One-dimensional positive displacement pump description in cavitating conditions

PhD Thesis

Francesco Rizzuto

Research Group

Mechanical and Aerospace Department

University of Strathclyde, Glasgow

Declaration of author's rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed:

Francesco Rizzo

Date: 17/09/2019

Abstract

The scope of the research contained in this thesis is to develop a new mathematical algorithm to model complex pump systems, capturing cavitation and pump behaviour. The approach consists of the solution of the hyperbolic wave equation, including compressibility and multiphase conditions. The second phase contains non-condensable gas and vapour formation, both important in the density and speed of sound variation. The adopted solution scheme is a finite volume method with a Monotonic Upwind Scheme for Conservational Law (MUSCL). This algorithm is second-order accurate in time and space, with a total variation diminishing (TVD) scheme to prevent spurious oscillation. In order to introduce a dissipation due to friction at the wall in a quasi-steady formulation, a source term is solved with a splitting method. To validate the code, the new simulation methodology was first applied to transient flow in a straight pipe with water hammer. The results were compared with results from pre-existing methodologies available in the literature. Thereafter, the algorithm was applied to a single chamber positive displacement diaphragm pump and then to a triplex diaphragm pump and the results compared with experimental data from an industrial test rig for both single chamber pump and multiple triplex pump network. The simulations coped with a wide range of working pump conditions and were capable of giving information on pressure pulsation, mass flow rate and volume fraction of the vapour formation inside the entire domain. The results modelled correctly the main pump behaviour especially for low cavitation formation, although cavitation was underestimated in same cases. Moreover, differences were evident out in the case of high pump rotation speed where the vapour formation also affected the discharge phase. For that condition, the algorithm was not able to perform correctly, limiting the use of the code. The algorithm may be easily extended to different positive displacement pump configurations, including a diaphragm pump where different liquids are on the driven and driving sides of the diaphragm. Such a hydraulically driven diaphragm pump requires an intermediate flow which transfers the information from the piston to the membrane. This may be embedded in the algorithm. The capability of the new algorithm to cope with different design layouts to work as a pre-design tool has been highlighted as has its ability to simulate not only the pump behaviour but also the system network response to which the pump is attached. From an industrial point of view, a reduction in terms of simulation effort with high fidelity results permits a reduction in costs for the design process and an improvement in the knowledge of a pump's operating process. Moreover, it is possible to include the practical operation characteristic, often neglected, permitting a better estimate of the $NPSH_R$ by simulation.

Acknowledgements

I would like to express my gratitude to my primary and secondary supervisors for their help throughout my years at the University of Strathclyde: Dr Matthew Stickland and Dr Bill Dempster. I am thankful to the European Union for giving me the opportunity to be part of the Marie Curie fellowship. Weir Group Netherlands, with Ralph van Rijswick always busy but always helping and sharing experience. Stephan Hannot who helped during the Wier experience and a special thanks to Geert Copper with whom I shared the Brazilian experience and without whom the experimental test would never have been performed. I would also like to thank those who helped me throughout my PhD, especially in life, and who made the difference in the two places where I undertook my PhD studies: Glasgow and Eindhoven. Marco Mannisi, who, no matter what, was always with me. Andrea Dell'Isola a good friend and climbing partner. Alex Josifovic, a colleague always with a good joke. Ashwin, who is still helping me. Alejandro, the Mexican guy that shared with me the Netherlands's life experience. My family and Maria Luisa always with me to push me further to open my mind. I am grateful to my old friends that are the proof that distance is a matter of kilometres not of feeling. Professor Arris Tijsseling who kindly invited me to the TUe University and gave me the possibility to work with him. Thanks to all that I had the opportunity to know and work with over this amazing PhD period.

Contents

Nomenclature

Acronism

∂	partial derivative
$APESA$	Advanced Pump Engineering for Severe Applications
API	American Petroleum Institute
BDC	Bottom dead centre
$DGCM$	Discrete Gas cavity model
DNS	Direct Numerical Simulation
$DVCM$	Discrete Vapour cavity model
FV	Finite Volume
LES	Large Eddy simulation
LPM	Lumped parameter method
MOC	Method of Characteristic
$MUSCL$	Monotone upwind scheme for conservative law
NCG	Non condensible gas
NCG_p	Non condensible gas parameter

$NPSH$	Net positive suction head
ODE	Ordinary differential equation
PD	Positive displacement
PDE	Partial differential equation
$RANS$	Raynold Average Navier-Stokes
SPM	Stroke per minute
TDC	Top dead centre
TVD	Total Variation diminishing

Greek

$\mathbf{\Lambda}$	Transformation matrix
Δ	Variation
η_{vol}	Volumetric efficiency
Γ	Non dimensional friction function
Λ	Eigenvalue
λ	rod length and crankshaft radius ratio
μ_s	friction factor between pipe and particles
ν	Viscosity $\left[\frac{m^2}{s}\right]$
Ω	Transformed variable
ω	angular speed $\left[\frac{rad}{s}\right]$
ω_s	Particle settling velocity $\left[\frac{m}{s}\right]$
ϕ_h	Hydraulic diameter

Ψ	Pressure force coefficient
ρ	density $\left[\frac{kg}{m^3}\right]$
θ	Crankshaft angle [deg]
ε	roughness surface
ζ	loss coefficient

Subscription

0	Initial condition
c	Chamber value
d	Discharge value
g	gas
L	Liquid
s	Suction value
v	Vapour value

Superscription

-1	Inverse matrix
L	Left
n	Time step
R	Left

Variables

$\boldsymbol{S}$	Source term
$\boldsymbol{u}$	Fluid velocity $\left[\frac{m}{s}\right]$

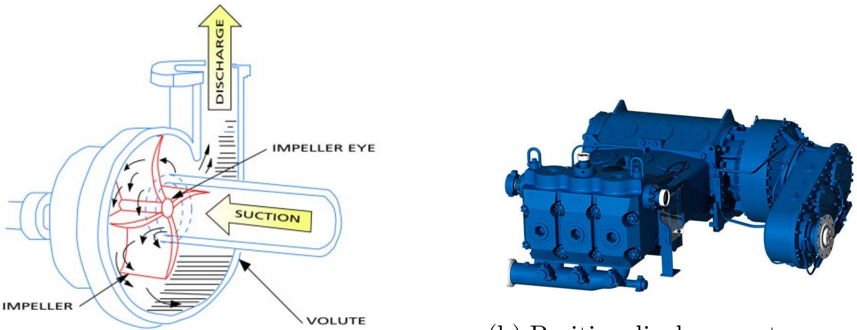
$\dot{\boldsymbol{u}}$	Fluid velocity time derivative $[\frac{m}{s^2}]$
A	Area $[m^2]$
C	Lumped Capacitance
c	Speed of sound $\left[\frac{m}{s}\right]$
C_d	Damping force coefficient of the valve $\left[\frac{Ns}{m}\right]$
C_R	Courant Number
E	Bulk Modulus [Pa]
$f(U)$	Flux function
f_n	natural frequency [Hz]
g	Gravity acceleration $\left[\frac{m}{s^2}\right]$
H	Head pressure [m]
J	Jacobian matrix
K	Comprehensive coefficient
k	Brunone friction term, decay factor
K_s	Spring coefficient $\left[\frac{N}{m}\right]$
L	Lumped Inductance
l	Rod length [m]
M	Mach number
N	Number of chambers, number of second phase
p	pressure [Bar]
Q	Volume flow rate $\left[\frac{m^3}{s}\right]$

R	Lumped resistence
r	Crankshaft radius [m]
r_{xy}	Cross correlation factor
Re	Reynold Number
S	Surface Tension
T	Temperature [K]
t	time [sec]
T_D	Time scale of radial diffusion [sec]
U	Variables vector
V	Volume $[m^3]$
V_C	Volume compressible $[m^3]$
V_L	Leakage volume $[m^3]$
V_{BF}	Backflow volume $[m^3]$
$V_{S/P}$	Piston or stroke Volume $\left[\frac{m^3}{s}\right]$
x_p	Piston displacement [m]

Chapter 1

Introduction

Fluids have always played an important role in the industrial world, where they are employed to produce and transform energy or displace material. Although the uses of fluids are wide spread, the understanding and the prediction of their behaviour is non-trivial. In the mineral and Oil & Gas industry, high-pressure fluid is employed to transport liquid, frequently containing particles or ore material. This process is commonly performed by heavy-duty pumps. Although there are a wide variety of these machines, in size, design and working principle, for the considered applications only two types are extensively used: the *Volumetric Pump* and the *Hydraulic pump*.



(a) Centrifugal pump (Hydraulic)

(b) Positive displacement pump (Volumetric)

Figure 1.1: Example of pumps.

The first one displaces pressurized fluid, after compressing it within a specific volume, whereas the second one uses an induction motion to transfer energy to the fluid. Two clear examples are shown in figure 1.1. The configuration of these machines can change in relation to the application and the required working pressure. Centrifugal pumps have a volumetric efficiency that is pressure dependent and is typically optimized for one specific design point. Their application is commonly restricted to situations where a high flow rate is required at modest pressure value. On the other hand, volumetric pumps are more suitable to handle high-pressure fluid at a constant flow rate. Ideally, a positive displacement pump's volume flow rate is independent of the pressure load. Although both types of pumps are employed for heavy-duty performance, in this research, only the positive displacement pump is considered and is shown in figure 1.2.

Positive displacement pumps with high-pressure performance are primarily used in hydraulic fracturing, to break the rock in the subsoil and extract natural gas and petroleum. Diaphragm configuration pumps are applied in the minerals environment where they are employed mainly for [1]:

- long-distance pipeline transportation,
- process feed, and
- tailing disposal.

Therefore, it is commonly required to process slurry, mud or ore fluid, and the presence of a membrane inside the machine creates two separate working chambers. This configuration reduces the wear of the pump's material in contact with mineral particulates, improving the pump reliability. Moreover, in cases of toxic, corrosive and aggressive environments (like caustic or acid fluid), the diaphragm creates a hermetically sealed chamber preventing possible contamination.

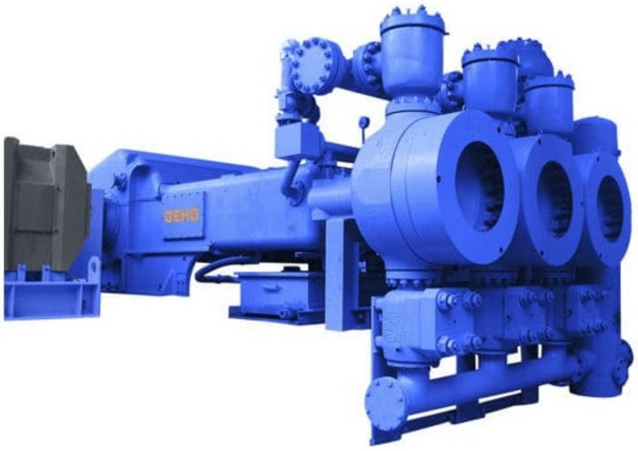


Figure 1.2: Geho Diaphragm Pump.

Considering the historical pump industry trends, the extraction of raw material has had a huge impact and improvement in the industry over the last few decades [1].

The increase in transported mineral materials from mines was accomplished by increasing at the same time the vector fluid, normally water, to keep the ratio fluid and suspended particle constant. The intention was to increase mineral production but not also increase the wear effect due to mineral concentration. However, the extraction industries have started to increase the concentration of the particles in slurries due to the scarcity of water where ores are typically produced. When the density changes, the fluid rheology is modified and the energy required to pump the fluid increases. The pump manufacturing industries responded to this variation by increasing pump size. This approach is no longer feasible. Significant cavitation phenomenon, erosion and fatigue failure appeared in these larger pumps. These phenomena are still under study, as shown by this APESA project. Therefore, a different approach must be adopted, requiring the redesign of the pump with a more detailed knowledge of the fluid behaviour. To achieve this goal an optimization procedure resulting from a better description of the pump phenomena must be embraced.

1.1 Motivation and objectives

This research study is part of the Horizon 2020, APESA Marie Skłodowska-Curie European project. The main purpose of this European funding is to improve the connection between the industrial and the academic worlds. Therefore, the purpose of APESA is to mitigate the gap between the industry and the academic environment. The APESA program, consists of different areas of research dealing with different aspects of the diaphragm pump such as fatigue and corrosion. The research developed in this thesis is focused on the simulation of the fluid dynamics and the cavitation phenomenon.

The cavitation research requirement is due to the combination of a lack of literature on the subject and industry needs. As is shown in the literature review in section 3, there are no suitable algorithms available to cope at the same time with modelling the pump fluid dynamics with high accuracy and reasonable computational efficiency. Fluids in the pump interact with the components, and in the case of multiple-chambers, the interaction with the other chambers is crucial. Even more complicated is the interaction within a system, where a multi-pump network is used. This aspect must be linked with the changing fluid phase phenomenon. Bubble formation is an undesirable effect that creates noise, vibration, pressure variation and potential pump damage. In conclusion, the creation of a broad range of tools capable of modelling cavitation in a one dimensional analysis, in a multi chamber pump and in a complex pipe networks, is the challenge of this research. The objective can be summarized as:

Development of one dimensional hydraulic modelling tools in a Matlab environment and validated by experimental data, capable of predicting cavitation *in a PD pump.*

1.2 Outline of the thesis

In chapter 2 a description of the positive displacement pump is given. The description is focused on the diaphragm pump, although the idea can be extended to all positive displacement pump configurations.

In chapter 3 the state of the art in positive displacement pump modelling and simulation is presented. In the same context, the cavitation algorithm used to simulate bubble formation is analysed. At the same time, the methodology available for modelling the fluid dynamics in a one dimensional analysis is presented.

In chapter 4 the mathematical theory for fluid simulation by the finite volume method is presented. In addition, the pump components, as well as pump models with cavitation is considered.

In chapter 5 the numerical algorithms used to evaluate the pump performance are analysed. The focus is given to the finite volume solution of partial differential equations as well as the ordinary differential equation formulations. Furthermore, the stability conditions are introduced.

In chapter 6 the validation of the code for water hammer in a long pipe is given. Comparison with experimental data taken from the literature is performed.

In chapter 7 the main research objectives are explored. Positive displacement pumps are simulated and validated with one chamber and three chamber pumps. The algorithm is analysed to understand its benefits and limitations. In addition, the model is used to analyse the pump characteristics and compared with other algorithms.

In chapter 8 conclusions with recommendations for further improvement work is given.

Chapter 2

Positive Displacement Pump

Pumps are machines widely employed in industry and their design is different depending on their application. Hydraulic pumps are machines that displace a fixed amount of fluid contained in a closed volume. Their characteristics can be classified by their working principle as rotary, reciprocating or linear. In the rotary typology, where a rotating volume is used, examples are given by vane or a screw pumps. Reciprocating pumps are subdivided into piston, plunger or diaphragm pumps and work by alternately sucking and discharging fluid. Meanwhile, linear pumps are machines where rectilinear motion is performed to create the pumping action.

