A pipe flow network model with incorporation of a junction pressure loss model

E.A. Meulman

July 5, 2016

	Bachelor Thesis
Assessment committee:	Dr.ir. J.E. Romate
	Prof.dr. R.F. Mudde
	Prof.dr.ir. A. W. Heemink
	Dr. J.M. Thijssen
	Dr. J.A.M. de Groot
Research institute:	Delft University of Technology
Faculties:	Electrical Engineering, Mathematics and Computer Science
	Applied Sciences
Contact:	e.a.meulman@student.tudelft.nl

Contents

Abstract

This paper introduces two models. First, it introduces a model for geometric pressure loss in horizontal pipe junctions with arbitrary amounts of in-/outlets, under assumption of steady-state, incompressible, single phase flow and no wall friction. The model is shown to be in agreement with experimental data via another model.

Second, the junction model is incorporated in a pipe flow network model under the same assumptions as the junction model. The resulting pipe flow network model is compared to a pipe flow network model in which pressure loss due to junctions is neglected. The results show that for long pipes $L \gg 600$ m the addition of junction pressure has negligible influence on the total static pressure drop in the network. However, for $L < 600$ m the influence is bigger than 10%.

1 Introduction

Between 1983 and 2013 the oil consumption worldwide has increased by 44%, which resulted in a total oil consumption of $4.5 \cdot 10^9$ liter oil per day[1]. The distribution of these huge amounts of oil is done by vast *pipe flow networks*, large networks of pipes used for distribution of fluids between different points of interest. Other examples of flow networks are the water distribution network in a city, the gas distribution system in a car or the networks used for transport of chemicals in factories.

The development and construction of such a pipe flow network is a very time consuming and costly endeavor. Therefore it is paramount to be able to calculate critical properties, such as mass flow rates through pipes and static pressures at nodes in the pipe flow network, beforehand. To calculate these properties mathematical models have been developed. These models find their origin in underlying physical principles such as the laws of fluid dynamics.

In 1936 Hardy Cross published his article "*Analysis of Flow in Networks of Conduits or Conductors*" in which he describes an at that point revolutionary method for solving pipe flow networks[7]. The method is based on the two observations that both the sum of the mass flow through a node, for an incompressible fluid, and the sum of pressure difference along a loop amount to zero. Furthermore it assumes that all ingoing and outgoing characteristics are known, and then it uses the linearization of the Darcy-Weisbach equation[1] to iteratively solve for the internal characteristics. Though this method was first developed in the pre-computer era, due to its iterative nature the method was well suited for a computer implementation. In 1957, Hoag and Weinberg implemented an adapted version of the Hardy Cross method in a computer application and applied it to the water distribution network of the city Palo Alto, California[12].

The limitations of the Hardy Cross method, such as slow convergence and the limitation to closed loop systems, ignited the spark to research algorithms which were better suited for large networks. In 1963, Martin and Peters were the first researchers ensuing the present day form of pipe flow network analysis algorithms, namely producing a system of equations which characterize the network and solving that system of equations using Newton-Rhapson or some other root finding algorithm[12].

One of the returning aspects in most, if not all, of the pipe flow network models published up until now is the assumption that pressure differences at the nodes due to geometric effects of the junctions are negligible. Now if the pipes in the network are long enough this is undoubtedly true as the pressure differences due to the long pipes will dominate the characteristics of the network. But what happens in a network with short pipes, and at what pipe length will the influences of the junctions play a significant role in a network? In this paper a model is derived to determine the static pressure loss as a result of fluid flowing through a junction. This junction pressure loss model is constructed in such a way that it can be used in a pipe flow network model, such that the mass flow rate and the static pressure distribution of a pipe flow network may be calculated without neglecting the junction pressure loss. Points of special interest concerning the resulting pipe flow network model are the increase in numerical complexity, increment of accuracy of the model, in what situations the model is more accurate than the current model and the difference in information needed for both models.

Because the flow direction of the inner pipes of a network are, in most cases, not directly evident, the junction model has to be flow-direction independent. More specifically the model should still work even if the direction of flow in all branches of the junction are chosen at random, provided that the mass continuity equation[2] is satisfied. Furthermore, the model should not influence the well-posedness, in the sense of Hadamard, of the pipe flow network model. To keep the model as general as possible it

[1]The Darcy-Weisbach equation is a phenomenological equation relating the pressure drop over a pipe containing flowing fluid to the characteristics of the pipe such as length, friction factor and diameter.

[2]The mass continuity equation simply states that for incompressible flow the mass in a certain volume should always be constant. Thus there should be equal inflow as outflow.

should be valid for arbitrary cross sectional areas and for subsonic laminar- and turbulent flow. The model shall furthermore assume a steady-state situation with an incompressible single-phase fluid.

The subsequent chapter gives an introduction into pipe flow networks with a general description of such networks and the model which is currently used to solve pipe flow networks. The second chapter elaborates on junctions, with emphasis on finding a general pressure loss model and the derived model is compared to experimental data via an existing model. In the third chapter the general pressure loss model is incorporated in a pipe flow network model, and some characteristics of this model are derived. Thereafter, an implementation of the pipe flow network model with the general pressure loss model is presented in python. And lastly, the newly derived model is compared to a pipe flow network model neglecting pressure loss due to junctions, and some recommendations for a follow-up study are conferred.

2 Introduction to pipe flow networks

This section gives an introduction to pipe flow networks with first a short general description, and after that a mathematical description of such networks. The section continues with the introduction of two laws of fluid mechanics, mass continuity and the Darcy-Weisbach equation. The Darcy-Weisbach equation is derived merely to introduce the reader to the general approach of derivations in fluid mechanics. Lastly the physical laws will be used to derive a simple model to solve pipe flow networks.

But first a short description. Pipe flow networks range from simple systems, such as a watering system of a garden, to vast complex networks, such as the water distribution network of a city. In our case a pipe flow network is a network of pipes interconnected by junctions and filled with an incompressible fluid. A pipe flow network can have points of external inflow as well as points of external outflow. The aim of this text is to derive a method to calculate the mass flow and static pressure distribution of any arbitrary network given a set of boundary conditions, which could be either ingoing or outgoing mass flow rates, predefined static pressures or combinations of the two. To be able to calculate characteristics of such networks a mathematical analysis of the problem is needed.

2.1 Mathematical description

To allow for mathematical analysis of a pipe flow network, it is convenient to represent the network in a more abstract mathematical form. A convenient form for any network is a mathematical graph, as it already contains the characteristics of a network. However because the flow through the pipes of a network have a direction a *directed graph* is needed. Therefore consider a directed graph $G = (V, E)$ where the nodes $i \in V$ represent junctions, points where two or more pipes are joined together, and points of external in-/outflow, where mass flow is added or removed from the network. The arrows of the graph $(i, j) \in E$ depict the pipes of the network, where the direction of the arrow may be arbitrarily chosen. It is very important to note that the direction of the edge does not correlate in any way with the direction of the flow through the pipe it represents. This can be formally summarised as

Definition 2.1: Graph representation of a pipe flow network

A pipe flow network can be represented as directed graph $G = (V, E)$, where the nodes $i \in V$ represent the junctions or external in-/outflows and arrows $(i, j) \in E$ represent the pipes.

Now to fully describe a network using the graph representation, it has to account for the geometric characteristics of the network. As a graph, per definition, has no geometrical properties, the properties have to be added externally. Therefore we have to define the cross sectional area, $A_{i,j} \in \mathbb{R}$, and length, $L_{i,j} \in \mathbb{R}$, for each pipe $(i, j) \in E$. As the angles between the branches of a junction do not play any role in this section, they are left unconsidered for the moment. Therefore formally the network can be described by defining

$$A_{i,j} := \text{The cross sectional area of pipe } (i,j) \text{ in } \left[m^2\right] \qquad \forall (i,j) \in E \tag{2.1}$$

and

$$L_{i,j} := \text{The length of pipe } (i,j) \text{ in } [m] \qquad \forall (i,j) \in E. \tag{2.2}$$

Do note however that to get a graph which fully describes the network, the angles between pipes as well as many more characteristics should be accounted for.

As previously stated the main objective is to describe the flow through the network. To that end characteristics of the flow have to be defined on the graph representation of the network. Firstly, the characteristics of the fluid flowing through the network, such as the density ρ and the dynamic viscosity

μ have to be defined[3]. In addition, variables describing the flow itself have to be incorporated. Two important quantities concerning the flow through a network are mass flow rate, $\dot{m}$, and static pressure, P. As we will later discover due to the assumptions we make in this text $\dot{m}$ is constant throughout a pipe, therefore it is convenient to define it as a variable on the pipes of the graph. Which can formally be defined as

$$\dot{m}_{i,j} := \begin{pmatrix} \text{The mass flow rate through pipe } (i,j) \text{ in } \left[\frac{kg}{s}\right] \text{ which is positive if the} \\ \text{fluid flows from } i \text{ to } j \text{ and negative if the fluid flows in the opposite} \\ \text{direction.} \end{pmatrix} \qquad \forall (i,j) \in E \tag{2.3}$$

However, static pressure can vary throughout the pipe and is only constant in a point of of the network. As we, for now, take nodes to be infinitisimally small points, it is convenient to define static pressure as a variable on the nodes of the graph. Which can formally be defined as

$$P_i := \text{The static pressure at node } i \text{ in } \left[\frac{kg}{m \cdot s^2}\right] \qquad \forall i \in V \tag{2.4}$$

Lastly we must not forget external in-/outflow, otherwise there would be no flow at all[4]. To incorporate this in the description external in-/outflow variables, s, are defined on the nodes[5]. Formally defined by

$$s_i := \text{The fluid in-/outflow to the whole system at node } i \text{ in } \left[\frac{kg}{s}\right] \qquad \forall i \in V \tag{2.5}$$

Now, if all these distributions were known our mathematical description of the pipe flow network would be finalized. But in most real-life scenarios not all distributions consist of solely known variables, they can be a combination of both known and unknown variables. If this is the case we speak of a *partial distribution*, which only consists of numerical values for the known variables and maintains symbolic variables for unknown quantities.

To better illustrate this subsection example 2.1 drafts a mathematical description for a network of three pipes connected to a junction.

[3]We assume incompressible fluid in this text and therefore these fluid characteristics are constant throughout a network.

[4]Note that pumps are not considered in this text, and therefore friction would make the steady-state the trivial one.

[5]Note that one could simply introduce a node to create an in-/outlet and therefore this does not narrow the applicability of the description.

Example 2.1: Three pipe junction

Consider the pipe flow network in figure 2.1 filled with water at $25°C$ which implies density $997.08\frac{kg}{m^3}$ and dynamic viscosity $9.00 \cdot 10^{-4}\ Pa\,s$.[9]

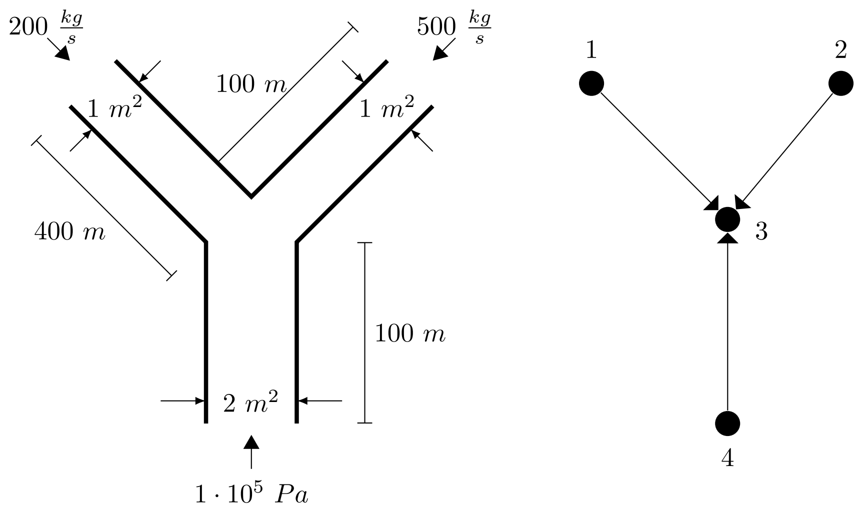


Figure 2.1: Schematic and graph representation of the example network. Note the density is $997.08\frac{kg}{m^3}$ *and the dynamic viscosity* $9.00 \cdot 10^{-4}\frac{kg}{m \cdot s}$

Using the notation introduced in section 2.1, we can deduce

$$\begin{aligned} V &= \{1, 2, 3, 4\}, \\ E &= \{(1, 3), (2, 3), (4, 3)\}, \\ A &= (A_{1,3} : 1, A_{2,3} : 1, A_{4,3} : 2), \\ L &= (L_{1,3} : 400, L_{2,3} : 100, L_{4,3} : 100), \\ \rho &= 997.08, \\ \mu &= 9.00 \cdot 10^{-4}, \end{aligned}$$

such that the mathematical description of the pipe flow network is given by $G = (V, E)$. Furthermore, the initial conditions of the flow are described by

$$\begin{aligned} s_k &= (s_1 : 200, s_2 : 500, s_3 : 0), \\ P &= (P_4 : 1 \cdot 10^5), \\ \dot{m} &= (), \end{aligned}$$

which in totality describes the situation. The unknowns of this network are

$$\{s_4, P_1, P_2, P_3, \dot{m}_{1,3}, \dot{m}_{2,3}, \dot{m}_{4,3}\}$$

and our aim is to solve these.

To solve the unknowns in a given network we need a model that describes the physical phenomena in the scenario. But in order to arrive at such a model a thorough understanding of the physical laws involved with pipe flow networks and pipe flow itself are a prerequisite. These will be addressed in the following subsection.

2.2 Physical laws

To solve the mass flow and pressure distribution of a pipe flow network, the behaviour in such a network has to be described by physical laws. In this section two such laws will be described, one fundamental law of fluid mechanics, mass continuity, and a law called the Darcy-Weisbach equation which relates the static pressure difference over a pipe to the mass flow rate through the pipe.

2.2.1 Mass continuity in a steady-state system

In fluid mechanics there are three fundamental conservation laws, mass continuity, conservation of momentum and conservation of energy. These laws describe the behaviour of quantities inside a control volume. A control volume is a, possibly time-dependent, volume in 3-dimensional space. Its closed boundary is referred to as the control surface. In this text both mass continuity as well as conservation of momentum will be encountered. The formal definition of mass continuity is

Definition 2.2: Mass continuity [2]

Given a control volume the law of **mass continuity** states that the rate of change of mass inside the control volume is exactly equal to the total mass transfer through the control surface. In mathematical form this yields

$$\frac{dM}{dt} = \dot{m}_{in} - \dot{m}_{out}, \tag{2.6}$$

where M is the total mass of the control volume in $[kg]$, t is time in $[s]$, $\dot{m}_{in}$ is the total ingoing mass flow rate crossing the control surface in $\left[\frac{kg}{s}\right]$ and $\dot{m}_{out}$ is the total outgoing mass flow rate crossing the control surface in $\left[\frac{kg}{s}\right]$

Note: The law of mass continuity holds to good approximation when the control volume does not contain a chemical reaction. However, the law does not hold when the control volume contains a nuclear reaction.

This is, in itself, a very intuitive physical phenomenon. Consider a bottle of water, the mass in that bottle of water will not change unless you pour some water out via the hole on the top. In this case the control volume is the bottle. When the bottle is closed the water mass in the bottle stays the same as no water goes out of the bottle or crosses the control surface. But when water is poured out of the bottle the mass of the water in the bottle is decreased by precisely the amount of water poured out of the bottle, which by going out of the bottle crossed the control surface.

In case of the bottle the mass in the bottle changes over time. This is a perfect example of a time dependent, or transient, system. In the case of pipe flow networks we are more interested in time independent solutions, which are more commonly called steady-state solutions. Therefore we can assume

Assumption 2.1: Steady-state

A **steady-state** situation is a time invariant situation. In other words it does not change in time. This mathematically manifests itself as

$$\frac{d\bullet}{dt} = 0 \tag{2.7}$$

where $\bullet$ can substituted with any situation defining quantity and t is time.

Now equation (2.6) in combination with equation (2.7) leaves us with

$$\dot{m}_{in} = \dot{m}_{out}, \tag{2.8}$$

which is true for all control volumes in steady-state, given the law of mass continuity holds.

Now the questions remains how this can be applied in a pipe flow network. As discussed above, a network consists of pipes and junctions. Therefore, first consider a pipe along the x-axis with a control volume from $x = a$ to $x = b$ such as schematically drawn in figure 2.2a. Consider a fluid flowing through the pipe along in the positive x-direction. The control surface consists of three parts, the circular surface at $x = a$ and $x = b$, respectively A_a and A_b, and the remaining cylindrical surface situated along the wall, A_{wall}. Now, obviously, there is no flow though A_{wall}. However there is flow through A_a and A_b, call these mass flow rates $\dot{m}_a$ and $\dot{m}_b$ respectively. Then as the fluid flows in the positive x-direction, we have inflow at $x = a$ and outflow at $x = b$, thus considering equation (2.8) we can conclude that

$$\dot{m}_a = \dot{m}_b.$$

But as a and b are arbitrarily chosen, this means that the mass flow rate is constant throughout the pipe if it is in steady-state.

In the same fashion a somewhat similar expression can be derived for a junction. Therefore consider a junction with an arbitrary amount of $N \geq 2$ branches. Now consider a spherical control volume such that the junction is contained in the control volume and the control surface only intersects the branches of the junction. This control volume is schematically drawn for a four branch junction in figure 2.2b. Now fluid can only enter or exit at the surfaces where the control surface intersect with one of the branches, call these surfaces $A_1, \dots, A_N$. The rest of the control surface does by definition not intersect with anything else.

Now we can divide the branches into two sets, one containing all branches with ingoing flow and one containing the rest of the branches. Call the set with ingoing flow I and the other set O. Formally

$$I = \{i : \text{Branch } 1 \leq i \leq N \text{ has a flow going into the junction}\}$$

and

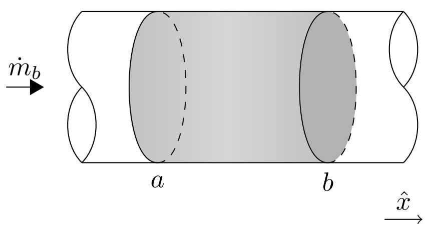
$$O = \{j : \text{Branch } 1 \leq j \leq N \text{ has either a flow out of the junction or no flow at all}\}.$$

Furthermore consider the external in-/outflow of the network at the junction, s, in $\left[\frac{kg}{s}\right]$ where s is positive with inflow and negative with outflow.

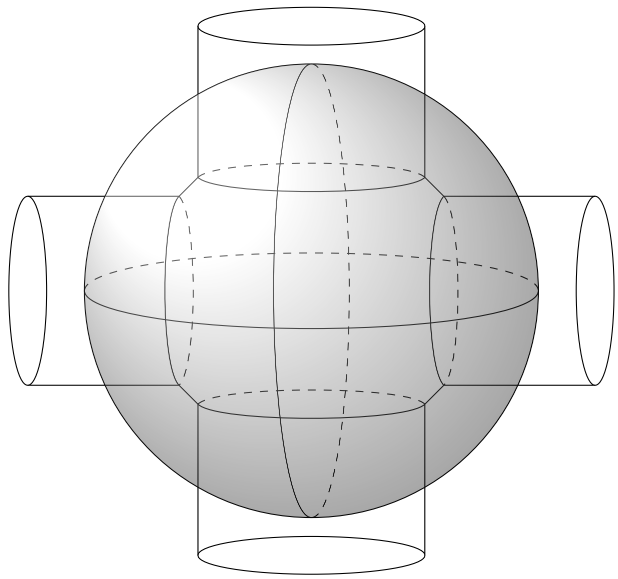
Then using equation (2.8) we get

$$s + \sum_{i \in I} \dot{m}_i = \sum_{j \in O} \dot{m}_j \tag{2.9}$$

which is a very convenient equation when solving the mass flow distribution of a pipe flow network.



(a) A schematic representation of a cylindrical pipe with its axis along the x-axis, and a control volume along the pipe from pipe $x = a$ to $x = b$.



(b) A schematic representation of a junction with 4 branches with a spherical control volume such that the junction is contained in the control volume and the control surface only intersects the branches of the junction.

Figure 2.2

2.2.2 Darcy-Weisbach equation

Just like the mass continuity equation, we can also use the law of conservation of momentum, which is defined as

Definition 2.3: Law of conservation of momentum

Given a control volume the law of **conservation of momentum** states that the rate of change of momentum in a control volume is exactly equal to the sum of the momentum transfer through the control surface **and** the forces applied to the control volume. The mathematical form yields

$$\frac{d\left(M\vec{U}\right)}{dt} = \dot{m}_{in}\vec{u}_{in} - \dot{m}_{out}\vec{u}_{out} + \sum \vec{F} \tag{2.10}$$

where $\vec{U}$ is the total velocity of the fluid in the control volume in $\left[\frac{m}{s}\right]$, $\vec{u}_{in}$ and $\vec{u}_{out}$ are the average velocities respectively in and out of the control volume in $\left[\frac{m}{s}\right]$, weighted by mass flow rate, and $\sum \vec{F}$ the sum of all forces working on the control volume in $[N] = \left[\frac{kg \cdot m}{s^2}\right]$.

Notice that the law of conservation of momentum is a vector equation, which in 3-dimensional space would produce three separate equations. In most cases we will only use one of those. Furthermore, by definition a control volume is required to obey the law of conservation of momentum.

