also be expressed as $B = [B_t \; B_c]$, where B_t contains the edges of the spanning tree and B_c contains the edges of the co-tree. Furthermore B_t can be expressed using the branch-nodal incidence matrix:

$$B_t^T = -A_t^{-1} A_c$$

4 Steady-state analysis

As shown before, hydraulic networks can be described by a directed graph. Here, steady-state flow analysis of gas networks is used to compute the nodal pressures and pipe flows for the given values of the reference nodes and the demand of the load nodes. A steady-state flow analysis is an analysis independent of time, in which steady flow is assumed (Abeysekera et al, 2016, p.992). The goal is to determine the pressures p at the nodes and the flow rate q through the edges, given the pressure at the reference nodes p_1 and the loads L at the nodes. It is important to note that for pressure it holds that the pressure drops along a pipe in the direction of the flow. This is due to the loss of pressure to overcome friction. For this reason it is important to define the pressure drop Δp as the drop of pressure along a pipe. All these values need to satisfy the pipe flow equations and the first and second Kirchhoff's laws, which will be described below (Abeysekera et al, 2016, p.993).

4.1 Kirchhoff's first law

The flow rate q on the edges need to satisfy Kirchhoff's first law. This law states that the algebraic summation of the flow at any given node is zero. That is, the load at any node is equal to the sum of the edge flows into and out of the node. In matrix form this can be written as

$$L = A_2 q \tag{1}$$

Here, L is the load vector of dimension n, A_2 is the reduced branch-nodal incidence matrix with n rows and e columns, and q is a vector of dimension e that gives the flow rate in each branch (Osiadacz & Pienkosz, 1988, p.1312).

4.2 Kirchhoff's second law

The pressures p at the (load) nodes need to satisfy Kirchhoff's second law. This law states that the pressure drops around any closed loop is zero. In matrix form this can be written as

$$B\Delta p = 0 \tag{2}$$

Here, B is the $(k \times e)$ loop-branch incidence matrix and Δp is the vector of length e that expresses the pressure drops across the edges. If the pressure drop over an edge that goes from node i to node j is defined as $\Delta p = p_i - p_j$, the pressure drop can be written as

$$\Delta p = -A_2^T p$$

Here, A_2 is the nodal-branch incidence matrix with n rows and e columns and p is the vector of dimension n which gives the pressure at each node (Osiadacz & Pienkosz, 1988, p.1312).

4.3 Pipe flow equations

The flow of gas through a pipe can be described by many different pipe flow equations. Since the effects of friction are difficult to quantify, none of these pipe flow equations are universal (Osiadacz, 1987, p.69). The pipe flow equations describe the relation between the pressure drops Δp over the pipes and the flow q through the pipes. This relation is derived from the common pressure loss and continuity equations (Kuosa et al, 2013, p.454). To derive general flow equations, several simplifying assumptions need to be made (Osiadacz, 1987, p.73):

1. Steady flow, so that the flow is constant along the pipes;

2. Isothermal flow due to heat transfer with the surrounding through the pipe wall;

3. Negligible kinetic energy change in the pipe;

4. Constant compressibility of the hydraulic over the length of the pipe;

5. Validity of Darcy friction loss relationship;

6. Constant friction coefficient along the pipe length.

The flow rate can be expressed as a function ϕ of the pressure drops:

$$q = \phi(\Delta p) \tag{3}$$

Conversely, the pressure drops can be expressed as a function ψ of the flow rates:

$$\Delta p = \psi(q) \tag{4}$$

The relation ϕ or ψ depends on the type of hydraulic network that is described. The flow of gas can vary in low-pressure, medium-pressure and high-pressure distribution systems (Osiadacz, 1987, p.73-74). For hydraulic network problems the relation is generally nonlinear (Hamam & Brameller, 1971, p.1608). There are some commonly used pipe flow equations, dependent on the pressure level that characterizes the gas network (Ociadacz & Pienkosz, 1988, p.1313).

4.3.1 Lacey's equation

For low-pressure networks which operate between 0-75 mbar gauge, Lacey's equation is often used. This equation is given by

$$\Delta p = p_i - p_j = Kq^2$$

Here, K is Lacey's coefficient $K = 11.7 \cdot 10^3 \cdot \frac{L}{D^5}$, with L the length of the pipe in meters and D the diameter in millimeters of the pipe, and p is given in mbar and q is given in m^3h^{-1} (Osiadacz, 1987, p.76-77).

4.3.2 The Polyflo equation

For medium-pressure networks which operate between 0.75-7.0 bar gauge, the Polyflo equation is often used. This equation is given by

$$\Delta P = P_i - P_j = p_i^2 - p_j^2 = Kq^{1.848}$$

Here, K is a constant given by $K = 27.24\frac{L}{E^2D^{4.848}}$, with L the length of the pipe in meters, E an efficiency factor and D the diameter of the pipe in millimeters, and p is given in bar and q is given in m^3h^{-1} (Osiadacz, 1987, p.77-78).

4.3.3 The Panhandle 'A' equation

For high-pressure networks which operate above 7.0 bar gauge, the Panhandle 'A' equation is often used. This equation is given by

$$\Delta P = P_i - P_j = p_i^2 - p_j^2 = Kq^{1.854}$$

Here, K is a constant given by $K = 18.43\frac{L}{E^2D^{4.854}}$, with L the length of the pipe given in meters, E an efficiency factor and D the diameter of the pipe given in millimeters, and p is given in bar and q is given in m^3h^{-1} (Osiadacz, 1987, p.78).

4.3.4 The Weymouth equation

Another equation used for low-, medium-, and high-pressure networks is the Weymouth equation (Osiadacz, 1987, p.79). The general pipe flow equation is given by:

$$\Psi(q) = K \cdot q^{m_1} \tag{5}$$

Here, K is a pipe constant and m_1 is the flow exponent given by :

$$m_1 = \begin{cases} 2, & \text{for low-pressure networks} \\ 1.848, & \text{for medium-pressure networks} \\ 1.854, & \text{for high-pressure networks} \end{cases}$$

We see that for low-pressure networks we have:

$$\Delta p = p_i - p_j = Kq^2 \tag{6}$$

or

$$q = S_{ij} \left(\frac{S_{ij}(p_i - p_j)}{K} \right)^{\frac{1}{2}} \tag{7}$$

where p_i is the absolute pressure in node i and

$$S_{ij} = \begin{cases} 1, & \text{if } p_i > p_j \\ 0, & \text{if } p_i = p_j \\ -1, & \text{if } p_i < p_j \end{cases}$$

For the medium and high pressure networks we have

$$\Delta P = P_i - P_j = p_i^2 - p_j^2 = Kq^{m_1} \tag{8}$$

or

$$q = S_{ij} \left(\frac{S_{ij}(p_i^2 - p_j^2)}{K} \right)^{\frac{1}{m_1}} \tag{9}$$

where p_i is the absolute pressure in node i

$$S_{ij} = \begin{cases} 1, & \text{if } P_i{>}P_j(p_i{>}p_j) \\ -1, & \text{if } P_i{<}P_J(p_i{<}p_j) \end{cases}$$

5 Solving a system of nonlinear equations

In this thesis, the Weymouth equation (5) is used as a pipe flow equation in the steady-state flow analysis. The Weymouth equation is used, since it holds for low, medium and high pressure networks. The Weymouth equation leads to a system of nonlinear equations that need to be solved iteratively, which will be shown in section 6. There are different methods that can be used to solving a system of nonlinear equations iteratively, such as the Newton-Raphson method and the Hardy-Cross method. Since the Newton-Raphson method is the most commonly used method, this method will be used in this thesis (Osiadacz, 1987, p.83).

5.1 Newton-Raphson method

Let the system of nonlinear equations be given by

$$f(x) = 0$$

for $f : \mathbb{R}^n \to \mathbb{R}^n$ and $x \in \mathbb{R}^n$. For the nonlinear equations in this thesis, the variable x are the pressures p or the flow q, depending on the method used (see section 6). The function f is then given by equation (3) or equation (4). Then for f_i the function and variable $x = (x_1, ..., x_n)$, for $i = 1, ..., n$ we have:

$$f_i(x_1, ..., x_n) = 0$$

To solve this system, the Newton-Raphson method requires a starting value for x, an initial estimation to the final solution, called x^0. Then, in each iteration, the first step is to calculate $f(x^h)$. The second step is to determine the Jacobian matrix $J(\mathbf{x}^h)$. Here, the Jacobian matrix is given by (Osiadacz, 1987, p.86):

$$J(x) = \begin{pmatrix} \frac{\delta f_1}{\delta x_1} & \frac{\delta f_1}{\delta x_2} & \cdots & \frac{\delta f_1}{\delta x_n} \\ \frac{\delta f_2}{\delta x_1} & \frac{\delta f_2}{\delta x_2} & \cdots & \frac{\delta f_2}{\delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta f_n}{\delta x_1} & \frac{\delta f_n}{\delta x_2} & \cdots & \frac{\delta f_n}{\delta x_n} \end{pmatrix}$$

The third step is to determine the next approximation of the final solution:

$$x^{h+1} = x^h - (J(x^h))^{-1} f(x^h)$$

After this, the next iteration is started and the method returns to the first step. The Newton-Raphson method continues making new approximations until the difference between the solutions of the iteration h and the iteration $h+1$ is close enough. Depending on the method used, the error can, for example, be given by $||x^h - x^{h+1}||_2 < \epsilon$, for some ϵ small (Osiadacz, 1987, p.87). In this thesis, the epsilon is taken as $\epsilon = 10^{-5}$. Furthermore, the iteration process in the Newton-Raphson method is stopped when it reaches the 100th iteration, in order for the process not to take too much time. Otherwise, the iteration process could take an infinite amount of time, because the starting value could cause the process to diverge.

6 Solving the steady-state problem

There are several methods for solving the steady-state problem as described in section 4. In this thesis, we take a look at the most used methods, which include the nodal method, the loop method and the loop-node method. In this thesis, theory and computer simulations will be used to compare the nodal method, the loop method, and the loop-node method.