Depending on the pressure and the flow rate requirement, industries use one of these categories. In detail, piston pumps are commonly used to displace high-pressure flow with high efficiency. Limitations are imposed by the motion of the seal connected to the piston and the piston stresses. For higher pressure the plunger layout performs better. Referring to figure 2.1, the seal for the plunger is much smaller and restricted to the plunger size.

The diaphragm pump is employed when the pumped fluid needs to be separated from the moving parts of the pump. Membrane pumps can operate mechanically or hydraulically depending on the driving mechanism. In the first case, the crankshaft is directly connected to the diaphragm by a

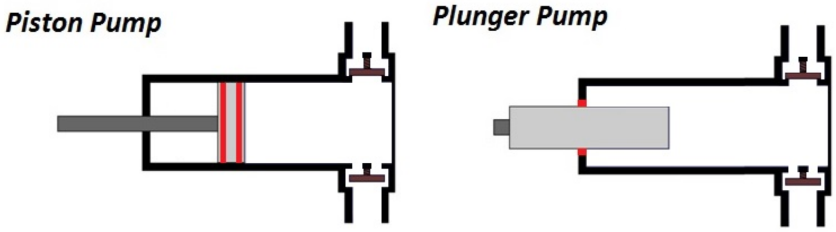


Figure 2.1: Scheme of piston and plunger, where in red is highlight the seal position.

rod, which transmits the movement and the forces directly to the diaphragm. This configuration is commonly found in small pumps where the stresses on the rubber components are small. In the case of a bigger pump, the diaphragm motion is generated by an intermediate fluid between the piston and the diaphragm; called the propelling liquid. The advantage of the combination of piston and diaphragm allows the pressure to be distributed to the entire membrane, reducing the local stresses. Diaphragm pumps have two separate volumes, hermetically sealed which allows the handling of a large number of different fluids.

2.1 Working principle

As already mentioned, the applications of piston, plunger and diaphragm pumps are different. However, their working principle is similar and the description can be given only for a diaphragm pump. Therefore, taking as a reference the cross-section of the diaphragm pump shown in figure 2.2, the main components are:

1. *Crankshaft*, the mechanical part capable of transforming the rotating motion given by an external engine to a linear reciprocating motion.

2. *Connecting rod*, the shaft that connects the crankshaft to a piston or to the crosshead.

3. *Cross head*, mechanical joint, capable of coupling the motion of the piston and the connecting rod.

4. *Connecting bar*, if present, connect the piston to the crosshead.

5. *Piston bar* is the stem that moves into the chamber. It could be considered as a single part when the diameter of the piston head is constant for the entire part, creating the plunger pump configuration.

6. *Piston head*, part of the piston system, sealed within the chamber during its motion.

7. *Safety relief valve*, a component that limits the pressure inside the system and guarantees the right operating pressure.

8. *Monitoring rod*, a system for a diaphragm configuration which aligns the membrane, avoiding asymmetric bending.

9. *Diaphragm*, rubber part that generates the displaced chamber volume. It can be moved by the propelling liquid or a connecting bar in the case of a mechanically driven configuration.

10. *Accumulator*, a hydraulic component capable of damping the pressure variation.

11. *Discharge valve.*

12. *Suction valve.*

13. *Air vessel*, different to the the accumulator, is related to the atmosphere.

14. *Chamber*, the variable volume, contained between the two valves, that hosts the diaphragm or the piston.

15. *Propelling liquid*, normally a mineral oil, is the liquid that transmits the forces from the piston to the diaphragm.

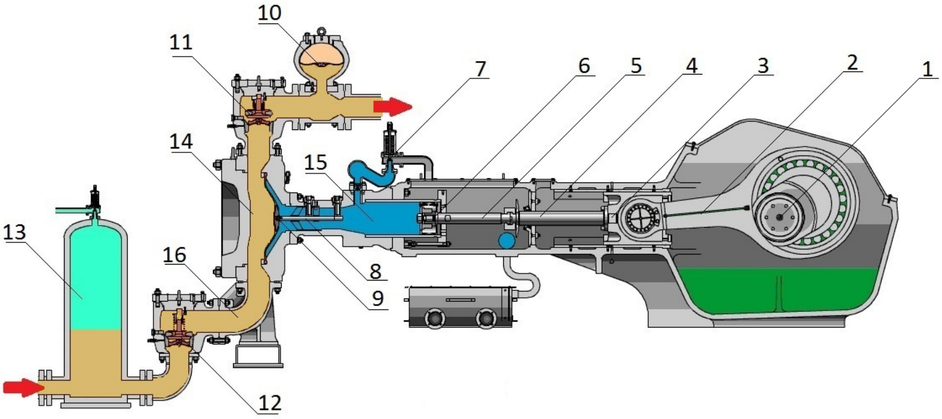


Figure 2.2: Diaphragm positive displacement pump configuration [2].

16. *Slurry fluid*, the load fluid to be delivered.

Mechanically speaking the functionality of the pump takes place from the crankshaft(1) moved by an electrical or internal combustion engine. Between these components, a gearbox is commonly interposed for reducing the rotation speed and increasing the torque. The motion of the engine is transmitted to the propelling liquid by a serial mechanical chain created respectively by a connecting rod(2), crosshead(3), connecting bar(4), a piston(5) and piston head(6). The propelling liquid(15), guided by the piston head, transmits the pressure to the diaphragm(9) held in position by the monitoring rod(8) that can move linearly into a specific bearing. Attached to the propelling liquid system, there is often a safety relief valve(7) active only when the pressure increases over a safety value. The slurry fluid(16) is pumped by the diaphragm displacement, and it is pushed into the chamber(14) across the suction valve(12) and vented through the discharge valve(11). Typically, two more hydraulic components are introduced to reduce pressure fluctuations: the air vessel(13), and the accumulator(10). This working principle can be extended to a piston or plunger pump when the propelling liquid between the piston and the diaphragm is not present.

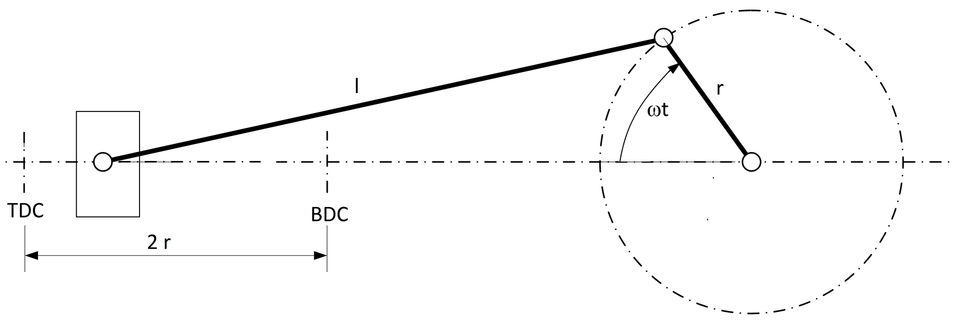


Figure 2.3: Crank mechanism.

2.2 Pump kinematics

The pump cycle is driven by the motion of the crankshaft, schematically represented in figure 2.3. The motion of the piston, considering a constant rotation speed ω, is commonly described by a sinusoidal function. The piston displacement, velocity and acceleration are given respectively by equations 2.1, 2.2, 2.3. The value of θ_0 is the initial crankshaft angle, different for each chamber of multi chambers pumps. Meanwhile, the parameter λ is the ratio between the crank radius r and the rod length l. A typical value of λ is restricted by the pump design and has values in the interval [0.1–0.2] [3].

$$x_p = r\Big[1 - cos(\omega t + \theta_0) - \frac{\lambda}{2} sin^2(\omega t + \theta_0)\Big] \tag{2.1}$$

$$\dot{x}_p \approx r\omega\Big[sin(\omega t + \theta_0) - \frac{\lambda}{2} sin(2\omega t + 2\theta_0)\Big] \tag{2.2}$$

$$\ddot{x}_p \approx r\omega^2[cos(\omega t + \theta_0) - \lambda cos(2\omega t + 2\theta_0)] \tag{2.3}$$

The displacement of the piston is not significantly affected by the value of λ, as shown in figure 2.4. However, it plays an essential role in the velocity and the acceleration profile.

Figure 2.5 shows the piston velocity of one complete crankshaft rotation.

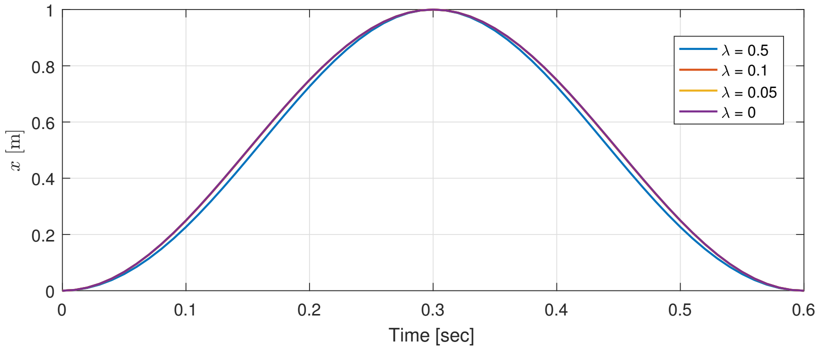


Figure 2.4: Piston position for different λ.

The performance of the pump is directly dependent on the piston velocity. As is shown in the next section, cavitation is also dependent on the piston velocity profile. The piston forces are related to the fluid inertia, and thereby its acceleration.

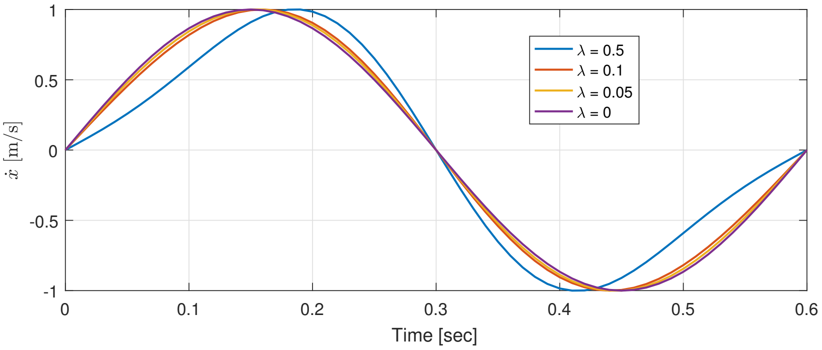


Figure 2.5: Piston Velocity for different λ.

The acceleration profile of the piston is shown in figure 2.6, where the pure sinusoidal function ($\lambda = 0$) shows minimum acceleration variation.

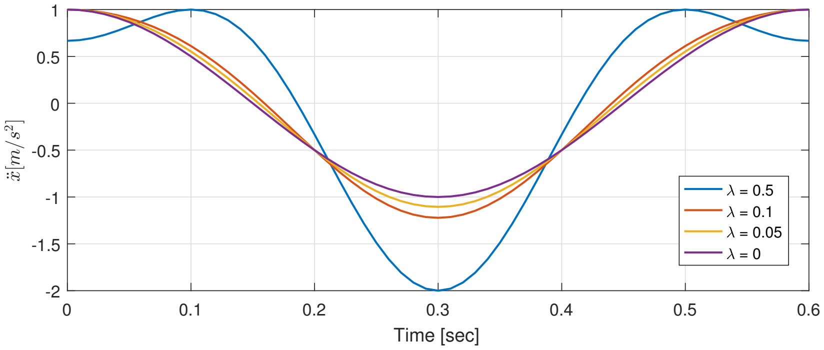


Figure 2.6: Piston accelleration for different λ.

The effect of the parameter λ is crucial in defining the velocity profile of the piston and consequently the fluid behaviour. The performance of the pump and load stresses can be improved with this parameter. However, λ has design constraints, namely rod length and crankshaft radius, that strongly limits its value to lower than 0.2.

2.3 Pump Cycle

The PD pump discharges fluid from a suction pressure to a load pressure value, theoretically with a regular and repetitive cycle. A typical pressure history in the chamber for an entire crank rotation without cavitation is shown in figure 2.7. Cavitation is a phenomenon in which the fluid changes phase and is described in details in section 3.3.

For simplicity, the fluid cycle description starts from the suction phase, where the piston position is at the maximum extension, defined as Top Dead Centre (TDC). At that point, the motion of the piston increases the volume size of the chamber and reduces the overall pressure. Ideally, the suction valve opens immediately at the start of the cycle. However, due to the forces and the fluid dynamics, there is an opening time delay. As soon as the differential pressure across the valve reaches the breakpoint value, the valve starts to lift. When the valve opens, fluid from the suction line passes through the

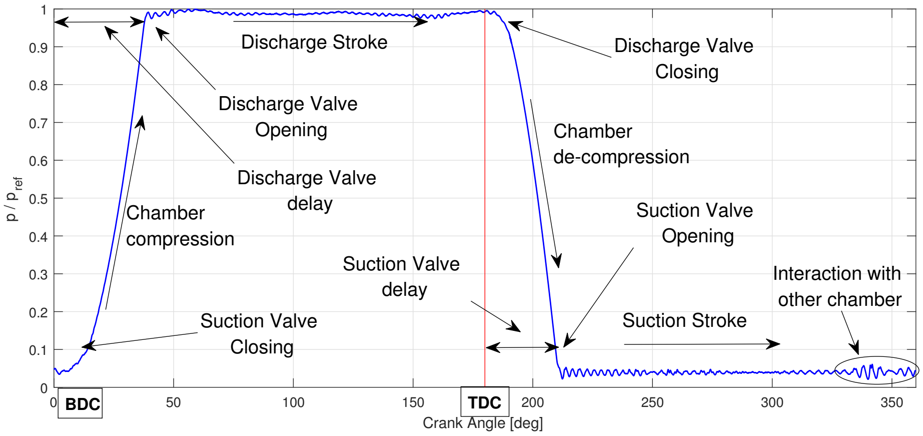


Figure 2.7: Pressure cycle in the chamber.

valve gap formed between the valve and the valve seat. The suction phase continues until the piston reaches the other extreme, Bottom Dead Centre (BDC). At that time, the piston changes its motion and decreases the volume of the chamber, starting the compression phase. However, the suction valve is retarded by an inevitable closure delay. Therefore, it is possible that part of the fluid sucked into the chamber is discharged again into the suction line, creating a back-flow. This phenomenon is a drawback that affects the volumetric efficiency. The fluid compression begins as soon as the suction valve closes. When the pressure reaches the discharge breakpoint for the discharge valve, the discharge phase occurs. However, the same issues described for the suction valve occur in the discharge manifold and back-flow can be created. The cycle is theoretically repeated in the same way for every pump rotation. However, differences made by the accumulator or external factors could slightly change the cycle. For instance, a dampener superimposes a low-frequency pulsation that is commonly smaller than the pump frequency. In the case of multi-chamber pumps, mutual interaction between pump chambers affects each chamber. This phenomenon can be seen in figure 2.8 and in detail in figure 2.7 around 330 degrees.