Again, consider a pipe segment along the x-axis but this time of length L and constant diameter D, thereby making the segment cylindrical. Let a fluid flow through the segment with in-/outgoing speed respectively $u_{in}(r)$ and $u_{out}(r)$ in the positive x-direction, where r is the radial coordinate. The static pressures at the in- and outlet of the segment are respectively p_{in} and p_{out} and the pipe is completely filled with fluid. This situation is schematically drawn in figure 2.3. Furthermore, let the situation

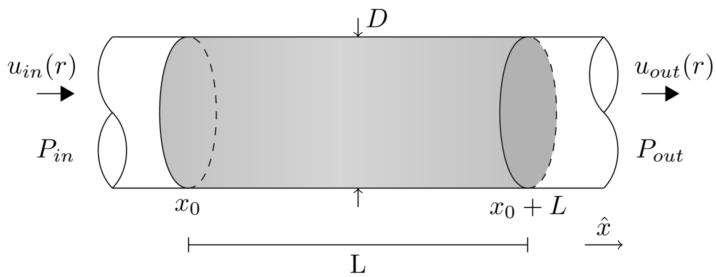


Figure 2.3: Schematic representation of a cylindrical pipe segment of length L and diameter D with in-/outgoing flow velocity respectively $u_{in}(r)$ and $u_{out}(r)$ and in- and outlet pressure respectively P_{in} and P_{out}.

be steady-state. Then according to section 2.2.1 the mass flow rate is constant throughout the pipe segment. Now because

$$\dot{m} = A \cdot \rho \cdot \overline{u}, \tag{2.11}$$

where A is the cross sectional area in $\left[m^2\right]$, ρ is the density of the fluid in $\left[\frac{kg}{m^3}\right]$ and $\overline{u}$ is the average speed in $\left[\frac{m}{s}\right]$, this results in

$$\left[A \cdot \rho \cdot \overline{u}\right]_{in} = \left[A \cdot \rho \cdot \overline{u}\right]_{out}. \tag{2.12}$$

As the diameter of the pipe was assumed constant and thereby cylindrical pipes as well we get

$$A_{in} = A_{out} = \frac{1}{4} D^2 \pi.$$

Additionally, assume that the fluid is incompressible, which is formally defined as

Assumption 2.2: Incompressible fluid

An **incompressible fluid** is a fluid with constant density. Which is mathematically described as

$$\rho = constant \tag{2.13}$$

where ρ is density in $\left[\frac{kg}{m^3}\right]$.

Then it follows that

$$\rho_{in} = \rho_{out} := \rho,$$

which, together with equation (2.12), results in

$$\overline{u}_{in} = \overline{u}_{out} := \overline{u}. \tag{2.14}$$

Now the law of conservation of momentum, equation (2.10), in the x-direction in combination with steady-state and equation (2.14) yields

$$A\rho(\overline{u}^2 - \overline{u}^2) + \sum F_x = 0,$$

thus

$$\sum F_x = 0, \tag{2.15}$$

where $\sum F_x$ is the sum of all x-components of forces working on the control volume.

To solve this equation a listing of the forces working on the control volume is needed. Fluid dynamics and general mechanics tells us there are three forces working on the control volume

Static pressure force The force as a result of static pressure p working on a surface A ($F_P = A \cdot P$)

Shear stress force The force as a result of shear stress between the wall and the fluid ($F_\tau = A_{wet} \cdot \tau$)

Gravitational force The force as a result of a height difference ($F_g = g \cdot M$)

where A_{wet} is the area of the wall that is in contact with the fluid in $\left[m^2\right]$, τ is the shear stress in $\left[\frac{N}{m^2}\right]$ and g is the gravitational constant in $\left[\frac{m}{s^2}\right]$. As the aim of this derivation is to find the static pressure difference as a result of the frictional forces due to the pipe wall we assume a horizontal pipe and therefore

$$[F_g]_x = 0, \tag{2.16}$$

as there can be no gravitational pull in the x-direction if the x-axis is perpendicular to the "downward" direction.

To find the shear stress force a development of the shear stress, τ, is needed to complete the research into the forces acting on the control volume. In theory, if we assume the fluid to be newtonian it follows[2] that

$$\tau = -\mu \frac{\partial u_x}{\partial r},$$

in which r is the radial direction orthogonal to the x-axis and μ is the dynamic viscosity of the fluid. However, for this to be useful the full flow velocity profile in the r-direction should be known, which it is not. Therefore we are left with dimension analysis. For that purpose assume

Assumption 2.3: Dimension analysis of shear stress

Assume that the shear stress τ in $\left[\frac{kg}{m \cdot s^2}\right]$ is a function of ρ in $\left[\frac{kg}{m^3}\right]$, D in $[m]$, $\overline{u}$ in $\left[\frac{m}{s}\right]$ and μ in $\left[\frac{kg}{m \cdot s}\right]$, such that

$$\tau \propto \rho^\alpha \cdot D^\beta \cdot \overline{u}^\gamma \cdot \mu^\delta \tag{2.17}$$

Dimensional analysis reveals

$$\left.\begin{array}{rc} \textbf{m:} & -3\alpha + \beta + \gamma - \delta = -1 \\ \textbf{kg:} & \alpha = 1 - \delta \\ \textbf{s:} & \gamma = 2 - \delta \end{array}\right\} \beta = -\delta$$

and thus

$$\tau \propto \rho \cdot \overline{u}^2 \cdot \left(\frac{\mu}{\rho \cdot D \cdot \overline{u}} \right)^\delta = \rho \cdot \overline{u}^2 \cdot (\mathrm{Re})^{-\delta} \tag{2.18}$$

where Re is the Reynolds number, one of the various *dimensionless quantities* in fluid mechanics.

To complete the derivation all parts have to be substituted into equation (2.15), which leaves

$$A \cdot P_{in} - A \cdot P_{out} - A_{wet} \cdot \rho \cdot \overline{u}^2 \cdot f\,(\mathrm{Re}) = 0 \tag{2.19}$$

where the flow in the positive x-direction results in the shear stress force being negative and

$$f(\mathrm{Re}) = k \cdot (\mathrm{Re})^{-\delta}$$

with $k, \delta \in \mathbb{R}$ dependent on the situation, this $f(\mathrm{Re})$ is called the *Darcy friction factor*.

Lastly, because the whole pipe is filled with water

$$\frac{A_{wet}}{A} = \frac{L \cdot D \cdot \pi}{\frac{1}{4} \cdot D^2 \cdot \pi} = \frac{4 \cdot L}{D}.$$

Which leads us to the Darcy-Weisbach equation,

$$P_{in} - P_{out} = \frac{L}{D} \cdot f(\text{Re}) \cdot \frac{1}{2} \cdot \rho \cdot \overline{u}^2 \tag{2.20}$$

where a factor 8 is absorbed into $f(\text{Re})$ for historical reasons, which concludes the derivation and gives us an expression for pressure loss in a pipe due to wall friction.
Note: It is possible to analytically derive $f(\text{Re}) = \frac{64}{\text{Re}}$ for laminar flow, however for turbulent flow $f(\text{Re})$ cannot be explicitly expressed.

2.3 Elementary pipe flow model

Using the physical laws in the previous subsection, an elementary pipe flow model can be derived. Though in order to use these laws they have to be rewritten in the variables of the mathematical description of a pipe flow network.

2.3.1 Mass continuity

The first law was mass continuity for which equation (2.9) yields the useful result. At every junction

$$s + \sum_{i \in I} \dot{m}_i = \sum_{j \in O} \dot{m}_j$$

holds[6], where s is the in-/outflow of the total network at the junction in $\left[\frac{kg}{s}\right]$, I and O are the sets of branches with in- and outgoing flow respectively and $\dot{m}_i$ is the mass flow rate through branch i in $\left[\frac{kg}{s}\right]$.

Now consider an arbitrary pipe flow network $G = (V, E)$ and an arbitrary node $k \in V$. Then s_k is defined, by equation (2.5), as the in- or outflow at node k. Furthermore, define $I_k \subset V$ as the set of nodes $i \in V$ such that there is an arrow in the graph from i to k, in graph notation

$$I_k = \{i \in V : (i, k) \in E\}. \tag{2.21}$$

In much the same way $O_k \subset V$ can be defined as the set of nodes $j \in V$ such that there is an arrow in the graph from k to j, again in graph notation

$$O_k = \{j \in V : (k, j) \in E\}. \tag{2.22}$$

It is **very important** to note that I_k and O_k are **not** the equivalent of I and O in the flow network, I_k and O_k are only dependent on the direction of edges in the graph and do not depend on the direction of flow of their elements in any way.

The direction of flow can be recovered using equation (2.3),

$$\dot{m}_{i,j} := \left(\begin{array}{l}\text{The mass flow rate through pipe } (i,j) \text{ in } \left[\frac{kg}{s}\right] \text{ which is positive if the} \\ \text{fluid flows from } i \text{ to } j \text{ and negative if the fluid flows in the opposite} \\ \text{direction.}\end{array}\right) \qquad \forall (i,j) \in E$$

[6]This is only true if the fluid in the network is *incompressible* and the situation is *steady-state* but for the rest of this section we consider these assumptions to be assumed.

implying that for each $i \in I_k$ there exists a $\dot{m}_{i,k}$ such that $\dot{m}_{i,k}$ is positive when the flow through (i,k) is towards k and negative when the fluid flow through (i,k) is away from k.

In the same fashion, for each $j \in O_k$ there exists a $\dot{m}_{k,j}$ such that $\dot{m}_{k,j}$ is positive when the flow through (k,j) is away from k and negative when the fluid flow through (k,j) is towards k.

From this analysis can be concluded that, if I, O and $\dot{m}_l$ are defined as in equation (2.9),

$$\sum_{i\in I} \dot{m}_i - \sum_{j\in O} \dot{m}_j = \sum_{i\in I_k} \dot{m}_{i,k} - \sum_{j\in O_k} \dot{m}_{k,j}.$$

and obviously $s = s_k$. Therefore, equation (2.9) is equivalent to

$$s_k + \sum_{i\in I_k} \dot{m}_{i,k} - \sum_{j\in O_k} \dot{m}_{k,j} = 0 \qquad \forall k \in V. \tag{2.23}$$

2.3.2 Darcy-Weisbach

The second physical law is the Darcy-Weisbach equation, equation (2.20), for every pipe

$$P_{in} - P_{out} = \frac{L}{D} \cdot f(\text{Re}) \cdot \frac{1}{2} \cdot \rho \cdot \overline{u}^2$$

holds, where P is the static pressure, the subscripts *in* and *out* refer to the in- and outlet respectively, L is the length of the pipe in $[m]$, D is the diameter of the pipe in $[m]$, $f(\text{Re})$ is the Darcy friction factor which is dimensionless, ρ is the density of the fluid in $\left[\frac{kg}{m^3}\right]$ and $\overline{u}$ is the mean flow velocity in $\left[\frac{m}{s}\right]$ which is constant throughout the pipe by equation (2.14).

Again, consider an arbitrary pipe flow network $G = (V,E)$ and in that network an arbitrary pipe $(i,j) \in E$. For clarity, the selected pipe is directed from node $i \in V$ to node $j \in V$ but the flow can also be from j to i if $\dot{m}_{i,j}$, as defined in (2.3), is negative. Therefore, define

$$\Delta P_{i,j} = \begin{cases} P_i - P_j & \text{if } \dot{m}_{i,j} \geq 0 \\ P_j - P_i & \text{if } \dot{m}_{i,j} < 0 \end{cases} \tag{2.24}$$

where P_i and P_j are defined by equation (2.4).

For the pipe itself the cross sectional area $A_{i,j}$, length $L_{i,j}$ are defined by definition 2.1 and ρ and μ are the density and dynamic viscosity respectively. To rewrite the left hand side of equation (2.20), some elementary relations have to be derived. Firstly the diameter D can easily be expressed in terms of $A_{i,j}$ by

$$D = 2\sqrt{\frac{A_{i,j}}{\pi}}.$$

Furthermore, using equation (2.11) simple algebra shows

$$\overline{u} = \frac{\dot{m}_{i,j}}{A_{i,j} \cdot \rho}.$$

Now by definition of the Reynolds number[2] in combination with the previous two conversions

$$\text{Re} = \frac{\rho \cdot \overline{u} \cdot D}{\mu} = 2\frac{|\dot{m}_{i,j}|}{\mu \cdot \sqrt{A_{i,j} \cdot \pi}}$$

where $|\dot{m}_{i,j}|$ is absolute because the Reynolds number is independent of flow direction.

Note that there is no closed form for $f(\text{Re})$, and the approximation of this factor is outside the scope of this section.

However, as an example, for linear flow there exists an analytica solution to $f(\text{Re})$,[2]

$$f(\text{Re}) = \frac{64}{\text{Re}} = 32\frac{\mu \cdot \sqrt{A_{i,j} \cdot \pi}}{\dot{m}_{i,j}}. \tag{2.25}$$

Using all the expressions derived above in combination with equation (2.20), we can derive

$$\Delta P_{i,j} - 8 \cdot \frac{\mu \cdot \pi}{\rho} \cdot \frac{L_{i,j} \cdot |\dot{m}_{i,j}|}{A_{i,j}^2} = 0$$

Notice $\Delta P_{i,j} \propto |\dot{m}_{i,j}|$, but the definition of $\Delta P_{i,j}$ is dependent on the sign of $\dot{m}_{i,j}$. Therefore we can rewrite the expression to

$$P_i - P_j - 8 \cdot \frac{\mu \cdot \pi}{\rho} \cdot \frac{L_{i,j} \cdot \dot{m}_{i,j}}{A_{i,j}^2} = 0 \tag{2.26}$$

which is the Darcy-Weisbach equation for laminar flow.

2.3.3 Final model

In conclusion, any pipe flow network $G = (V, E)$ with (partial) mass flow rate distribution $\dot{m}$, (partial) static pressure distribution P, (partial) in-/outflow distribution s and laminar flow can be solved using algorithm 1. Do note, the function call Solve(S) simply denotes that the system of equations, S, is solved, this can be done using a multitude of methods. Most often this is done using a numerical root-finding algorithm which approximates the distributions.

Algorithm 1 Algorithm to solve the mass flow rate, static pressure and in-/outflow distribution of a pipe flow network.

$S := \varnothing$
for $k \in V$ **do**
  $I_k := \{i \in V : (i,k) \in E\}$
  $O_k := \{j \in O : (k,j) \in E\}$
  $S := S \cup \left\{ s_k + \sum_{i \in I_k} \dot{m}_{i,k} - \sum_{j \in O_k} \dot{m}_{k,j} = 0 \right\}$
end for
for $(i,j) \in E$ **do**
  $S := S \cup \left\{ P_i - P_j - 8 \cdot \frac{\mu \cdot \pi}{\rho} \cdot \frac{L_{i,j} \cdot \dot{m}_{i,j}}{A_{i,j}^2} = 0 \right\}$
end for
return Solve(S)

Notice that in this model there are $\#V + \#E$ equations[7], which suggests it can solve a system with the same amount of unknowns. However, the three distributions, $\dot{m}$, P and s, without any prior knowledge about any of the distributions have a total amount of $2 \cdot \#V + \#E$ unknowns. Therefore, to be able to solve the distributions using the presented model, a total of $\#V$ variables in the distributions should be known. Lastly, notice that the static pressures $P_i \in P$ are only encountered in the system of equations in relative form, thus $P_i - P_j$, therefore to get an absolute distribution at least one static pressure should be known. Concluding, to solve the three distributions in a pipe flow network, a total of $\#V$

[7]Here the # set-operator signifies the cardinality operator, thus $\#A$ is equal to the amount of elements in set A.

weights in those distributions should be known beforehand one of which has to be a static pressure weight.

Lastly, to illustrate its usage the model described in this section is applied to the three pipe junction of example 2.1 in example 2.2

Example 2.2: Three pipe junction (continued)

Consider the situation from example 2.1. We can use the simple model to calculate the total distribution.
As node $3 \in V$ is most illustrative for the algorithm we will give a full account of the algorithm for that node. So first I_3 and O_3 have to be generated. As $(1,3)$, $(2,3)$ and $(3,4)$ are the pipes in E of the form $(i,3)$,

$$I_3 = \{1, 2, 4\}.$$

There are no pipes in E of the form $(3,i)$, thus

$$O_3 = \emptyset.$$

Now equation (2.23) gives us

$$\mathbf{s_3} + \dot{m}_{1,3} + \dot{m}_{2,3} + \dot{m}_{4,3} = 0 \tag{2.27}$$

where bold font signifies the weight is known. The same method for the other nodes result in

$$\mathbf{s_1} - \dot{m}_{1,3} = 0, \tag{2.28}$$
$$\mathbf{s_2} - \dot{m}_{2,3} = 0, \tag{2.29}$$
$$s_4 - \dot{m}_{4,3} = 0, \tag{2.30}$$

which concludes the analysis of the nodes.
Next up in the algorithm the pipes of the network are concerned. Consider $(1,3) \in E$, then equation (2.26) gives us

$$P_1 - P_3 - 8 \cdot \frac{\mu \cdot \pi}{\rho} \frac{\mathbf{L_{1,3}}}{\mathbf{A^2_{1,3}}} \dot{m}_{1,3} = 0 \tag{2.31}$$

The same method for the other pipes result in

$$P_2 - P_3 - 8 \cdot \frac{\mu \cdot \pi}{\rho} \frac{\mathbf{L_{2,3}}}{\mathbf{A^2_{2,3}}} \dot{m}_{2,3} = 0 \tag{2.32}$$
$$P_4 - P_3 - 8 \cdot \frac{\mu \cdot \pi}{\rho} \frac{\mathbf{L_{4,3}}}{\mathbf{A^2_{4,3}}} \dot{m}_{4,3} = 0 \tag{2.33}$$

Now our system of equations, S in the algorithm, are equations (2.27) to (2.33). The last step of the algorithm is to solve this system, as it is a linear system, solving this system is left as an exercise for the reader.
The solution is

$$s = (s_1 : 200, s_2 : 500, s_3 : 0, s_4 : -700)$$
$$P = (P_1 : 100049.31, P_2 : 100033.17, P_3 : 100031.37, P_4 : 10000.00)$$
$$\dot{m} = (\dot{m}_{1,3} : 200, \dot{m}_{2,3} : 500, \dot{m}_{4,3} : -700)$$

Note the negative $\dot{m}_{4,3}$, it is negative because the graph edge and the flow have opposite directions.

3 Pressure loss model for pipe junctions with n-branches

In the previous section a pipe flow network model was developed, however it neglects pressure loss due to junctions. To accomodate for that, this section will concentrate on pressure loss models for junctions, nodes in a pipe flow network where multiple pipes meet. In the first subsection multiple possible types of junction models are considered, and their advantages and disadvantages discussed. The second subsection will dive deeper into one of those types, the pressure coefficient model, a popular model in fluid mechanics. The subsection thereafter contains the derivation of a new model, based on a model published by Basset[3]. The section is concluded by a summary of the new model and its domain of validity.

3.1 Possible models

Throughout the years several models have been used to get a more thorough understanding of junctions in a pipe flow network. This subsection will list a variety of common or interesting models, to get a better understanding of what is important in a junction model.

3.1.1 No friction model

In the no friction model only pressure losses due to the pipe segments in the junction are considered, thereby neglecting all geometrical effects of the junction. The mass flow rates through the in-/outlets are related to eachother using mass continuity, equation (2.9).

By neglecting the geometrical effect of the junction an inaccuracy is introduced into the model, which is its main disadvantage. This neglection is justified if the network on which it is used has very long pipes. In that case the frictional losses due to the pipes will dominate the pressure losses in the network.

In addition, by completely neglecting the geometrical effects of the junction, there is no need for knowledge about the geometry of the junction. This is not its main advantage. The main advantage is that this model is computationally cheap, it introduces far fewer calculation steps than any other junction model.

3.1.2 Balance method

The balance method is a very general method to analyse fluid mechanical problems. When using this method on a junction, one would consider the volume of the junction to be a control volume. By using the conservation laws, conservation of mass, momentum and energy, the pressure drop over different in-/outlets of the junctions can be calculated.

The biggest disadvantage of this method is that every junction has to be analysed separately, and the analysis of a junction is hard to automate. The method however is physically completely sound, and it allows for very good management of assumptions made in the analysis of a junction.

3.1.3 CFD

Computational fluid dynamics, is a very popular branch of fluid mechanics that uses numerical analysis to solve problems that involve fluid flow. As already discussed with the Balance method, fluid mechanics is governed by conservation laws. These laws produce systems of partial differential equations.

Using numerical schemes the solution of these systems can be approximated[8]. This approach can also be used for pipe junctions.

Using this approach gives very detailed solution of the problem, it is the virtual equivalent of doing an experiment. It is however very computationally expensive, calculating the pressure loss over a single T-junction could easily take a few minutes using this approach. Furthermore, just like the Balance method every junction in a network would have to be analysed separately, making the approach simply impractical for use in a pipe flow network.

3.1.4 Pressure coefficient

A very popular model in the field of pipe flow networks is the pressure coefficient model, the model assumes that the pressure loss due to a component in the network can be given by

$$\Delta P = C \cdot \rho u^2,$$

where C is a dimensionless, component dependent factor called the *pressure coefficient* and ρ and u are the density and flow velocity of the fluid through the component.

The pressure coefficients are generally quite hard to obtain in closed form, therefore most coefficients are empirically obtained. Moreover in the case of closed form pressure coefficients, they are often subject to a myriad of assumptions. However, when a sufficiently general closed form pressure coefficient can be derived this approach is computationally inexpensive. However, the solution is probably less accurate than the balance method or CFD. We would like to generalize this model for use in case of a junction. Mainly because, this is the only one of the four models which does consider pressure loss due to junction, but does not involve very junction specific computations.

3.2 Pressure coefficient model

From the models examined in the previous subsection the pressure coefficient model is the only one showing potential for use with pipe flow networks. Therefore this subsection is dedicated to get a thorough understanding of that model.

First of all the model seems to find its origin in the Darcy-Weisbach equation, as derived in section 2.2.2. Although the derivation of the Darcy-Weisbach equation is specifically for flow through a pipe, it is not that much of a leap to say an equivalent derivation is possible along a streamline[8] in a different type of component.