6.1 Nodal method

In the steady-state problem the pressure p and the flow rate q needs to be calculated. The nodal method focuses on the nodes in the network. This means that this method takes the pressure at the nodes p as unknowns. For the flow equations, this method uses equation (3). Using Kirchhoff's first law (1), this gives the following set of equations (Osiadacz & Pienkosz, 1988, p.1318):

$$\begin{cases} L = A_2 q \\ \Delta p = -A^T p \\ q = \phi(\Delta p) \end{cases}$$

where L is a vector of length n, q is a vector of length e, Δp is a vector of length e, and p is a vector of length $n+m$. Combining these equation gives

$$L = A_2 \phi(\Delta p) = A_2 \phi(-A^T p)$$

Thus the nonlinear system of equations which needs to be solved is

$$A_2 \phi(-A^T p) - L = 0$$

In the Newton-Raphson method, the first step is to make an initial approximation of the pressures. There are several ways to do this (Osiadacz, 1987, p.106). The best way to do this will be investigated in section 8.1.1.

The nodal error gives the imbalance of the approximation of the pressures. It is a function of the nodal pressures, except for the reference nodal pressure (which is fixed). It is given by

$$F(p_2) = A_2\phi(-A^T p) - L$$

and it will tend to zero as the pressures approach their true values. As stated before, the Newton Raphson method is used to approach the pressures p iteratively until the nodal errors are small enough. That is, the Newton-Raphson process is continued until either $||F(p_2)||_2 < \epsilon$ or 100 iterations are reached. The iterative scheme is given by (Osiadacz, 1987, p.106):

$$p_2^{h+1} = p_2^h - (J^{-1})^h (F(p_2))^h$$

with h the iteration number. Here, J is the Jacobi matrix defined by $J = -A_2 \cdot \text{diag}(\frac{1}{m_1}\frac{q}{\Delta p}) \cdot A_2^T$ (Osiadacz, 1987, p.108).

6.2 Loop method

The loop method focuses on the pipes in the network. This means that this method takes the flow rate at the edges q as unknowns. For the flow equations, this method uses equation (4). Using Kirchhoff's first law (1), this gives the following set of equations (Osiadacz & Pienkosz, 1988, p.1314):

$$\begin{cases} L = A_2 q \\ \Delta p = -A^T p \\ \Delta p = \psi(q) \\ B\Delta p = 0 \end{cases}$$

where L is a vector of length n, q is a vector of length e, Δp is a vector of length e, and p is a vector of length $n + m$.

In the Newton-Raphson method, the first step is to make an initial approximation of the flow pattern. In the loop method, the first initial flow pattern must be chosen to satisfy Kirchhoff's first law (Brkic, p.2956). In section 8.1.2 it is shown how this is done. Since the flow is an approximation of the true values, a loop flow is introduced. This is the flow correction which is added to the first approximation to attain their true values. In general, the branch flows q are a function of their initial approximations Q^0 and of all the loop flows q_k, given by

$$q = Q^0 + B^T q_k \tag{10}$$

where q is the approximation of the branch flows, Q^0 is the initial approximation of the final solution, and q_k is the flow trough the pipes in the loops (Osiadacz, 1987, p.126).

The advantage of the loop method is that Kirchhoff's second law (2) can be used, since this decreases the number of equations that need to be solved. The loop method requires that a set of loops is defined in the network (Osiadacz, 1987, p.125). This can be done by dividing the matrix A_2 in two matrices, namely $A_2 = [A_{2s}\ A_{2k}]$, where A_{2k} is the nodal-branch incidence matrix that consists of all edges that are in the loops. The same can be done for the flow, so $q = [q_s\ q_k]$, where q_k are the flows of the branches in the loops and q_s are the remaining branches (Osiadacz & Pienkosz, 1988, p.1314). Combining Kirchhoff's second law (2), equation (4), and equation (10) we obtain:

$$B\Delta p = B\psi(q) = B\psi(Q^0 + B^T q_k) = 0$$

where B is the branch-loop incidence matrix of dimension k rows by e columns. With this equation the flow in the loops q_k can be determined (Osiadacz, 1987, p.126). In each loop there is a loop error, which is a function of all the loop flows. The vector of the loop errors is given by

$$F(q) = B\psi(Q^0 + B^T q_k)$$

These errors will tend to zero as the loop flows approach their true values.

To solve the equations, the iterative Newton-Raphson method is used. The Newton-Raphson process is continued until either $||F(q)||_2 < \epsilon$, or 100 iterations are reached. The scheme that needs to be solved is:

$$q^{h+1} = q^h - (J^{-1})^h (F(q))^h$$

where h gives the iteration number (Osiadacz, 1987, p.127). J gives the Jacobi matrix defined by $J = BMB^T$ with $M = \text{diag}(m_1 K_i |q_i|^{m_1 - 1})$ (Osiadacz, 1987, p.129).

6.3 Loop-node method

The loop-node method essentially solves the set of loop equations, just like the loop method, so: $f(q) = B\psi(q)$ (Osiadacz, 1987, p.148). For the loop-node method, the spanning tree needs to be constructed. So here we obtain: $A_2 = [A_{2t}\ A_{2c}]$ and $q = [q_t\ q_c]$, where A_{2c} is the nodal-branch incidence matrix that consists of all edges that are in the co-tree, A_{2t} is the nodal-branch incidence matrix that constists of all edges that are in the spanning tree, q_c are the flows of the branches in the co-tree, and q_t are the flows of the branches in the spanning tree. Kirchhoff's first law (1) then becomes:

$$L = A_2 q = A_{2t} q_t + A_{2c} q_c$$

Which can be rewritten as:

$$q_t = (A_{2t})^{-1}(L - A_{2c}q_c) = (A_{2t})^{-1}L - (A_{2t})^{-1}A_{2c}q_c \tag{11}$$

This can be solved as long as $(A_{2t})^{-1}$ exists (Osiadacz, 1987, p.149).

The loop-node method does not actually solve the loop equations, but the loop equations are transformed to an equivalent set of nodal equations which are then solved to give the nodal pressures. These nodal pressures are used to calculate the corrections to the chord flows Δq_c (the branches of the co-tree), and the branch flows are then obtained from them by using equation (11). Using the Newton-Raphson method, this process is repeated iteratively until the final solution is obtained. This happens when the corrections to the chord flows are less then some error, namely when $||\Delta q_c|| < \epsilon$, or when 100 iterations are reached. For each iteration the correction Δq is given by

$$\Delta q^h = (R^h)^{-1}(\Delta p^{h+1} - \Delta p^h) = (R^h)^{-1}(-A^T p^{h+1} - \Delta p^h) \tag{12}$$

where h is the iteration number and R is given by $R = m_1 K|q|^{m_1 - 1}$, with K a pipe constant (Osiadacz, 1987, p.151). Furthermore, we know that:

$$-A^T p = -A_1 p_1 - A_2 p_2 = \Delta p$$

where p_1 is a vector of unknown pressures in the load nodes and p_2 is a vector of the known pressure(s). Using this and equation (12) we obtain than the nodal pressures p^{h+1} can be calculated using the Jacobian matrix $J = -A_2\text{diag}(R^{-1})A_2^T$, namely

$$p^{h+1} = J^{-1}(A_2 R^h(\Delta p^h + A_1^T p_1))$$

where p_1 is the pressure of the reference node and A_1 is the matrix A with only the rows of the reference nodes. The new values of the chord flows are then given by

$$q_c^{h+1} = q_c^h + \Delta q_c^h$$

where Δq_c^h is the part of Δq^h that is in the co-tree (Osiadacz, 1987, p.152).

7 Theoretical comparison

In order to determine when best to use which method, a comparison will be made between the nodal method, the loop method, and the loop-node method. The goal is to determine what type of network needs what type of method. In this section, a theoretical comparison will be made. After this, a comparison will be made by doing some computer simulations, which is done in section 8.

7.1 Computer storage

Every computer program requires storage from the computer. The larger the data used, the more storage is needed. For large networks, it can be worthwhile to use methods to decrease the amount of storage needed. One of these methods is to use the fact that the Jacobi matrix in the three methods is sparse. A matrix is sparse if it contains a large quantity of zero elements. In that case specials methods can be used to operate on the non-zero elements only and not save the zero elements to the computer (Osiadacz & Pienkosz, 1988, p.1987). This can become more worthwhile when the number of nonzero elements increases, such as for the Jacobi matrix that is used in the different methods. To determine the sparsity of the Jacobi matrix, the sparsity coefficient can be used. This gives the proportion of zero coefficients in the Jacobi matrix (Osiadacz, 1987, p.110).

7.1.1 Nodal method

The Jacobi matrix of the nodal method is square ($n \times n$) and symmetrical. An advantage of the nodal method is that its Jacobi matrix will contain many elements with a zero coefficient. For the nodal Jacobi matrix, its sparsity coefficient depends only on the structure of the network and is therefore a constant (Osiadacz & Pienkosz, 1988, p.1318). In fact, the sparsity coefficient of the nodal coefficient matrix is given by

$$1 - \frac{n + 2b}{n^2}$$

where b is the number of branches in the network that connect any two load nodes (i.e. without the branches that connect to the reference node) (Osiadacz, 1987, p.110).

7.1.2 Loop method

The Jacobi matrix of the loop method is square ($k \times k$) and symmetrical. The loop method defines loops and the sparsity in the Jacobi matrix is highly dependent on the method of loop generation. For networks without loops, the loop Jacobi matrix is not defined. This is because for networks without loops the loop method does not require any number of iterations in the iterative process, as will be shown later in this thesis. Therefore, the sparsity coefficient for networks without loops is not defined. For networks with loops, the defined loops should be the ones which are least interconnected, leading to a Jacobi matrix with a high degree of sparsity. The sparsity coefficient of the loop method is given by

$$\frac{k + 2i_k}{k^2}$$

where i_k is the number of loops that share one or more common branches. Thus, to ensure a high degree of sparsity in the Jacobi matrix the definition of the loops is important (Osiadacz, 1987, p.130). However, in general it is very difficult to define the loops that will give maximum sparsity in an automatic and efficient way (Osiadacz, 1987, p.132).

7.1.3 Loop-node method

The Jacobi matrix of the loop-node method is square ($n \times n$) and symmetrical. In fact, this matrix has identical symmetry and degree of sparsity as the nodal Jacobi matrix of the nodal method.

Thus the Jacobi matrix of the loop-node method contains many zero elements (Osiadacz, 1987, p.155).