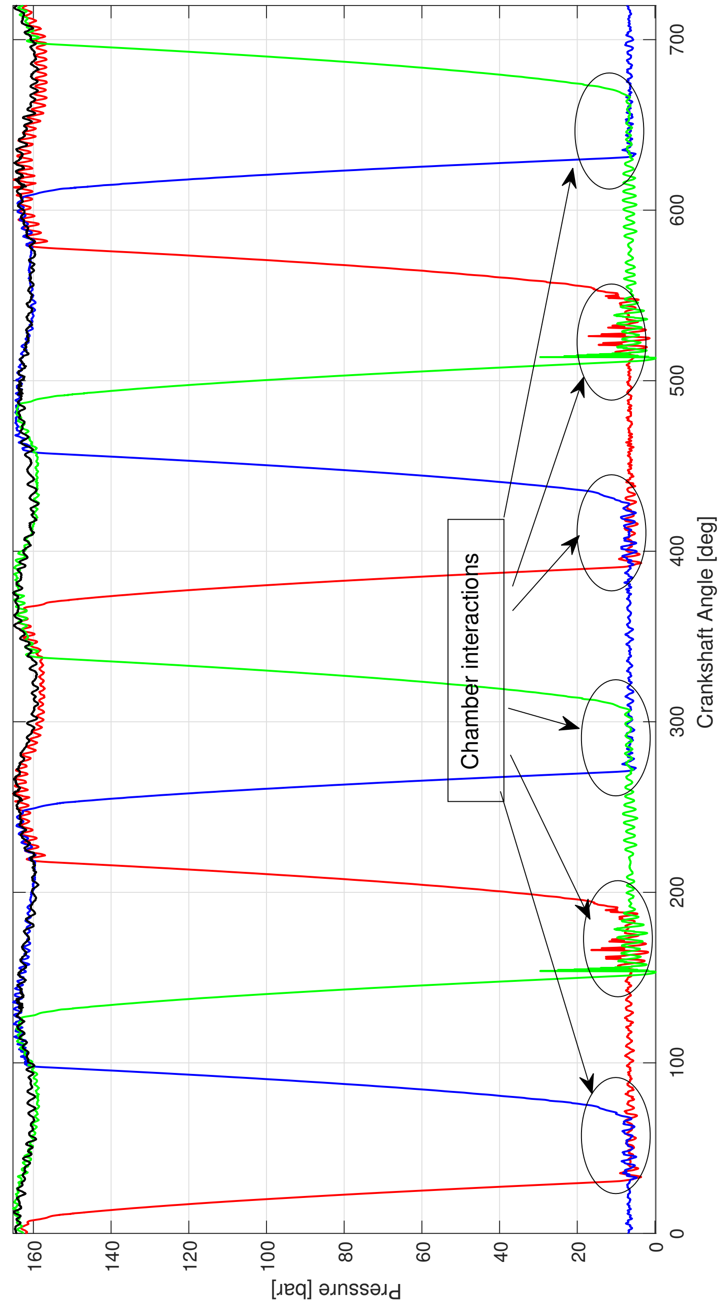


Figure 2.8: Example of pressure history for three chamber pump where the data where collected from the field.

The performance of a PD pump is defined by the volumetric efficiency, given as the ratio between the ideal and the real discharged flow. The ideal flow is equal to the piston volume displacement defined as the stroke volume V_S. The real flow is the ideal flow affected by all the losses. Therefore, the drop-in discharge flow can be caused by:

- leakages,
- back-flow phenomenon,
- cavitation.
- compressibility

Leakages can occur all around the system, especially where a seal is involved. The back-flow is related to the valves and their dynamics depending on pump speed and the working pressure. Compressibility is always present but becomes significant only for high-pressures or bigger pump sizes. Furthermore, the pump can produce vapour due to the cavitation. When the pump cavitates, the sucked volume flow rate does not vary (if not in a choked condition); however the mass flow rate decreases due to the variation in density. In this condition, before the discharge of the fluid, compressibility will be affected until the vapour is condensed. Thereby reducing the ideal discharged fluid volume. However, for small cavitation conditions, this effect is harmless. Summarising, the volumetric efficiency can be expressed with the formulation in equation 2.4.

$$\eta_{vol} = \frac{Q_{real}}{Q_{Ideal}} = 1 - \frac{V_L + V_{BF} + V_C}{V_P} \tag{2.4}$$

The compressibility of the fluid is related to the pressure by the equation of the bulk moduls 2.5 (for non cavitating conditions):

$$K = -V\frac{\partial p}{\partial V} \tag{2.5}$$

Ideally, the fluid discharged is equal to the total volume of the pump chamber. However, this condition is impossible to achieve due to engineering restrictions. Therefore, a dead volume, V_0 is formed. This volume is equal to the remaining fluid trapped in the chamber at the end of the discharge phase. The V_0 has a parasite effect on the compressibility. Considering equation 2.6 given by Tackett et al. [4], the overall volumetric efficiency can be calculated.

$$\eta_{vol} = 1 - \left(\frac{\Delta p}{K}\frac{V_S + V_0}{V_S} + V_L + V_{BF}\right) \tag{2.6}$$

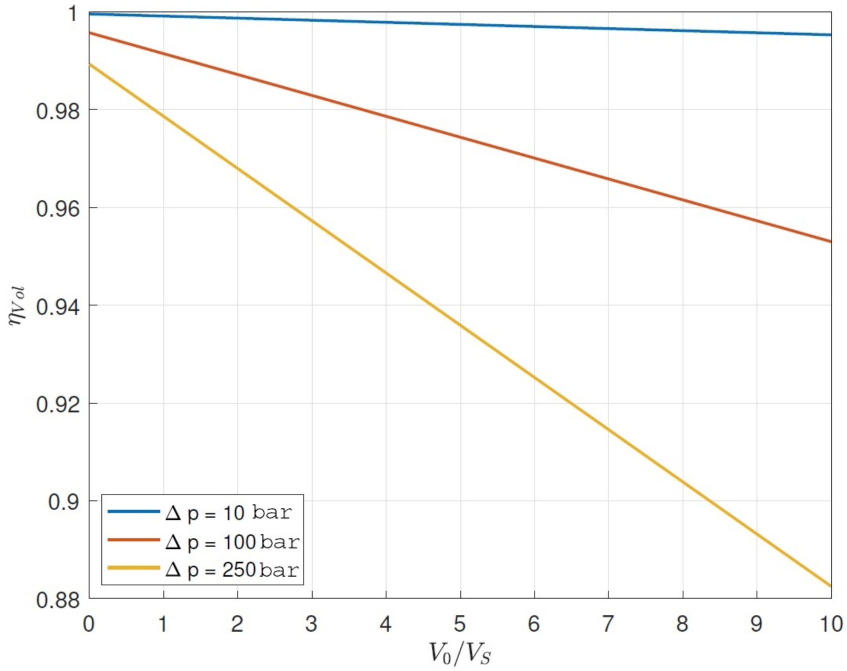


Figure 2.9: Volumetric efficiency versus the Dead Volume.

In figure 2.9 the variation of the volumetric efficiency is shown. To generate the curve, only the two compressibility terms, namely Δp and $\frac{V_S+V_0}{V_S}$ of the equation 2.6 are used. The first parameter is dominant for positive displacement pumps used for hydraulic fracking. For these pumps,

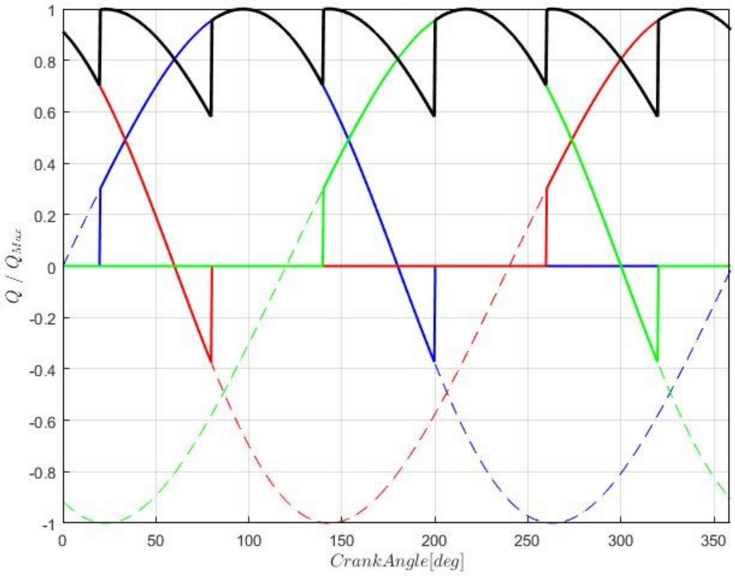


Figure 2.10: Ideal flow rate cycle.

the pressure variation can reach several hundred bar, and the dead volume is normally half the stroke volume size[3]. The volume effect is dominant in the diaphragm pump configuration, where the dead volume size can be ten times bigger than the stroke volume[1].

For multi-chambers pumps, the interaction with the chambers must be included to calculate the flow rate, pressure pulsation and efficiency. The theoretical mean flow rate displaced by a three-chamber pump is shown in figure 2.10. The overall pump flow rate is the sum of the single chamber behaviour.

However, in reality, the flow rate is affected by the fluctuation of the pressure, given by the entire system. This is a significant limitation because such a complex system is hard to model in its entirety.

Chapter 3

Literature Review

Positive displacement (PD) pumps are reciprocating machines that generate a controlled flow that is theoretically independent of the source and load pressure. Although, mechanically they are simple and the working principle is easily understood, from the perspective of the fluid dynamics their description is not trivial. PD pumps are influenced by fluid behaviour factors that affect the volumetric efficiency such as leakage, back-flow, cavitation and, in a minor way, by compressibility. They have been studied mostly with simple models and only rarely with the complete 3D Navier-Stokes equations. Three-Dimensional analysis is only worthwhile in particular cases where a detailed understanding of the fluid dynamics is required. For this reason, there is limited information in the published literature regarding the numerical simulation of fluid flow in a complete PD pump.

Following the logic of the tree structure shown in figure 3.1, the literature was analysed considering three main topics: positive displacement pumps, fluid transient model and cavitation. In the first case, the state-of-art in the pump simulation is considered. At the same time a detailed description of pump components such as valves and accumulators is given. In the second and the third section, fluid behaviour and cavitation are analysed. In all cases, the focus is directed at a one-dimensional description and the most

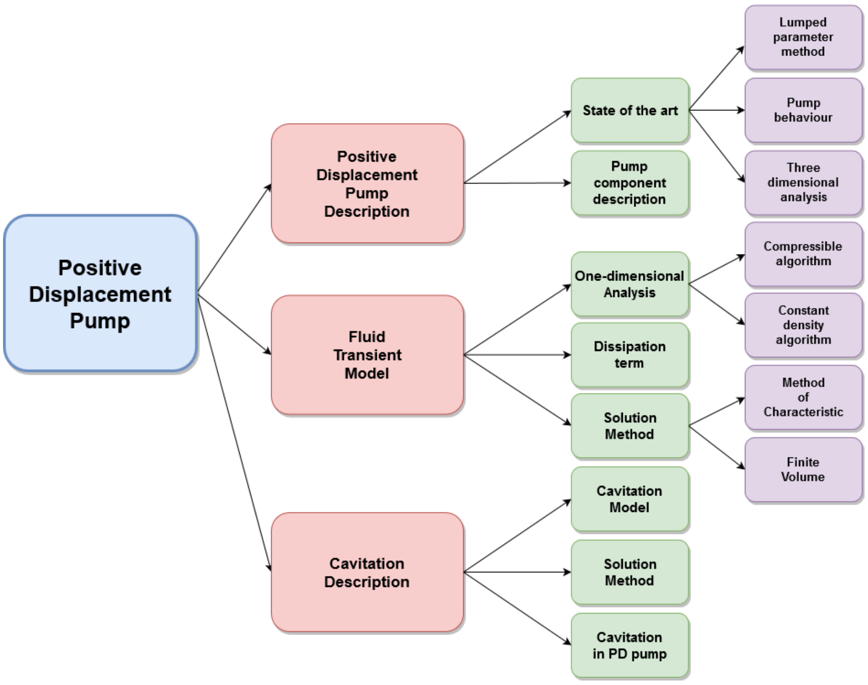


Figure 3.1: Tree Structure of the literature review.

relevant solution methodology. At the end, only brief details are given for the fluid rheology and multidimensional approaches.

3.1 Positive displacement pump: modelling behaviour

PD pumps are made of several different components that work cyclically to displace fluid at a specific pressure. The mechanical complexity is not high, and the overall fluid motion is easy to describe. However, locally in order to understand the fluid behaviour an accurate model is required. This section is subdivided considering the State of the art of the positive displacement pump description and in details the main component.

3.1.1 State of the Art

Research involving positive displacement pumps can commonly be subdivided into two categories; a complete three-dimensional analysis and a lumped parameter method (LPM). The main reason for this large gap is due to the pump's size. In order to describe the overall pump phenomena a fine numerical grid is required. This makes the three dimensional description computationally impracticable for big pumps. On the other hand, in order to predict the main behaviour of a pump, a simple model can be used. This method neglects important factors like wave reflection. In addition, there is only a limited amount of information in the published literature regarding one or two-dimensional analysis, especially with cavitation conditions. Although, there were a significant number of studies of positive displacement pumps published in the early nineties, in the last decade the amount of research in this sector has reduced. With all of the available commercial software and computational power, a three-dimensional analysis is still far too expensive for an industrial application. On the other hand, one-dimensional codes are restricted to pipe networks[5] and often neglect or oversimplify the pump behaviour[6, 7]. The only available research involving wave transfer formulation is given by Vetter and Schweinfurter[7]. In the study, the ROLAST software with the method of characteristic solver was performed, where some of the components were simulated with empirical curves. The algorithm was one-dimensional isentropic flow, neglected the fluid-structure interaction, and was used for understanding pump pulsation. It was highlighted that the amplitudes of the higher pump harmonics increased with decreasing the volumetric efficiency. Van Rijswick [8] instead, studied the frequency of field pump response of double acting (duplex) and single acting pump (triplex), simulated with the lumped parameter model. The research involved the analysis of the frequency domain for both pump types. The duplex configuration contains all the uneven and multiples of the fourth harmonics and the first harmonic is always present. Meanwhile, the single acting pump has only multiples of the number of the pistons as harmonics. In order to calculate the frequency response of each component, Van Rijswick related all the parts with three characteristics behaviours:

- capacitance,
- inductance, and
- resistance.