Another point of view is considering the ρu^2 factor, this is the momentum flux[9] through the component, and one could argue that due to frictional losses momentum should be reduced by passing the component. To counter these effects the pressure downstream has to decrease, and as this is a countering effect it makes sense that the amount of difference should be proportional to the momentum flux.

However, these arguments are mere interpretations of the pressure coefficient model. The actual rationale for using this model is that it seems to be the standard in pipe flow network models, and in most texts about this subject, such as [3, 10, 13], the relation is posited without any real reasons as to why this is justified.

[8] A streamline is a line that follows the direction of the fluid flow, just like following a single particle along its path from one end of the component to the other.

[9] The flux of a quantity, simply means the quantity per unit area.

3.2.1 Coefficients in literature

Before deriving a pressure coefficient it is important to get a feeling for what variables influence such a pressure coefficient. This insight is most amply acquired by a brief chronological account of pressure coefficients for pipe junctions in literature.

In 1963, Blaisdell and Manson[5] published a paper with a comprehensive account of the research in the field before 1963. According to that publication the research of pipe junctions was limited to T-junctions, a junction consisting of a straight main pipe and a branch at variable angles. Moreover the research up until and including the work of Blaisdell and Manson consisted of mostly empirical data, and in the case of a theoretical analysis only the general form of the coefficient is derived and then fitted to experimental data. Although not supplying a purely theoretical pressure coefficient, the works described in [5] do lend some insight into possible influential quantities such as the ratio of mass flow rate, angle and ratio of cross sectional area between two in-/outlets.

In 1971 a book was published by Miller[10] titled "*Internal flow systems: A guide to losses in pipe and duct systems*", which is viewed as "the reference" for losses in pipe systems by many people in the field of pipe flow networks. Though Miller proposes a purely theoretical derivation of a pressure coefficient for T-junctions, he concludes that the resulting model does not confer with experimental data and subsides to an empirically determined relation. In the book he does however make interesting notions about negligible geometric quantities. For example, he states "*The cross-sectional shape of the pipes forming a tee has only a secondary effect compared to the flow and area ratios.*"[10] as well as "*If the components are separated by a spacer of more than 30 diameters, interaction effects are not important*"[10].

In 2001, Bassett et al. proposed a closed form pressure coefficient for flow through a T-junction[4], based on balance model analyses of a general T-junction with a variable angle between the main pipe and the branch and a variable cross sectional area ratio between the main pipe and the branch. The general junctions are analysed separately for all possible flow direction configurations, which for a junction with three in-/outlets are 6 possible configurations. For each of these, two pressure coefficients are derived, one for each streamline[10]. Resulting in a total of 12 different pressure coefficients. The generalisation of this model to a junction with an arbitrary amount of in-/outlets would lead to an exponential increase of pressure coefficients, which is simply impractical. However, the model proposed by Bassett et al. is supported by experimental data, and therefore it makes for a good reference model to check validity of any other model in the case of T-junctions.

Lastly in 2003, Bassett et al. proposed a generalisation of the model published in 2001[3]. The new model can be used to calculate the pressure loss for flow through a junction with an arbitrary amount of in-/outlets. It is based on a single closed form pressure coefficient for pressure loss from the inlet with highest mass flow to an arbitrary outlet. The pressure coefficient is a function of the mass flow ratio, cross sectional area ratio and angle between the inlet and the outlet. Furthermore the model assumes the pressures at all inlets to be equal, which finalises the model. Do note that this model does have its drawbacks. For one, it completely ignores any incoming flows other than that of the most significant inlet[11]. Furthermore, the assumption that the pressure is equal at all inlets is not supported by experimental data, such as the cross junction data of Sharp[15]. The model can however function as a good basis for a more advanced model.

3.3 The new junction model

In this section a new junction model will be developed. This junction model is heavily based on the model proposed by Bassett et al. in 2003. The first subsection will aim to derive a more general

[10] In this case a streamline is a flow from an inlet to an outlet

[11] The inlet with the highest mass flow rate

version of the pressure coefficient first derived by Bassett et al.[3] After the derivation additions to the model will be proposed to improve the model. After that the newly derived model will be compared to the model Bassett et al. published in 2001, which shows agreement with experiment.

3.3.1 Derivation of pressure coefficient

Consider an arbitrary junction with $n \in \mathbb{N}$ inlets and $m \in \mathbb{N}$ outlets. Then two sets can be defined I the set of inlets and O the set of outlets, where in the special case of no flow we speak of an inlet. Now for any combination of an inlet and outlet, $(i, j) \in I \times O$, a *stagnation pressure coefficient*, $K_{i,j}$, is introduced, defined by

$$K_{i,j} = \frac{(p_o)_i - (p_o)_j}{\frac{1}{2}\rho_i u_i^2} = \frac{P_i + \frac{1}{2}\rho_i u_i^2 - \left(P_j + \frac{1}{2}\rho_j u_j^2\right)}{\frac{1}{2}\rho_i u_i^2}, \tag{3.1}$$

where p_o is stagnation pressure in $\left[\frac{kg}{m \cdot s^2}\right]$, ρ is the density of the fluid in $\left[\frac{kg}{m^3}\right]$, u is the flow velocity in $\left[\frac{m}{s}\right]$ and P is the static pressure in $\left[\frac{kg}{m \cdot s^2}\right]$.

To continue the derivation, the fluid is assumed to be incompressible. Specifically

Assumption 3.1: Incompressible fluid

An **incompressible fluid** is a fluid with constant density. Which is mathematically described as

$$\rho = constant \tag{3.2}$$

where ρ is density in $\left[\frac{kg}{m^3}\right]$.

Now (3.1) can be rewritten as follows

$$K_{i,j} = \frac{P_i + \frac{1}{2}\rho u_i^2 - \left(P_j + \frac{1}{2}\rho, u_j^2\right)}{\frac{1}{2}\rho u_i^2},$$

$$K_{i,j} = \frac{P_i - P_j}{\frac{1}{2}\rho u_i^2} + 1 - \frac{u_j^2}{u_i^2},$$

$$P_i - P_j = \frac{1}{2}\rho u_i^2 \left(K_{i,j} - 1 + \frac{u_j^2}{u_i^2}\right), \tag{3.3}$$

which leaves an expression for the pressure difference between the in- and outlets. Due to practicalities it is more convenient to rewrite the expression in terms of mass flow ratio defined as

$$q_{i,j} = \frac{\dot{m}_j}{\dot{m}_i}, \tag{3.4}$$

and cross-sectional area ratio defined as

$$\psi_{i,j} = \frac{A_i}{A_j}, \tag{3.5}$$

where $\dot{m}$ is the mass flow rate in $\left[\frac{kg}{s}\right]$ and A the cross-sectional area in $\left[m^3\right]$. Now using the relation between mass flow rate and flow velocity[2]

$$\dot{m} = \rho A u, \tag{3.6}$$

and using equation (3.2) we can express u_j in terms of u_i,

$$
\begin{aligned}
u_j &= \frac{\dot{m}_j}{\rho \cdot A_j} \frac{u_i}{u_i}, \\
&= \frac{A_i}{A_j} \frac{\dot{m}_j}{\dot{m}_i} u_i, \\
u_j &= q_{i,j} \psi_{i,j} u_i,
\end{aligned} \tag{3.7}
$$

which leaves us with

$$
\frac{u_j}{u_i} = q_{i,j} \psi_{i,j}, \tag{3.8}
$$

with this (3.3) results in

$$
\begin{aligned}
P_i - P_j &= \frac{1}{2} \rho \frac{u_j^2}{q_{i,j}^2 \psi_{i,j}^2} \left(K_{i,j} - 1 + q_{i,j}^2 \psi_{i,j}^2 \right), \\
P_i - P_j &= \frac{1}{2} \rho u_j^2 \left(\frac{K_{i,j}}{q_{i,j}^2 \psi_{i,j}^2} - \frac{1}{q_{i,j}^2 \psi_{i,j}^2} + 1 \right).
\end{aligned}
$$

Now, the pressure coefficient $C_{i,j}$ is defined as

$$
C_{i,j} = \frac{P_i - P_j}{\rho u_j^2} = \frac{1}{2} \left(\frac{K_{i,j}}{q_{i,j}^2 \psi_{i,j}^2} - \frac{1}{q_{i,j}^2 \psi_{i,j}^2} + 1 \right), \tag{3.9}
$$

which leaves us with an expression for the pressure coefficient between an inlet and an outlet of a junction given the stagnation pressure coefficient, $K_{i,j}$.

Now we are left with the task of finding the stagnation pressure coefficient. Therefore, take a random combination of an inlet and an outlet, $(i, j) \in I \times O$, then our goal is to find the stagnation pressure difference between the inlet, i, and the outlet, j. Both i and j are schematically drawn in figure 3.1, where subscript "others" signifies all in-/outlets of the junction other than i and j. Note that the angle between i and j is $0 < \theta_{i,j} \leq \pi$, because of this angle and the momentum of the fluid a recirculation area is induced, which is depicted by D-R-P in figure 3.1. Furthermore $q_{i,j}$ is defined as in (3.4), and $0 < \xi < 1$ is the ratio between the minimal free flow area, the cross-sectional area of flow where the pipe is most restricted by the recirculation area, $A_{RR'}$, and the total pipe area, A_j, thus

$$
\xi = \frac{A_{RR'}}{A_j}. \tag{3.10}
$$

Now to calculate the stagnation pressure coefficient, $K_{i,j}$, as defined by (3.1), we consider two control volumes D-D'-R'-R and R-R'-P'-P separately.

3.3.2 Control volume: D-D'-R'-R

Consider the control volume D-D'-R'-R, note that D'-R' is a wall and D-R is a boundary to the recirculation area which imply there is no mass transfer through these areas. Therefore there can only be mass flow through D-D' and R-R' and mass continuity (2.8) states that $\dot{m}_{RR'}$ and $\dot{m}_{DD'}$ are equal in size. To be precise this only holds in a steady-state which we will assume

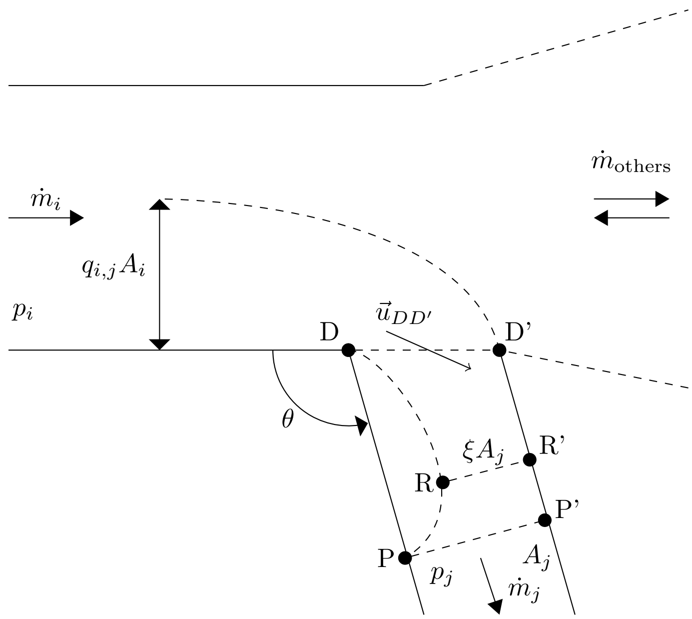


Figure 3.1: Schematic representation of a junction of N-pipes

> **Assumption 3.2: Steady-state**
>
> A **steady-state** situation is a time invariant situation, in other words it does not change in time. This mathematically manifests itself as
>
> $$\frac{d\bullet}{dt} = 0, \tag{3.11}$$
>
> where $\bullet$ can substituted with any situation definining quantity and t is time.

Now again using equation (2.8) we can deduce

$$\dot{m}_{RR'} = \dot{m}_{DD'} = \dot{m}_j. \tag{3.12}$$

Furthermore, because D-D'-R'-R is a closed control volume the law of conservation of momentum should hold, which is defined in definition 2.10. The law of conservation of momentum is a vector equation, which in our 2-dimensional representation[12] leads to two equations. Only one of them is needed to find $K_{i,j}$. In the rest of this subsection we will consider the law of momentum conservation in the direction of mass flow through outlet j, $\hat{\dot{m}}_j$, which in combination with a steady-state and equation (3.12) yields

$$\left(\dot{m}_j\vec{u}_{DD'} - \dot{m}_j\vec{u}_{RR'} + \left(\sum \vec{F}\right)_{DD'R'R}\right) \cdot \hat{\dot{m}}_j = 0, \tag{3.13}$$

where $\vec{u}_{DD'}$ and $\vec{u}_{RR'}$ are the velocity vectors of the flow through D-D' and R-R' respectively and $(\sum \vec{F})_{DD'R'R}$ represents the sum of the forces acting on the control volume. Do note the only way mass can enter or exit the control volume is via D-D' and R-R' and the dot-product at the end of the left hand side of the equation ensures only the components in the $\hat{\dot{m}}_j$ direction are taken into account.

[12] Note that we can use a 2-dimensional representation because according to Miller the effect of cross sectional shape is negligible[10]

To solve this equation the magnitude and the direction of the flow velocities crossing D-D' and R-R' have to be known. Bassett et al. state that experimental observations made using T-junctions suggest that the velocity magnitude of the flow entering j from i, thereby crossing D-D', is equal to u_i[3]. Furthermore, Bassett et al. suggest that the direction of $\vec{u}_{DD'}$ makes an angle of $\frac{\pi-\theta}{4}$ with the "*horizontal*"[3], meaning $\hat{\vec{m}}_i$. Thus assume

Assumption 3.3: Magnitude and direction of $\vec{u}_{DD'}$

The magnitude of $\vec{u}_{DD'}$ is equal to $|u_i|$ and the direction of $\vec{u}_{DD'}$ makes an angle of $\frac{\pi-\theta}{4}$ with $\hat{\vec{m}}_i$. The approximate direction is drawn in figure 3.1.

which implies

$$\vec{u}_{DD'} \cdot \hat{\vec{m}}_j = u_i \cos\left(\frac{3}{4}(\pi - \theta)\right). \tag{3.14}$$

To determine the direction of $\vec{u}_{RR'}$ note that the at R-R' there is no compression or expansion of free flow area, therefore the flow direction is parallel to the outlet axis, thus

$$\vec{u}_{RR'} \cdot \hat{\vec{m}}_j = u_R, \tag{3.15}$$

where u_R represents the magnitude of $\vec{u}_{RR'}$.

Now to solve equation (3.13) we need to analyse the forces acting on the control volume. To significantly simplify the derivation we make four assumptions

Assumption 3.4: Horizontal junctions

The junction is assumed to lie in a horizontal plane, thereby neglecting any gravitational influences.

Assumption 3.5: No wall friction

The junction is assumed to have no wall friction, the model only accounts for geometrical effects of the junction.

Assumption 3.6: Static pressure in recirculation area D-R-P

The static pressure in D-R-P is assumed to be constant and equal the pressure along R-R', P_R. **Rationale:** As there is no mass transfer through D-R-P, the pressure has to be constant. A pressure difference would lead to flow towards the point of lower pressure implying mass transfer through D-R-P. Furthermore, as R-R' is the point of smallest free flow area, $P_{DRP} = P_R$. If not R-R' would either become smaller or larger as the pressure difference would move the position of R.

Assumption 3.7: Static pressure at D-D'

The static pressure at D is equal to the pressure at the inlet,

$$P_D = Pi.$$

Furthermore, the static pressure at D' is equal to the stagnation pressure at the inlet. Thus

$$P_{D'} = (p_o)_i = P_i + \frac{1}{2}\rho u_i^2$$

where P_i is the static pressure at i and u_i is the flow speed at i. Lastly, the mean static pressure over D-D' is equal to the mean of the static pressure at D and at D'. Thus

$$P_{DD'} = \frac{P_D + P_{D'}}{2} = P_i + \frac{1}{4}\rho u_i^2 \tag{3.16}$$

Rationale: As there is no frictional loss and no hindrance to the flow there is no energy conversion from or to potential energy between the inlet and D, furthermore all gravitational influences are neglected thus there can be no pressure difference between those points.
At point D' all flow from the inlet is halted, Bernoulli's principle then states that the kinetic energy is transformed to potential energy. As we are talking about a horizontal system the potential can only be stored in static pressure.
Considering the first two parts of this rationale it is not a leap to assume the amount of conversion of kinetic energy to potential energy goes linearly between point D and D' which would account for the choice of $P_{DD'}$.

Due to the first two assumptions gravitational and frictional forces are neglected, leaving only the pressure terms to contribute to the sum of forces. As D-R' is parallel to $\hat{\dot{m}}_j$ the pressure force due to D-R' is perpendicular to $\hat{\dot{m}}_j$ and therefore the inner product is zero. Now note that D-R and R-R' have the same static pressure by the third assumption. Furthermore, considering figure 3.1 a rather straight forward geometrical argument implies

$$P_R \vec{A}_{DR} \cdot \hat{\dot{m}}_j = -P_R(1-\xi)A_j, \tag{3.17}$$

where $\vec{A}_{DR}$ has a direction perpendicular to D-R and magnitude A_{DR}. Therefore, because R-R' is perpendicular to $\hat{\dot{m}}_j$ combining the pressure forces due to D-R and R-R' yields

$$P_R\left(\vec{A}_{DR} + \vec{A}_{RR'}\right) \cdot \hat{\dot{m}}_j = -P_R A_j\left((1-\xi)+\xi\right) = -P_R A_j. \tag{3.18}$$

Revisiting figure 3.1 in combination with the fourth assumption results in

$$P_{DD'}\vec{A}_{DD'} \cdot \hat{\dot{m}}_j = \left(P_i + \frac{1}{4}\rho u_i^2\right) A_j. \tag{3.19}$$

Note the subtle geometrical argument that $\vec{A}_{DD'} \cdot \hat{\dot{m}}_j = A_j$, which is best understood by closely inspecting figure 3.1.

To finalize the analysis of forces equations (3.18) and (3.19) are used to form

$$\left(\sum \vec{F}\right)_{DD'R'R} \cdot \hat{\dot{m}}_j = \left(P_i + \frac{1}{4}\rho u_i^2 - P_R\right) A_j. \tag{3.20}$$

Using equations (3.14), (3.15) and (3.20) we are able to fill in (3.13), resulting in

$$\dot{m}_j\left(u_i \cos\left(\frac{3}{4}(\pi-\theta)\right) - u_R\right) + \left(P_i + \frac{1}{4}\rho u_i^2 - P_R\right) A_j = 0. \tag{3.21}$$

To further solve this equation we need another assumption. Bassett et al. proposes that, "*as flow converges (in the region D-D'-R'-R), the stagnation pressure remains almost constant*"[3] and thereby that the overall stagnation pressure loss occurs as the flow diverges, in R-R'-P'-P. Therefore assume

Assumption 3.8: No stagnation pressure loss in D-D'-R'-R

There is no stagnation pressure loss in D-D'-R'-R, thus

$$(p_o)_{DD'} = (p_o)_{RR'} \Rightarrow P_i + \frac{3}{4}\rho u_i^2 = P_R + \frac{1}{2}\rho u_R^2, \tag{3.22}$$

where the $\frac{3}{4}$ factor on the right hand side is a consequence of equation (3.16).

Using equation (3.22) we can derive an expression for P_R

$$P_R = P_i + \frac{3}{4}\rho u_i^2 - \frac{1}{2}\rho u_R^2. \tag{3.23}$$

Substitution in equation (3.21) yields

$$\begin{aligned}
\dot{m}_j \left(u_i \cos\left(\frac{3}{4}(\pi - \theta)\right) - u_R \right) + \left(P_i + \frac{1}{4}\rho u_i^2 - P_i - \frac{3}{4}\rho u_i^2 + \frac{1}{2}\rho u_R^2 \right) A_j &= 0, \\
A_j \rho u_j \left(u_i \cos\left(\frac{3}{4}(\pi - \theta)\right) - u_R \right) + \left(-\frac{1}{2}u_i^2 + \frac{1}{2}u_R^2 \right) A_j \rho &= 0, \\
u_j \left(u_i \cos\left(\frac{3}{4}(\pi - \theta)\right) - u_R \right) - \frac{1}{2}u_i^2 + \frac{1}{2}u_R^2 &= 0. \qquad (3.24)
\end{aligned}$$

To further analyse this expression, two conversions have to be considered. The first one was already used in the derivation, namely equation (3.8) or

$$u_i = \frac{u_j}{\psi_{i,j} q_{i,j}}. \tag{3.25}$$

The other conversion is derived using equation (3.12) as follows

$$\begin{aligned}
u_R &= \frac{\dot{m}_R}{\xi A_j \rho}, \\
&= \frac{A_j \rho u_j}{\xi A_j \rho}, \\
u_R &= \frac{u_j}{\xi}. \qquad (3.26)
\end{aligned}$$

Substitution of these relations into equation (3.24) yields

$$\begin{aligned}
u_j \left(\frac{u_j}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right) - \frac{u_j}{\xi} \right) - \frac{1}{2}\frac{u_j^2}{\psi_{i,j}^2 q_{i,j}^2} + \frac{1}{2}\frac{u_j^2}{\xi^2} &= 0, \\
\frac{1}{2}u_j^2 \left(\frac{2}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right) - \frac{2}{\xi} - \frac{1}{\psi_{i,j}^2 q_{i,j}^2} + \frac{1}{\xi^2} \right) &= 0, \\
\frac{1}{\xi^2} - \frac{2}{\xi} + \frac{2}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right) - \frac{1}{\psi_{i,j}^2 q_{i,j}^2} &= 0, \\
\left(1 - \frac{1}{\xi}\right)^2 - 1 + \frac{2}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right) - \frac{1}{\psi_{i,j}^2 q_{i,j}^2} &= 0
\end{aligned}$$

which leads us to the end result of our endeavour with this control volume

$$\left(1 - \frac{1}{\xi}\right)^2 = 1 + \frac{1}{\psi_{i,j}^2 q_{i,j}^2} - \frac{2}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right). \tag{3.27}$$

3.3.3 Control volume: R-R'-P'-P

Now that we have an expression for $\left(1 - \frac{1}{\xi}\right)^2$ we can use this to derive an expression for the stagnation pressure loss, and with that an expression for $K_{i,j}$. First of all, again consider conservation of momentum in the $\hat{\dot{m}}_j$ direction. In much the same way as equation (3.13) for R-R'-P'-P this yields,

$$\left(\dot{m}_j \vec{u}_{RR'} - \dot{m}_j \vec{u}_{PP'} + \left(\sum \vec{F}\right)_{RR'P'P}\right) \cdot \hat{\dot{m}}_j = 0. \tag{3.28}$$

Using equation (3.15)

$$\dot{m}_j \vec{u}_{RR'} \cdot \hat{\dot{m}}_j = \dot{m}_j u_R, \tag{3.29}$$

and as P-P' is perpendicular to $\hat{\dot{m}}_j$, and there are no expansion effects at P-P', the flow speed is parallel to $\hat{\dot{m}}_j$. Therefore

$$\dot{m}_j \vec{u}_{PP'} \cdot \hat{\dot{m}}_j = \dot{m}_j u_j. \tag{3.30}$$

Which leaves the task of finding $\left(\sum \vec{F}\right)_{RR'P'P}$. Due to assumptions 3.4 and 3.5, again only pressure forces are taken into account. As in this case R'-P' is parallel with the $\hat{\dot{m}}_j$ direction, there is no force component due to R'-P'.