7.1.4 Final remarks

As shown, the three methods use two different Jacobi matrices. Namely, the nodal Jacobi matrix for the nodal method and the loop-node method, and the loop Jacobi matrix for the loop method. The sparsity coefficient of the nodal Jacobi matrix is larger than the sparsity coefficient of the loop Jacobi matrix in each case. This means that the nodal method and the loop-node method require less computer storage for large networks than the loop method. For a good loop-generation method, the loop method will still require more computer storage, but the difference with the nodal method and the loop-node method will be smaller than in the case of a bad loop-generation method. However, to make a good loop-generation method is very difficult. Thus it is reasonable to conclude that in terms of computer storage the loop method is worse than the nodal method and the loop-node method for every size network (Osiadacz, 1988, p.30).

7.2 Sensitivity to starting values

For all methods, one of the requirements is to have a starting value. For the nodal method this means an initial estimation of the pressures at the nodes. For the loop method and the loop-node method this means an initial estimation of the flow through the pipes. A "good" starting value can lead to fast convergence to the final solution, whereas "bad" starting value can lead to a slow convergence to the final solution or even divergence.

7.2.1 Nodal method

For the nodal method, it is required to make an initial estimation to the pressures in the nodes. This estimation does, in theory, not have to follow any requirements. Thus the initial approximation can, in theory, be anything. However, in practice, this is not the case. The nodal equations described in section 6.1 lead, with the Weymouth equation, to a pipe flow equation of the form of equations (7) and (9). The terms in these equations are square-root for low-pressure networks and close to square-root for medium- and high-pressure networks. This means that if the initial values are too far removed from the final solution, the iterative process will diverge (Osiadacz, 1987, p.109). Therefore, for the nodal method it is important to take a good starting value.

7.2.2 Loop method

The loop method has good convergence characteristics. The loop equations, with the Weymouth equation, gives the pipe flow equation as in equations (6) and (8). The terms in these equations are quadratic for low-pressure networks, and nearly quadratic for medium- and high-pressure networks. This leads to fast convergence and insensitivity to the starting values (Osiadacz, 1987, p.136). Thus it is not that important what the starting value is for the loop method, as long as it satisfies Kirchhoff's first law (Brkic, 2011, p.1956).

7.2.3 Loop-node method

For the estimation of the flow through the pipes in the loop-node method it is only required to estimate the flow in the chords. To make this estimation, Osiadacz (1987, p.155) takes the value of $0.5\frac{m^3}{h}$ for the chord flows of a specific network (network 6 in appendix A). In section 8.1.3 we will see that this is actually a reasonable value to use for other networks as well. Then, using Kirchhoff's first law, the flow trough all pipes can be determined, making sure that the initial approximation satisfies the flow balance at each node. It is important that the initial values of the chord flows are not zero, because if they were zero the value of R (see section 6.3) will be zero, resulting the Jacobi matrix to have entry values that are infinity, which is impossible (Osiadacz, 1987, p.155).

The loop-node method corrects the flows through the chords, which are equivalent to the loop flows of the loop method. This means that the fast convergence of the loop method also hold for the loop-node method (Osiadacz, 1987, p.150). This means that for the loop-node method it also is not that important what the starting value is, as long as the flows through all the pipes satisfies Kirchhoff's first law.

7.2.4 Final remarks

Thus, we see that the loop method and the loop-node method are not sensitive to starting values. The loop-node method even has a preset starting value, which leads to fast convergence. The nodal method on the other hand, is very sensitive to starting values. If the starting values are too far removed from the final solution, the iterative process will even diverge. It is important to investigate how sensitive the nodal method is. Furthermore, it is interesting to see if there is a reasonable starting value that works, independent of the network that needs to be solved.

7.3 Computational time

The computational time gives the amount of time needed by the computer to do a simulation. For the nodal, loop, and loop-node method this is dependent on several properties of the methods. The first property of the methods that influences the computational time is the sparsity of the Jacobian matrices, as stated in section 7.1. This would lead us to conclude that the loop method requires the most computer storage and thus more computational time than the nodal method and the loop-node method.

Another property that influence the computational time is the number of equations that needs to be solved in the iterative process. In the loop method and the loop-node method, which are both based on the loop equations, the number of equations that need to be solved is the total number of loops k, whereas the number of equations that need to be solved for the nodal method is the total number of load nodes n. This means that the number of equations that needs to be solved for the nodal method is the most of the three methods (Stevanovic et al, 2007, p.1537). Furthermore, the loop-node method has more matrix-vector equations than the loop method (Osiadacz & Pienkosz, 1988, p.1317). Thus, the loop method has the least amount of equations that needs to be solved. This means that the loop method should be the fastest in solving the iterative Newton-Raphson process, whereas the nodal method should be the slowest in solving this process.

Something else that influences the computational time is that the loop method and the loop-node method both require a loop generation method. This requires computations, which for a "bad" loop generation method can take much longer. The loop method also needs to compute the branch-loop matrix B, which also requires extra computations (Osiadacz & Pienkosz, 1988, p.1318).

8 Computer simulations

So far we have seen the theoretical comparison between the nodal method, the loop method, and the loop-node method. In this section computer simulations will be done to show the differences between the methods. In order to do this, several different networks are used that represent different types of networks. These networks are shown in Appendix A.

8.1 Sensitivity to starting values

The theoretical comparison in section 7 showed that there is a difference between the nodal method, the loop method, and the loop-node method when talking about sensitivity to starting values. According to this, the nodal method is very sensitive to starting values, whereas the loop method and the loop-node method are not sensitive to starting values. Computer simulations can show if this is really the case. Furthermore, computer simulations can show what would be a good starting value for each method, independent of type of network.

8.1.1 Nodal method

As stated before, the nodal method diverges if the initial values are too far removed from the final solution. This means that for the nodal method it is important to take a good starting value. In order to investigate what would be a good starting value for the nodal method, there are different types of starting values to look at. Note that in a network there is at least one node at which the pressure is known. It is important not to vary this given pressure value in the starting value. If this pressure is set at a different value, a different solution will be reached. What the other values can be will be investigated below.

One type of starting value is to take every value in the starting value equal to the known pressure in the reference node. This will result in a vector in which every value is the same (and equal to the known pressure). Doing this will result in a singular Jacobian matrix, which means that no solution will be reached. The Jacobian matrix will be singular since, by the Weymouth equation, the vector of the flows q and the vector of the pressure drops Δp will both be a vector consisting of zeros, see equation (9). As a consequence, the nodal Jacobian matrix, given by $J = -A_2 \cdot \text{diag}(\frac{1}{m_1} \frac{q}{\Delta p}) \cdot A_2^T$, will be a matrix containing NaN's, since we are dividing by zero. This will lead to a singular Jacobian matrix.

Another type of starting value is to take every value in the starting value, except the known pressure at the reference node, equal to each other, but not equal to the pressure at the reference node. This will also result in a singular Jacobian matrix, which means that no solution will be reached. This is also a result of the Weymouth equation (7), since this will result in a vector of the flows q and a vector of the pressure drops Δp, both with many zero values, except at the edges that are connected to the reference node. As a consequence, the nodal Jacobian matrix, given by $J = -A_2 \cdot \text{diag}(\frac{1}{m_1} \frac{q}{\Delta p}) \cdot A_2^T$, will be a matrix containing NaN's, since we are dividing by zero. This will lead to a singular Jacobian matrix.

Now, it is interesting to look at what happens when every value in the starting value is taken differently. It is reasonable to assume that the pressure will decrease in the pressure vector p. So, first, let the pressures be a vector of dimension $n + m$ in which all values are equal to the known pressure, denoted by $p_1 \cdot \text{ones}(n+m)$. Then, we subtract from this vector a vector from 0 to n, with increments of 1, denoted by $(0 : n)$, multiplied by a percentage of the known pressure. The results of this are shown in figure 3 and figure 4.

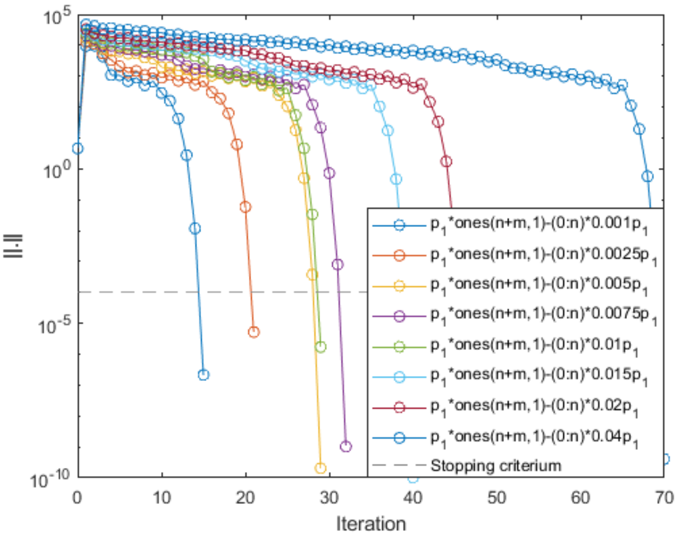


Figure 3: Convergence plot of the starting values for the nodal method using a vector with only the known pressure, denoted as $p_1 \cdot \text{ones}(n+m, 1)$, minus a vector from 0 to n, with increments of 1, denoted as $(0 : n)$ times a percentage of the known pressure, for a network with loops (network 6)

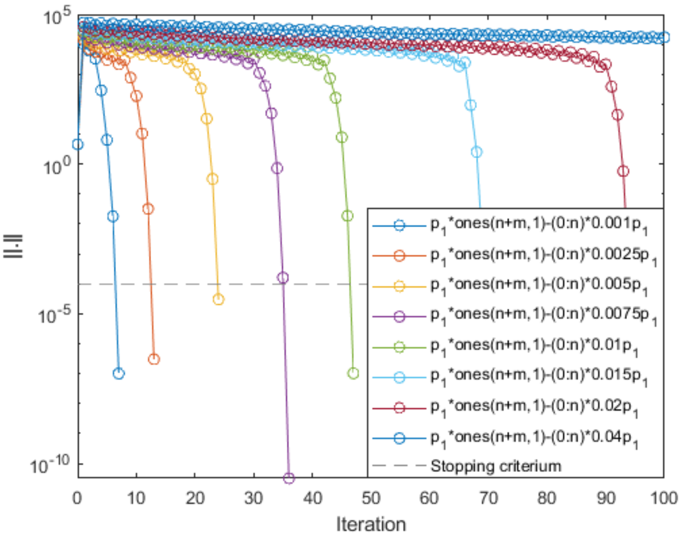


Figure 4: Convergence plot of the starting values for the nodal method using a vector with only the known pressure, denoted as $p_1 \cdot \text{ones}(n+m, 1)$, minus a vector from 0 to n, with increments of 1, denoted as $(0 : n)$ times a percentage of the known pressure, for a network without loops (network 7)

From these figures, it is visible that for a greater percentage of the known pressure, the iteration number increases fast. This is also visible in table 2. From this table it is also clear that all these number of iterations are not small. In fact, in most cases the number of iterations is large. Furthermore, the number of iterations differ between the different starting values, even though the difference between the starting values is small. This would lead us to believe that none of these starting values are “good” staring values to use in the nodal method for arbitrary networks.