In the capacitance term C, all of the effect caused by the compressibility of gas and liquid was simulated. In the inductive term L, the inertia of the fluid was considered. Resistance simulated the friction and all the losses inside the system. This method permits the calculation of the natural frequency of the components with the equation 3.1.

$$f_n = \frac{1}{2\pi}\sqrt{\frac{1}{LC}} \tag{3.1}$$

The triplex pump has more advantages than the duplex pump with the same flow rate. The lower harmonic of the triplex pump is three-times higher than the duplex making it more difficult to reach resonance frequency. Moreover, Rijswick[8] analysed the beating behaviour, due to offset in frequency of the waves interactions. The beat phenomenon occurs when two waves with similar phases interact, creating a pulsation wave effect. An example is shown in figure 3.2.

This behaviour occurs in a multi-pump network, where crankshaft synchronization is performed to adjust and reduce this effect. For instance, two pumps connected to a single discharge line with similar velocity create pressure waves that interact producing the beat phenomenon. This effect is not predicted by the LPM model. In the same investigation,[8], the limitation

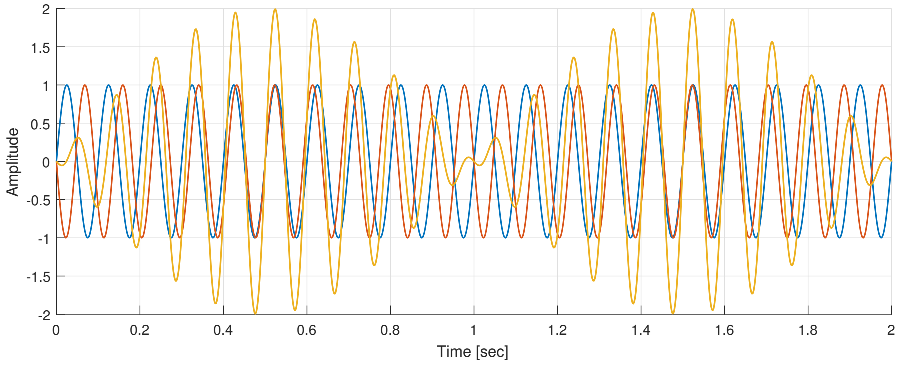


Figure 3.2: An example of the beat phenomenon, for two frequency wave of 10 and 11 Hz with 180 degrees of shift.

of the lumped and distributed parameter simulation was brought to light; the two algorithms are unable to predict high-frequency fluid response.

Although for a different application, an explanation of why an even number of chambers is worse than an odd number has been discussed by Manring[9]. Considering an idealized flow, a pump with an even number of chambers has pistons that simultaneously are at zero flow position, namely zero and 180 degrees. That is not present for pumps with an odd number of chambers that always have some pistons in a different phase. When an even chamber pump is at zero or 180 degrees, only the $\frac{N-2}{2}$ chambers are contributing to the net flow.

Considering a different technique for simulating the pump behaviour Singh and Madavan[10], used the transfer matrix technique to calculate the system in the frequency domain. The technique allowed the analysis of the frequency spectrum of the pump and network, but only for the steady condition. Interesting is the fact that the parameter $\lambda = \frac{l}{r}$ affected the fluctuations in the flow rate. When the value of the λ ratio goes to zero, the piston displacement is a pure sinusoid. In this condition, the primary flow harmonics are a multiple of the number of plungers. Mathematically the incompressible flow with high λ creates only even multiple flow harmonics.

When the connecting rod effect is included, odd multiples occur. A more straightforward approach was applied to predict the amount of cavitation in the pump by Opitz et al.[11]. The cavity calculation used the unsteady energy equation and assumed that:

- the water level in the reservoir is constant,
- the static pressure in the chamber is equal to the vapour pressure, and
- the suction manifold has a constant area.

In other words, the velocity of the fluid in the reservoir can be neglected while the velocity in the suction manifold is constant due to the constant area. The energy function has the form as an inhomogeneous first order non-linear ordinary differential equation 3.2.

$$\boldsymbol{u}^2\frac{\rho}{2}\big(\frac{fl}{2}+\sum_{i}^{n}\zeta_i+1\big)+\rho l\frac{d\boldsymbol{u}}{dt}=p_s-p_v \tag{3.2}$$

Equation 3.2 can be simplified by introducing loss, impedance and source terms to equation 3.3, which is able to be solved analytically.

$$A\boldsymbol{u}^2(t)+B\dot{\boldsymbol{u}}(t)=C \tag{3.3}$$

The methodology compares the velocity of the fluid to the piston velocity profile, for further information reference[11] is given. This algorithm required as input the valve opening time delay. Figure 3.3 gives an example of a guessed valve opening at 30 degrees crankshaft angle.

The main limitations of this method are:

- the guessing of the valve opening (required simulation or pump design experience)
- the simple layout of the pump suction line, and

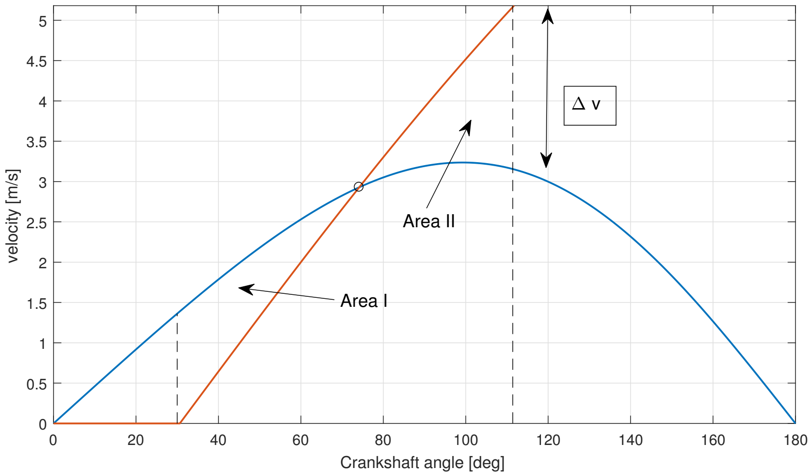


Figure 3.3: Example of Opitz[11] algorithm result, where the blue line is the piston velocity and the red line is the fluid velocity.

- the neglecting of pressure pulsation interaction.

Pei et al.[12] compared two different approaches, the approximation equation and the U. Adolph Theory. These two methods are too simple to be applied to the modern standards of the pump industry. As an example, the first method assumes the valve to be massless, to have a constant spring force and incompressible fluid. Therefore, these two methods can be considered only as a first pump approximation. Although simple, Lee et al.[13] described a positive displacement pump by following the assumption of incompressible fluid behaviour with a constant rotation of the crankshaft. As a further simplification, the suction valve was closed when the piston reached the bottom dead centre (BDC). Meanwhile, the discharge was closed when the piston reached the Top dead centre (TDC). The suction and the discharge pressure were taken as a constant value, whilst the pressure in the chamber was calculated using bulk modulus relation, with the equation 3.4.

$$\frac{dp_c}{d\theta} = -\frac{E}{\omega}\frac{\boldsymbol{Q}_c}{V_p} \tag{3.4}$$

The accumulator in the discharge line was simulated with a similar expression. Although the scope of the simulation was to compare the behaviour of normal and abnormal working of the pump, this method of simulation does not predict the behaviour of the fluid accurately. The simplification of the pump description required the presence of tuning parameters that makes this procedure impractical. A comparison between lumped and distributed parameter models was performed by Shu et al.[14]. Both methodologies could capture the main pulsation characteristic of the system. Even the peak to peak amplitudes were comparable with the measurements. The main drawback of lump parameter simulation is the neglect of the wave propagation. The travel of the pressure wave and its reflection changes the valve response as well as the chamber's dynamics. That is one of the reasons why this methodology loses the mutual interference between the different parts of the pump. However, as reported by Rijswick[8] the lumped parameter model fails to predict the high-frequency oscillations although they are better depicted in the distributed parameter model. Furthermore, the inefficient simulation in a multi chambers condition was brought to light . For an applied optimization purpose, Josifovic et al.[15] used a simple transfer function to describe the pressure cycle, but the results were too simple and not accurate. In order to cope with the simulation of a three-dimensional multi chamber pump, Josifovic et al.[6] combined together two different commercial codes. The combination of two software was used to overcome the limitation given by using each one separately. The three-dimensional analysis was used to calculate the steady valve coefficients, even in back-flow. The one-dimensional algorithm was used to analyse the overall pump behaviour. The hybrid algorithm provided a better time solution for multi-chambers than simple three-dimension analysis. PD pump simulation strongly depends on the valve model and the extrapolation of data created with three-dimensional software from a steady-state approach is not always accurate. Nevertheless, the algorithm agrees with the experimental result within a reasonable engineering approximation ($r_{xy} = 0.7474$) and in a reasonable time. However, the time required for the simulation of the three-dimensional analysis of the steady valve condition should also be considered. In this research, only PD pumps with a piston or plunger were examined. Research covering the diaphragm pump was carried out by van Rijswick et al.[16] and Blanco et al.[17]. The first based the research on the fluid-structure interaction within the pump chamber, considering a three-dimensional analysis and incompressible fluid. While, the second used commercial software to describe the unsteady operation of a double acting diaphragm pump. Van Rijswick[1] has developed a three-dimensional analysis with implicit large eddy simulation (ILES). He considered an immersed boundary method, combining a finite volume method with a non-linear finite element analysis. From his research, the non-dimensional analysis could be considered, thereby evaluating the importance of single fluid properties. In a diaphragm pump, hydraulically driven, the slurry and the propelling liquid can be treated as equal when their density and viscosity are comparable. Although this is not common in the real application where a slurry can cover a wide range of properties, the API 674/ ISO 13710 regulation for pump test, only consider water as the operating fluid. Therefore, the pump can be simulated as one fluid thanks to the similar properties of mineral oil and water[18]. This approach permits the presence of the membrane to be neglected[1, p. 2]. In conclusion, for an overview of the positive displacement pump component and design, the research by Tackett et al.[4] is given as a reference.

3.1.2 Pump component

To describe PD pumps correctly the interaction between all of the components must be included which makes the resolution of the problem even more complicated. As an example, valves are influenced by the surrounding pipe network which can affect the time delay of opening and closing. In addition, PD pumps are often working in cavitation conditions that is numerically expensive to model. For that reason, some researchers focused their attention only in a specific area where cavitation is mainly formed. Such is the case of Iannetti[3], in which the fluid vapour around the valve was modelled. Fluent ANSYS software was used, and the validation of the results was conducted with a closed loop test rig. The performance and the accuracy achieved with the three dimensional RANS method and k-ε turbulence model with moving mesh agreed with the experimental results. This methodology could be used as a potential tool in the design and optimization phase. However, due to the immense amount of computational power required, it was only feasible for limited parts of the pump. The motion of the valve was taken into consideration with a User Defined interface function (UDF), which provided the actuating forces and dynamic numerical solution. Furthermore, three main factors were highlighted:

- the importance of the mutual interaction between the chambers which were neglected in the numerical modelling,
- the cavitation algorithm accuracy and,
- the introduction of the air-gas phase in the system.

The simulation required several days[19] to perform only the suction phase for a single chamber, this makes the strategy impracticable for diaphragm positive displacement pumps, multi-chamber interaction or for a multi-pump network. At the other extreme, Johnston[20] created a modular code to predict the behaviour of a reciprocating pump with a lumped parameter method to obtain the working condition for the valve model. His research also involved cavitation formation. Although the result was in

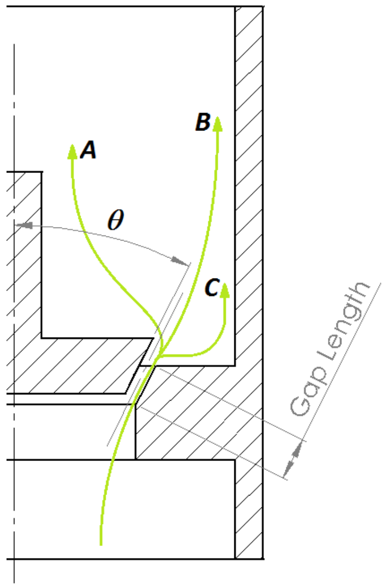


Figure 3.4: Schematic flow path in the valve[22].

agreement with experiment, the research highlighted the weakness of multi-chamber simulation. Empirical coefficients extrapolated from experimental data were required to accurate simulate the valve dynamics. However, the complete closure of the valve created numerical issues, due to the linear interpolation of these coefficients, therefore a small gap was left, simulating a small leakage. The same problem was faced and solved in a similar way by Iannetti[21] where the valve gap was simulated with a few grid layers to permit the creation of the moving mesh. With the same research, Johnston[20] brought to light the extreme importance of a good cavitation model.

In addition, from an experimental point of view, Johnston et al.[22, 23] considered the fluid response for different valve designs. Referring to figure 3.4 the experiment focused on:

- θ the cone angle of the valve,
- the seat area and
- the different shape of the valve.

The range of Reynolds number considered was in the range of 2500 to 35000, much lower than the PD pump working range. The results of the experiment provided a flow force coefficient, C_F and its variation with the valve variables. The coefficient can be used for a one dimensional empirical description of the pressure force around the valve. Furthermore, the flow pattern from the valve gap was studied. Figure 3.4 shows three main flow paths, where the Coanda effect plays an essential role in the pattern definition. For a large conical angle and small valve opening the jet is attached to the valve seat, pattern C. While, pattern A was dominant for a small valve angle. The prediction of the flow direction cannot be easily guessed from the valve design and position. Commonly, the jet takes a path in between the two paths A and C. Although the description of the flow around the valve is beyond the scope of this research, it is crucial for an accurate pump simulation. First and foremost, the one-dimensional analysis simplifies all the information in an average axial direction description. Secondly, the valve behaviour strongly depends on the shape characteristics that, in the one-dimensional simulation, are challenging to introduce directly, unless with coefficients. Therefore, these conditions must be taken into consideration to evaluate the resulting accuracy. A numerical experiment of the same kind was performed by Bernad and Resiga[24] where the ANSYS Fluent code was used. The main achievement was given by the cavitation algorithms and the evaluation of the vortex surrounding the valve chamber. Also, they highlighted the possibility of using three-dimensional analysis for a specific part but not for the whole pump simulation. In conclusion, research focusing on the pump dynamics were performed primarily for three reasons:

- to study the frequency response of the system,
- to study the main effect produced by cavitation, and
- to study the effect of pressure pulsation.

To achieve these goal a good accuracy of all the components and their dynamics must be embraced and properly modelled considering the introduction of conveniently simplification.