Now using the same geometrical argument for R-P as used for D-R results in

$$P_R \vec{A}_{RP} \cdot \hat{\dot{m}}_j = P_R(1 - \xi) A_j, \tag{3.31}$$

and using the same argument for R-R' as used in the previous subsection results in

$$P_R \vec{A}_{RR'} \cdot \hat{\dot{m}}_j = P_R \xi A_j. \tag{3.32}$$

Do note that $\vec{A}_{RR'}$ in this control volume has its direction exactly opposite to its direction in the D-D'-R'-R control volume, this is because area vectors are chosen to be directed inward in case of a control volume. All in all this leads to

$$\left(P_R \vec{A}_{RP} + P_R \vec{A}_{RR'}\right) \cdot \hat{\dot{m}}_j = P_R(1 - \xi) A_j + P_R \xi A_j = A_j P_R, \tag{3.33}$$

just like in the D-D'-R'-R control volume.

Furthermore, the fact that D-D' is perpendicular to the $\hat{\dot{m}}_j$ direction and situated at the outlet results in

$$P_j \vec{A}_{DD'} \cdot \hat{\dot{m}}_j = -P_j A_j. \tag{3.34}$$

Using the last two equations and the fact that R'-P' does not supply a pressure force term, we can conclude that

$$\left(\sum \vec{F}\right)_{RR'P'P} \cdot \hat{\dot{m}}_j = A_j (P_R - P_j). \tag{3.35}$$

Substitution of this expression and equations (3.29) and (3.30) in equation (3.28) yields

$$\begin{aligned} \dot{m}_j (u_R - u_j) + A_j (P_R - P_j) = 0, \\ P_R - P_j = \rho u_j (u_j - u_R), \end{aligned}$$

where to make the step the definition of $\dot{m}_j = A_j \rho u_j$ is used.

To retrieve the stagnation pressure coefficient, consider its definition (3.1).

$$K_{i,j} = \frac{P_i + \frac{1}{2}\rho u_i^2 - P_j - \frac{1}{2}\rho u_j^2}{\frac{1}{2}\rho u_i^2}. \tag{3.36}$$

Now, as a result of assumption 3.8, that all stagnation pressure loss between i and j is the stagnation pressure loss in R-R-P'-P, we can rewrite

$$\begin{aligned}
K_{i,j} &= \frac{P_R + \frac{1}{2}\rho u_R^2 - P_j - \frac{1}{2}\rho u_j^2}{\frac{1}{2}\rho u_i^2}, \\
K_{i,j} &= \frac{\rho u_j (u_j - u_R) + \frac{1}{2}\rho u_R^2 - \frac{1}{2}\rho u_j^2}{\frac{1}{2}\rho u_i^2}.
\end{aligned}$$

Combining the conversions of the previous subsection, equations (3.25) and (3.26), results in

$$\begin{aligned}
K_{i,j} &= \frac{\rho u_j (u_j - \frac{u_j}{\xi}) + \frac{1}{2}\rho \frac{u_j^2}{\xi^2} - \frac{1}{2}\rho u_j^2}{\frac{1}{2}\rho \frac{u_j^2}{\psi_{i,j}^2 q_{i,j}^2}}, \\
&= 2\psi_{i,j}^2 q_{i,j}^2 \frac{\left(1 - \frac{1}{\xi} + \frac{1}{2}\frac{1}{\xi^2} - \frac{1}{2}\right)\rho u_j^2}{\rho u_j^2}, \\
&= \psi_{i,j}^2 q_{i,j}^2 \left(1 - \frac{2}{\xi} + \frac{1}{\xi^2}\right), \\
K_{i,j} &= \psi_{i,j}^2 q_{i,j}^2 \left(1 - \frac{1}{\xi}\right)^2.
\end{aligned}$$

Note the $(1 - \frac{1}{\xi})^2$ factor, substituting equation (3.27) yields

$$K_{i,j} = \psi_{i,j}^2 q_{i,j}^2 \left(1 + \frac{1}{\psi_{i,j}^2 q_{i,j}^2} - \frac{2}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right)\right), \tag{3.37}$$

which concludes our encounter with this last control volume.

Now that we have an expression for $K_{i,j}$ we can easily find an expression for the pressure coefficient using equation (3.9)

$$\begin{aligned}
C_{i,j} &= \frac{1}{2}\left(\frac{K_{i,j}}{q_{i,j}^2 \psi_{i,j}^2} - \frac{1}{q_{i,j}^2 \psi_{i,j}^2} + 1\right), \\
C_{i,j} &= \frac{1}{2}\left(\frac{\psi_{i,j}^2 q_{i,j}^2 \left(1 + \frac{1}{\psi_{i,j}^2 q_{i,j}^2} - \frac{2}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right)\right)}{q_{i,j}^2 \psi_{i,j}^2} - \frac{1}{q_{i,j}^2 \psi_{i,j}^2} + 1\right), \\
C_{i,j} &= \frac{1}{2}\left(1 + \frac{1}{\psi_{i,j}^2 q_{i,j}^2} - \frac{2}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right) - \frac{1}{q_{i,j}^2 \psi_{i,j}^2} + 1\right), \\
C_{i,j} &= 1 - \frac{1}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right).
\end{aligned} \tag{3.38}$$

Which finally gives us an expression for the pressure coefficient and concludes the derivation based on the work of Bassett et al.[3].

3.3.4 Flow independent generalization

Now that we have an expression, we can analyse it to see when it is valid. First of all taking a close look at figure 3.1 tells us immediately that by choice of control volume there is no room in the outlet for any other inflow than that of the observed inlet. However, in a pipe flow network it is quite common to have combining flow. Thus, it is critical to account for multiple inflows into one outlet.

Therefore, consider the control volume in figure 3.2. In the figure we look solely at i and j again, where i is an inlet and j is an outlet. But we assume that all flow from i goes to j. Thereby, it follows that

$$\dot{m}_j \geq \dot{m}_i,$$

which, by equation (3.4), implies that

$$q_{i,j} > 1.$$

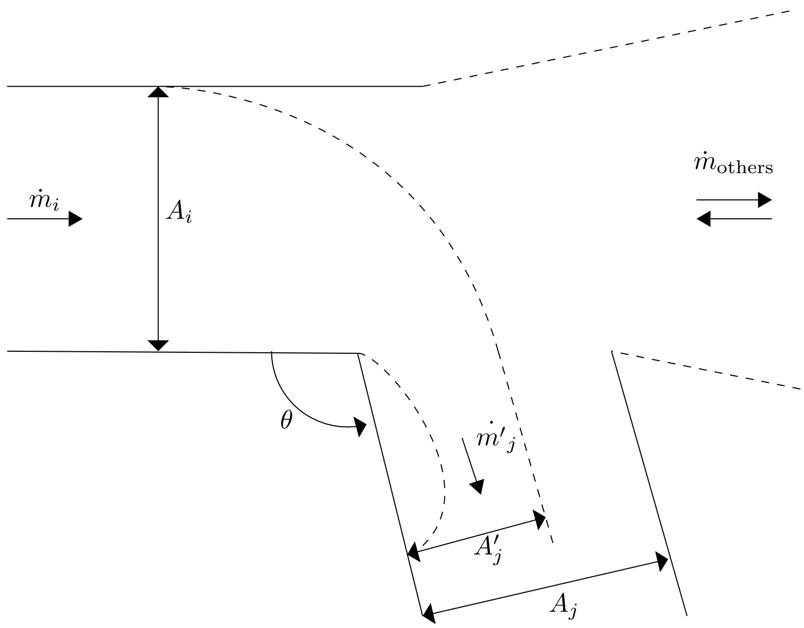


Figure 3.2: A schematic representation of an inlet i and outlet j of a junction. A control volume, represented by dashed lines, is drawn which accounts for multiple inflows.

Now, the derivation above does not account for this kind of flow. However, we can pretend the outlet has a different cross-sectional area A'_j. By choosing A'_j such that

$$\dot{m}'_j = \dot{m}_i, \tag{3.39}$$

we are able to apply the result of the derivation above. Note that equation (3.4) and (3.6) imply

$$
\begin{aligned}
q_{i,j} &= \frac{\dot{m}_j}{\dot{m}_i}, \\
&= \frac{\dot{m}_j}{\dot{m}'_j}, \\
&= \frac{\rho A_j u_j}{\rho A'_j u_j}, \\
q_{i,j} &= \frac{A_j}{A'_j}, \\
A'_j &= \frac{A_j}{q_{i,j}}.
\end{aligned}
\tag{3.40}
$$

Therefore, by equation (3.5)

$$
\psi'_{i,j} = \frac{A_i}{A'_j} = \frac{A_i}{A_j} q_{i,j} = \psi_{i,j} q_{i,j}, \tag{3.41}
$$

where the accent means in our pretence that j has cross-sectional area A'_j. Furthermore, equation (3.4) in combination with (3.39) results in

$$
q'_{i,j} = \frac{\dot{m}'_j}{\dot{m}_i} = 1. \tag{3.42}
$$

Now substitution of $\psi_{i,j}$ and $q_{i,j}$ by $\psi'_{i,j}$ and $q'_{i,j}$ in 3.38 yields

$$
\begin{aligned}
C'_{i,j} &= 1 - \frac{1}{\psi'_{i,j} q'_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right), \\
&= 1 - \frac{1}{\psi_{i,j} q_{i,j} \cdot 1} \cos\left(\frac{3}{4}(\pi - \theta)\right), \\
C'_{i,j} &= C_{i,j}.
\end{aligned}
\tag{3.43}
$$

So, we can conclude that equation (3.38) is valid for all $q_{i,j} > 0$. Note that $q_{i,j} \leq 0$ do not exist as $\dot{m}_i \geq 0$ and $\dot{m}_j > 0$ by definition of O, or more accurately I "*where in the special case of no flow we speak of an inlet.*" Therefore, we can finally conclude that we have found a pressure coefficient between an arbitrary combination of inlet and outlet which is defined for all possible flow configurations.

A similar derivation as above is presented in appendix A to show that the expression of $C_{i,j}$ holds for junctions with n inlets and m outlets. However, this derivation assumes that any inlet can flow to any outlet, which in practice does not have to hold.

Now that we have such a pressure coefficient it is time to consider how we can put them together to create a junction model. First, it is important to consider what is needed from a junction model. Remember that to allow for pressure loss between inlets and outlets, additional pressure variables at those in-/outlets have been introduced. Specifically, if a junctions consists of n inlets and m outlets, $n + m - 1$ additional pressure points have been introduced[13]. In general, for the mathematical model to be consistent it should introduce the same amount of equations as variables. Therefore, the pressure coefficients should be manipulated in such a way that n+m-1 independent equations remain.

In the current state the model has $n \cdot m$ equations, namely

$$
P_i - P_j = C_{i,j} \rho u_j^2 \qquad \forall (i,j) \in I \times O. \tag{3.44}
$$

[13]The -1 is due to the pressure point which would be already present in a no-friction model.

Because,

$$n \cdot m \geq n + m - 1 \qquad \forall n, m \geq 1$$

usage of the "raw" equations, given by equation (3.44), would lead to an overdetermined system which in general means that it does not , with certainty, yield a solution. However, in some cases an overdetermined system can be transformed to a linearly independent system by making linear combinations of the system of equations. Therefore, for all $i \in I$ consider $a_i \in \mathbb{R}$ and equally for all $j \in O$, $b_j \in \mathbb{R}$ such that

$$\sum_{i \in I} a_i = 1 \qquad \text{and} \qquad \sum_{j \in O} b_j = 1. \tag{3.45}$$

Then we can define a linear combination of equations (3.44) by

$$\begin{aligned} \sum_{i \in I} \sum_{j \in O} a_i b_j \left(P_i - P_j\right) &= \sum_{i \in I} \sum_{j \in O} a_i b_j C_{i,j} \rho u_j^2, \\ \sum_{i \in I} a_i P_i - \sum_{j \in O} b_j P_j &= \sum_{i \in I} \sum_{j \in O} a_i b_j C_{i,j} \rho u_j^2. \end{aligned} \tag{3.46}$$

Now consider an arbitrary $i \in I$ then by setting $a_i = 1$ and thereby $a_k = 0$ for all $k \in I \setminus \{i\}$. Then (3.46) becomes

$$P_i - \sum_{j \in O} b_j P_j = \sum_{j \in O} b_j C_{i,j} \rho u_j^2. \tag{3.47}$$

This equation can be interpreted by noting that

$$\sum_{j \in O} b_j P_j$$

is a weighted average pressure of the outlets. Now consider two outlets with equal mass flow, it makes sense that these two should add equally to this average pressure. However, when considering two outlets where one has far more mass flow than the other, it is natural to make the one with more mass flow contribute more to the weighted average then the one with less flow. To incorporate this insight define

$$M = \sum_{j \in O} \dot{m}_j = \sum_{k \in I} \dot{m}_k, \tag{3.48}$$

where the second equality is a direct consequence of the mass continuity, equation (2.9). Then by specifying

$$b_j = \frac{\dot{m}_j}{M}, \tag{3.49}$$

equation (3.47) becomes

$$P_i - \sum_{j \in O} \frac{\dot{m}_j}{M} P_j = \sum_{j \in O} \frac{\dot{m}_j}{M} C_{i,j} \rho u_j^2, \tag{3.50}$$

which is the *mass flow rate weighted average* of equation (3.44), concerning an arbitrary $i \in I$.

Equivalently, for an arbitrary $j \in O$ we can deduce the *mass flow rate weighted average*, which results in

$$\sum_{i \in I} \frac{\dot{m}_i}{M} P_i - P_j = \sum_{i \in I} \frac{\dot{m}_i}{M} C_{i,j} \rho u_j^2. \tag{3.51}$$

Do note however that this method generates $n + m$ equations, which is still one too many.

To reduce the size of the system, matrix analysis is used. The system of equations (3.50) and (3.51) gives rise to the following system matrix

$$\left[\begin{array}{cccccccc|c} 1 & 0 & \dots & 0 & -\frac{\dot m_{j_1}}{M} & \dots & -\frac{\dot m_{j_m}}{M} & B_{i_1} \\ 0 & 1 & \dots & 0 & -\frac{\dot m_{j_1}}{M} & \dots & -\frac{\dot m_{j_m}}{M} & B_{i_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\frac{\dot m_{j_1}}{M} & \dots & -\frac{\dot m_{j_m}}{M} & B_{i_n} \\ \frac{\dot m_{j_1}}{M} & \frac{\dot m_{j_2}}{M} & \dots & \frac{\dot m_{j_m}}{M} & -1 & \dots & 0 & B_{j_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\dot m_{j_1}}{M} & \frac{\dot m_{j_2}}{M} & \dots & \frac{\dot m_{j_m}}{M} & 0 & \dots & -1 & B_{j_m} \end{array}\right], \tag{3.52}$$

where

$$B_{i_k} = \sum_{j\in O} \frac{\dot m_j}{M} C_{i_k,j}\rho u_j^2 \qquad \text{and} \qquad B_{j_k} = \sum_{i\in I} \frac{\dot m_i}{M} C_{i,j_k}\rho u_{j_k}^2,$$

and which after Gaussian elimination[14] gives the equivalent matrix

$$\left[\begin{array}{ccccccc|c} 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & \dots & 0 & 0 & \dots & 0 & B_{i_2} - B_{i_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & 0 & \dots & 1 & 0 & \dots & 0 & B_{i_n} - B_{i_1} \\ 1 & 0 & \dots & 0 & -1 & \dots & 0 & B_{j_1} - \sum_{i\in I\setminus i_1} \frac{\dot m_i}{M}\left(B_i - B_{i_1}\right) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & \dots & 0 & 0 & \dots & -1 & B_{j_m} - \sum_{i\in I\setminus i_1} \frac{\dot m_i}{M}\left(B_i - B_{i_1}\right) \end{array}\right]. \tag{3.53}$$

Note that the system matrix has rank $n + m - 1$ which is precisely what we need, and that there are no restrictions to which inlet is i_1 therefore i_1 can be chosen arbitrarily in I.

Now pick ref $\in I$ arbitrarily, then using equation (3.53) we get the following system

$$P_i - P_{\text{ref}} = \sum_{j\in O} \frac{\dot m_j}{M}\left(C_{i,j} - C_{\text{ref},j}\right)\rho u_j^2 \qquad \forall i \in I \setminus \{\text{ref}\} \tag{3.54}$$

$$P_{\text{ref}} - P_j = \sum_{i\in I} \frac{\dot m_i}{M} C_{i,j}\rho u_j^2 - \sum_{i\in I\setminus\{\text{ref}\}} \frac{m_i}{M} \sum_{k\in O} \frac{\dot m_k}{M}\left(C_{i,k} - C_{\text{ref},k}\right)\rho u_k^2 = C_{\text{ref},j}\rho u_j^2 \qquad \forall j \in O \tag{3.55}$$

where the last equality is a result of equation (3.44). This system is linearly independent as shown in equation (3.53) and has $n + m - 1$ equations. Therefore this system constitutes a model conforming to all requirements.

3.3.5 Validity

Now that we have a model, it is time to confirm its validity. This is normally done by comparing it to experimental results. However, due to lack of experimental results this cannot be done. Therefore, the model is compared to the T-junction model published by Bassett et al. in 2001[4], which shows agreement with experimental data.

[14] This is simply using elementary row operations to get an equivalent matrix in row echelon form. (NL: "*matrix vegen*")

Consider a T-junction consisting of a main pipe and a branch attached at an angle θ to the main pipe. Now call the two in-/outlets of the main pipe a and c and the branch b. Furthermore, consider the cross sectional area of the main pipe to be constant thus

$$A_a = A_c$$

This situation is schematically drawn in figure 3.3.

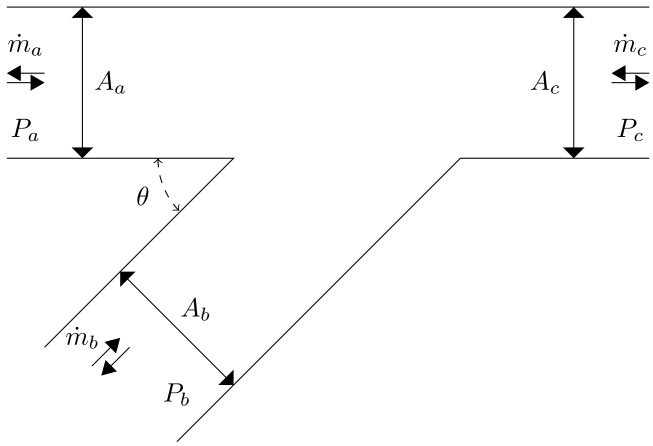
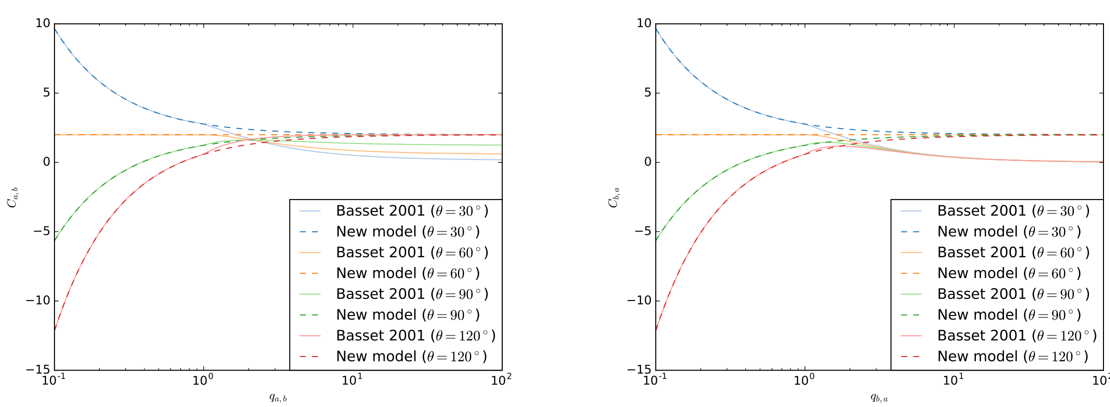


Figure 3.3: Schematic representation of a T-junction, consisting of a main pipe and a branch which makes an angle of θ with the main branch.

In 2001, Bassett et al. published a model which can describe the pressure coefficient, $C_{i,j}$ between an inlet, i, and outlet, j, for arbitrary flow conditions with good agreement to experimental results. Therefore, it is informative to compare that model to the new model derived here. However, due to symmetry only two flow types have to be examined, namely flow from a to b and flow from b to a. In figure 3.4a the pressure coefficient $C_{a,b}$ is plotted against the mass flow ratio $q_{a,b}$ for both Bassets model and the new model derived here. The same is done for $C_{b,a}$ and $q_{b,a}$ in figure 3.4b. Thus, in figure 3.4a the flow from a to b is observed, and in figure 3.4b the flow from b to a is observed.