Table 2: Number of iterations for the starting values for the nodal method approaching zero, using a vector with only the known pressure, denoted as $p_1 \cdot \text{ones}(n+m, 1)$, minus a vector from 0 to n, with increments of 1, denoted as $(0 : n)$ times a percentage of the known pressure

	Number of iterations	
Starting value	**Network with loops**	**Network without loops**
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.001 \cdot p_1$	15	7
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.0025 \cdot p_1$	21	13
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.005 \cdot p_1$	29	24
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.0075 \cdot p_1$	32	36
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.01 \cdot p_1$	29	47
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.015 \cdot p_1$	40	70
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.02 \cdot p_1$	46	94
$p_1 \cdot \text{ones}(n+m,1) - (0:n) \cdot 0.04 \cdot p_1$	70	100+

Thus, we have seen that taken a relative difference from the known pressure as a starting value does not result in a starting value we can take for an arbitrary network. Another possibility for a starting value is taking a vector consisting of only small values, or taking a vector consisting of only large values. Since the pressure will decrease in the network, taking large values will not result in a “good” starting value. Thus we will look at a vector consisting of small values.

In this case the unknown pressures are taken as a vector from 1 to n, with increments of 1, denoted by $(1 : n)$. This vector is then multiplied by something small, such that the vector consists of values close to zero. In this case the final solution is reached. Here, we look at the absolute difference. Since the known pressures for network 6 and network 7 are in the order of 10^3, this difference is large enough to know it reasonable to talk about. The results are shown for a network with loops and a network without loops, as shown in figure 5 and figure 6.

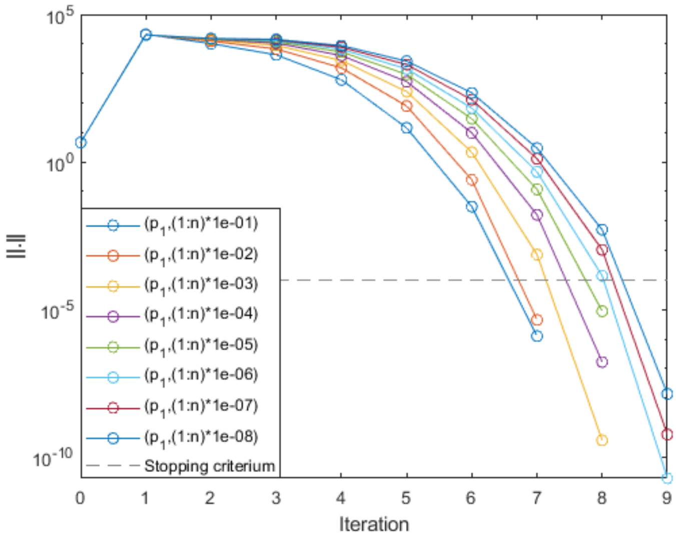


Figure 5: Convergence plot of the starting values for the nodal method approaching zero, using a vector from 1 to n, with increments of 1, denoted as $(1:n)$, for a network with loops (network 6)

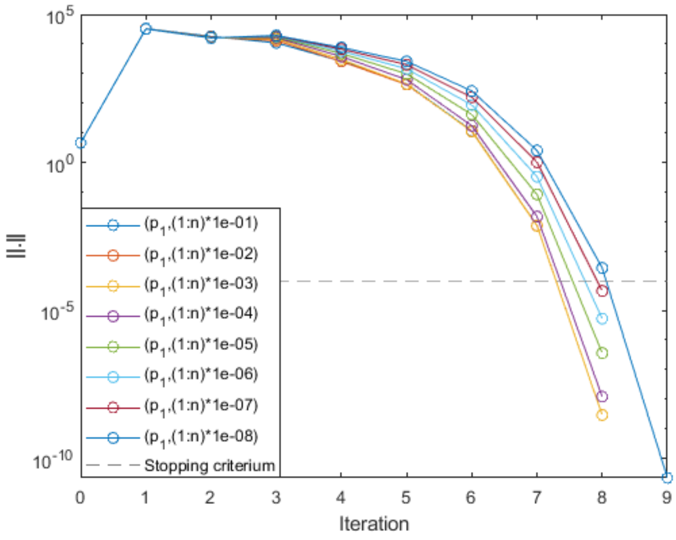


Figure 6: Convergence plot of the starting values for the nodal method approaching zero, using a vector from 1 to n, with increments of 1, denoted as $(1:n)$, for a network without loops (network 7)

In these figures it is clearly visible that the investigated starting values all need about the same number of iterations to reach the final solution. This is also visible in table 3. It is interesting to note that the number of iterations needed by all these starting values is very close to each other. Furthermore, the number of iterations needed to reach the final solution is not high. This means that taking one of these starting values as starting value will be a good initial approximation in many different networks. In fact, going forward in this thesis we will take $(p_1, (1:n) \cdot 1e-01)$ as the starting value for the nodal method for every network.

Table 3: Number of iterations for the starting values for the nodal method approaching zero, using a vector from 1 to n, with increments of 1, denoted as $(1:n)$

	Number of iterations	
Starting value	**Network with loops**	**Network without loops**
$(p_1, (1:n) \cdot 1e-01)$	7	8
$(p_1, (1:n) \cdot 1e-02)$	7	8
$(p_1, (1:n) \cdot 1e-03)$	8	8
$(p_1, (1:n) \cdot 1e-04)$	8	8
$(p_1, (1:n) \cdot 1e-05)$	8	8
$(p_1, (1:n) \cdot 1e-06)$	9	8
$(p_1, (1:n) \cdot 1e-07)$	9	8
$(p_1, (1:n) \cdot 1e-08)$	9	9

Another starting value for the nodal method to look at is when only a small deviation from the final solution is taken. In this case, the pressures from the final solution at all nodes is taken. Then for all these values, except values at the reference node, we have taken this value plus or minus some percentage of the known pressure at the reference node. First, we take a look at the final solution plus some percentage of the pressure at the reference node. The results of this are shown in figure 9 and figure 10.

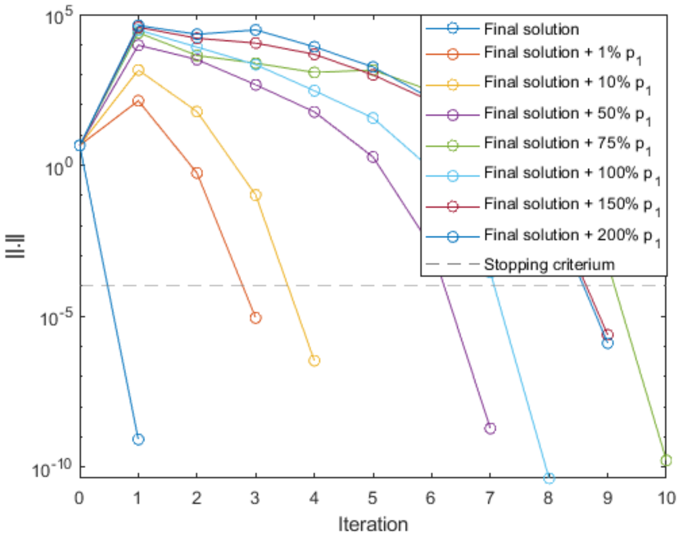


Figure 7: Convergence plot of the starting values for the nodal method: final solution plus some percentage of the reference pressure, for a network with loops (network 6)

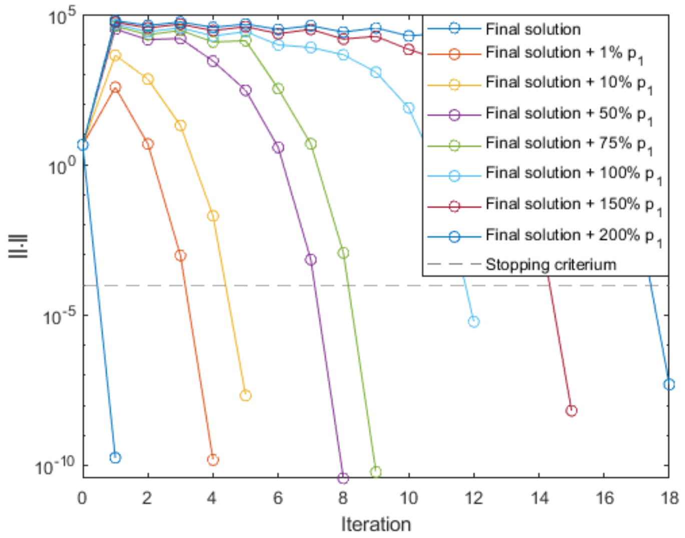


Figure 8: Convergence plot of the starting values for the nodal method: final solution plus some percentage of the reference pressure, for a network without loops (network 7)

From these figures it is visible that the number of iterations increases as the starting value differs more from the final solution. This is also visible from table 4. However, it is also visible that the process of the nodal method does not diverge for these networks, even though the starting value is taken far from the final solution.

Table 4: Number of iterations of starting values for the nodal method: final solution plus some percentage of the reference pressure

	Number of iterations	
Starting value	**Network with loops**	**Network without loops**
Final solution	1	1
Final solution + 1%	3	4
Final solution + 10%	4	5
Final solution + 50 %	7	8
Final solution + 75 %	10	9
Final solution + 100 %	8	12
Final solution + 150 %	9	15
Final solution + 200 %	9	18

That the nodal method does not diverge for these networks is also visible if we take a look at the final solution minus some percentage of the pressure at the reference node. The results of this are shown in figure 9, figure 10, and table 5. We see that similar results are obtained when the final solution is taken minus some percentage of the pressure at the reference node is used.