3.2 Fluid transient model

For describing the velocity, the pressure and the energy of the fluid, a complete fluid formulation is given by the Navier-Stokes equations. The solution of these equations is still an open question, and even a simplified formulation does not have a straightforward solution. However, for fluids that flow in closed ducts, one-dimensional equations can be used to describe the average behaviour and thereby simplify the problem. Considering the PD pump, its function is given by the change of the chamber volume. The compressing and the decompressing of the volume transfers the mechanical work to the fluid trapped between the suction and discharge valves. The variation of the volume size is generated by an actuating component and, mathematically it can be expressed as a velocity boundary condition. The variation of one of the fluid properties such as velocity or pressure creates a wave response in the fluid that travels at a specific speed. This type of system is defined as hyperbolic[25] and it is described by the eigenvalue function which provides the speed of the information update. This speed physically corresponds to the speed of sound. Therefore, every time there is a variation in the properties of the fluid, a wave travelling at the speed of sound is formed and propagated in the domain[26]. In PD pumps, each perturbation is a wave which transfers information inside the domain which then diffused due to dissipative phenomena. Similarly, the sudden open and closing of the valve, creates a water hammer phenomenon which propagates through the system. As a consequence, PD pumps can be described with hyperbolic equations. In addition, since the fluid is mainly forced in a single direction, a one-dimensional description can be used.

3.2.1 One-Dimensional Analysis

The easiest set of equations capable of capturing the wave propagation are the water hammer equations. Water hammer has been studied since the early nineteen nineties due to the potential damage to hydraulic systems[27]. Water hammer is formed when a moving fluid is suddenly stopped, creating an increase of pressure inside the system. This is potentially harmful. Also, wave reflection can be dangerous in terms of cavitation formation. The reflecting wave could decrease the head pressure and reach the vapour pressure. The water hammer equations described in the form of head pressure and velocity are given in the system 3.5[27].

$$\begin{cases} \frac{c^2}{g}\frac{\partial \boldsymbol{u}}{\partial x} + \frac{\partial H}{\partial t} = 0 \\ \frac{\partial \boldsymbol{u}}{\partial t} + g\frac{\partial H}{\partial x} = S_x \end{cases} \tag{3.5}$$

where c is the speed of sound, g the gravity's acceleration, $\boldsymbol{u}$ the fluid velocity, H the head pressure and S_x is the source term. The system of equations 3.5 contains all the physics necessary to model the wave propagation in a complex pipe system[28]. The dissipation term, S_x is included in the momentum equation as a wall friction term. In the practical review, Ghidaoui et al.[28] described the historical background and the numerical methods, used to solved these equations, before the advent of the finite volume method. The Method of Characteristic (MOC), was the most popular algorithm due to the simplicity and the limited numerical effort required. This method converges to the analytical solution for the frictionless problem, and it is commonly used by one-dimensional transient fluid software. In the same research[28], a software review found that eight of the eleven commercial codes analysed were using the MOC algorithms and only two a direct finite difference solver. More than a decade later, only one code had introduced the finite volume algorithms: FluidFlow. Commercial software present drawbacks that are not easy to overcome in order to simulate positive displacement pumps. A combination of cavitation and the variation of the speed of sound are not processed at the same time. Furthermore, new studies show the limitations given by the MOC strategy within a complex system compared to other methods that are available. Although water hammer is commonly described with one-dimensional equations, some research dealt with multi-dimensional approach. In the theoretical review made by Ghidaoui et al.[28] a two dimensional analysis was made. However, new research has employed multi-dimensional approaches and Saeml[36] used commercial software to study the two and three dimensional effect of the valve closure phenomenon. It emphasised that the presence of dissipation phenomena that involves all the directions due to the turbulence produced at the valve was important. Therefore, one-dimensional analysis underestimates the dissipation produced by the system. Pezzinga and Cannizaro[37] instead used a two-dimensional analysis in cylindrical coordinates, to describe the radial and the axial velocity in the transient flow. The result showed that a multi-dimensional description is more accurate than a one-dimensional description. However, these algorithms require a high computational effort[37], thereby reducing the benefit of using a multi dimensional treatment. Not withstanding the better properties of the finite volume methods, they are not applied widely, and MOC is still the simplest model applied, even in a complex network. For instance, Shin and Chen[5] was found to be the only application the MOC methodology in a complex multi junction network system. The research proved the feasibility of a one-dimensional description of a complex system although with accuracy limitations. Therefore, new algorithms with higher accuracy must be introduced.

3.2.2 Solution method

MOC is a mathematical approach to solve partial differential equations translating to a simpler ordinary differential equations taking advantages from their characteristic propagation information[27]. The fixed grid MOC scheme is the most accessible algorithm to be solved and requires a spatial grid that fits equally in all of the system pipe lengths. In the majority of cases, this is not possible due to the speed of sound and the stability condition required. For this reason, interpolation schemes and speed of sound adjustment have been used to improve computational efficiency at the cost of a more dissipative scheme[28]. Different researchers have dealt with the problem. Trikha[29] used a specific time step for each section of duct, interpolating only at the boundary. This approach can produce more inaccuracy when rapid changes occur in the system. Wylie et al.[27] considered a flexible method with a self-path adjustment calculated with the characteristic information (characteristic grid) and interpolating when the domain was no longer inside the physical boundary. This method is a combination of spatial, boundary and time interpolation. The major drawback of this algorithm, as with the Trikha methodology, is the simulation of dynamic components that require information from the system, for instance, a safety valve. Different methodologies were developed to improve water hammer simulation, and implicit schemes were also studied. Wylie et al.[27] considered an implicit finite difference method, solved with a sparse matrix and Newton-Raphson procedure. Although the algorithm allows the use of larger time steps than an explicit scheme, the computational effort to invert the matrix can reduce the numerical advantages. In addition, mathematically it seems that wave propagation problems are better described by explicit schemes due to the nature of updating the information[28]. The accuracy of the results improved with the introduction of the finite volume (FV) method. Pioneering research was made by Guinot[30] who applied the finite volume scheme solving the Riemann problem with the first order accuracy method. The first approach was similar to the linear interpolation scheme developed for the MOC. Further steps were made only in recent years where more accurate and computationally efficient methods were introduced with a higher order scheme. The introduction of the latest numerical algorithms, such as the Total Variation Diminishing (TVD) scheme, have also permitted an improvements in the numerical stability. An example of second-order accuracy in time and in space can be found in Zhou et al.[31, 32] where a Monotonic Upwind scheme for Conservative Law (MUSCL) was used. The MUSCL scheme has shown a significant benefit in terms of computational efficiency and simplicity. This algorithm performes as well as the first-order MOC when the Courant number (C_R) is equal to one[31]. Meanwhile, for $C_R < 1$, MUSCL performed better than the MOC producing less numerical dissipation phenomena. In addition, for frictionless simulation, the MUSCL scheme converges to the analytical solution. Moreover, it has been shown that for finer grids, the finite volume method avoids unrealistic pressure spikes commonly present in water hammer simulations performed with the MOC. A further advantage of the second order method is less storage memory required. Compared with the first order scheme, the accuracy is higher and requires a less refined grid to converge[33]. Water hammer is a pressure variation phenomenon that depends of the nature of the fluid, and in many cases it is required a multi-phase treatment for the presence of different component (particles, other liquid or gasses). Even in pure liquid phase, a the second phase can be present in form non-dissolved gas (for instance, air) or vapour or both and, depending on the algorithm, the simulated fluid response can be different. The presence of the other phases affects the speed of sound, making even more complicated the use of the interpolation scheme or the MOC with constant time intervals. Wylie et al.[27] dealt with this issue by considering the second-phase to be lumped at the computational section, considering the liquid within these grid points free of gas. With this assumption the speed of sound can be considered constant in the system, and the numerical stability of this scheme was demonstrated by Liou[34]. Although the algorithm performs fast and is easy to implement, it can only consider a small amount of vapour and/or gas to be produced. Even more researchers are moving in the direction of using finite volume solvers for multiphase flow. Guinot[35] compared the first-order and second-order approximation of the Riemann invariants with the exact solution. Results showed that the approximation of the Riemann problem solution was a reasonable simplification for wave propagation in multiphase flow. Starting from that point, further investigations were conducted[30], introducing the free gas phase into the system. This result has allowed water hammer phenomenon with cavitating conditions, to be solved with a finite volume method. The application of a peer reviewed discrete vapour model was carried out by Zhou et al.[31]. The researchers applied the discrete vapour cavity model developed for MOC[27] in combination with a finite volume treatment. The solution algorithm used a second order in time and space MUSCL scheme with the TVD method. The same idea was applied to the water hammer phenomenon with a Discrete Gas Cavity Model[31]. Further discussion of second phase is given in the section 3.3.

3.2.3 Dissipation term

So far this section has only described the flow model and the dissipation term was not discussed, although it plays an important role. The importance of a correct dissipation model formulation was highlighted at the beginning of the nineteen seventies. Since then, several types of research involving fluid transient flow in ducts have been developed with different strategies. The review of Ghidaoui et al.[28] evaluated the importance of the wall friction term considering the system response. The research gave the equation 3.6, further details in the appendix C, where:

- L is the length,
- M the Mach number,
- T_D the time scale of the radial diffusion of the vorticity,
- c the speed of sound,
- ϕ the diameter and
- ζ a real positive parameter used to adjust the formulation.

$$\Gamma = \frac{\zeta L M f}{2\phi} + \frac{\zeta T_D}{\frac{L}{c}} \tag{3.6}$$

It has been highlighted that the correct friction term description is important when the non-dimensional value Γ becomes greater than unity. This commonly happens when the simulation time exceeds the first wave reflection, namely for a long simulation. In addition, wall stress is a key factor for a long pipeline or when the cross section area of the pipe is small. All of these factors are usually present in a positive displacement pump or in a pipe network. Therefore, different strategies for correctly interpreting the unsteady dissipation term must be considered. Bergant et al.[38] listed the mathematical formulation in relation to the formulations term. Six different categories were defined:

- mean velocity,
- mean velocity and acceleration,
- mean velocity, instantaneous acceleration and convective acceleration,
- mean velocity and diffusion,
- mean velocity and previous time changing, and

- cross-section distribution of the velocity.

For instance, the Brunone model[39] is a function of mean flow velocity, the advective acceleration and a delay coefficient as shown in equation 3.7.

$$f_B = f_s + \frac{kD}{|\boldsymbol{u}|\boldsymbol{u}}(\frac{\partial \boldsymbol{u}}{\partial t} - c\frac{\partial \boldsymbol{u}}{\partial x}) \tag{3.7}$$

Research[38, 40, 41] dealt with the Brunone formulation showing a better agreement with experiment. Shu et al.[42], to prevent the limitation given by the method of characteristics, used a Galerkin finite element method applied only to the spatial variable. Important was the formulation of the unsteady friction dependency calculated as a variation of the Zielke formulation[43]. This approach differs from Brunone as it is required to store previous numerical data and solve the friction term as a series of frequency factors. Also Urbanowicz et al.[44] used the series expansion of the velocity history. However, these methodologies seem more computationally inefficient and complex to be implemented compared to Brunone's method. Moreover, even from the friction point of view, in the earlier years, research has highlighted the drawbacks of using the method of characteristic rather than the finite volume method[31, 32, 41, 45, 46].

Considering the fluid properties, the compressibility is a dissipation factor that is usually neglected or linearised in most of the cases. This approach can be considered feasible if the pressure variation is of the order of a hundred bar and the amount of any second phase does not affect the density and the speed of sound. Daude et al.[47] introduced a compressibility model in the study of water hammer, solving a set of equations similar to the Euler system. The presence of the energy equation is a complexity that can be neglected since the fluid transient can be considered isothermal in most of the cases[27]. The same point can be seen by the result of Daude et al.[47], where the temperature did not change drastically during a simulation with cavitation. This approach agrees nicely with the experimental results, also allowing the computation of the dynamics of the bubble formation, growth and collapse. On the other hand, the strategy is not easy to implement in the case of a complex system where multiple T junctions are present. As pointed out[48], it is necessary to solve a three-dimensional junction analysis.

Although modelling slurry is beyond the purpose of this research, it seems reasonable to have an idea of the research that dealt with particles suspended in the fluid. Positive displacement pumps work mainly with slurry or fluidised ore, despite the fact that they are tested using water. Suspended particles dissipate energy by friction with the wall and subtracting energy from the fluid to be suspended. Wang et al.[49] studied the water hammer effect in a slurry pipeline created by pump failure. Considering the slurry's formulation, equation 3.8 was added to the momentum equation.

$$F_{slurry} = \frac{sign(\boldsymbol{u})}{\boldsymbol{u}_0}\big(KC_v\frac{\rho_S - \rho_M}{\rho_M}g\mu_s\omega_s\big) \tag{3.8}$$

Where K is a comprehensive coefficient to match the experimental data, C_V is the volume concentration of solids in the slurry, ρ_S is the density of the particles, ρ_m is the density of the slurry, μ_s is the friction coefficient between particles and the pipe, ω_s is the particle settling velocity. Given the simplicity of this formulation it could be introduced in further to improve the performance of the pump model. Considering, the slurry flow condition, an important contribution of the slurry regime in transient flow was given by Bbosa et al.[50]. The research described three main regimes and their implication: stable turbulent regime, the stable laminar regime and the unstable regime. In the first regime, the suspension of the particles is guaranteed by the high flow velocity and the eddy forces. This regime is the most appropriate to be treated as a homogeneous fluid. The second condition consists of the suspension of the particle in the core of the fluid giving a heterogeneous flow (the most common regime). Whilst, the unstable regime occurs when there are no forces capable of suspending the particle and results in the sinking of the particles. This regime is not constant in terms of particle concentration, neither in time nor in space. The transition velocity was also studied by Bbosa et al.[50] and gave critical values for particle deposition and non-Newtonian fluid flow.

3.2.4 Summary of fluid transient model

Summarizing, this section dealt with the formulation and the fluid transient description principally in the one-dimension formulation and the methodology to reach a solution. Although different approaches were analysed, the finite volume strategy seems to be the most efficient in terms of accuracy and computational results. In addition, an improvement can be introduced with the use of a TVD method avoiding numerical issues given by higher order methods. The correct formulation of the friction term is a crucial point to describe the dissipation phenomenon in the wave travel problem. Compressibility research was also considered, and current studies on slurry were identified.

3.3 Cavitation

Cavitation is a phenomenon of nucleation, growth and collapse of a bubble. American Petroleum Institute (API) regulations define cavitation limits for centrifugal pumps. The maximum flow rate in the pump performance must be three per cent of the steady condition, the so-called $NPSH_3$. However, the same regulation defines that cavitation in the PD pump must be avoided at any time. Unfortunately, this operating condition is almost impossible to achieve. Therefore the real effect of the cavitation formation must be known and predicted. The change of the liquid to the gas phase can be caused by a variation in temperature (boiling) or pressure (cavitating) or

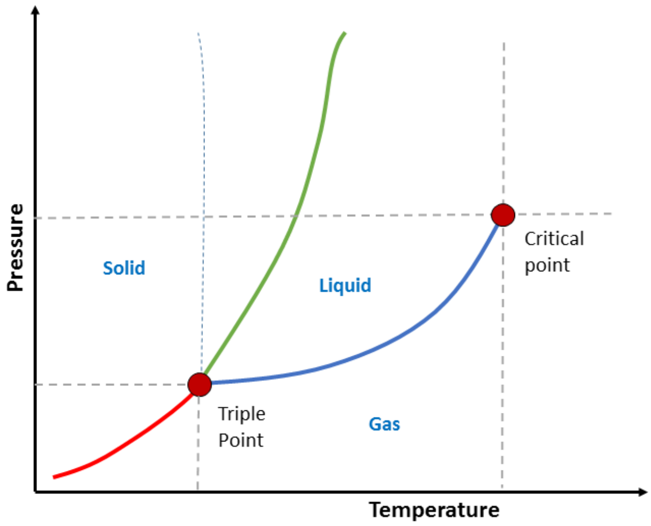


Figure 3.5: Phase change diagram[51].

their combination as shown in figure 3.5. During the pump's cycle, the temperature of the fluid is almost constant while the pressure can change drastically. Although, during the collapse of the cavity, the temperature can locally reach thousands of degrees instantaneously[51], the time scale of the phenomenon allows the cavity to be considered as isothermal.