(a) Plot of $C_{a,b}$ against $q_{a,b}$ from the model proposed by Bassett et al.[4] and the new model for various angles θ. Note the logarithmic scale.

(b) Plot of $C_{b,a}$ against $q_{b,a}$ from the model proposed by Bassett et al.[4] and the new model for various angles θ. Note the logarithmic scale.

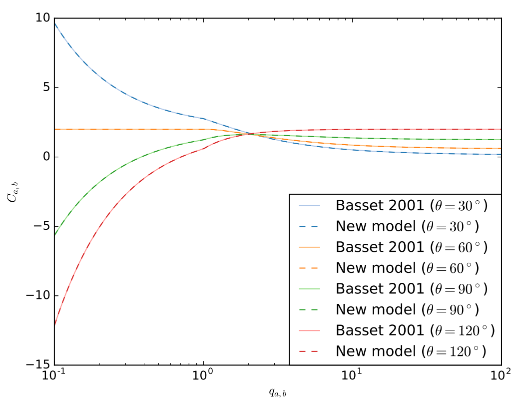
Figure 3.4: Plots to compare the new model to the model of Bassett et al.[4].

Both figures seem to follow the same pattern, for $q < 1$ the two models seem to match perfectly. Note that in for $q < 1$ there is only one inlet due to mass continuity. However, for $q > 1$ the new model does not seem to agree with the old model, this would imply that the new model does not hold for multiple inlets. But it should, after all we have specifically looked into that case. No, the difference finds its nature in an assumption of Bassett et al, namely "*it will be assumed that the static pressures in the inflow branches are equal*"[4]. This assumption is a fundamental difference between the new model and that derived by Bassett et al. In addition, Bassets model seems to deviate from experimental data as $q > 1$ increases. A possible explanation for this phenomenon could be the assumption of equal static pressure.

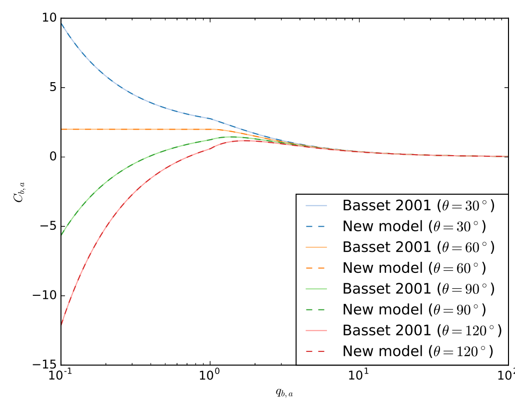
To further compare the models the assumption by Bassett et al. can be *emulated* in the new model by considering equation (3.51)

$$\sum_{i \in I} \frac{\dot{m}_i}{M} P_i - P_j = \sum_{i \in I} \frac{\dot{m}_i}{M} C_{i,j} \rho u_j^2. \tag{3.56}$$

By considering $\sum_{i \in I} \frac{\dot{m}_i}{M} P_i$ to be the static pressure for all inlets the assumption of equal static pressure for inlets is incorporated. In figures 3.5a and 3.5b, the pressure coefficients are again plotted against the mass flow ratio but with the assumption of equal static pressure at the inlets.



(a) Plot of $C_{a,b}$ against $q_{a,b}$ from the model proposed by Bassett et al.[4] and the new model with equal static inlet pressure for various angles θ. Note the logarithmic scale.



(b) Plot of $C_{b,a}$ against $q_{b,a}$ from the model proposed by Bassett et al.[4] and the new model with equal static inlet pressure for various angles θ. Note the logarithmic scale.

Figure 3.5: Plots to compare the new model with the assumption of equal static pressure for all inlets to the model of Bassett et al.[4].

The figures show that both models follow exactly the same curves for both flow types. An argument showing that this equivalency is analytical can be made rigorous by equating the expressions of the pressure coefficients. From this analysis can be concluded that the difference between both models is purely due to the assumption of equal static pressures at the inlets.

To understand what this means a specific configuration is examined. Consider the situation where

$$\begin{aligned}
\theta &= 90^\circ, \\
A_a = A_b = A_c &= 1\ m^2, \\
P_a &= 10^5\ Pa, \\
I &= \{b, c\}, \\
O &= \{a\}, \\
\dot{m}_a = \dot{m}_c &= 100\ \frac{kg}{s}, \\
\dot{m}_b &= 0\ \frac{kg}{s}, \\
\rho &= 1000\ \frac{kg}{m^3}.
\end{aligned}$$

Note that

$$\dot{m}_a + \dot{m}_b - \dot{m}_c = 100 - 100 = 0$$

thus the situation satisfies mass continuity. Now the newly introduced junction model equations (3.54) and (3.55) imply

$$\begin{aligned}
P_c - P_b &= (C_{c,a} - C_{b,a})\,\rho u_a^2, \\
P_b - P_a &= C_{b,a}\rho u_a^2,
\end{aligned}$$

where according to equation (3.38),

$$\begin{aligned}
C_{c,a} &= 1 - \frac{1}{\psi_{c,a} q_{c,a}} \cos\left(\frac{3}{4}(\pi - \theta_{c,a})\right), \\
C_{c,a} &= 1 - \frac{1}{\psi_{c,a} q_{c,a}} = 1 - 1 = 0.
\end{aligned}$$

where we can make the step because by definition $\theta_{c,a} = 180^\circ$. Furthermore, equation (3.38) implies that

$$\begin{aligned}
C_{b,a} &= 1 - \frac{1}{\psi_{b,a} q_{b,a}} \cos\left(\frac{3}{4}(\pi - \theta_{b,a})\right), \\
C_{b,a} &= 1 - \frac{\dot{m}_b}{\psi_{b,a} \dot{m}_a} \cos\left(\frac{3}{4}(\pi - \theta_{b,a})\right), \\
C_{b,a} &= 1 - 0 = 1.
\end{aligned}$$

In addition note that according to equation (2.11)

$$\begin{aligned}
\rho u_a^2 &= \rho \left(\frac{\dot{m}_a}{\rho \cdot A_a}\right)^2, \\
\rho u_a^2 &= 1000 \left(\frac{100}{1000}\right)^2, \\
\rho u_a^2 &= 10.
\end{aligned}$$

Therefore, the new model gives us

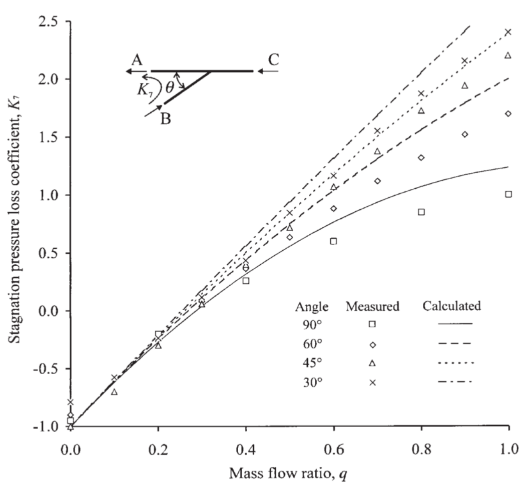
$$P_c - P_b = -\rho u_a^2,$$
$$P_b - P_a = \rho u_a^2,$$
$$P_c - P_a = 0.$$

By following Bassett et al. [3], where equal inlet pressure is assumed we get

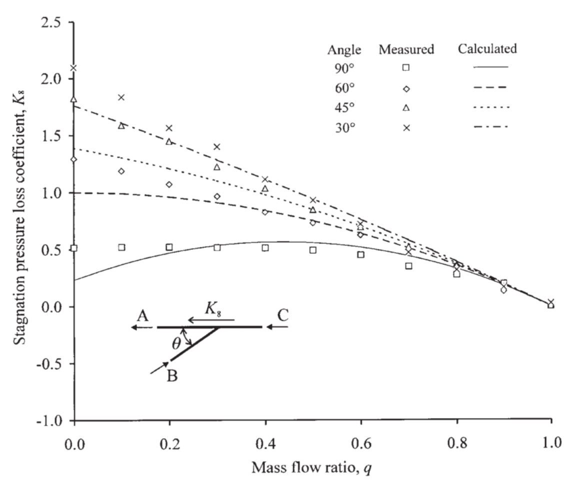
$$P_c - P_b = 0,$$
$$P_b - P_a = \rho u_a^2,$$
$$P_c - P_a = \rho u_a^2,$$

where $P_c - P_b = 0$ follows from the assumption of equal inlet pressure.

Consider figure 3.6a and 3.6b which were taken from Bassett et al. 2001[4]. In 3.6a $q = \frac{\dot{m}_b}{\dot{m}_a}$ and in 3.6b $q = \frac{\dot{m}_c}{\dot{m}_a}$. Therefore for 3.6a $q = 0$ is of interest and for 3.6b $q = 1$ is of interest in this situation.



(a) *Plot of the total pressure coefficient K7 as defined in Bassett. 2001.*



(b) *Plot of the total pressure coefficient K8 as defined in Bassett. 2001.*

Figure 3.6: Plots of Bassett et al. showing experimental data.[4].

Now as K7 is the total pressure normalized by ρu_a^2 we get for both models that K7$= -1$ which is conform figure 3.6a, however for Basset 2003 we get that K8 is 1 where the new model is 0 and only the new model conforms to figure 3.6b. Therefore we can conclude that by allowing for pressure difference between the inlets the model has better agreement with experimental results in this situation than the model with equal static pressure for all inlets. However, more experimental data is required to generalize this statement.

3.4 Summary of the model

To conclude this section, a short summary of the model and its domain of validity is given. This section describes that in case of

Single phase flow The fluid is in a single phase

Incompressible fluid A fluid with constant density

Steady-state The system is time invariant.

Horizontal junction The junction does not encounter gravitational effects

No wall friction There are no losses due to wall friction

and some more derivation specific assumptions, the pressure differences in a junction with inlets I and outlets O are described by

$$P_i - P_{\text{ref}} = \sum_{j \in O} \frac{\dot{m}_j}{M} \left(C_{i,j} - C_{\text{ref},j} \right) \rho u_j^2 \qquad \forall i \in I \setminus \{\text{ref}\} \tag{3.57}$$

$$P_{\text{ref}} - P_j = C_{\text{ref},j} \rho u_j^2 \qquad \forall j \in O \tag{3.58}$$

where

$\mathbf{P_k}$ the static pressure at in-/outlet k in $\left[\frac{kg}{m \cdot s^2}\right]$

$\dot{\mathbf{m}}_\mathbf{k}$ the mass flow rate through in-/outlet k in $\left[\frac{kg}{s}\right]$

M the total mass flow through the junction defined by $M = \sum_{i \in I} \dot{m}_i = \sum_{j \in O} \dot{m}_j$

ρ the density of the fluid in $\left[\frac{kg}{m^3}\right]$

$\mathbf{u_k}$ the flow velocity at in-/outlet k in $\left[\frac{m}{s}\right]$

Furthermore ref $\in I$ can be arbitrarily chosen and

$$C_{i,j} = 1 - \frac{1}{\psi_{i,j} q_{i,j}} \cos \left(\frac{3}{4} \left(\pi - \theta_{i,j} \right) \right) \tag{3.59}$$

where

$\psi_{\mathbf{i,j}}$ the cross sectional area ratio between inlet i and outlet j defined by $\psi_{i,j} = \frac{A_i}{A_j}$

$\mathbf{A_k}$ the cross sectional area of in-/outlet k in $\left[m^2\right]$

$\mathbf{q_{i,j}}$ the mass flow rate between inlet i and outlet j defined by $q_{i,j} = \frac{\dot{m}_i}{\dot{m}_j}$

$\theta_{\mathbf{i,j}}$ the angle between inlet i and outlet j.

4 Pipe flow network model with incorporation of junction model

In this section the junction model introduced in section 3 will be incorporated in a pipe flow network model, which was introduced in section 2. Thereafter, the section will be concluded with a short digression to numerical methods which can be used to solve the pipe flow network model.

4.1 Addition of junction model

To incorporate the junction model in a pipe flow network model some changes to the mathematical description as proposed in section 2.1 have to be made. Namely, the junction model introduces $n+m-1$ new pressure variables to the network for a junction with n inlets and m outlets. In other words, every pipe $(i,j) \in E$ in the network now has an ingoing pressure, $(P_{in})_{i,j}$, and an outgoing pressure, $(P_{out})_{i,j}$. Where $(P_{in})_{i,j}$ represents the static pressure of pipe (i,j) at node i, and $(P_{out})_{i,j}$ represents the static pressure of pipe (i,j) at node j. Thus the *in* and *out* refer to the direction of the pipe in the graph, not the direction of flow. Formally,

$$(P_{in})_{i,j} = \text{The static pressure of pipe } (i,j) \in E \text{ at node } i \in V \qquad \forall (i,j) \in E, \tag{4.1}$$

$$(P_{out})_{i,j} = \text{The static pressure of pipe } (i,j) \in E \text{ at node } j \in V \qquad \forall (i,j) \in E. \tag{4.2}$$

Furthermore, to calculate the pressure coefficient between an inlet i and an outlet j the angle $\theta_{i,j}$ between i and j has to be known. Therefore it has to be added to the mathematical description. This is in no way an easy task as it links two pipes in a junction to each other. On first sight this would lead to something awkward like $\theta_{(a,b),(c,d)}$ which would represent the angle between pipes $(a,b),(c,d) \in E$. However, by giving every node a predetermined axis we can specify an angle $(\theta_{in})_{i,j}$ which represents the angle between pipe $(i,j) \in E$ and the reference axis of node i.[15] In an equivalent manner $(\theta_{out})_{i,j}$ can be defined. Thus, formally

$$(\theta_{in})_{i,j} = \text{The angle between } (i,j) \in E \text{ and the predetermined axis of node } i \quad \forall (i,j) \in E, \tag{4.3}$$

$$(\theta_{out})_{i,j} = \text{The angle between } (i,j) \in E \text{ and the predetermined axis of node } j \quad \forall (i,j) \in E. \tag{4.4}$$

For good measure, note that the axis of a node is solely there to create a reference axis on which the angles can be based and the angles should be measured counter-clockwise[16].

Now that the necessary variables are added to the mathematical description the new pipe flow network model can be introduced. To be rigorous, consider an arbitrary pipe flow network. Using section 2.1 an equivalent graph, $G = (V, E)$, can be generated where by equations (2.1) to (2.3) and (2.5) the variables for pipe cross sectional area, $A_{i,j}$, pipe length, $L_{i,j}$, mass flow rate through a pipe, $\dot{m}_{i,j}$, and external in-/outflow at a node, s_i, are defined. Furthermore using equations (4.1) to (4.4) the static pressures at both ends of a pipe, $(P_{in})_{i,j}$ and $(P_{out})_{i,j}$, and the angles $(\theta_{in})_{i,j}$ and $(\theta_{out})_{i,j}$ with respect to the reference axis of the node at each respective end are defined. Note that all variable distributions are presumed known, except for $\dot{m}$, s, (P_{in}) and (P_{out}) which are presumed to be *partial distributions* as defined in section 2.1.

To complete the model we need equations to govern the unknown variables in the partial distributions. By assuming steady-state, equation (2.23) gives us

$$s_k + \sum_{i \in I_k} \dot{m}_{i,k} - \sum_{j \in O_k} \dot{m}_{k,j} = 0 \qquad \forall k \in V, \tag{4.5}$$

[15] Note that the assumption that the network lies in a single flat plane has been made here. A 3-dimensional network would lead to the introduction of two angles the polar and azimuth angle as in the spherical coordinate system.

[16] The choice of direction does not matter, it only matters that the same convention is used for the whole network.

where I_k and O_k are defined as in equations (2.21) and (2.22),

$$I_k = \{i \in V : (i,k) \in E\}, \tag{4.6}$$
$$O_k = \{j \in V : (k,j) \in E\}. \tag{4.7}$$

If we assume cylindrical pipes with constant diameter and incompressible fluid, equation (2.20) gives us

$$(P_{inflow})_{i,j} - (P_{outflow})_{i,j} = \frac{L_{i,j}}{D_{i,j}} \cdot f((\text{Re})_{i,j}) \cdot \frac{1}{2} \cdot \rho \cdot \overline{u}_{i,j}^2 \quad \forall (i,j) \in E, \tag{4.8}$$

where P_{inflow} and $P_{outflow}$ are the static pressure at the side of inflow and outflow respectively, $D_{i,j}$ is the diameter of pipe (i,j), $f((\text{Re})_{i,j})$ is the Darcy friction factor which is depend on the Reynolds number[17] which is defined as

$$(\text{Re})_{i,j} = \frac{\rho \cdot \overline{u}_{i,j} \cdot D_{i,j}}{\mu}, \tag{4.9}$$

and $\overline{u}$ is the average flow velocity through pipe (i,j). To use this in the model the equation has to be rewritten in quantities defined in the mathematical description. First of all, as $(P_{inflow})_{i,j}$ and $(P_{outflow})_{i,j}$ are static pressures at both ends of the pipe then can be related to $(P_{in})_{i,j}$ and $(P_{out})_{i,j}$. Do note that $(P_{in})_{i,j}$ and $(P_{out})_{i,j}$ are related to the direction of (i,j) in the graph and not to the flow direction. However, $\dot{m}_{i,j}$ relates the flow direction to the direction of (i,j) in the graph by its sign. Therefore, we can deduce

$$(P_{inflow})_{i,j} - (P_{outflow})_{i,j} = \begin{cases} (P_{in})_{i,j} - (P_{out})_{i,j} & \text{if } \dot{m}_{i,j} \geq 0 \\ (P_{out})_{i,j} - (P_{in})_{i,j} & \text{if } \dot{m}_{i,j} < 0 \end{cases}. \tag{4.10}$$

The next undefined quantity is $D_{i,j}$, but as we have already assumed the pipes to be cylindrical we can simply use the area of a circle to rewrite $D_{i,j}$ in terms of $A_{i,j}$ resulting in

$$D_{i,j} = 2\sqrt{\frac{A_{i,j}}{\pi}}. \tag{4.11}$$

Lastly, we have to rewrite $\overline{u}_{i,j}$ which can easily be done using equation (2.11), resulting in

$$\overline{u}_{i,j} = \frac{\dot{m}_{i,j}}{\rho \cdot A_{i,j}}. \tag{4.12}$$

By substituting all conversions found above into equation (4.8) we get

$$\left\{ \begin{matrix} (P_{in})_{i,j} - (P_{out})_{i,j} & \text{if } \dot{m}_{i,j} \geq 0 \\ (P_{out})_{i,j} - (P_{in})_{i,j} & \text{if } \dot{m}_{i,j} < 0 \end{matrix} \right\} = f(\text{Re}) \cdot \frac{L_{i,j} \cdot \dot{m}_{i,j}^2}{\rho \cdot A_{i,j}^2} \cdot \sqrt{\frac{\pi}{A_{i,j}}} \qquad \forall (i,j) \in E. \tag{4.13}$$

Note that $f(\text{Re})$ is still present in the equation even though it is not specified in the mathematical description. That is done because the conversion of $f(\text{Re})$ is more complex than the other conversions, and therefore deserves special attention.

Many a paper has been written about the Darcy friction factor. The problem with this friction factor is that it is dependent on the flow regime, so whether the fluid flow is laminar or turbulent. This is a problem because in the transition area between laminar and turbulent, $2000 < \text{Re} < 4000$, the flow regime is undefined. But apart from this hiccup, the factor for commercial pipes in turbulent flow is governed by the Colebrook-White equation[16]. The main problem with this equation is that it is an implicit equation. Therefore, there exist many papers that approximate the Colebrook-White equation

[17] The Reynolds number is a dimensionless quantity indicating the *regime* of the flow.

in explicit form, whose analysis is outside the scope of this text. One of those approximations is the Blasius correlation[17],

$$f(\mathrm{Re}) = \frac{0.079}{\mathrm{Re}^{\frac{1}{4}}} \qquad \text{for turbulent flow } (\mathrm{Re} > 4000), \tag{4.14}$$

which we will use in this paper because of its simplicity. The friction factor is much simpler for laminar flow, it can be derived that

$$f(\mathrm{Re}) = \frac{64}{\mathrm{Re}} \qquad \text{for laminar flow } (\mathrm{Re} < 2000). \tag{4.15}$$

To accommodate for the undetermined transition area, $2000 < \mathrm{Re} < 4000$, the sigmoid function[18] defined by

$$\mathrm{Sig}(x) = \frac{1}{1+e^{-x}}, \tag{4.16}$$

will be used to stitch the turbulent and laminar approximations together to one function yielding

$$f(\mathrm{Re}) = \mathrm{Sig}\left(-4.5 \cdot \frac{\mathrm{Re}-3000}{1000}\right) \cdot \frac{64}{\mathrm{Re}} + \mathrm{Sig}\left(4.5 \cdot \frac{\mathrm{Re}-3000}{1000}\right) \cdot \frac{0.079}{\mathrm{Re}^{\frac{1}{4}}}, \tag{4.17}$$

where the factor 4.5 inside the sigmoid function comes from the fact that $\mathrm{Sig}(4.5) \approx 0.99$. It is very important to note that this definition of $f(\mathrm{Re})$ is merely chosen for its favourable properties, such as continuity. Readers interested in explicit approximations of the Darcy friction factor are urged to read a paper specifically about this topic, such as "*Review of explicit approximations to the Colebrook relation for flow friction*" by Dejan Brkić[6], as such approximations are out of the scope of this text.