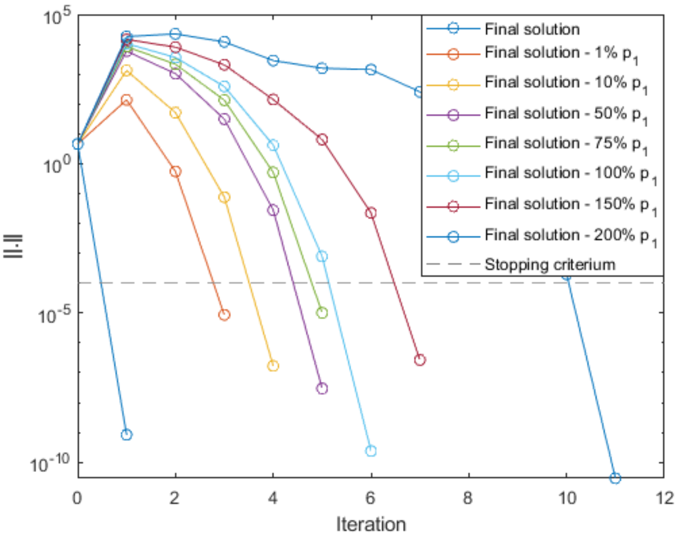


Figure 9: Convergence plot of the starting values for the nodal method: final solution minus some percentage of the reference pressure, for a network with loops (network 6)

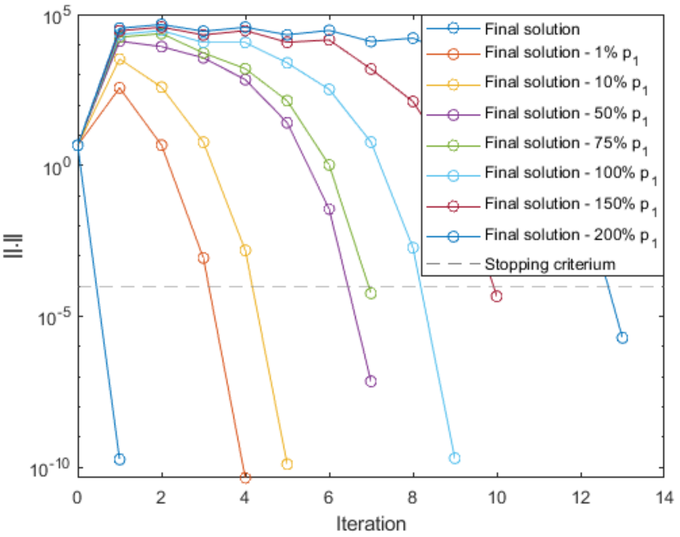


Figure 10: Convergence plot of the starting values for the nodal method: final solution minus some percentage of the reference pressure, for a network without loops (network 7)

Table 5: Number of iterations of starting values for the nodal method: final solution minus some percentage of the reference pressure

	Number of iterations	
Starting value	**Network with loops**	**Network without loops**
Final solution	1	1
Final solution - 1%	3	4
Final solution - 10%	4	5
Final solution - 50 %	5	7
Final solution - 75 %	5	7
Final solution - 100 %	6	7
Final solution - 150 %	7	10
Final solution - 200 %	11	13

From these last figures and tables, we see that the nodal method is not as sensitive to the starting values for these networks. This contradicts what stated in section 7.2, where we stated that the nodal method is sensitive to the starting values. This can lead us to think that network 6 and network 7 are very robust networks, that do not give a good representation of the sensitivity of the nodal method to the starting values.

8.1.2 Loop method

To estimate the flow through the pipes for the loop method there is a simple method to use. This method is to use Kirchhoff's first law (1). Since the loads at the nodes are known and the branch nodal incidence matrix is known, it is possible to obtain an initial approximation to the flow through the pipes, namely one of the infinite number of flow combinations. In this thesis, this is done by using:

$$Q^0 = A_2 \backslash L$$

Consequently, this initial approximation ensures that a flow balance exists at each node, which is a requirement of the loop method.

8.1.3 Loop-node method

As stated before, Osiadacz (1987, p.155) takes the chords flows at the value of $0.5\frac{m^3}{h}$. After that, using Kirchhoff's first law, the flow trough all pipes can be determined, making sure that the initial approximation satisfies the flow balance at each node. To show that this indeed makes a reasonable initial approximation, we can take a look at different starting values for the chord flows for a network with loops (network 6), as shown in figure 11. Here it is visible that for values smaller than $0.5\frac{m^3}{h}$ the convergence is similar. For larger values however, the convergence becomes less fast for values larger than or equal 1000. This means that taking too large values for the starting value of the chord flows would not be optimal.

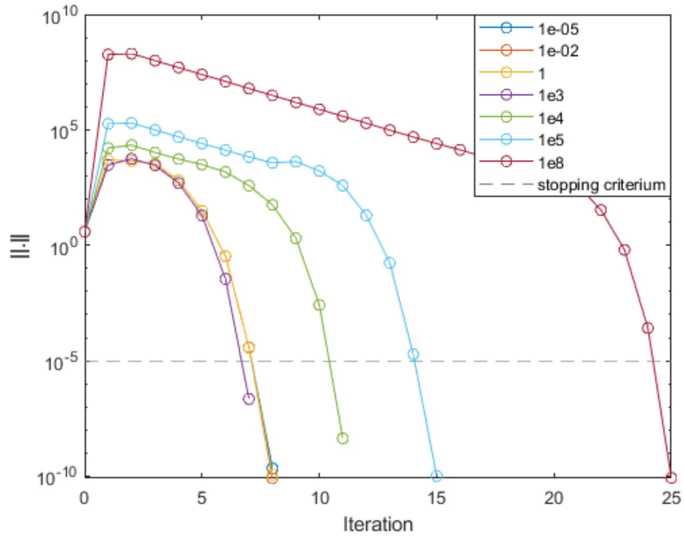


Figure 11: Convergence plot of Newton-Raphson for different starting values of the chords for the loop-node method (for network 6)

Furthermore, to show that $0.5\frac{m^3}{h}$ would indeed be a reasonable starting value for the chords for different type of networks we can look at the number of iterations needed to obtain the final solution as shown in table 6. Here only the networks with loops are taken into consideration, since otherwise there would be no chords. In table 6 it is shown that on average $0.5\frac{m^3}{h}$ would be a reasonable starting value. Furthermore, in most cases, taking values of $1000\frac{m^3}{h}$ or larger results in a longer iteration process.

Table 6: Iteration of different starting values of the chords for the loop-node method

	Number of iterations				
Starting value	Network 2	Network 3	Network 4	Network 5	Network 6
1e-05	4	5	6	7	8
1e-02	4	5	6	7	8
1	4	5	6	7	8
1e3	8	8	9	10	7
1e4	12	11	12	13	11
1e5	14	14	16	17	15
1e8	24	24	26	27	25

8.1.4 Final remarks

In this section we have seen the sensitivity of the starting values of the different methods. The nodal method is the only method of the three that is very sensitive in taking starting values, even though this was not visible in the computer simulations. A very good approximation of the final solution will lead to fast convergence to the final solution. However, making a good approximation of the final solution in the nodal method is difficult. A slight difference from the final solution will lead a longer iterative process, with more iterations. The loop method and the loop-node method on the other hand both have straight-forward methods to get a good starting value.

8.2 Computational time

It is interesting to investigate the computational time of the different methods, both in terms of the iterative process and the whole method. To do this a network with loops (network 6) and a network without loops (network 7) are investigated. All networks have used maximum sparsity techniques. To determine the computational time, each method is run a 1000 times, after which the mean is taken and stated as the computational time.

8.2.1 Network with loops

The results for the network with loops is given in table 7. From this it is clear that the loop method would be the fastest method to use in a network with loops. The loop method is also the fastest when only looking at the Newton-Raphson iterative process. Since the number of iterations of the three different methods does not differ much, this means that the iterative process of the loop method is in fact the fastest method in the iterative process.

Table 7: CPU time (in seconds) of the different methods for a network with loops (network 6)

Method	Whole program	Only Newton-Raphson	**Iterations**
Nodal method	0.011	0.0094	7
Loop method	0.0019	0.00070312	8
Loop-node method	0.011	0.0016	8

It is also interesting to note, that even though the nodal method requires the least amount of iterations, it takes the most time. Thus, for this network with loops, the nodal method is a very slow method.

8.2.2 Network without loops

The results for the network without loops is given in table 8. From this table it is clear that the loop-node method is the fastest method for solving a network without loops. The loop-node method does not even have to start the iterative process before reaching the correct final solution.

Table 8: CPU time (in seconds) of the different methods for a network without loops (network 7)

Method	Whole program	Only Newton-Raphson	**Iterations**
Nodal method	0.0097	0.0087	8
Loop method	0.001	0.00015625	1
Loop-node method	0.000875	0	0

Again, the nodal method is the slowest method in solving the network. This method also needs several iterations before reaching the correct final solution. Both the loop method and the loop-node method do not have to make several, if any, iterations. Thus, for this networks without any loops, the nodal method is a very slow method.

8.2.3 Final remarks

In this section we have seen the computational time needed by each method for a network with loops and for a network without loops. In doing this, we distinguished between only the iterative Newton-Raphson process and the whole method. From these investigations it followed that the loop method is the fastest method for a network with loops, and that the loop-node method is the fastest method for a network without loops. In both the case of a network with loops and a network without loops the nodal method is the slowest method. Thus, we can conclude that the nodal method is the slowest method of the three methods, when speaking about computational time.

8.3 Convergence

In order to determine how fast each method converges to the final solution it is interesting to look at different types of networks. For the nodal network, the starting values are taken as a small value, since section 8.1.1 showed this to be a good starting value.

8.3.1 Network without any loops

First, we take a loop at a simple network without any loops. These networks are given by a straight line, as in network 1, or by a tree, as in network 7. To look at how fast the three different methods converge to the final solution we take a look at convergence plots, as shown in figure 12 and figure 13.