3.3.1 Cavitation model

The description of single bubble dynamics is given by Rayleigh-Plesset. The complete formulation can be seen in equation 3.9.

$$\frac{p_B(t) - p_\infty(t)}{\rho_L} = R\frac{d^2R}{dt^2} + \frac{3}{2}\left(\frac{dR}{dt}\right)^2 + \frac{4\nu_L}{R}\frac{dR}{dt} + \frac{2S}{\rho_L R} \tag{3.9}$$

Each term plays an essential role in the dynamics and the time scale of the bubble. The left-hand side of the equation is the driving term or far field value, while the right-hand side is related to the bubble and fluid characteristics. The formulation considers the viscosity effect ν_L, the bubble surface tension S, and the variation of the size $\frac{dR}{dt}$. In this form, the Rayleigh-

Plesset equation does not have an analytic solution[51] therefore a simplified version is commonly used to evaluate the cavity formation. A useful review of the theory and the description of the bubble dynamics can be found in Brennen[51], who studied cavitation in depth.

Starting from equation 3.9 and making the assumption of constant temperature and steady state conditions, it is possible to calculate the bubble pressure value, $p_v(T)$. In that condition, for the fluid system to be in balance requires that the external liquid pressure has a higher pressure than the bubble. The reason for this difference is given by the surface tension as seen in equation 3.10.

$$p_\infty(t) = p_V(T) + \frac{2S}{R} \tag{3.10}$$

The fluid is generally in contact with an atmosphere where gas can be trapped in solution in the liquid phase. This can happen even in a controlled environment. The presence of the second component further increases the far-field pressure of cavity formation, as highlighted in equation 3.11.

$$p_\infty(t) = p_V(T) + p_g + \frac{2S}{R} \tag{3.11}$$

Therefore, in the presence of a non-condensable gas, cavitation is produced at a higher pressure compared to the usual cavitation pressure. Thus, to ensure a correct formulation of the cavity formation, the correct amount of non-condensable gas must be included in the model. Here, the considerations for the cavity nucleation are made considering homogeneous formation, and all the bubbles are assumed to be spherical. However, more complexities are introduced in the case of heterogeneous nucleation. For instance on a flat hydrophobic surface. The presence of a cloud of bubbles makes the analysis even harder, where there is also the interaction between the bubbles, which is not included in the Rayleigh-Plesset equation.

Different algorithms were developed for modelling cavitation formation. However, all approaches can be categorized as either homogeneous treatment or a two phase model[52]. The second method consists of describing N phases (liquid and gas) with N Navier-Stokes equations and N-1 interaction functions. This approach is computationally expensive and requires complex mathematical formulation. Two examples are given by Guillard and Murrone[53], and Liuzzi[54]. On the other hand, the homogeneous description considers only one Navier-stokes system of equations. It averages all phases as one overall fluid mixture and only one function is needed to describe the fluid properties.

3.3.2 Solution method

The common description of the fluid dynamics is either described with a discrete model or a continuum formulation. Discrete gas and vapour cavity models are part of the discrete treatment. The continuum can be modelled by the transport equation model (TEM) and the equation of state (EoS). Dwelling on the transport equation, a further equation for the transport of vapour is integrated into the Navier-Stokes system. The transportation of the volume or the mass fraction is given by an inhomogeneous transport equation where the source term performs the variation of the liquid phase. Zwart et al.[55] described the transition of the vapour to liquid and vice versa starting from a simplification of the Rayleigh-Plesset equation, neglecting the second order and the surface tension terms. In addition, the assumption of no bubble interaction was made, which is plausible only at the beginning of the cavity phenomenon. With the same logic, Singhal et al.[56] developed strategies to also include the non-condensable gas, keeping constant the phase amount but evaluating its volume variation. Sumam et al.[57] applied a variation of the source term in the transport equation formulation to the water hammer phenomenon. The research has highlighted instability issues and the time step required to capture the cavity dynamics is much smaller than the Courant stability value. These three algorithms required two empirical coefficients (collapse and evaporation parameters) to tune the liquid-vapour transformation phenomena. A comparison of the transport equation models was made by Zhao et al.[52], where an overview of the overall formulation was given. The majority of the formulations were derived from thermodynamic behaviour or from a simplified description of the Rayleigh-Plesset formulation. The first method has proportionality with the pressure variation, while the second with the square root of the pressure. Due to this difference, all the methodologies gave different results and required carefully study for each application. The equation of state model, consisted of evaluating the variation of the fluid properties with the fluid pressure value. Goncalves et al.[58, 59] used an arcsine barotropic formulation to model the fluid characteristics in the mixture regime, while Zheng et al.[60] used an isentropic formulation.

3.3.3 Cavitation in positive displacement pumps

Cavitation in a positive displacement pump can occur during the valve opening phase, where a small gap is formed making the fluid accelerate and at the same time reducing the static pressure. This is not the only mechanism for vapour cavity formation. Schlücker and Opitz[11] categorized the cavitation formation in relation to the pump cycle phase. In that research, a PIV analysis of the cavitation was related to pressure measurement, recording the bubble growth and collapses history. A classification related to the pump condition and the water hammer effect was produced. Considering the amount of vapour produced, cavitation could be categorize as:

- inception,
- partial,

- full cavitation.

When the cavitation pressure is reached only for a small portion of time, and at the beginning of the suction phase, the cavitation is referred to as inception cavitation. The cause of this cavitation could be related to the volume expansion in the chamber. When a constant pressure is established for a more extended period, the cavity is defined as partial cavitation. If the cavity occurs within the first half of the phase of the cycle, the cavity is called incipient partial cavitation or advance partial cavitation. When the cavitation reaches the second half of the suction phase, the cavitation takes the name of distinctive partial cavitation. Full cavitation formation, occurs when the cavitation occurs after the suction phase and extending into the discharge cycle. The inception and partial cavitation seem to be harmless in most cases[61]. Meanwhile, full cavitation significantly affects the volumetric efficiency. An application of a cavitation algorithm in a positive displacement pump was undertaken by Iannetti[3] who applied the Singhal et al.[56] formulation in a three dimensional analysis to evaluate valve cavitation with good results. A different approach for a hydraulic system was performed by Maiga et al.[62], where a multi-bubble model formulation was used. The importance of an accurate bubble interaction model was pointed out especially during the incipient cavitation. A cavitation model should also consider the nature of the fluid. In PD pumps slurry application the fluid can barely be considered Newtonian due to the suspended particles. In the case of solid-liquid fluid, the presence of particles changes the fluid response. In that field, Brujan[63] gave an example of the phenomenon for the case of polymeric fluid. In conclusion, cavitation can be modelled with several different approaches that have implications on complexity and computational effort. For that reason, simulation of the cavity formation is still an active field of research, although more and more accurate models have been developed. The discrete cavity formulation is still the most suitable algorithm available which does not overload the numerical computational capacity. This method produces a good balance of the simulation effort required and the accuracy of the model.

3.4 Summary

As pointed out from the literature review, modelling a positive displacement pump is still a challenge. Although there is considerable research involving cavitation and positive displacement pumps, they are restricted due to the computational time required. On the other hand, simulation of a complete pump are difficult, and research to simplify the problem in order to produce meaningful results is required. The research in this thesis tries to fill this void, considering at the same time compressibility, transient fluid description and cavitation. The path will include the MUSCL scheme solver with TVD algorithm to produce accurate results without significant computational effort thanks to its straightforward implementation. For the cavitation phenomenon a base line of the Discrete Gas Cavity Method is considered due to its simple solution even for complex network systems. The scope of this thesis is therefore, to create an algorithm capable of predicting the behaviour of a pump within a reasonable computational time. In addition, the algorithm should overcome the limitations in the commercial software by including complex network simulation.

Chapter 4

Theory of pressure waves

The complete description of fluid transient behaviour is given by the Navier-Stokes system of equations. The system consists of a continuity equation, the vector momentum equation and the energy equation. Solving the complete Navier Stokes equations is exceptionally complex, and only numerical techniques can be used. An example is the Direct Numerical Simulation approach (DNS), where no turbulence model is applied, and the fluid dynamics at all time and space scales is solved. To use DNS, the grid mesh should be small enough to capture all the dissipation phenomenon that takes place in the Kolmogorov scale[64]. For that condition, the computation required means an extremely long time for solution, making the algorithms inapplicable for industrial applications[1]. For that reason, different approaches were developed to reduce the amount of computational effort required[65]. Even if these approaches are getting more and more accurate, the schemes still require significant time to solve and convergence is not always guaranteed. The Reynolds Averaged Navier Stokes (RANS), simplify the equations and time-average the turbulence phenomenon. In other words, the turbulence term is decoupled into steady and fluctuating quantities. On the other hand, Large Eddy Simulation (LES) applies a spatial filter to solve for the turbulence. However, all of these methodologies need substantial time to reach the solution. In the case of multiphase flow, the complexity is increased and requires further computational resourses. However, for fluid flowing in a duct, a one-dimensional formulation can be used with reasonable accuracy. This approach simplifies the system without compromising the global description of the fluid dynamics. In addition, with this strategy, a more extensive analysis of different phenomena may be undertaken thanks to its versatility. Therefore, in this section, the system of equations to evaluate the wave formulation for transient fluid flow in one-dimension is given.

4.1 One dimensional model

In a hydraulic system, the changing of the fluid properties due to variations of speed, pressure or density, creates a wave which propagates through the domain. Mathematically, problems that are transient in nature are classified as hyperbolic. The most common transient phenomenon described in one dimension is the so-called water hammer, which is caused by a sudden stop of the fluid. Under this condition, the perturbated fluid creates a high-pressure wave that can potentially damage the entire system. For this reason, water hammer has been studied extensively since the beginning of the twentieth century. Joukowsky[28] gives the most straightforward formulation to evaluate the peak pressure from the fluid velocity variation. The equation is a direct consequence of the continuity equation, and it relates the pressure increase to the speed of sound and the density, as shown in equation 4.1.

$$\Delta p = \rho c \Delta \boldsymbol{u} \tag{4.1}$$

From the Joukowsky formulation, it is possible to evaluate two essential characteristics. The pressure response depends on the property of the fluid in terms of the density and the speed of sound. For instance, comparing the peak pressure variation with a fluid speed equal to one meter per second for water and for mineral oil, the pressure rises ≈ 15 bar and ≈ 11 bar respectively[18]. However to describe the full physics of the phenomenon, a more accurate model is required. Referring to the control volume in the figure 4.1, the continuity equation and the momentum equation are obtained.

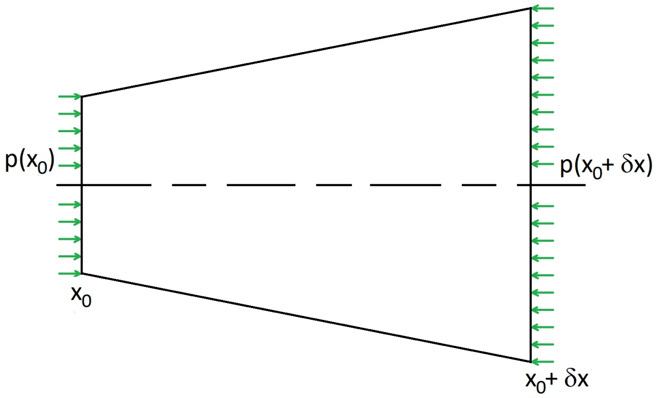


Figure 4.1: Contiuity equation for Water Hammer.

The continuity equation is expressed as equation 4.2. The parameter $\delta U_C(t)$ is the rate of increase of mass in time and equal to the flux mass $\delta F_C(t)$. These parameters are defined by equations 4.3 and 4.4 respectively.

$$\delta U_C(t_0 + \delta t) - \delta U_C(t_0) = \delta F_C(x_0 + \delta x) - \delta F_C(x_0) \tag{4.2}$$

$$\delta U_C(t) = \int_{x_0}^{x_0+\delta x} \rho(x,t)A(x,t)dx \tag{4.3}$$

$$\delta F_C(t) = \int_{t_0}^{t_0+\delta t} \rho(x,t)\boldsymbol{u}(x,t)A(x,t)dt \tag{4.4}$$

The velocity of the fluid is $\boldsymbol{u}$, whereas ρ is the fluid density and A the cross-sectional area. By differentiating equation 4.2 in time and space, the final continuity equation in one dimension is obtained (equation 4.5).

$$\frac{\partial(\rho A)}{\partial t} + \frac{\partial(\rho \boldsymbol{u} A)}{\partial x} = 0 \tag{4.5}$$

Regarding the momentum equation, Newton's second law is applied to the system, given by equation 4.6.

$$\delta U_M(t_0+\delta t)-\delta U_M(t_0) = \delta F_M(x_0+\delta x)-\delta F_M(x_0)+\delta P(x_0)-\delta P(x_0+\delta x)+\delta F_E+\delta F_P \tag{4.6}$$

Where the terms are:

- $\delta U_M(t)$ the overall fluid momentum given by the equation 4.7.
- $\delta F_M(t)$ the momentum of the fluid that passes position x over the time interval calculated with equation 4.8.
- δF_P the pressure force given by 4.9.
- δF_E the external forces acting on the fluid control volume.

$$\delta U_M(t) = \int_{x_0}^{x_0+\delta x} \rho(x,t)\boldsymbol{u}(x,t)A(x,t)dx \tag{4.7}$$

$$\delta F_M(t) = \int_{x_0}^{x_0+\delta x} \rho(x,t)\boldsymbol{u}^2(x,t)A(x,t)dx \tag{4.8}$$

$$\delta F_P(t) = \int_{t_0}^{t_0+\delta t} p(x,t)A(x,t)dt \tag{4.9}$$

The complete momentum equation is given by combining all the integral formulations and differentiating with respect to space and time, yielding equation 4.10.

$$\frac{\partial(\rho\boldsymbol{u}A)}{\partial t} + \frac{\partial(\rho\boldsymbol{u}^2A)}{\partial x} + \frac{\partial(pA)}{\partial x} = F_{ext} \tag{4.10}$$

Summarizing, the complete fluid transient system in one dimension is given by the system of equations 4.11.