Using the prior conversions and equation (4.9) yields

$$\mathrm{Re} = 2\frac{|\dot{m}_{i,j}|}{\mu \cdot \sqrt{A_{i,j} \cdot \pi}}, \tag{4.18}$$

where the absolute of $\dot{m}_{i,j}$ is taken because Re is per definition always positive. And with this the analysis of the Darcy-Weisbach equation for this model is concluded. For good measure, the procedure to generate a Darcy-Weisbach equation for an arbitrary pipe $(i,j) \in E$ is provided in algorithm 2.

Algorithm 2 Procedure to generate a Darcy-Weisbach equation for an arbitrary pipe $(i,j) \in E$.

$\mathrm{Re} := 2\frac{|\dot{m}_{i,j}|}{\mu \cdot \sqrt{A_{i,j} \cdot \pi}}$
$f = \mathrm{Sig}\left(-4.5 \cdot \frac{\mathrm{Re}-3000}{1000}\right) \cdot \frac{64}{\mathrm{Re}} + \mathrm{Sig}\left(4.5 \cdot \frac{\mathrm{Re}-3000}{1000}\right) \cdot \frac{0.079}{\mathrm{Re}^{\frac{1}{4}}}$
$\Delta P = (P_{out})_{i,j} - (P_{in})_{i,j}$
if $\dot{m}_{i,j} \geq 0$ **then** $\Delta P = (P_{in})_{i,j} - (P_{out})_{i,j}$
end if
return $\Delta P - f \cdot \frac{L_{i,j} \cdot \dot{m}_{i,j}^2}{\rho \cdot A_{i,j}^2} \cdot \sqrt{\frac{\pi}{A_{i,j}}} = 0$

The one thing that remains is incorporating a junction model. Note that up until this point the derivation is independent of junction model and the above equations make up the basis of the *Frictionless model*. Such a Frictionless model would be finalized by introducing, for all $k \in V$, equations

$$(P_{out})_{\mathrm{ref},k} - (P_{out})_{i,k} = 0 \qquad \forall i \in I_k \setminus \{i\}, \tag{4.19}$$

$$(P_{out})_{\mathrm{ref},k} - (P_{in})_{k,j} = 0 \qquad \forall j \in O_k, \tag{4.20}$$

[18] The sigmoid function is a continuous approximation of the Heaviside function with the property that $\mathrm{Sig}(x) + \mathrm{Sig}(-x) = 1$)

for an arbitrarily chosen ref $\in I_k$ where I_k and O_k are again defined by equations (2.21) and(2.22)

$$I_k = \{i \in V : (i,k) \in E\}, \tag{4.21}$$

$$O_k = \{j \in V : (k,j) \in E\}. \tag{4.22}$$

However, section 3 introduces a new junction model and this derivation has been working towards implementing that model. So, without further ado, for each node $k \in V$ consider I_k and O_k as defined by equations (4.21) and(4.22). To adhere to the notation used in section 3.4 define the set of inlets, I as

$$I = \{i \in I_k : \dot{m}_{i,k} \geq 0\} \cup \{j \in O_k : \dot{m}_{k,j} \leq 0\}. \tag{4.23}$$

Equivalently the set of outflows, O as

$$O = \{i \in I_k : \dot{m}_{i,k} < 0\} \cup \{j \in O_k : \dot{m}_{k,j} > 0\}. \tag{4.24}$$

Note that I and O are disjunct sets and $I \cup O = I_k \cup O_k$. Moreover, it is very important to note that where I_k and O_k are defined by graph direction, I and O are defined by flow direction.

To further convert the notation used here to the notation used in section 3.4 consider an arbitrary inlet $i \in I$. We can define the mass flow rate, $\dot{m}_i$, cross sectional area, A_i, static pressure, P_i, and angle to the reference axis of the junction θ_i of the inlet by

$$\dot{m}_i = \begin{cases} \dot{m}_{i,k} & \text{if } i \in I_k \\ -\dot{m}_{k,i} & \text{if } i \in O_k \end{cases}, \tag{4.25}$$

$$A_i = \begin{cases} A_{i,k} & \text{if } i \in I_k \\ A_{k,i} & \text{if } i \in O_k \end{cases}, \tag{4.26}$$

$$P_i = \begin{cases} (P_{out})_{i,k} & \text{if } i \in I_k \\ (P_{in})_{k,i} & \text{if } i \in O_k \end{cases}, \tag{4.27}$$

and

$$\theta_i = \begin{cases} (\theta_{out})_{i,k} & \text{if } i \in I_k \\ (\theta_{in})_{k,i} & \text{if } i \in O_k \end{cases}. \tag{4.28}$$

Equivalently, for an outlet, $j \in O$, $\dot{m}_j$, A_j, P_j and θ_j are defined by

$$\dot{m}_j = \begin{cases} -\dot{m}_{j,k} & \text{if } j \in I_k \\ \dot{m}_{k,j} & \text{if } j \in O_k \end{cases}, \tag{4.29}$$

$$A_j = \begin{cases} A_{j,k} & \text{if } j \in I_k \\ A_{k,j} & \text{if } j \in O_k \end{cases}, \tag{4.30}$$

$$P_j = \begin{cases} (P_{out})_{j,k} & \text{if } j \in I_k \\ (P_{in})_{k,j} & \text{if } j \in O_k \end{cases}, \tag{4.31}$$

and

$$\theta_j = \begin{cases} (\theta_{out})_{j,k} & \text{if } j \in I_k \\ (\theta_{in})_{k,j} & \text{if } j \in O_k \end{cases}. \tag{4.32}$$

Note that $\dot{m}_i, \dot{m}_j \geq 0$ just like in section 3.4.

Now pick ref $\in I$ at random, then according to section 3.4 the pressure distribution of the junction is governed by equations (3.57) and (3.58),

$$P_i - P_{\text{ref}} = \sum_{j\in O} \frac{\dot{m}_j}{M} \left(C_{i,j} - C_{\text{ref},j}\right) \rho u_j^2 \qquad \forall i \in I \setminus \{\text{ref}\}, \tag{4.33}$$

$$P_{\text{ref}} - P_j = C_{\text{ref},j} \rho u_j^2 \qquad \forall j \in O, \tag{4.34}$$

where M is the total mass through the junction defined by

$$M = \sum_{i\in I} \dot{m}_i = \sum_{j\in O} \dot{m}_j, \tag{4.35}$$

and u_j is the flow speed in outlet $j \in O$ defined by

$$u_j = \frac{\dot{m}_j}{\rho \cdot A_j}. \tag{4.36}$$

Furthermore, $C_{i,j}$ is the pressure coefficient for flow between inlet i and outlet j defined by equation (3.59),

$$C_{i,j} = 1 - \frac{1}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}\left(\pi - \theta_{i,j}\right)\right), \tag{4.37}$$

where $\psi_{i,j}$ is the cross sectional area ratio defined by

$$\psi_{i,j} = \frac{A_i}{A_j}, \tag{4.38}$$

and $q_{i,j}$ is the mass flow rate ratio defined by

$$q_{i,j} = \frac{\dot{m}_j}{\dot{m}_i}. \tag{4.39}$$

In addition, the angle between an in- and outlet, $\theta_{i,j}$, is defined as

$$\theta_{i,j} \equiv \theta_i - \theta_j \mod \pi, \tag{4.40}$$

where the modulo π is to ensure that $0 \le \theta_{i,j} < \pi$, which is necessary for use in the model of section 3.4.[19]

Consideration of all equations, mass continuity, Darcy-Weisbach and the equations as a result of the junction model, results in a system of equations governing the variables. Solving this system of equations yields the end result where all unknowns have been solved. However this model cannot be used on an arbitrary set of unknowns, for example if all variables are unknown it is obvious that the network can not be solved. For a network to be solvable by this model, it should have an equal amount of unknowns and independent equations in the model. We start by counting the total amount of equations. Looking at pipes, the Darcy-Weisbach equation supplies one equation for each pipe thus in total $\#E$ equations. For the nodes it is a little more complex. Equations on the nodes are the mass continuity equation which supplies one equation per node and the junction model which supplies the $n_k - 1$ equations for each node $k \in V$ where n_k is the amount of pipes connected to k. Thus in total each node $k \in V$ supplies n_k equations. By noting that each pipe is by definition connected to two nodes, we can deduce that

$$\sum_{k\in V} n_k = 2 \cdot \#E \tag{4.41}$$

[19] In layman's terms $0 \le \theta_{i,j} < \pi$ is defined such that $\theta_i - \theta_j = k \cdot \pi + \theta_{i,j}$ with k an integer, it can be proven that such a $\theta_{i,j}$ always exists.

which brings the total amount of equations to $3 \cdot \#E$.

To count the variables consider that only the $\dot{m}$, (P_{in}), (P_{out}) and s distributions are allowed to be partial and therefore contain unknowns. As $\dot{m}$, (P_{in}) and (P_{out}) are defined on the pipes they amount to $3 \cdot \#E$ variables. The s distribution is defined on the nodes and therefore introduces $\#V$ variables. This brings the total to $\#V + 3 \cdot \#E$ variables. As we have $\#V + 3 \cdot \#E$ variables and $3 \cdot \#E$ equations, the model becomes solvable if $\#V + 3 \cdot \#E - 3 \cdot \#E = \#V$ variables are known.

Hereby, we have completed the pipe flow network model with incorporated junction model. For clarity, the whole model is summarized in algorithm 3.

Algorithm 3 Pipe flow network model with junction model incorporated

---

Require: The mathematical description of a pipe flow network consisting of a graph $G = (V, E)$, variables for each pipe $(i,j) \in E$ for cross sectional area $A_{i,j}$, pipe length $L_{i,j}$, mass flow rate through pipe $\dot{m}_{i,j}$, static pressure at in- and outflow $(P_{in})_{i,j}$, $(P_{out})_{i,j}$ respectively and angles with respect to the reference axis at the nodes of the in- and outflow $(\theta_{in})_{i,j}$ and $(\theta_{out})_{i,j}$ respectively. In addition, the external in-/outflow variable s_i for $i \in V$ has te be given, and the fluid density ρ and dynamic viscosity μ. Note that the mass flow rate, static pressure and external in-/outflow distributions can be partial distributions as long as these partial distributions combined have precisely $\#V$ knowns and atleast one static pressure is known.

$S = \varnothing$

for $k \in V$ **do**

  $I_k := \{i \in V : (i,k) \in E\}$

  $O_k := \{j \in V : (k,j) \in E\}$

  $S := S \cup \left\{ s_k + \sum_{i \in I_k} \dot{m}_{i,k} - \sum_{j \in O_k} \dot{m}_{k,j} = 0 \right\}$

  $I := \{i \in I_k : \dot{m}_{i,k} \geq 0\} \cup \{j \in O_k : \dot{m}_{k,j} \leq 0\}$

  $O := \{i \in I_k : \dot{m}_{i,k} < 0\} \cup \{j \in O_k : \dot{m}_{k,j} > 0\}$

  for $i \in I \cup O$ **do**

$$A_i := \begin{cases} A_{i,k} & \text{if } i \in I_k \\ A_{k,i} & \text{if } i \in O_k \end{cases}$$

$$P_i := \begin{cases} (P_{out})_{i,k} & \text{if } i \in I_k \\ (P_{in})_{k,i} & \text{if } i \in O_k \end{cases}$$

$$\theta_i := \begin{cases} (\theta_{out})_{i,k} & \text{if } i \in I_k \\ (\theta_{in})_{k,i} & \text{if } i \in O_k \end{cases}$$

$$\dot{m}_i = \begin{cases} \dot{m}_{i,k} & \text{if } i \in I_k \text{ and } i \in I \\ -\dot{m}_{k,i} & \text{if } i \in O_k \text{ and } i \in I \\ -\dot{m}_{i,k} & \text{if } i \in I_k \text{ and } i \in O \\ \dot{m}_{k,i} & \text{if } i \in O_k \text{ and } i \in O \end{cases}$$

  end for

  Take ref $\in I$ randomly

  for $j \in O$ **do**

$\psi_{\text{ref},j} = \frac{A_{\text{ref}}}{A_j}$

$q_{\text{ref},j} = \frac{\dot{m}_j}{\dot{m}_{\text{ref}}}$

$\theta_{\text{ref},j} \equiv \frac{\theta_{\text{ref}} - \theta_j}{\pi} \mod 1$

$C_{\text{ref},j} = 1 - \frac{1}{\psi_{\text{ref},j} q_{i,j}} \cos\left(\frac{3}{4}\left(\pi - \theta_{\text{ref},j}\right)\right)$

$u_j = \frac{\dot{m}_j}{\rho \cdot A_j}$

$S := S \cup \left\{ P_{\text{ref}} - P_j - C_{\text{ref},j} \rho u_j^2 \right\}$

  end for

Algorithm 3 Pipe flow network model with junction model incorporated (continued)

$M = \sum_{j\in O} \dot{m}_j$
for $i \in I \setminus \{\text{ref}\}$ **do**
  for $j \in O$ **do**
    $\psi_{i,j} = \frac{A_i}{A_j}$
    $q_{i,j} = \frac{\dot{m}_j}{\dot{m}_i}$
    $\theta_{i,j} \equiv \frac{\theta_i - \theta_j}{\pi} \mod 1$
    $C_{i,j} = 1 - \frac{1}{\psi_{i,j} q_{i,j}} \cos\left(\frac{3}{4}\left(\pi - \theta_{i,j}\right)\right)$
  end for
  $S := S \cup \left\{ P_i - P_{\text{ref}} - \sum_{j\in O} \frac{\dot{m}_j}{M}\left(C_{i,j} - C_{\text{ref},j}\right)\rho u_j^2 \right\}$
end for
end for
for $(i,j) \in E$ **do**
  $\text{Re} := 2\frac{|\dot{m}_{i,j}|}{\mu\cdot\sqrt{A_{i,j}\cdot\pi}}$
  $f = \text{Sig}\left(-4.5\cdot\frac{\text{Re}-3000}{1000}\right)\cdot\frac{64}{\text{Re}} + \text{Sig}\left(4.5\cdot\frac{\text{Re}-3000}{1000}\right)\cdot\frac{0.079}{\text{Re}^{\frac{1}{4}}}$
  $\Delta P = \left(P_{out}\right)_{i,j} - \left(P_{in}\right)_{i,j}$
  if $\dot{m}_{i,j} \geq 0$ **then**
    $\Delta P = \left(P_{in}\right)_{i,j} - \left(P_{out}\right)_{i,j}$
  end if
  $S := S \cup \left\{ \Delta P - f\cdot\frac{L_{i,j}\cdot\dot{m}_{i,j}^2}{\rho\cdot A_{i,j}^2}\cdot\sqrt{\frac{\pi}{A_{i,j}}} = 0 \right\}$
end for
return Solve(S)

4.2 Numerical methods

To solve the system of equations which results from the model numerical root-finding algorithms can, and most often will, be used. These algorithms come in a lot of flavours, probably the most intuitive algorithm is the bisection method which will be discussed first. However the bisection method can only be used for univariate equations, equations with one variable. The model consists of a system of multivariate equations. Therefore, we have to look for an algorithm for multivariate equations. The Newton method is such an algorithm, probably the most well known of its sort, and will be discussed next. After that the subsection will be concluded with a notion about pre-implemented algorithms.

4.2.1 Bisection method

As already said the bisection method is an algorithm to find the root[20] of a univariate continuous real function f. It is a very robust method, meaning that it will always find a root, $x \in \mathbb{R}$ if it exists. It does however request an interval $[a, b] \subseteq \mathbb{R}$ from the user in which the root is situated, or more precisely such that $f(a)\cdot f(b) < 0$. Then the iterative step is pretty straight forward, consider $c = a + \frac{(b-a)}{2}$, which is just the middle of a and b, then if $f(c) = 0$ stop iterating, else if $f(c)$ has the same sign as $f(a)$, $a = c$ and otherwise $b = c$. This approach is summarized in algorithm 4. The general idea is that the width of the interval is halved with each iteration, while keeping the root in the interval and thereby approximating the root with the centre of the interval.

[20] In mathematics a root of a real function, $f : \mathbb{R} \to \mathbb{R}$, is an $x \in \mathbb{R}$ such that $f(x) = 0$.

Algorithm 4 The Bisection method.

Require: A univariate continuous real function $f : \mathbb{R} \to \mathbb{R}$, an interval $(a, b) \subset \mathbb{R}$ such that $f(a) \cdot f(b) \leq 0$ and a threshold value $\varepsilon > 0$.

while $b - a > \varepsilon$ **do**
  $c = a + \frac{(b-a)}{2}$
  if $f(c) = 0$ **then**
    $a := c$
    $b := c$
  else if $f(c) \cdot f(a) > 0$ **then**
    $a := c$
  else
    $b := c$
  end if
end while

4.2.2 Newton method

Another method, more appropriate to the model, is the multivariate Newton method. Consider a vector of real multivariate differentiable functions $\vec{f} : \mathbb{R}^k \to \mathbb{R}^m$. The aim is to derive a method to approximate an $\vec{x} \in \mathbb{R}^k$ such that $\vec{f}(\vec{x}) = \vec{0}$ given an initial guess $\vec{x}_0 \in \mathbb{R}^k$. Consider the first order Taylor expansion of $f_i \in \vec{f}$,

$$f_i(\vec{x} + \delta\vec{x}) = f_i(\vec{x}) + \sum_{x_n \in \vec{x}} \frac{\partial f_i}{\partial x_n}(\vec{x}) \cdot \delta\vec{x} + \mathcal{O}(\delta\vec{x}^2), \tag{4.42}$$

which in vector form yields

$$\vec{f}(\vec{x} + \delta\vec{x}) \approx \vec{f}(\vec{x}) + \vec{J}_f(\vec{x})\delta\vec{x}, \tag{4.43}$$

where J_f is the Jacobian of $\vec{f}$ defined as

$$\vec{J}_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_k} \end{bmatrix}. \tag{4.44}$$

Consider the situation where we have an approximation of the root $\vec{x}_n$ and we want a better approximation $\vec{x}_{n+1}$ such that

$$\vec{f}(\vec{x}_{n+1}) = 0, \tag{4.45}$$

then using equation (4.43) we get

$$\begin{aligned} \vec{f}(\vec{x}_{n+1}) &\approx \vec{f}(\vec{x}_n) + \vec{J}_f(\vec{x}_n)(\vec{x}_{n+1} - \vec{x}_n), \\ 0 &\approx \vec{f}(\vec{x}_n) + \vec{J}_f(\vec{x}_n)\vec{x}_{n+1} - \vec{J}_f(\vec{x}_n)\vec{x}_n, \\ \vec{J}_f(\vec{x}_n)\vec{x}_{n+1} &\approx \vec{J}_f(\vec{x}_n)\vec{x}_n - \vec{f}(\vec{x}_n). \end{aligned}$$

Thus we get an approximation of $\vec{x}_{n+1}$ by solving

$$\vec{J}_f(\vec{x}_n)\vec{x}_{n+1} = \vec{J}_f(\vec{x}_n)\vec{x}_n - \vec{f}(\vec{x}_n), \tag{4.46}$$

which is a linear equation and therefore can easily be solved using linear algebra. So using this as iterative step, we can deduce algorithm 5. It is important to note that this method is not robust like the bisection method, meaning it can diverge and not find the root. Therefore it is important to specify a maximum amount of iterations $N \in \mathbb{N}$. For a more extensive account of the Newton method and other numerical root-finding algorithms the reader is recommended to have a look at chapter 9 of the book "*Numerical recipes in C: The art of scientific computing*" by Saul Teukolsky and William H. Press [14].

Algorithm 5 Newton method

Require: A real multivariate differentiable functions $\vec{f} : \mathbb{R}^k \to \mathbb{R}^m$, the Jacobian matrix of $\vec{f}$, $\vec{J}_f$, an initial guess of the root $\vec{x}_0$ and $N \in \mathbb{N}$ the amount of iterations.

for $i \in \{1, \dots, N\}$ **do**

Solve($\vec{J}_f(\vec{x}_{i-1})\vec{x}_i = \vec{J}_f(\vec{x}_{i-1})\vec{x}_{i-1} - \vec{f}(\vec{x}_{i-1})$)

end for

4.2.3 Pre-implemented algorithms

Nowadays, most programming languages have libraries for scientific work. Which most often include a wide range of implemented root-finding algorithms. Using one of these implementations has the big advantage that the implementation is generally highly optimized and therefore runs very fast. In addition, a big advantage over implementing a root-finding algorithm yourself is that it is less prone to bugs, as the code is checked by many more people than just yourself. In the rest of this text a pre-implemented root-finding algorithm will be used, the next section will elaborate on the implementation of the model and the algorithm used.

5 Implementation

The model described in section 4 is implemented in a programming language called Python. This chapter gives a short introduction to Python, after that two libraries used in the implementation are discussed. The section concludes with a description of the actual implementation.

5.1 Python

Python is one of the most popular programming languages in the world[11]. The language was developed in the 1980s at the Centrum voor Wiskunde en Informatica in Amsterdam (Netherlands) by Guido van Rossum. Nowadays it is used by big companies such as Google, large open-source projects such as Blender[21] and well known research institutions like CERN.

The main advantage of Python is the relative ease of use when compared to languages such as C and Rust due to higher-level data structures and the fact that it is can be used as interpreted language. Because of this ease of use it is an ideal language for fast prototyping and proof-of-concept programs. Furthermore, because of the popularity of Python in combination with being open-source, many functionalities not readily built in into the standard Python library are implemented in external libraries such as the scientific ecosystem SciPy.

5.1.1 SciPy suite

According to SciPy.org "*SciPy (pronounced 'Sigh Pie') is a Python-based ecosystem of open-source software for mathematics, science, and engineering.*" It consists of multiple open-source Python libraries and is widely accepted as the golden standard for computational- and scientific work in the Python community. It can be viewed as the pythonic equivalent of Matlab, but combined with the powerful constructs inherent to the Python language. Its features range from simple linear algebra to signal processing to symbolic integration, and it also contains a comprehensive plot library.