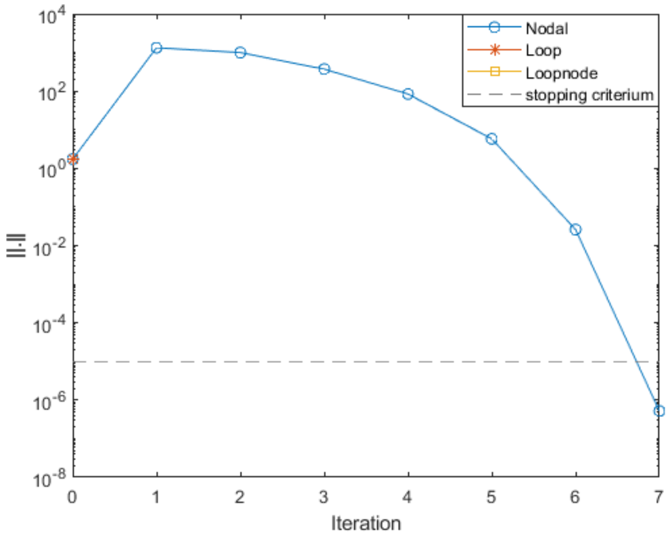


Figure 12: Convergence plot of a network without any loops: network 1

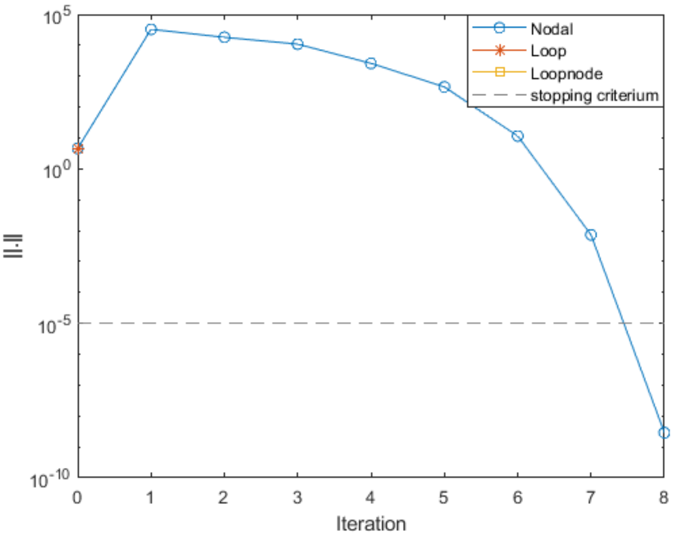


Figure 13: Convergence plot of a network without any loops: network 7

From these figures, we see that the nodal method is the only method that needs several iterations to approach the final solution. The loop method needs one iteration, whereas the loop-node method needs no iterations. This is also shown in table 9.

Table 9: Total number of iterations needed to determine the final solutions for the networks without any loops

	Number of iterations	
method	Network 1	Network 7
Nodal method	7	8
Loop method	1	1
Loop-node method	0	0

The slow convergence of the nodal method for networks without loops is also visible when looking at the number of iterations needed to reach the final solution, as shown in table 9. From these iterations it is clear that the loop-node never has to use the iterative process to approximate the final solution. This is because, since there are no loops, the error of the chord flows is empty since there are no chord flows. The loop method, on the other hand, needs the iterative process exactly one time in both networks to reach the final solution. The nodal method however needs several iterations to approximate the final solution. The amount of iterations needed for the nodal method even increases slightly for a more complicated network (i.e. network 7).

8.3.2 Network consisting of a few of loops

Next, we take a look at small networks with only a few loops, namely network 2, network 3, and network 4. These networks have one or two loops. This means that in this case the loop method and the loop-node method will also have some iterations to reach the final solution, and not only the nodal method. This is also visible in the convergence plots shown in figure 14, figure 15, and figure 16.

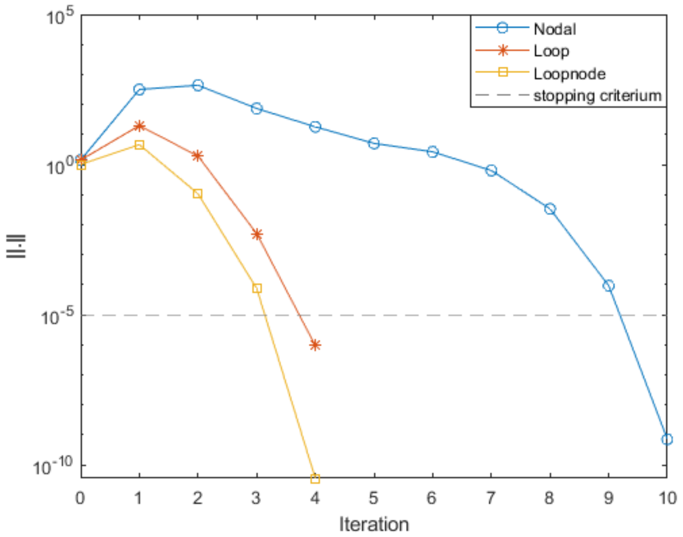


Figure 14: Convergence plot of a network with a few loops: network 2

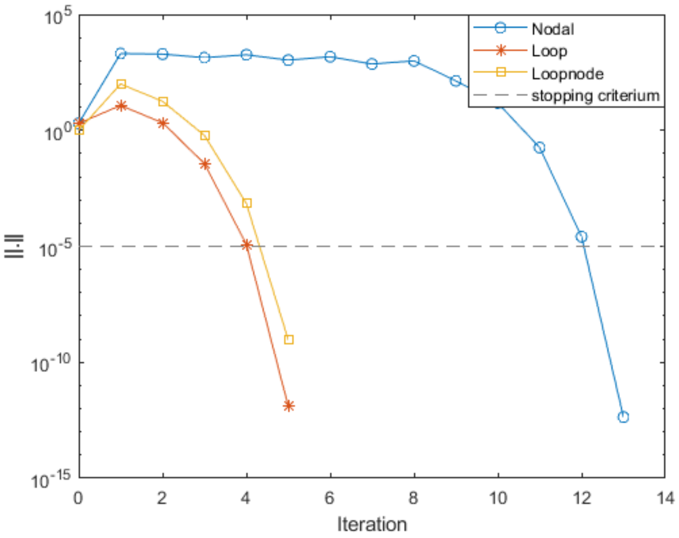


Figure 15: Convergence plot of a network with a few loops: network 3

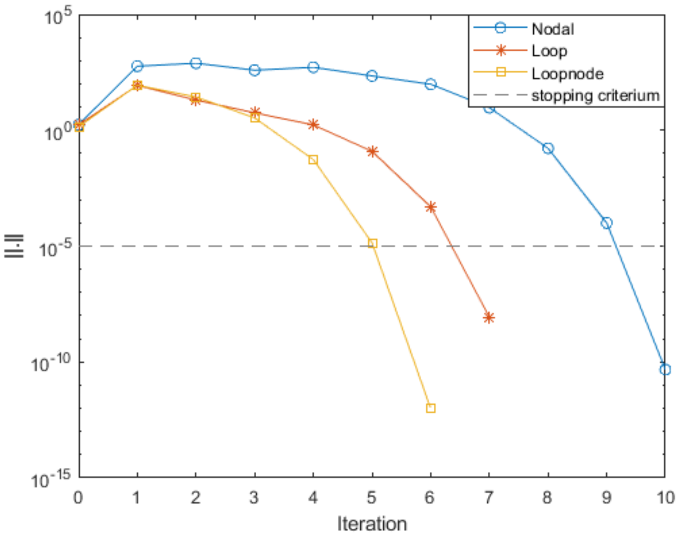


Figure 16: Convergence plot of a network with a few loops: network 4

From these figures, it is clearly visible that the nodal method needs the most iterations to reach the final solution for all three networks. Thus, the nodal method is the slowest method in reaching the final solution. The curve of the nodal method even flattens in network 3. Whereas the other two methods approach the final solution quicker. From table 10 it also becomes clear that the nodal method needs the most iterations for all three networks. The loop method and loop-node method, on the other hand, need the same number of iterations in network 2 and network 3, and the loop method only needs one iterations more in network 4. Thus, the loop method and the loop-node method are close in terms of convergence.

Table 10: Total number of iterations needed to determine the final solutions for the networks with a few loops

	Number of iterations		
method	Network 2	Network 3	Network 4
Nodal method	10	13	10
Loop method	4	5	7
Loop-node method	4	5	6

Network 4 is a network with two loops, whereas network 2 and network 3 are networks with one loop. From the figures, it is clear that in the case of network 4 the convergence of the three methods is closer to each other than in the other two networks. This is also shown in table 10. This can imply that the more loops there are in the network, the closer the nodal method is in terms of convergence to the loop method and the loop-node method.

8.3.3 Network consisting of multiple loops

Now, we take a look at networks that have multiple loops, namely network 5, which has 4 loops, and network 6, which has 15 loops. The convergence plots of these two networks are shown in figure 17 and figure 18.

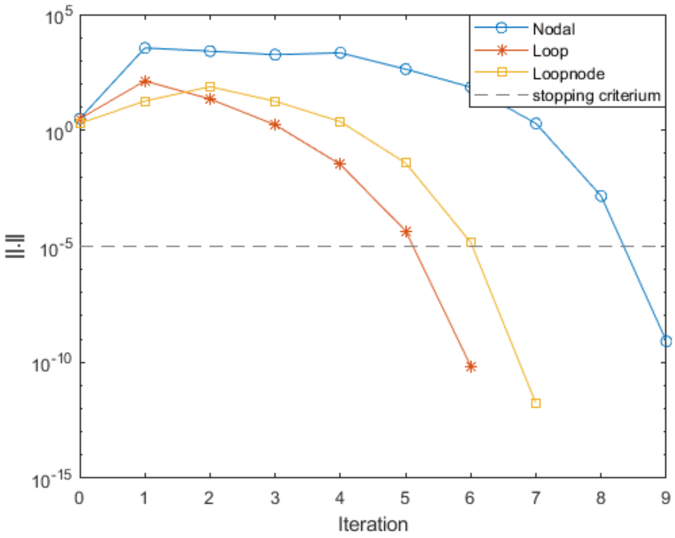


Figure 17: Convergence plot of a network with multiple loops: network 5

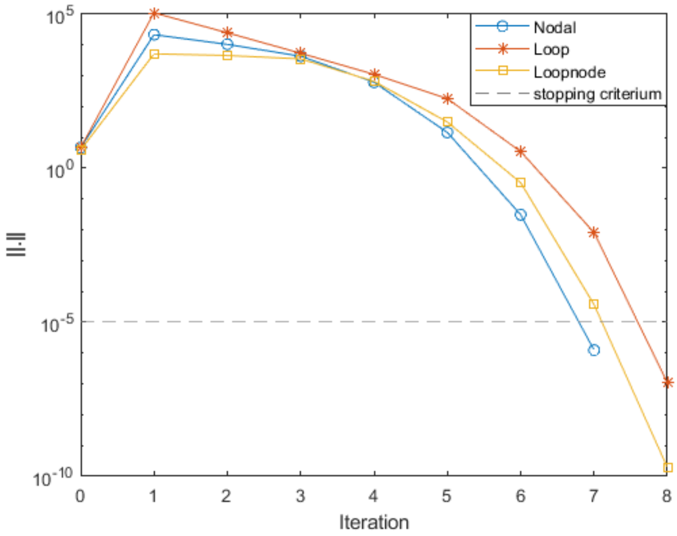


Figure 18: Convergence plot of a network with multiple loops: network 6

From these figures, we see the same as in the previous subsection 8.3.2. Namely, we see that the more loops, the closer the convergence of the three networks is, and the fewer iterations the nodal method requires. This is also shown in table 11.