$$\begin{cases} \frac{\partial(\rho A)}{\partial t} + \frac{\partial(\rho \boldsymbol{u} A)}{\partial x} = 0 \\ \frac{\partial(\rho \boldsymbol{u} A)}{\partial t} + \frac{\partial(\rho \boldsymbol{u}^2 A)}{\partial x} + \frac{\partial(pA)}{\partial x} = F_{ext} \end{cases} \tag{4.11}$$

In this form, it is required to solve three different variables: pressure, density and velocity. Normally this can be done by the introduction of the energy equation or by iterative methods. However, it is easier to use the mechanical description of the speed of sound to couple the pressure and density variation. Using the form $c^2 = \frac{\partial p}{\partial \rho}$, a further simplification can be obtained. In this condition, the system of equations becomes a function of only the pressure and the mass flow rate $[p, \rho u A]$, as shown in the system of equations 4.12.

$$\begin{cases} \frac{\partial(pA)}{\partial t} + c^2 \frac{\partial(\rho \boldsymbol{u} A)}{\partial x} = 0 \\ \frac{\partial(\rho \boldsymbol{u} A)}{\partial t} + \frac{\partial(\rho \boldsymbol{u}^2 A)}{\partial x} + \frac{\partial(pA)}{\partial x} = F_{ext} \end{cases} \tag{4.12}$$

4.1.1 External force

The system of equations 4.12 does not include the dissipation term directly in the formulation. The viscous effect is neglected, and hence dissipation is introduced only from the source term. Referring to Guinot[66] the external force can be considered as the sum of two phenomena. The first is given by the variation in the cross sectional area, namely the $\frac{\partial A}{\partial x}$ term, while the second term is the dissipation that takes place between the fluid and the wall in the form of friction. Therefore, the overall formulation is given by equation 4.13.

$$F_{ext} = \int_{t_0}^{t_0+\delta t} \Big(\int_{x_0}^{x_0+\delta x} \big(F_P(x,t) + F_w(x,t) \big) dx \tag{4.13}$$

The wall friction force $F_w(x,t)$ is opposite to the flow, and it can be considered as proportional to the velocity by the relation 4.14[27, 67].

$$F_w = \tilde{f} = -\frac{1}{2}\phi\pi\rho|\boldsymbol{u}|\boldsymbol{u} \tag{4.14}$$

The overall expression of the source term is then rewritten as a function of the wall stress and the variation of the area. In addition, for simplicity, the pipe angle is neglected considering only horizontal ducts and no gravity forces. The complete description of the transient flow is hence given in the system of equations 4.15.

$$\begin{cases} \frac{\partial(pA)}{\partial t} + c^2\frac{\partial(\rho\boldsymbol{u}A)}{\partial x} = 0 \\ \frac{\partial(\rho\boldsymbol{u}A)}{\partial t} + \frac{\partial(\rho\boldsymbol{u}^2A)}{\partial x} + \frac{\partial(pA)}{\partial x} = p\frac{\partial A}{\partial x} - \tilde{f} \end{cases} \tag{4.15}$$

4.2 Linearised and characteristic form

The formulation 4.15 is the complete formulation that takes into account:

- the advective term,
- the variation of the area, and
- the wall friction dissipation.

In a pipeline network and in pump systems, the cross sectional area is usually constant. Therefore the system can be further simplified in the system of equations 4.16.

$$\begin{cases} \frac{\partial p}{\partial t} + \frac{c^2}{A}\frac{\partial(\rho\boldsymbol{Q})}{\partial x} = 0 \\ \frac{\partial(\rho\boldsymbol{Q})}{\partial t} + \frac{\partial(\boldsymbol{u}\rho\boldsymbol{Q})}{\partial x} + A\frac{\partial p}{\partial x} = -\tilde{f} \end{cases} \tag{4.16}$$

To obtain a linearized formulation, the Riemann formulation is used. The system is written in the form of variable U and its flux, as shown in equation 4.17.

$$\frac{\partial U}{\partial t} + \frac{\partial f(U)}{\partial x} = \boldsymbol{s}(U) \tag{4.17}$$

Introducing the Jacobian matrix $[J] = \frac{\partial f(U)}{\partial U}$, given as a partial derivative of the flux, $\frac{\partial f(U)}{\partial x} = \frac{\partial f(U)}{\partial U}\frac{\partial U}{\partial x}$, the system becomes a function only of the primitive variable U. In this specific case, the Jacobian formulation of the system 4.15 is equal to the matrix 4.18.

$$[J] = \begin{bmatrix} 0 & \frac{c^2}{A} \\ A & u \end{bmatrix} \tag{4.18}$$

Therefore, using the variable column vector $[p, \rho Q]^T$ the system can be rewritten in the extended version 4.19.

$$\begin{bmatrix} p \\ \rho Q \end{bmatrix}_t + \begin{bmatrix} 0 & \frac{c^2}{A} \\ A & u \end{bmatrix} \begin{bmatrix} p \\ \rho Q \end{bmatrix}_x = \begin{bmatrix} 0 \\ \tilde{f} \end{bmatrix} \tag{4.19}$$

This system is non-homogeneous due to the source term. However, it is possible to use the Godunov splitting algorithms, described in section 4.4, to decouple the system into homogeneous and ordinary differential equations. Hyperbolic homogeneous systems are easier to solve thanks to their properties. However, to be classified as hyperbolic the Jacobian matrix must be diagonalizable. This means that the eigenvalues $\Lambda_{1,\dots,n}$ of the system 4.19 must be real. With this condition, the system can be decoupled into n independent advection equations[25] which are resolvable independently. The eigenvalues of the system 4.16 are given in the system of equations 4.20. At the same time, the corresponding eigenvectors are shown in equations 4.21.

$$\begin{cases} \Lambda_1 = \frac{\boldsymbol{u} - \sqrt{\boldsymbol{u}^2 + 4c^2}}{2} \\ \Lambda_2 = \frac{\boldsymbol{u} + \sqrt{\boldsymbol{u}^2 + 4c^2}}{2} \end{cases} \tag{4.20}$$

$$\begin{cases} \left[1; \frac{A}{c^2} \frac{\boldsymbol{u} - \sqrt{\boldsymbol{u}^2 + 4c^2}}{2}\right] \\ \left[1; \frac{A}{c^2} \frac{\boldsymbol{u} + \sqrt{\boldsymbol{u}^2 + 4c^2}}{2}\right] \end{cases} \tag{4.21}$$

The eigenvalues are always real $\forall c$, therefore each advective function can be described in characteristic form equal to the function 4.22. Further description of the algorithm may be found in[25] and[26].

$$\frac{\partial \Omega}{\partial t} + \mathbf{\Lambda} \frac{\partial \Omega}{\partial t} = K^{-1} S \tag{4.22}$$

In equation 4.22, $\mathbf{\Lambda}$ is the transformation matrix given by $\mathbf{\Lambda} = K^{-1} J K$. $\mathbf{\Lambda}$ is a matrix with the eigenvalues on its diagonal, while K^{-1} is the inverse matrix with the eigenvector as column, given by the equations 4.23 and 4.24 respectively.

$$\mathbf{\Lambda} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \tag{4.23}$$

$$[K]^{-1} = \begin{bmatrix} -\frac{\Lambda_2}{\Lambda_1 - \Lambda_2} & -\frac{c^2}{A(\Lambda_2 - \Lambda_1)} \\ \frac{\lambda_1}{\lambda_1 - \lambda_2} & \frac{c^2}{A(\lambda_2 - \lambda_1)} \end{bmatrix} \tag{4.24}$$

Introducing a new set of variables defined as $\Omega = K^{-1} U$, it easier to handle the transformation process. Moreover, the function can be expressed in terms of eigenvalues by equation 4.25.

$$[\Omega] = \begin{bmatrix} -\frac{\Lambda_2}{\Lambda_1 - \Lambda_2} p - \frac{c^2}{A(\Lambda_2 - \Lambda_1)} \rho Q \\ \frac{\Lambda_1}{\Lambda_1 - \Lambda_2} p + \frac{c^2}{A(\Lambda_2 - \Lambda_1)} \rho Q \end{bmatrix} \tag{4.25}$$

The primitive variables can be obtained by inverting the characteristic formulation as shown in the system of equations 4.26.

$$\begin{cases} p = \Omega_1 + \Omega_2 \\ \rho Q = \frac{A \Lambda_2}{c^2} \left(\Omega_2 + \frac{\Lambda_1}{\Lambda_2} \Omega_1 \right) \end{cases} \tag{4.26}$$

The benefit of this approach is given in the numerical solution for the cell interface calculations. In detail, thanks to the equations 4.26 the interface Riemann problem can be solved analytically as explained in chapter 5.

4.3 Water Hammer

The system of equations 4.12 developed for the transient fluid motion description, reduces to the common water hammer formulation with the following assumptions:

- small variation of the fluid speed compared with that of the pressure;
- the compressibility of the fluid can be neglected, introducing a linearised equation form.

With these conditions and using the piezometric head pressure, $H = \frac{p}{\rho g}$ the system of equations becomes equal to that shown in 4.27 given in [27].

$$\begin{cases} \frac{\partial H}{\partial t} + \frac{c^2}{g}\frac{\partial \boldsymbol{u}}{\partial x} = 0 \\ \frac{\partial \boldsymbol{u}}{\partial t} + g\frac{\partial H}{\partial x} = -\frac{f\boldsymbol{u}|\boldsymbol{u}|}{2D} \end{cases} \tag{4.27}$$

This system can be transformed, with the same linearisation procedure described in section 4.2, into the the characteristic form. Two distinct characteristic functions that propagate with a velocity of ±c, are generated. At this stage, the system of equations can be solved in terms of pressure head and flow rate by equations 4.28 extrapolated by Wylie et al.[27].

$$\begin{cases} H = C_P – B_P Q \\ H = C_M + B_M Q \end{cases} \tag{4.28}$$

The parameters $C_{M/P}$ and $B_{M/P}$ in the system of equation 4.28 depend on the numerical algorithm used and are described in section 5.5.

4.4 Source Term

To reach the solution in the linearised form of the equation, the source term was neglected. However, in order to reintroduce its effect, the splitting method or fractional-step method is performed[25, 26]. The methodology consists of dividing the problem into two subroutines that are solved consecutively. The system 4.17 is decoupled into a homogeneous partial differential equation and into an ordinary differential equation, as shown in equations 4.29.

$$\begin{cases} \frac{\partial U}{\partial t} + \frac{\partial f(U)}{\partial x} = 0 \\ \frac{\partial \tilde{U}}{\partial t} = \boldsymbol{s}(\tilde{U}) \end{cases} \tag{4.29}$$

The first equation follows the procedure described in section 4.2. The ODE, requires the results from the PDE to be solved. In terms of accuracy, the fractional splitting method is always first-order, no matter how well the subsystems are discretized and solved[25]. However, it is possible to improve the accuracy of the model with a Strang[25] splitting algorithm. The methodology consists of updating the variables two times in the same iteration. Therefore, the calculation of the variable is performed every half time step. Nevertheless, Leveque[25] proved that the accuracy of the first order method is comparable with the second order method. In addition, the boundary conditions are more easily computed with the first-order method than with a Strang splitting algorithm. Therefore the solving method used in this research is focused on first order theory. The solving technique could use either standard explicit or implicit methods for the ODE. However, in this context, the explicit Runge-Kutta method was used.

4.4.1 Frictional Term

As already highlighted, the source term can be defined as a function of the velocity and the friction factor. The latter parameter is a non-dimensional parameter used to describe the losses in a pipe flow. The exact value of this term is difficult to evaluate due to its dependence on the velocity. However three main approaches can be used:

- constant factor
- quasi-steady factor
- unsteady factor

The first method consists of a constant value of the friction factor which oversimplifies the dissipation. In the second method, the friction term varies only in relation to the Reynolds number. The last method is described in depth in[38] and evaluates the friction term with different factors. An example is given by Brunone[39], equation 3.7. Considering the quasi-steady formulation used in this research, the Colebrook function[68] is used. The equation is given in 4.30 where ϕ_h is the hydraulic diameter, ε is the surface roughness and the Re is the Reynolds number.

$$\frac{1}{f} = -2log\Big(\frac{\varepsilon}{3.7\phi_h} + \frac{2.51}{Re\sqrt{f}}\Big) \tag{4.30}$$

The formulation of the friction factor is implicitly defined and hence requires an iterative solution. Figure 4.2 is a representation of the friction factor as a Moody diagram. Two distinct behaviours are defined depending on the Reynolds number. The laminar condition is given by a linear relation for $Re_{Transition} < 2300$. While, for $Re > 4000$ the turbulent condition requires the solution of 4.30. However, this distinction creates a discontinuity that could affect the solution technique. In between the laminar and turbulent regimes an interpolation scheme is performed to obtain the friction value. Nevertheless, the Reynolds number in PD pumps is commonly high, and the transition regime is easily exceeded.

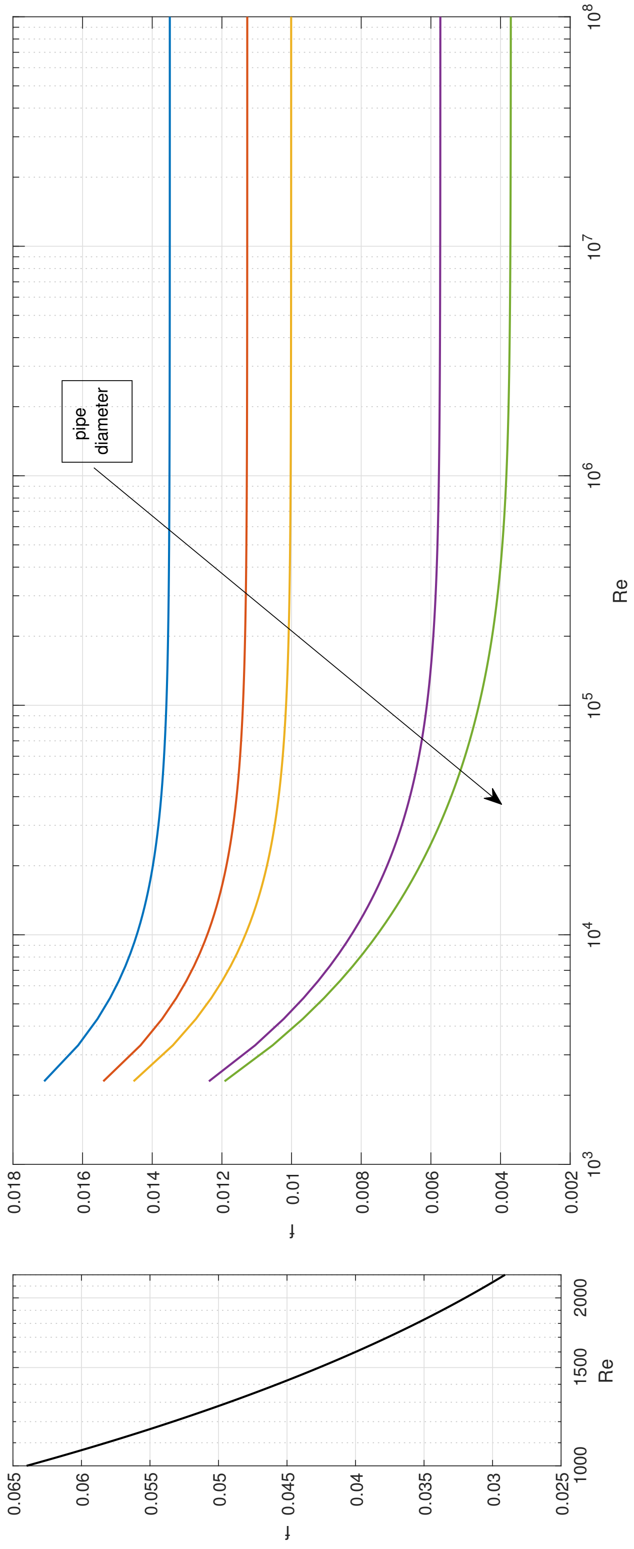


Figure 4.2: Moody's diagram performed with the developed friction algorithm.