5.1.2 NetworkX

According to NetworkX.github.io "*NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.*" It is an open-source Python library implementing data structures for graphs, directed graphs and even multi graphs. Its data structures are very general purpose, any node and any edge can hold any type of data, from single weights to large XML records. Though in this text only the data structures are of interest, the package also allows for advanced analysis of graphs.

5.2 Pipe flow model implementation

In appendix B the source code of the implementation discussed here is given along with a usage example. To make the implementation applicable for different pipe flow networks it is written as a framework. Meaning the program is not written to solve one specific network, but rather to solve a user specified network. This framework was implemented in the `FlowNetwork` class, so to create a new network a new instance of this class has to be made. During initialization the fluid properties, density `(float) rho` and dynamic viscosity `(float) mu`, can be supplied, they default to the properties of water at $25°C$ and can later on be changed using their respective set methods.

[21] Blender is a very popular open-source 3D content creation suite. Link: www.blender.org

The initializer returns a `FlowNetwork` instance which has methods such as `addnodes` and `addcomponents` to add nodes and pipes to the network. A node can have two variables the external in- or outflow on the node, `(float) s` in $\left[\frac{kg}{s}\right]$, and whether or not the node is a junction, `(bool) junction`. Note that for each node one has to specifically state that it is a junction, this is done because the junction model requires the angles of the pipes connected to this junction to be specified. Therefore, by introducing the `junction` variable the user can choose to use the friction or no-friction model on a per junction basis.

A component (or pipe) should have at least two variables, the cross sectional area of the pipe `(float) A` in $\left[m^2\right]$ and the length `(float) L` in $[m]$. A component can however have up to 5 additional variables, the in- and outgoing pressure `(float) pin` and `(float) pout` respectively in $\left[\frac{kg}{m \cdot s^2}\right]$, the in- and outgoing angle with the node reference axis, `(float) thetain` and `(float) thetaout` respectively in $[°]$, and lastly the mass flow rate through the pipe `(float) m` in $\left[\frac{kg}{s}\right]$. The variables of a component can all be easily correlated with the variables in section 4.

After the network has been input properly in the `FlowNetwork` instance the network can be prepared for numerical solving. This is done using the `getunknowns` method, this method takes the network and checks for any missing variables. If any are found an initial value is added to the *initials list* and their place in the graph in combination with their index in the initials list is saved to a *dictionary object*[22]. The method returns the dictionary object and the initials list.

The initials list is used to solve the network using a numerical solver such as `scipy.optimize.root` of the SciPy suite. Such a solver needs a residue function to find the root of, for the `FlowNetwork` instance this residue function is given by the `residue` method. This method is an implementation of algorithm 3 with some additional functionalities such as falling back to a no-friction model when the junction is not specified or the reference inlet cannot be established.

After the numerical solver is done, the `getresult` method returns a new `FlowNetwork` instance containing the numerically determined values. The `info` method of the resulting `FlowNetwork` instance can then be used to get the info from the nodes or the pipes or both. Which concludes the elaboration on a typical usage of the implementation.

To conclude this section an implementation choice of the framework has to be highlighted. Namely, both the pipe model as the junction model have been implemented using static functions. This implies that they can be interchanged with other models by calling their respective set methods `setPipeModel` and `setJunctionModel`. This functionality allows for testing of other pipe flow- or junction models with this framework.

[22]In layman's terms a dictionary object is a list which relates specific indices to data, just like an ordinary dictionary.

6 Comparison of the simple model and the model with junction integration

In this section the pipe flow network model with junction model, introduced in section 3.4, will be compared to the pipe flow network model without junction model, or more precisely with the no friction model. This is important because from the comparison the added value of incorporating the junction model can be determined. And even more important, using the data presented here a feeling can be developed for which situations lead to a significant difference to use the junction model. For the comparison the implementation as described in section 5 will be used.

6.1 Results

To obtain results consider a T-junction with a straight horizontal main pipe from a to c with a constant cross sectional area of 1 m^2. In the middle of the main pipe a branch pipe is connected making an angle of θ with the side of a, and implicitly an angle $\pi - \theta$ with the side of c. The cross sectional area of b is A in $\left[m^2\right]$. The centre of the junction which arises is called j. The lengths of the pipes are all L in $[m]$, thus,

$$L_{a,j} = L_{b,j} = L_{c,j} = L \tag{6.1}$$

Now that the structure of the network is set, we can focus our attention on the fluid flow inside the network. The network is filled with water at $25°C$ from which we can derive that the density $\rho = 997.08 \frac{kg}{m^3}$ and dynamic viscosity $\mu = 9.00 \cdot 10^-4\ Pa\,s$ where we assume to work in the order of 1 atmospheric pressure[9]. We also assume the external in-/outflow to be known at a and b thus s_a and s_b are known and we assume the static pressure at the entrance of pipe (a, j) to be $(P_{in})_{a,j} = 10^5\ Pa$. Lastly, we assume that there is no external in-/outflow at the junction so $s_j = 0\ \frac{kg}{s}$. This situation is summarized in figure 6.1.

Using this network we can get an insight in when the incorporation of a junction model into a pipe flow network model yields substantial differences in the result. This is done by solving the network using the implementation in section 5. On first sight the most influential factor will probably be the length of the pipes in the network L because the total static pressure loss in the network is made up out of the pressure loss due to junctions and due the pipe friction which is proportional to the pipe length. In figure 6.2a the pressure loss $\Delta P = (P_{in})_{a,j} - (P_{out})_{b,j}$ is plotted against the length of the pipes in the network L for different values of the external in-/outflow at b, s_b. In the figure $s_a = 10$ $\frac{kg}{s}$, $A = 1\ m^2$ and $\theta = 90°$.

Figure 6.2a clearly shows that, at least for higher values of s_b, there is a significant difference in pressure loss between the two models. The junction model increases the pressure loss over the network, which is to be expected as the fluid loses energy due to geometric friction in the junction. On the other hand, the pipe length, L, seems to have no influence on the absolute pressure loss difference between the two models. However, as the total static pressure loss increases linearly with the pipe length and the pressure loss difference is constant throughout, the pipe length should have an impact on the relative pressure loss difference.

To further investigate this the relative pressure loss difference, defined by

$$\frac{\Delta P_{friction} - \Delta P_{no-friction}}{\Delta P_{no-friction}}, \tag{6.2}$$

is plotted against L in figure 6.2b. Note that the lines of this figure start at $L = 20\ m$ because the relative pressure loss difference becomes very big as L goes to zero because the pipe friction goes to zero. Figure 6.2b clearly shows what was expected, namely that the relative difference goes to zero as

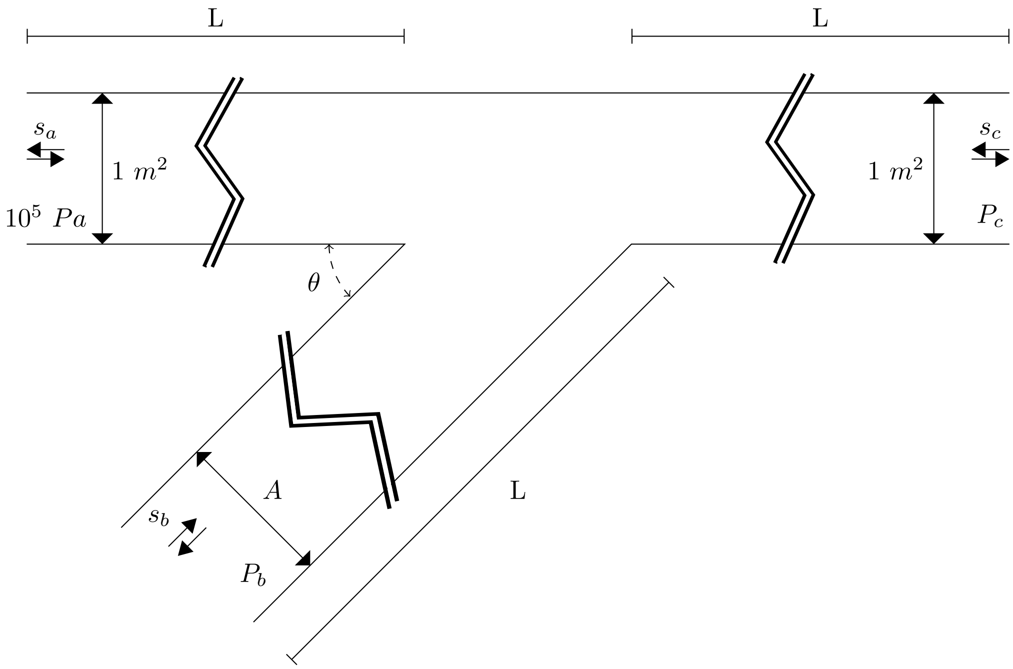
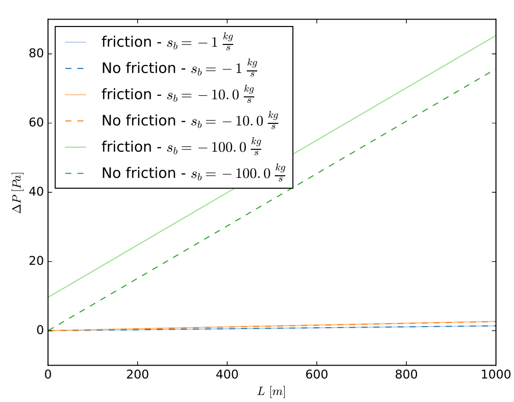
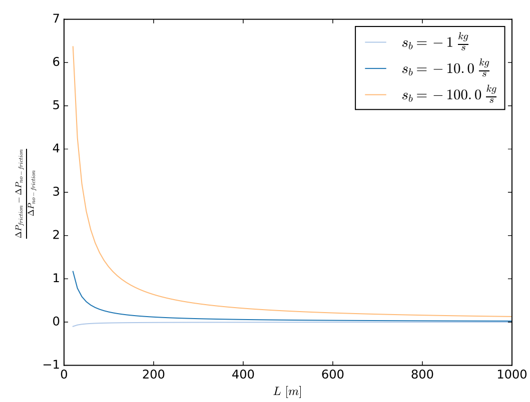


Figure 6.1: A schematic representation of the network used for the comparison between the pipe flow network model with junction model and without.



(a) The pressure loss over a and b plotted against the length of the pipes in the network L for different values of s_b. In the plot $s_a = 10\ \frac{kg}{s}$, $A = 1\ m^2$ and $\theta = 90°$.



(b) The relative pressure loss difference over a and b plotted against the length of the pipes in the network L for different values of s_b. In the plot $s_a = 10\ \frac{kg}{s}$, $A = 1\ m^2$ and $\theta = 90°$.

Figure 6.2

L becomes large. However, especially for large s_b, the relative pressure difference is very significant, more than 10%, for $L < 600\ m$ and even longer than that.

Apart from the pipe length, other quantities are believed to play a role in the pressure loss difference. For example, for smaller angles between a and b, θ, the fluid has to make a sharper bend and therefore the pressure loss would intuitively become bigger. To illustrate these effects the relative pressure loss difference is plotted against the angle between a and b, θ, in figure 6.3a for various values of s_b. In the figure $s_a = 10\ \frac{kg}{s}$, $A = 1\ m^2$ and $L = 200\ m$. The aforementioned effect is clearly visible in the figure, especially for $s_b = -10\ \frac{kg}{s}$ where the junction model adds approximately 30% extra pressure loss to the system for small angles. Another thing to note is that for $s_b = -10\ \frac{kg}{s}$, where there is only flow from a to b, the relative pressure loss difference goes to 0 as θ approaches 180°. Thismplies that the junction model does not impose any geometrical effects which is intuitive because the "junction", geometrically, approaches a pipe. However, note that for $s_b \neq -10\ \frac{kg}{s}$ this is not the case which is probably due to the interaction of the flow from a to b with the flow between a and c or b and c. Lastly, another interesting result is the negative relative pressure loss for $s_b = -1\ \frac{kg}{s}$ at large angles. Which could be due to a pressure applied by the 9 times bigger flow from a to c which has to be countered or it could be an artefact in the model, further comparison with experiments, which is out of the scope of this text, should bring closure in this matter.

Another quantity of interest is the cross sectional area ratio between pipe a and pipe b,

$$\psi_{a,b} = \frac{1}{A}, \tag{6.3}$$

because a pipe flow network can be made up of a lot of different sized pipes. Therefore, the relative pressure loss difference is plotted against the cross sectional area of pipe b, A, in figure 6.3b, again for different values of s_b. The figure shows that the relative pressure loss difference has a peak cross sectional area which is dependent on s_b. This can be explained by considering that, according to the Darcy-Weisbach, the pressure loss over a pipe is proportional to u^2, where u is the average flow velocity through a the pipe. Now because

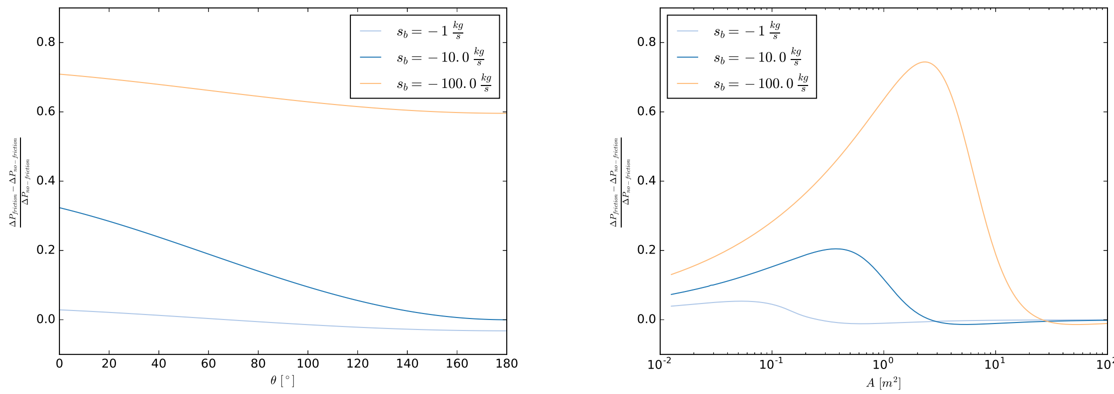
$$u = \frac{\dot{m}}{A \cdot \rho} \tag{6.4}$$

the pressure loss due to pipes for $A \ll 1$ is approximately proportional[23] to A^{-2}. Now by considering the junction to be a contraction for $A \ll 1$ literature [9] states that the pressure coefficient is proportional to A. Therefore, the pressure loss due to the junction is more in the order of A^{-1}. This implies that the pressure loss due to the pipes will dominate the pressure loss of the system, and therefore the relative pressure loss difference will go to 0 for $A \ll 1$. For $A \gg 1$ the junction functions as an enlargement. Then according to literature[9] the pressure coefficient is approximately proportional to $(1 - \frac{1}{A})^2$ which implies that the pressure loss due to the junction is approximately proportional to A^{-2} and goes to zero for $A \gg 0$. This inherently means that the relative pressure loss difference goes to zero for $A \gg 0$.

We conclude this section by observing one last quantity. From the plots above we observe that s_b has a large influence on the relative pressure loss difference. Therefore, figure 6.4 shows a plot of the relative pressure loss difference over a and b versus the external mass outflow in b, s_b. In the figure $s_a = 10\ \frac{kg}{s}$, $A = 1\ m^2$ and $\theta = 90°$. From the figure it is clearly observed that the relative pressure loss difference increases with the decrease of s_b[24]. This can be explained by considering that the mass that flows out of the network at s_b has to go through the junction. In the junction it changes direction, by 90° to be exact, and in that process energy is lost in the form of a pressure loss. The higher the mass flow rate the more momentum the fluid has in the junction and the more force has to be exerted by the junction to "*bend*" the flow. Thus the more energy in the form of pressure is dissipated in the junction.

Figure 6.4 also shows negative relative pressure loss difference for $s_b > -5$. Note that because $s_a = 10$, $s_b > -5$ implies that $s_c < -5$ thus the flow in the junction from a towards c is bigger than the flow from a to b. Then it could be that additional pressure has to be applied at b to keep the fluid from flowing towards it. Note that this is only an interpretation and experiments should establish whether this kind of this negative relative pressure loss difference even occurs in practice.

[23] Note that this is a mere approximation and in most cases the Darcy friction factor is also a function of A.
[24] Note that s_b is negative in the plot.



(a) The relative pressure loss difference over a and b plotted against the angle between a and b, θ, for different values of s_b. In the plot $s_a = 10\ \frac{kg}{s}$, $A = 1\ m^2$ and $L = 200\ m$.

(b) The relative pressure loss difference over a and b plotted against the cross sectional area A for different values of s_b. In the plot $s_a = 10\ \frac{kg}{s}$, $L = 200\ m$ and $\theta = 90°$.

Figure 6.3

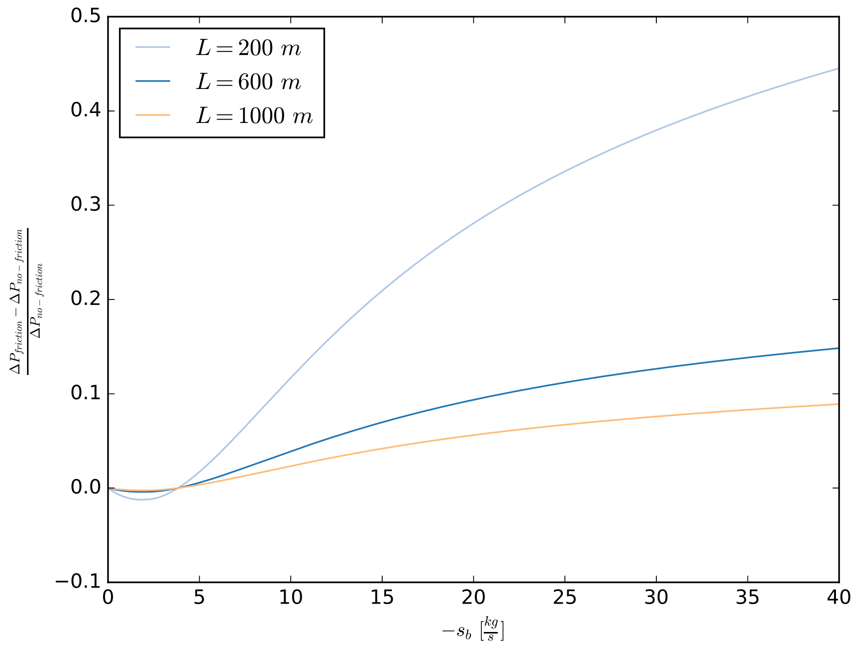


Figure 6.4: The relative pressure loss difference over a and b plotted against external mass outflow s_b for different values of L. In the plot $s_a = 10\ \frac{kg}{s}$, $A = 1\ m^2$ and $\theta = 90°$.

7 Conclusion

This text introduces a steady-state pressure loss model for flat horizontal junctions with arbitrary amounts of in-/outlets and incompressible single phase flow. From the literary research concerning junction models can be concluded that the quantities with most influence on the pressure loss due to a junction are the mass flow rate ratios, cross sectional area ratios and angles between inlets and outlets in the junction. The model that was derived using this information shows agreement with an existing model that is verified with experimental data. Furthermore the model shows promise of improving the existing model, this however should be verified with experimental data.

Using the aforementioned junction model, the mass continuity equation and the Darcy-Weisbach equation a *pipe flow network model* is derived. This model can solve the flow in an arbitrary pipe flow network if the geometric properties of the network, the fluid properties and enough data about the flow are provided.

Using a Python implementation of the pipe flow network model the model is compared to a model for pipe flow networks where the pressure loss due to junctions is neglected. The comparison shows that using the new pipe flow network model yields different results when pipe lengths in the network average to less than 600 meters, and the difference becomes greater with smaller angles and a higher mass flow rate. It should however be noted that the pipe flow network model is not compared to experimental results, and therefore it is hard to say whether the new model yields an improvement over the model without junction friction.

For future research it is highly recommended to compare the here proposed model to experimental results. This would help to verify the model and could lead to more insight in multi pipe junctions and their influence on flow in a pipe flow network. In addition, mathematical properties such as the well-posedness of the model should be determined to form rigorous notion about how well the model behaves under all circumstances. Lastly, the model could be used to solve large networks again to observe its behaviour in a more complex setting.

References

[1] International Energy Statistics - EIA.

[2] H.E.A. van den Akker and R.F. Mudde. *Fysische transportverschijnselen : denken in balansen.* Delft Academic Press, 2014.

[3] M. D. Bassett, R. J. Pearson, N. P. Fleming, and D. E. Winterbone. A Multi-Pipe Junction Model for One-Dimensional Gas-Dynamic Simulations. mar 2003.

[4] M. D. Bassett, D. E. Winterbone, and R. J. Pearson. Calculation of steady flow pressure loss coefficients for pipe junctions. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 215(8):861–881, aug 2001.

[5] Fred W Blaisdell and Philip W Manson. Loss of energy at sharp-edged pipe junctions in water conveyance systems. 1963.

[6] Dejan Brkić. Review of explicit approximations to the Colebrook relation for flow friction. 2011.

[7] Hardy Cross. Analysis of Flow in Networks of Conduits or Conductors, 1936.

[8] Dmitri Kuzmin. Introduction to Computational Fluid Dynamics.

[9] L.P.B.M. (Leon Peter Bernard Marie) Janssen and M.M.C.G. (Marinus Maria Cornelis Gerardus) Warmoeskerken. *Transport phenomena data companion.* VSSD, 2006.

[10] Richard Miller. *Internal flow systems: a guide to losses in pipe and duct systems.* 1996.

[11] Stephen O'Grady. The redmonk programming language rankings: June 2015, 2016.

[12] Lindell E Ormsbee. The History of Water Distribution Network Analysis: The Computer Age.

[13] Vasava Paritosh. Fluid flow in T-junction of pipes, 2007.

[14] William H. Press. *Numerical recipes in C : the art of scientific computing.* Cambridge University Press, 1992.

[15] Zachary B Sharp. Energy Losses in Cross Junctions. 2009.