Table 11: Total number of iterations needed to determine the final solutions for the networks with multiple loops

	Number of iterations	
method	Network 5	Network 6
Nodal method	9	7
Loop method	6	8
Loop-node method	7	8

We see that in network 5 the nodal method still requires the most iterations. However, in network 6 we see that the nodal method requires one iteration less than the loop method and the loop-node method. Thus, the more loops there are in a network, the better the nodal method works. However, the loop method and the loop-node method have about the same convergence as the nodal method. Thus, the nodal method does not work better than the loop method and the loop-node method.

8.3.4 Network consisting of two (or more) sources

Until this point we have only looked at networks with one source. However, in reality we often deal with networks that have multiple sources. Solving networks with two (or more) sources is more difficult than solving networks with one source. To show this difficulty, we will use network

8, which is a simple network with two sources.

For the nodal method, solving a network with multiple sources is not more difficult than solving a network with one source. Just like with other networks, this method reviews every node, with the pressures at the nodes unknown. The main difference is that the number of reference nodes is taken as two, since the number of sources is two, at which the pressure is known.

For the loop method and the loop-node method however, solving a network with multiple sources is much more difficult than solving a network with one source. In the loop method and the loop-node method the flow through the branches of the co-tree are determined in the iterative process. However, we have seen before that when there are no loops in the network, as in network 8, the iterative process is not even started. Thus, the solution these methods reach is the same as the starting value. However, this starting value does not have to be equal to the correct final solution.

To solve a network with multiple sources using the loop method or the loop-node method an addition to the method is needed. One way to do this is to add a branch to the network between the source nodes. Looking at network 8, this means adding a branch between the two sources (node 1 and 2). Then, the flow through this branch is set at zero. After doing this, there is a loop in the network and both methods can solve this network and reach the correct final solution. A consequence of this addition to the methods is that both methods will take more computational time and more computer storage, whereas this is not the case for the nodal method.

8.3.5 Non-pipe elements

Thus far we have looked at fairly simple networks but, in reality, most large networks contain non-pipe elements. These non-pipe elements can be valves, pumps, pressure regulators, and compressor stations. It is important to note how networks containing one or more of these non-pipe elements are modeled.

Valves and pumps are pressure controlling devices. Since the nodal method solves directly for such pressures, inclusion of such non-pipe elements is relatively straight forward. In the case of the loop method and the loop-node method, this is a bit more difficult. That is because in these methods one solves for flow rates, and pressures become available only after a second step (Haghighi, 1992, p.302). Often, in many types of calculations it is even justified to neglect the pressure losses through the different types of valves (Osiadacz, 1987, p.34).

In order to solve a network with compressor stations and pressure regulators, however, modifications to the methods are needed. Variables associated with compressor stations are the flow trough the compressor, the inlet pressure, and the outlet pressure. The compressors in the compressors station also have several constraints that need to be taken into account (Osiadacz, 1987, p.157). The variables associated with regulators is the outlet pressure. Furthermore, the regulator has several constraints (Osiadacz, 1987, p.158). The nodal method does not solve networks containing compressor stations or pressure regulators. There are modifications possible for the loop method and the loop-node method. Since these modifications are difficult, they will not be discussed further in this thesis. Osiadacz & Pienkosz (1988, p.1318) state that for gas networks with compressors and regulators the loop method is not as suitable as the loop-node method. Thus, the loop-node method would be the best method to use for a network with compressors or regulators.

8.3.6 Final remarks

In this section we looked at different type of networks to investigate in which case it is best to use which method when looking at convergence rate, a summary of this is shown in table 12. It became clear, as we have seen before, that the best method for a network without loops is the loop-node method, and the worst method is the nodal method. For a network with only a few loops the loop method or the loop-node method would be the best method to use. When looking at networks consisting of multiple loops however, the difference between convergence rates of the different methods became smaller. For such a network there is not one method significantly better than the other methods.

When looking at a network with two or more sources, it became clear that the nodal method is the best method to use. That is because this method does not require extra computations in order to be able to reach a correct final solution. These extra computations are needed for the loop method and the loop-node method, which makes these methods less preferred in the case of a network with multiple sources.

If a network also contains non-pipe elements, the nodal method is not as usable as the loop method and the loop-node method. This is because the nodal method can not be rewritten to solve networks with compressors. Both the loop method and the loop-node method can be rewritten to solve networks with these kind of non-pipe elements. Therefore, these methods are preferred for solving a network containing non-pipe elements. From Osiadacz & Pienkosc (1988, p.1318), it followed that the loop-node method is preferred over the loop method.

Table 12: When to use best the nodal method, the loop method, or the loop-node method, based on the findings of this thesis (+ + Suitable method, + good method, - bad method, - - not recommended method)

	Nodal method	**loop method**	**loop-node method**
Network without loops	- -	+	+ +
Network with a few loops	- -	+ +	+
Network with multiple loops	+	+	+
Network with multiple sources	+ +	-	-
Network with non-pipe elements	- -	+	+ +

9 Conclusion

In this thesis, we have looked at different methods to solve the steady-state flow problem of gas networks and district heating networks in order to determine when best to use which method. The networks are described as a directed graph. A steady-state flow analysis is performed in order to determine the pressures at the nodes and the flow trough the edges, given the pressure at the reference nodes and the loads at the load nodes. These values need to satisfy Kirchhoff's first law, Kirchhoff's second law, and a pipe flow equation. The pipe flow equation used in this thesis was the Weymouth equation. This leads to a system of nonlinear equations that needs to be solved using a iterative process. In this thesis, the Newton-Raphson method was used.

There are multiple ways to solve the steady-state flow problem. In this thesis, three methods were compared, namely the nodal method, the loop method, and the loop-node method. The nodal method focuses on the nodes in the network, and takes the pressures at the nodes as unknowns. The loop method and the loop-node method, on the other hand, focus on the pipes in the network, and therefore take the flow rate through the edges as unknowns. Both these methods solve a set of loop equations, which means that they only determine the flow trough the loops, in case of the loop method, or the flow trough the chords, in case of the loop-node method. The main difference between the loop method and the loop-node method is that the loop-node method does not actually solve the loop equations, but first transforms the loop equations to an equivalent set of nodal equations, which are then solved.

As stated before, the goal of this thesis was to determine for when best to use each method, by looking at different types of networks. First, a general analysis of the method was done. From this it followed that the nodal method and the loop-node method are better in terms of computer storage than the loop method. Furthermore, it followed that the nodal method is very sensitive to the starting values, whereas the loop method and the loop-node method are not. It also followed that the loop method is the fastest method in terms of CPU time for a network with loops, and that the loop-node method is the fastest in terms of computational time for a network without loops. When looking at the convergence rate of the different methods, the method best to use is dependent on the type of network. The findings of these investigations are used to determine for what type of network which method is best used.

For a network without any loops, it is clear from the computational time and the convergence rate that the loop-node method is the best method to use. The loop method however is a close second. The nodal method is worse than these methods, and thus is not recommended to use for a network without loops.

For a network with a few loops, it follows again from the computational time and the convergence rate that the loop method is the best method to use, with the loop-node method a close second. Again, the nodal method has worse convergence rate, and thus is not recommended to use for a network with a few loops.

For a network with multiple loops, the difference between convergence rates of the different methods became smaller. For such a network there is not one method significantly better than the other methods when talking about the convergence rates. Thus, it comes down to the other properties of the methods. An advantage of the nodal method is the good storage space usage, but it has the worst computational time, and is sensitive to the starting values. An advantage of the loop method is the good computational time, however it has the worst usage in storage space. An advantage of the loop-node method is the storage space usage, and the computational space is not the worst, but also not the best. Thus, the best method to use is dependent on what is more desired. We can not conclude which method is best to use for this type of network.

For a network with two or more sources, it is clear that the nodal method is the best method to use. This method does not require extra computations to reach a correct final solution. These extra computations are however needed for the loop method and the loop-node method, which makes these methods less preferred in the case of a network with multiple sources.

For a network that contains non-pipe elements, the nodal method is not recommended, since this method can not be rewritten to solve networks with compressors. Both the loop method and the loop-node method can be rewritten to solve networks with these kind of non-pipe elements. Therefore, these methods are preferred for solving a network containing non-pipe elements.

Furthermore, it followed that the loop-node method is preferred over the loop method.

Thus, there are clear recommendation in the usage of the three different methods. However, further research could make even more clear distinctions between the methods. Networks with two or more sources and networks containing non-pipe elements have not had an in-depth investigation in this thesis. Both these types of networks however are very likely to occur in reality for both district heating networks and gas networks. Furthermore, in this thesis we have not investigated the differences between district heating networks and gas networks. At the start we have assumed that gas networks have the same characteristics in design as district heating networks. However, the difference in the analysis is the pipe constant and the fact that for water the pressures are absolute pressures and for gas the pressures are gauge pressures. These difference could have influence on when to best use which method. This would be interesting to do further research on.