[16] Prabhata K. Swamee and Akalank K. Jain. Explicit eqations for pipe-flow problems. 102(5):657–664, 1976.

[17] Khanh Tuoc. On the Blasius correlation for friction factors.

A Junction model extension for n inlets and m outlets

In section 3 a pressure loss model for junctions with arbitrary amounts of in-/outlets is derived. However, the derivation only accounts for either one inlet and multiple outlets, or one outlet and multiple inlets. This appendix will give an extension to n inlets and m outlets. Note that during the derivation it will be assumed that any inlet can flow to any outlet, in reality this does not have to be true as flows can not *cross* each other.

Consider a junction with n inlets and m outlets, call the set of inlets I and the set of outlets O conform section 3. Then we can define the total mass flow to be

$$M = \sum_{i \in I} \dot{m}_i = \sum_{j \in O} \dot{m}_j, \tag{A.1}$$

where $\dot{m}_i$ is the mass flow rate in in-/outlet i and the last equality is a result of mass continuity in combination with an incompressible fluid.

Now consider an outlet $j \in O$. We can assume that every inlet $i \in I$ of the junction has a flow towards j proportional to its total mass flow rate. Thus an inlet with a large mass flow rate supplies more mass to i then an inlet with a small mass flow rate. Mathematically this can be formulated as

$$\dot{m}_{i,j} = \alpha \cdot \dot{m}_i, \tag{A.2}$$

where $\dot{m}_{i,j}$ is the flow from inlet i to outlet j and $\alpha \in \mathbb{R}$ is the proportionality factor. Note that the sum of the mass flow rates toward j has to be equal to the mass flow rate of j, thus

$$\begin{aligned}
\dot{m}_j &= \sum_{i \in I} \alpha \cdot \dot{m}_i, \\
\dot{m}_j &= \alpha \cdot \sum_{i \in I} \dot{m}_i, \\
\dot{m}_j &= \alpha \cdot M, \\
\alpha &= \frac{\dot{m}_j}{M}.
\end{aligned}$$

Therefore,

$$\dot{m}_{i,j} = \frac{\dot{m}_i \cdot \dot{m}_j}{M} \qquad \forall (i,j) \in I \times O. \tag{A.3}$$

Note that $\dot{m}_i \leq M$, which implies $\dot{m}_{i,j} \leq \dot{m}_j$ and similarly $\dot{m}_j \leq M$ implies $\dot{m}_{i,j} \leq \dot{m}_i$.

Now consider an arbitrary combination of inlet and outlet, $(i,j) \in I \times O$. Then the flow from i to j is given by equation (A.3). But as $\dot{m}_{i,j} \leq \dot{m}_j$ there must exist a part A'_j of the cross section A_j such that all flow from i goes through that cross sectional area A'_j. Similarly, there must exist and $A'_i \leq A_i$ such that all flow going towards j must go through that. This situation is schematically drawn in figure A.1. Note that we assume A'_i consists of one part and is as close to j as possible, and the same for A'_j and i. Then by assuming radially uniform flow velocity we get

$$\begin{aligned}
m_{i,j} &= \frac{\dot{m}_i \cdot \dot{m}_j}{M}, \\
A'_i \cdot \rho \cdot u_i &= A_i \cdot \rho \cdot u_i \frac{\dot{m}_j}{M}, \\
A'_i &= A_i \dot{m}_j M,
\end{aligned} \tag{A.4}$$

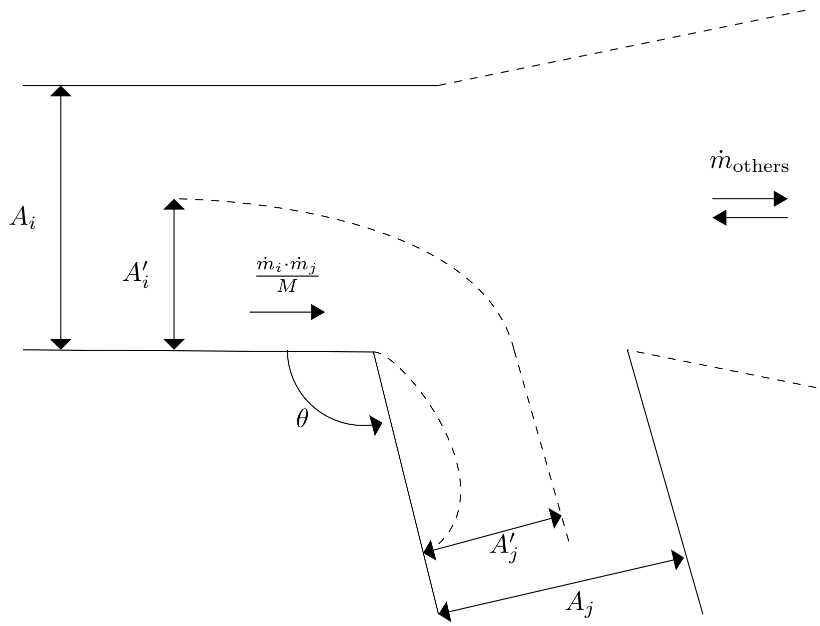


Figure A.1: A schematic representation of an inlet i and outlet j of a junction with n-inlets and m-outlets.

and using an equivalent manner we get

$$A'_j = A_j \frac{\dot{m}_i}{M}. \tag{A.5}$$

Using the definition of the area ratio we get

$$\begin{aligned}
\psi'_{i,j} &= \frac{A'_i}{A'_j}, \\
&= \frac{A_i \frac{\dot{m}_j}{M}}{A_j \frac{\dot{m}_i}{M}}, \\
&= \frac{A_i \dot{m}_j}{A_j \dot{m}_i}, \\
\psi'_{i,j} &= \psi_{i,j} q_{i,j}.
\end{aligned}$$

Now by considering the surface A'_i to be an inlet and A'_j to be an outlet of a junction with only combining or separating flow, which we can do because $q'_{i,j} = 1$ in that case, we can use the model of section 3. From which it follows that

$$\begin{aligned}
C'_{i,j} &= 1 - \frac{1}{\psi'_{i,j} q'_{i,j}} \cos\left(\frac{3}{4}(\pi - \theta)\right), \\
&= 1 - \frac{1}{\psi_{i,j} q_{i,j} \cdot 1} \cos\left(\frac{3}{4}(\pi - \theta)\right), \\
C'_{i,j} &= C_{i,j},
\end{aligned} \tag{A.6}$$

and therefore the model as derived in section 3 can be used for a junction with n-inlets and m-outlets, given that the crossing of flows is neglected.

B Implementation source-code

The pipe flow junction model with the junction model of section 3.4 incorporated has been implemented in Python. The implementation has been built as a framework allowing for easy solving of arbitrary networks. An example setup for a three pipe junction network is given in source-code 1. The framework itself is printed in source-code 2.

The code is also available at: **https://github.com/realtwister/FlowNetworkLibrary**

Source-code 1: Use case for the pipe flow junction model with junction model incorporation. Consisting of a three pipe junction network.

```
from FlowNetworkOnEdge import *                       # Import the framework
from scipy.optimize import root                       # Import the numerical solver

N = FlowNetwork()                                     # Create the class instance

N.addnodes([('a',{'s' : 1}), ('b',{}), ('c',{})])     # Create three nodes a,b,c with node c
# having inflow 1 kg/s

N.setJunction('j')                                    # Create a new node j and set it to be a
                                                      # junction, this implies that the inflow
                                                      # on this node is 0 kg/s

N.addcomponents([('a', 'j', {'A':1, 'pin':0,'thetaout': 0}),
    ('c', 'j',{'A': 1,'thetaout': 180}),
    ('b','j', {'A':1,'pin':0,'thetaout': 90})])       # Add the components with their
                                                      # respective angles

(dict, vec) = N.getunknowns(True)                     # Get the initials list

sol = root(N.residue, vec,method='krylov')            # Solve the system using the Newton-
                                                      # Krylov algorithm

result = N.getresult()                                # Get the results network

print result.info(True,True)                          # Print the final network.
```

Source-code 2: Implementation of the pipe flow network model with junction model incorporation as described in algorithm3

```
import networkx as nx
import collections
import numpy as np
from scipy.special import expit as sig
from collections import defaultdict
from copy import deepcopy


class FlowNetwork:
    """Base class for flow networks"""

    def __init__(self, rho=998.0, mu=8.9 * 10 ** -4, nodes=None):
        """Initialize a flowNetwork by creating a graph from the nodes and setting rho and mu.

        Args:
            rho (float): Density
            mu (float): Dynamic viscosity
            nodes (Nodes datatype): Nodes to add directly
        """
        self.rho = rho                          # Density of the fluid
```

```
        self.mu = mu                             # Dynamic viscosity of the fluid
        self.graph = nx.DiGraph(nodes)
        self.junctionmodel = self.standard_junctionmodel
        self.pipemodel = self.standard_pipemodel
        # Specify standard variables for nodes and components
        self.nodeVariables = {'s': 0}
        self.componentVariables = {'A': 1.0, 'm': 1.0, 'pin': 1.0*10**5, 'pout': 2.0*10**5}
        self.translation = {}
        self.vec=[]

    def addnodes(self, nodes):
        """Add nodes to the flowNetwork.
        :param nodes: the nodes to add
        """
        if not isinstance(nodes, collections.Iterable):
            # Add single node
            self.graph.add_node(nodes)
        elif isinstance(nodes, tuple):
            # Add nodes with data
            if not isinstance(nodes[0], collections.Iterable):
                # Single node with data
                self.graph.add_node(*nodes)
            else:
                # List of nodes with same date
                self.graph.add_nodes_from(nodes[0], **nodes[1])
        else:
            # Add any list of nodes
            self.graph.add_nodes_from(nodes)

    def addcomponents(self, edges):
        """Add components to the flowNetwork.

        Args:
            edges (multiple): the components to add
        """
        if isinstance(edges, tuple):
            if isinstance(edges[0], list):
                # Add multiple edges with same data
                self.graph.add_edges_from(edges[0], **edges[1])
            else:
                # Add single edge
                self.graph.add_edge(*edges)
        else:
            # Add multiple edges with or without own data
            self.graph.add_edges_from(edges)

    def setJunction(self, nodes):
        """Set a specific node to be a junction

        Args:
            nodes (String):
        """
        for node in list(nodes):
            self.addnodes((node, {'junction': True, 's': 0}))

    def setJunctionModel(self, func):
        """Set the junction model to use

        Args:
            func (reference): reference to the function to use.
        """
        self.junctionmodel=func

    def setPipeModel(self, func):
        """Set the junction model to use
```

```
        Args:
            func (reference): reference to the function to use.
        """
        self.pipemodel=func

    def getJunction(self, node, vec):
        """ Get the junction type as used by the junction model for the given node.

        Args:
            node (node type): The node to get the junction for
            vec (object): The variable vector
        """
        junction = {'in': [], 'out': []}
        for component in self.graph.in_edges_iter(node[0],data=True):
            direction = 'in'
            m = self.getval((component[0], component[1]), vec, 'm')
            if m < 0:
                direction = 'out'
                m *= -1
            junction[direction].append({'theta': (component[2]['thetaout'] if 'thetaout' in component[2] else 0
                                        'p': self.getval((component[0], component[1]), vec, 'pout'),
                                        'm': m,
                                        'A': self.getval((component[0], component[1]), vec, 'A')})

        for component in self.graph.out_edges_iter(node[0], data=True):
            direction = 'out'
            m = self.getval((component[0], component[1]), vec, 'm')
            if m < 0:
                direction = 'in'
                m *= -1
            junction[direction].append({'theta': (component[2]['thetain'] if 'thetain' in component[2] else 0
                                        'p': self.getval((component[0], component[1]), vec, 'pin'),
                                        'm': m,
                                        'A': self.getval((component[0], component[1]), vec, 'A')})
        return junction

    def getunknowns(self, vector=False):
        """Find the unknowns of the network and create a vector with initial values
        :return: unknown dict, (solution vector)

        Args:
            vector (bool): should the initial value vector be returned?
        """
        unknown = {'nodes': defaultdict(dict), 'components': defaultdict(dict)}
        vec = []
        i = 0
        for key, value in self.nodeVariables.iteritems():
            n=0;
            sum=0;
            for node in self.getnodes(True):
                if key in node[1]:
                    sum+=np.abs(node[1][key])
                    n+=1;
            if n>0:
                self.nodeVariables[key] = sum/n;
        #print self.nodeVariables
        for node in self.getnodes(True):
            # For every node check if all keys have a value, if not create a
            # vector entry
            for key, value in self.nodeVariables.iteritems():
                if key not in node[1]:
                    unknown['nodes'][node[0]][key] = i
                    vec.append(value)
                    i += 1
```

```
    for key, value in self.componentVariables.iteritems():
        n=0
        sum=0
        for component in self.getcomponents(True):
        # For every component check if all keys have a value, if not create a
        # vector entry
            if key in component[2]:
                sum+=np.abs(component[2][key])
                n+=1;
        if n>0:
            self.componentVariables[key] = sum/n;
        elif key is 'm':
            self.componentVariables[key] = self.nodeVariables['s'];
    #print self.componentVariables
    for component in self.getcomponents(True):
        # For every component check if all keys have a value, if not create a
        # vector entry
        for key, value in self.componentVariables.iteritems():
            if key not in component[2]:
                unknown['components'][(component[0], component[1])][key] = i
                vec.append(value)
                i += 1
    self.translation = unknown
    self.vec = vec
    if vector:
        return unknown, vec
    return unknown

def getresult(self, vec):
    result = deepcopy(self)
    if self.translation is {}:
        print 'Error: network unknowns are unknown.'
        return False
    for node in self.translation['nodes']:
        for (k,v) in self.translation['nodes'][node].iteritems():
            #print node
            #print k
            result.addnodes((node,{k:self.getval(node, vec, k)}))
    for component in self.translation['components']:
        for (k,v) in self.translation['components'][component].iteritems():
            result.addcomponents([(component[0],component[1],{k:self.getval((component[0], component[1]), 
    return result

def setrho(self, rho):
    """ Set the fluid density

    Args:
        rho (float): fluid density
    """
    self.rho = rho

def setmu(self, mu):
    """ Set the fluid dynamic viscosity

    Args:
        mu (float): Fluid dynamic viscosity
    """
    self.mu = mu

""" Cost Functions"""

def residue(self, vec=None):
    """
```

```
        The root function of the flowNetwork.

        Args:
            vec (list): the solution vector
        """
        cost = []
        self.vec = vec
        for node in self.getnodes(True):
            # Conservation of mass for every node.
            # add source
            masseq = self.getval(node[0], vec, 's')
            if 'junction' in node[1]:
                junceq = self.callJunctionModel(node, vec)
            else:
                junceq = self.callJunctionModel(node, vec, junction= False)
            for component in self.graph.in_edges_iter(node[0]):
                # add incoming mass flows
                masseq += self.getval((component[0], component[1]), vec, 'm')
            for component in self.graph.out_edges_iter(node[0]):
                # subtract outgoing mass flows
                masseq -= self.getval((component[0], component[1]), vec, 'm')

            #print 'continuity:'+str(masseq)
            cost.append(masseq)
            cost.extend(junceq)

        for component in self.getcomponents(True, False):
            # Pressuredrop over every component
            # calculate pressure drop (k*( m/A)^2)
            l = 1.0 if 'l' not in component[2] else component[2]['l']
            cost.append(self.pipemodel(
                pin=self.getval((component[0], component[1]), vec, 'pin'),
                pout=self.getval((component[0], component[1]), vec, 'pout'),
                m=self.getval((component[0], component[1]), vec, 'm'),
                A=self.getval((component[0], component[1]), vec, 'A'),
                l=l,
                rho=self.rho,
                mu=self.mu))
        return cost

    def noJunctionResidueModel(self, node, vec):
        """ Calculate the residue in case of no junction, all pressures should be the same.

        Args:
            node (node type): The junction node
            vec (list): The solution vector
        """
        ps = []
        for component in self.graph.in_edges_iter(node[0]):
            # add incoming mass flows
            ps.append(self.getval((component[0], component[1]), vec, 'pout'))
        for component in self.graph.out_edges_iter(node[0]):
            # subtract outgoing mass flows
            ps.append(self.getval((component[0], component[1]), vec, 'pin'))
        eq = []
        for i in range(len(ps) - 1):
            eq.append(ps[i] - ps[i + 1])
        return eq

    def callJunctionModel(self, node, vec, junction=True):
        """ Call the set junction model and return residue

        Args:
            node (node type): The junction node
            vec (list): The solution vector
```

```
        """
        if junction:
            return self.junctionmodel(self.getJunction(node, vec), self.rho)
        return FlowNetwork.nofriction_junctionmodel(self.getJunction(node, vec), self.rho)

    @staticmethod
    def nofriction_junctionmodel(junction, rho):
        """ Calculate the residue of the function according to the no friction model.

        Args:
            junction (junction type): The junction
            rho (float): Fluid density
        """
        eqs = []
        if len(junction['in'])>0:
            ref = junction['in'][0]
            junction['in'].pop(0)
        else:
            ref = junction['out'][0];
            junction['out'].pop(0)
        for i in junction['in']:
            eqs.append(ref['p']-i['p'])

        for j in junction['out']:
            eqs.append(ref['p']-j['p'])
        return eqs

    @staticmethod
    def standard_junctionmodel(junction, rho):
        """ Calculate the residue of the function according to the new model.

        Args:
            junction (junction type): The junction
            rho (float): Fluid density
        """
        junction = junction.copy()
        eqs = []
        if len(junction['in'])<1:
            #print 'no ref'
            return FlowNetwork.nofriction_junctionmodel(junction,rho)
        ref = junction['in'][0];
        junction['in'].pop(0)
        for j in junction['out']:
            uj = j['m'] / (j['A'] * rho)
            q = j['m'] / ref['m'] #TODO: Deling door nul
            psi = ref['A'] / j['A']
            theta = np.abs((180-np.abs(ref['theta'] - j['theta']) % 360)-180) * np.pi / 180.0
            #print 'theta:'+str(theta/np.pi*180)
            eqs.append(ref['p']-j['p']-FlowNetwork.C(uj, q, psi, rho, theta) * rho * uj ** 2)

        M = 0
        for j in junction['out']:
            M += j['m'] #TODO: kan nul worden

        for i in junction['in']:
            sol = i['p']-ref['p']
            for j in junction['out']:
                uj = j['m'] / (j['A'] * rho)
                q = j['m'] / i['m']
                psi = i['A'] / j['A']
                theta = np.abs((180-np.abs(i['theta'] - j['theta']) % 360)-180) * np.pi / 180.0
               # print 'theta:'+str(theta/np.pi*180)
                qref = j['m'] / ref['m']
                psiref = ref['A'] / j['A']
                thetaref = np.abs((180-np.abs(ref['theta'] - j['theta']) % 360)-180) * np.pi / 180.0
```

```python
            sol -= j['m'] / M * (FlowNetwork.C(uj, q, psi, rho, theta)-
                                 FlowNetwork.C(uj, qref, psiref, rho, thetaref)) * rho * uj ** 2
        eqs.append(sol)

    # return the equations
    return eqs

@staticmethod
def C(uj, q, psi, rho, theta):
    """ Calculate the pressure difference in a junction

    Args:
        uj (float): flow speed of fluid in outgoing pipe
        m (float): massflow ratio
        psi (float): area ratio between pipes
        rho (float): density of the fluid
        theta (float): the angle between the pipes
    """
    return 1.0-np.cos(3.0/4.0*(np.pi-theta))/(psi*q)

@staticmethod
def standard_pipemodel(pin, pout, m, A, l, rho, mu):
    """ Calculate the pressure difference in a pipe

    Args:
        pin (float): pressure at ingoing pipe
        pout (float): pressure at outgoing pipe
        m (float): massflow through pipe
        A (float): crossectional area of pipe
        l (float): length of pipe
        rho (float): density of the fluid
        mu (float): dynamic viscosity of fluid
    """
    Re = 2.0*np.abs(m)/(mu*np.sqrt(A*np.pi))
    #print Re
    f = sig(-4.5*(Re/1000.0-3.0))*64.0/Re+sig(4.5*(Re/1000.0-3.0))*0.079/Re**0.25
    f = 64.0/Re
    dp = pin - pout
    if m < 0:
        dp *= -1.0
    return dp - f*l*m**2/(rho*A**2)*np.sqrt(np.pi/A)

""" HELPER FUNCTIONS """

def getnodes(self, data=False):
    return self.graph.nodes(data=data)

def getcomponents(self, data=False, node_data=False):
    """
    get all components of the network
    :param data: Need data from components
    :param node_data: Need data from nodes
    :return: Components (with data)
    """
    components = self.graph.edges(data=data)
    if node_data:
        for n, component in enumerate(components):
            components[n] = ((component[0], self.graph.node[component[0]]),
                             (component[1], self.graph.node[component[1]]), component[2])
    return components

def getval(self, obj, vec, key):
    """
    Get value if in graph otherwise get from solution
    :return: the data asked for
```

```python
        Args:
            obj (object type): either the node or the component
            vec (list): Solution vector
            key (string): variable key

        """
        if not isinstance(obj, tuple):
            if key in self.graph.node[obj]:
                return self.graph.node[obj][key]
            return vec[self.translation['nodes'][obj][key]]

        (a, b) = obj
        if key in self.graph[a][b]:
            return self.graph[a][b][key]
        return vec[self.translation['components'][obj][key]]

    def info(self, nodes=False, edges=False):
        """ Print some general info about the network.

        Args:
            edges (bool): print edges
            nodes (bool): print nodes
        """
        # print(nx.info(self.graph))
        if edges and nodes:
            return (self.graph.edges(data=True),self.graph.nodes(data=True))
        if nodes:
            return {node[0]:node[1] for node in self.graph.nodes(data=True)}
        if edges:
            return {(comp[0],comp[1]):comp[2] for comp in self.graph.edges(data=True)}

        print(nx.info(self.graph))
```