10 Bibliography

Abeysekera, M., Wu, J., Jenkins, N., Rees, M. (2016). Steady state analysis of gas networks with distributed injection of alternative gas. *Applied energy, 164*, 991-1002.

Altman, T., Boulos, P. F. (1995). Convergence of Newton method in nonlinear network analysis. *Mathematical and computer modelling, 21*(4), 35-41.

Ancona, M. A., Bianchi, M., Branchini, L., Melino, F. (2014). District heating network design and analysis. *Energy Procedia, 45*, 1225-1234.

Arsene, C. T. C., Bargiela, A., Al-Dabass, D. (2004). *Modelling and simulation of water systems based on loop equations. IJ of Simulation, 5*(1–2), 61-72.

Brkić, D. (2011). Iterative methods for looped network pipeline calculation. *Water resources management, 25*(12), 2951-2987.

Haghighi, K., Mohtar, R., Bralts, V. F., Segerlind, L. J. (1992). A linear formulation model for pipe network components. *Computers and electronics in agriculture, 7*(4), 301-321.

Hamam, Y. M., Brameller, A. (1971, November). Hybrid method for the solution of piping networks. In *Proceedings of the Institution of Electrical Engineers* (Vol. 118, No. 11, pp. 1607-1612). IET Digital Library.

Kuosa, M., Kontu, K., Mäkilä, T., Lampinen, M., Lahdelma, R. (2013). Static study of traditional and ring networks and the use of mass flow control in district heating applications. *Applied Thermal Engineering, 54*(2), 450-459.

Osiadacz, A. (1987). Simulation and analysis of gas networks.

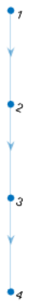
Osiadacz, A. J. (1988). Comparison of numerical methods for steady-state simulation of gas networks. *Civil Engineering Systems, 5*(1), 25-30.

Osiadacz, A. J., Pienkosz, K. (1988). Methods of steady-state simulation for gas networks, 19(7), 1311-1321.

Stevanovic, V. D., Prica, S., Maslovaric, B., Zivkovic, B., Nikodijevic, S. (2007). Efficient numerical method for district heating system hydraulics. *Energy Conversion and Management, 48*(5), 1536-1543.

A Networks

A.1 Network 1



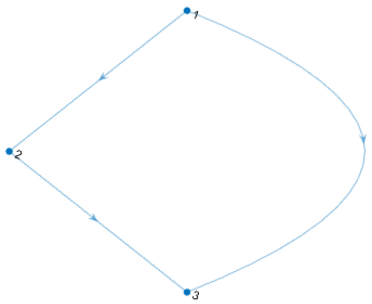
A.1.1 Parameters

```
L = [250;200;200]; % Loads (m^3/h)
p1 = 75; % Pressure source (mbar gauge)
diameter = [160;160;110]; % Diameter pipes (mm)
length = [300;300;300]; % Length pipes (m)
```

A.1.2 Solution

```
p = [75.0000;60.6155;55.1681;46.3013];
q = [650.0000;400.0000;200.0000];
```

A.2 Network 2



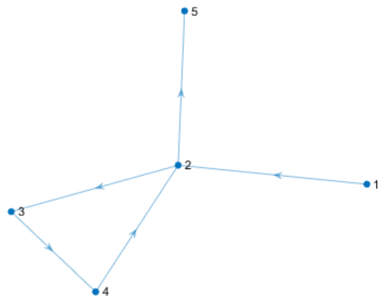
A.2.1 Parameters

```
L = [250;100]; % Loads (m^3/h)
p1 = 30; % Pressure source (mbar gauge)
diameter = [150;100;100]; % Diameter pipes (mm)
length = [680;500;600]; % Length pipes (m)
```

A.2.2 Solution

```
p = [30.0000;23.6602;23.6715];
q = [245.9905;104.0095;-4.0095];
```

A.3 Network 3



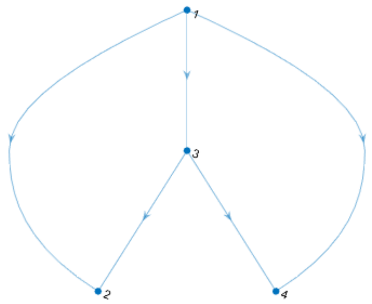
A.3.1 Parameters

```
L = [250;100;180;100]; % Loads (m^3/h)
p1 = 30; % Pressure source (mbar gauge)
diameter = [240;150;100;100;100]; % Diameter pipes (mm)
length = [420;680;500;600;340]; % Length pipes (m)
```

A.3.2 Solution

```
p = [30.0000;27.5506;24.7866;22.0512;21.7006];
q = [630.0000;162.4226;100.0000;62.4226;-117.5774];
```

A.4 Network 4



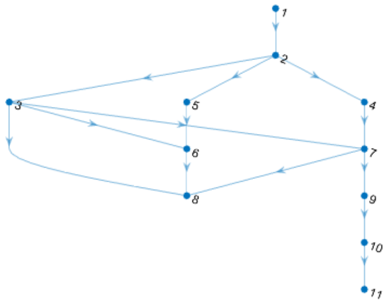
A.4.1 Parameters

```
L = [250;100;180]; % Loads (m^3/h)
p1 = 30; % Pressure source (mbar gauge)
diameter = [150;100;150;100;100]; % Diameter pipes (mm)
length = [680;500;420;600;340]; % Length pipes (m)
```

A.4.2 Solution

```
p = [30.0000;25.0354;25.7686;26.6575];
q = [217.6817;85.0475;227.2708;32.3183;−47.2708];
```

A.5 Network 5



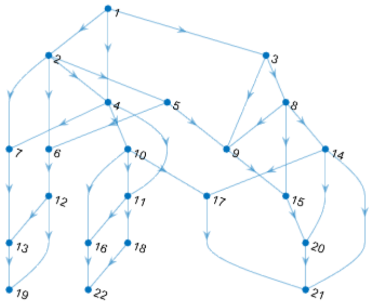
A.5.1 Parameters

```
L = [219;192;175;228;157;43.8;206;48;42;30]; % Loads (m^3/h)
p1 = 75; % Pressure source (mbar gauge)
diameter = [160;160;110;110;110;110;110;80;80;80;80;80;80;80];
    % Diameter pipes (mm)
length = [50;500;500;500;600;600;500;600;600;780;780;200;200;200];
    % Length pipes (m)
```

A.5.2 Solution

```
p = [75.0000;64.9704;43.7084;44.6198;38.9271;35.5398;36.4649;34.4635;
     26.1817;22.4798;21.8371];
q = 1.0e+03*[1.3408;0.6173;0.2367;0.2678;0.1369;0.1289;0.1595;0.0617;
     0.0398;0.0197;0.0268;0.1200;0.0720;0.0300];
```

A.6 Network 6



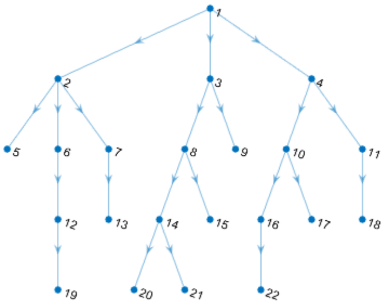
A.6.1 Parameters

```
L = 100*[17;12;17;12;20;16;15;18;10;9;16;10;10;12;20;17;8;13;9;16;15];
    % Loads (m^3/h)
p1 = 3431; % Pressure source (mbar gauge)
diameter = 100*[5;4;4;7;7;5;5;4;5;5;4;4;4;5;5;4;5;4;4;4;5;4;5;4;4;
     7;4;4;4;4;4;5;4;5;4;5]; % Diameter pipes (mm)
length = 1000*[45;15;43;52;50;40;60;17;27;52;60;42;85;35;31;27;41;53;22;
     35;32;42;45;27;27;52;31;31;17;22;28;53;27;22;27;17]; % Length pipes (m)
```

A.6.2 Solution

```
p = 1.0e+03*[3.4310;1.1710;1.4290;1.1525;1.1429;1.0375;1.0981;1.0789;
    1.1383;0.6952;0.6873;0.9898;0.9899;0.7257;0.9462;0.6177;0.6995;
    0.6154;0.9749;0.7316;0.7085;0.5992];
q = 1.0e+04*[1.1582;1.0808;0.6810;0.2258;0.2838;0.2985;0.1801;0.4245;
    0.5363;0.1672;0.2583;0.3114;0.1042;0.0596;0.2028;0.1872;-0.1967;
    0.2415;0.2297;0.2191;0.0815;0.1272;-0.0504;0.1502;0.1526;-0.0221;
    0.0648;0.0652;0.1162;-0.0482;0.0735;0.3289;0.0774;-0.1042;0.0726;
    0.1907];
```

A.7 Network 7



A.7.1 Parameters

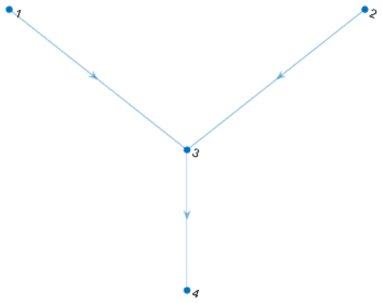
```
L = 100*[17;12;17;12;20;16;15;18;10;9;16;10;10;12;20;17;8;13;9;16;15];
   % Loads (m^3/h)
p1 = 8431; % Pressure source ((mbar gauge)
diameter = 100*[5;4;4;7;5;5;4;5;4;4;5;4;4;4;4;5;4;4;4;4;4]; % Diameter pipes (
length = 1000*[45;15;43;50;40;60;17;27;60;42;31;27;53;22;
    42;45;27;31;22;28;27]; % Length pipes (m)
```

A.7.2 Solution

```
p = 1.0e+03*[8.4310;6.6087;6.9804;3.9031;6.6037;6.2491;6.4569;6.2337;
```

```
    6.9476;1.2679;3.7644;6.1515;6.4260;5.4919;6.1975;0.6800;1.2192;
    3.7447;6.0917;5.4716;5.4100;0.6106];
q = 1.0e+04*[1.0400;0.9200;0.9600;0.1200;0.4900;0.2600;0.6200;0.1800;
    0.6200;0.1700;0.2900;0.1000;0.3500;0.1200;0.3500;0.1700;0.0800;
    0.1300;0.0900;0.1600;0.1500];
```

A.8 Network 8



No computations where made using this network.