4.5 Cavitation

Cavitation is difficult to describe mathematically. The main issues are related to the transformation of the liquid into vapour and vice versa. In addition, the interaction between the bubbles and the liquid is challenging. This issue was briefly introduced in the literature review, and for more details, the research by Brenner is cited[51]. However, in this section only the discrete cavity model is described due to its straightforward implementation. This method allows the vapour to be produced at a specific pipe location. Depending on the solver strategy, these positions can be the grid points for finite difference or the cells boundary interfaces for finite volume. Further explanation is given in section 5 where the combination with the numerical method is performed. Discrete cavity model consists of setting the pressure at the vapour pressure as soon as the calculated pressure is lower or equal to the vapour pressure. At that point, the vapour formation is calculated as a function of the net flow rate. The net flow is the difference across two adjacent cells. At this stage, the cell cannot accept further liquid flow due to the fact that the pressure should be smaller than the cavitation pressure to balance the equation. Therefore the surplus fluid is then formed as vapour, calculated with equation 4.31.

$$\dot{V_{cav}} = Q_{net} = Q_{out} – Q_{in} \tag{4.31}$$

Furthermore, the pressure is maintained to the vapour pressure value as long as the bubbles exists. Only when solution of equation 4.31 is zero is the pressure allowed to increase. In addition, this method permits the calculation of the peak pressure related to the water hammer formation due to bubble collapse. The reason is given by the pressure gradient given by the value that the fluid should have at the instance of bubble collapse and the forced cavitation pressure value. In other words, the system sees a jump in pressure equal to $p_{calculated}$–p_{vapour}.

4.6 Lumped Parameter model

In this section, the theory behind the lumped parameter method (LPM) is given. The LPM condenses all the characteristics of the control volume into one single point. Starting from the system of equations 4.15 and considering constant density and neglecting the advective term the formulation can be written as equation 4.32.

$$\begin{cases} \frac{\partial p}{\partial t} + \frac{\rho c^2}{A}\frac{\partial \boldsymbol{Q}}{\partial x} = 0 \\ \frac{\partial \boldsymbol{Q}}{\partial t} + \frac{A}{\rho}\frac{\partial p}{\partial x} = f(Q) \end{cases} \tag{4.32}$$

Resolving the spatial derivative as an incremental ratio, the formulation drops into the form of equations 4.33.

$$\begin{cases} \frac{\partial p}{\partial t} + \frac{\rho c^2}{A}\frac{\Delta \boldsymbol{Q}}{\Delta x} = 0 \\ \frac{\partial \boldsymbol{Q}}{\partial t} + \frac{A}{\rho}\frac{\Delta p}{\Delta x} = R \end{cases} \tag{4.33}$$

Therefore the lumped parameter model consists of solving two separate ordinary differential equations given in 4.34.

$$\begin{cases} \dot{p} = \frac{\rho c^2}{Al} Q_{Net} \\ \dot{\boldsymbol{Q}} = \frac{A}{\rho l}\Delta p – f(\boldsymbol{Q}) \end{cases} \tag{4.34}$$

The variation of the pressure is then related to the capacitance $C = \frac{Al}{\rho c^2}$. The value ρc^2 is the inverse of the compressibility, namely the bulk modulus E. However, the capacitance is different for liquid and gas. For the first case, the formulation is equal to the ratio between the volume and the bulk modulus, $C_{Liquid} = \frac{V}{E}$. For gases it is $C_{Gas} = \frac{V}{p\gamma}$, where γ is the transformation coefficient. In the case of a mixture flow, the effect is summed with the formulation 4.35.

$$\frac{1}{C_{Tot}} = \sum_{i=1}^{N} \frac{1}{C_N} \tag{4.35}$$

For the variation of the flow rate instead, the ordinary differential equation is a function of the inductance $L = \frac{\rho l}{A}$.

In order to consider the losses given by the friction factor, equation 4.36 relates the pressure and the flow rate.

$$\Delta p = F(\boldsymbol{Q}) \approx f\boldsymbol{Q}^2 \tag{4.36}$$

The lumped method is therefore a further simplification of the one-dimensional system of equations. Also the space discretization is collapsed to a finite difference and therefore, all the information related to the spatial fluctuations are simplified.

4.7 Components

The response of the fluid inside a system is not only related to its fluid properties but also on the interaction that it has with the physical components. In a positive displacement pump, there are different parts that drastically affect the fluid behaviour. Accumulators or hydraulic dampeners, as well as the valves, play a key role and mathematically can be seen as boundary conditions providing information into the domain. The response in terms of pressure and flow rate is then a function of the dynamic model of those parts.

4.7.1 Valve

The valve description is crucial for an accurate prediction of the fluid behaviour, and its description is not straightforward. For instance, during the first instant of the opening phase, when the gap is very small, the viscous effects are significant. This effect is not easy to model and is typically neglected. Therefore, the velocity of the fluid through the gap is only related to the pressure across the valve, the gap area, and the valve dynamics. In order to model the mass flow rate through the valve, the Euler equation, 4.37, for incompressible fluid in one dimension is used[11].

$$\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho\boldsymbol{u}\frac{\partial \boldsymbol{u}}{\partial x} + \frac{\partial p}{\partial x} - \rho g = 0 \tag{4.37}$$

Integrating along the streamline and neglecting the fluid weight, the formulation can be written in the form of equation 4.38.

$$\int_{l_1}^{l_2} \Big(\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho\boldsymbol{u}\frac{\partial \boldsymbol{u}}{\partial x} + \frac{\partial p}{\partial x}\Big) ds = 0 \tag{4.38}$$

The solution shown in equation 4.39 can be seen as an extension of the Bernoulli equation, where the losses of the valve and the gap are included.

$$(p_2 - p_1) + \rho\frac{\boldsymbol{u}^2}{2}\Big(1 + \sum \zeta_i\Big) + \rho\int_{l_1}^{l_2} \frac{\partial \boldsymbol{u}}{\partial t} ds = 0 \tag{4.39}$$

The term ζ_i is the loss value of each of the valve parts, while the friction loss is given as $\frac{\lambda l}{d}$[11]. The incompressible valve gap equation can be derived in the form of equation 4.40.

$$(p_2 - p_1) + \rho\frac{\boldsymbol{u}^2}{2}\Big(1 + \frac{f l_{gap}}{d_h} + \sum \zeta_i\Big) + \rho\frac{\partial \boldsymbol{u}}{\partial t} l_{gap} = 0 \tag{4.40}$$

Although the inertia of the fluid is small, it plays an essential role during the incipient valve lift moment. A limitation of this equation is given by the constant density formulation. In fact, this is a big assumption for the high cavitation regime, where the vapour can be formed in the valve gap. The use of the simple Bernoulli equation could also be used. However, simulations have shown this overestimates the gap velocity and is reflected in a higher pressure pulsation. In order to calculate the valve lift and the related gap area, the valve equation of motion is solved. The second Newtonian law is applied considering the following forces:

- Pressure force
- Spring force
- Preloaded force
- Viscous damping force
- Gravity force

The pressure force is affected not only by the valve area, but also by the valve reaction itself. This behaviour is difficult to model with a one-dimensional algorithm. Therefore a formulation that takes into account detailed valve parameters must be embraced[20]. Two different approaches from the literature were used, the Johnston[20] and Thiel[69] methods. Both methods consider the pressure forces as a function of the pressure difference multiplied by the area of the valve and a pressure coefficient Ψ. Although the methods are similar, the first approach considers a second order formulation given in equation 4.41.

$$\Psi = 1–2C_D\frac{A_{Gap}}{A_V}cos\theta + 2\Big(C_D\frac{A_{Gap}}{A_V}\Big)^2 = 1–2C_D\frac{x}{\phi_m}cos\theta + 2\Big(C_D\frac{x}{\phi_m}\Big)^2 \tag{4.41}$$

Where A_{Gap} is the orifice area, namely the gap area, A_V is the area of the valve face, shown in figure 4.3, C_D is the discharge coefficient, which is experimentally estimated. Differently, the Thiel method considers three different empirical parameters associated with the lift of the valve and the gap Reynolds number under the formulation 4.42.

$$\Psi = K_0 - K_1 \frac{x}{\phi_m} - K_2 log(Re_{gap}) \tag{4.42}$$

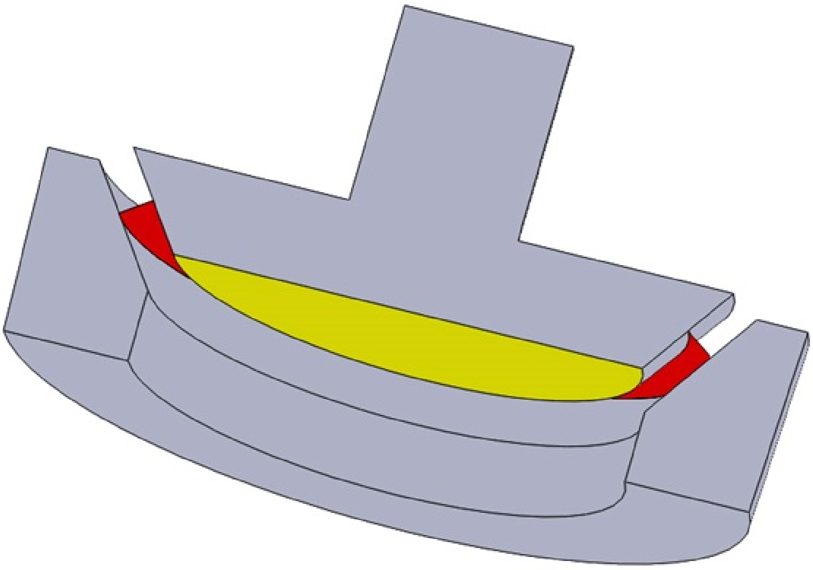


Figure 4.3: Valve and valve seat section, highlighting the valve gap area A_{Gap} in red and the valve area A_V in yellow.

4.7.2 Hydraulic dampener

Hydraulic dampeners are components that aim to reduce pressure pulsation and prevent pressure peaks. They are challenging to model in a one dimensional description. Although their function is simple, the mathematical description is not. Accumulators store the surplus fluid energy in the form of pressure that will be released when a pressure deficit occurs. The physical function requires a two fluid interaction with different compressibility. The compressibility of the gas works as the storage and dissipation, and the modelling of this behaviour can be performed by the perfect gas law. In spite of this description, there are two critical points: the maximum and the minimum gas volume. Considering the theoretical operation, when the gas is compressed to the minimum volume the pressure is at the maximum value. At that point, the gas cannot exchange any information with the fluid, and its behaviour is similar to a boundary wall. The same behaviour is found when the pressure is lowered to the minimum accumulator pressure, and the volume is at the maximum value. In this case, the behaviour of the accumulator can also be associated to a wall boundary. Therefore a sudden stop of the fluid due to the maximum pressure or due to the maximum gas volume being reached could create an non-physical pressure spike similar to water hammer. In reality, this behaviour is not present. The reason is given in the physical concept of gas compression that is not proportional during the process. In addition, the behaviour of the accumulator is affected by thermal hysteresis. In the research of Puddu and Paderi[70], and Pourmovahed and Otis[71] it is shown the differences betweent the experiment result and the ideal gas model. The differences in the results can result in error of 20% in the case of pressure higher than 100 bar. In this research, the issues were overcome by the introduction of an artificial function from a matlab subroutine (detail in the **mathworks website**). This method divides the accumulator into three volume sections, the fluid, the gas and the dead volume. The dead volume is the minimum volume permitted in the accumulator. The gas volume variation is calculated by the gas law equation 4.43, defining the process with the expansion coefficient γ (1 is isothermal process).

$$p_g = \sqrt[\gamma]{\frac{K_{gas}}{V_g}} \qquad (4.43)$$

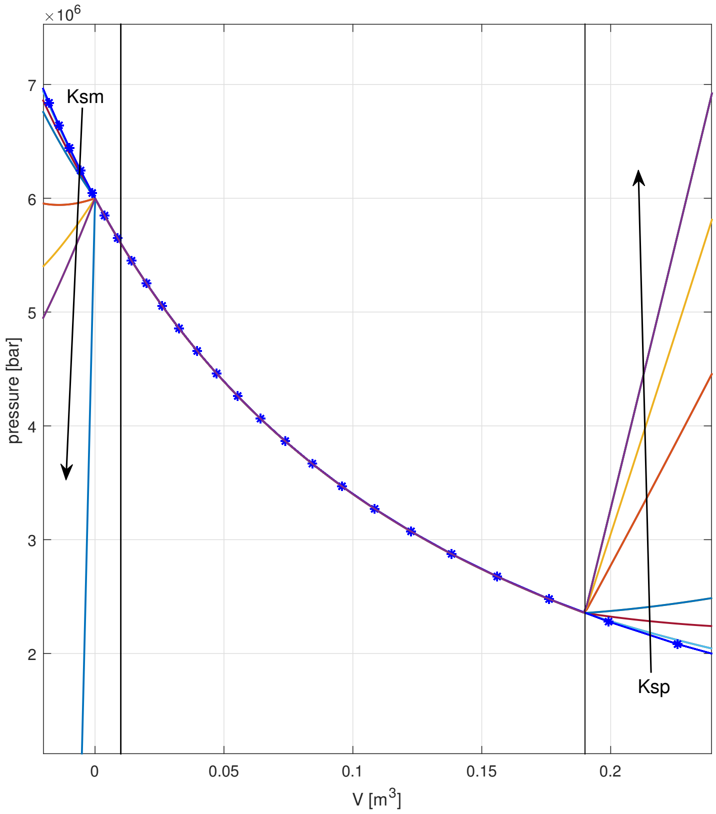


Figure 4.4: Hydraulic pressure response as function of the $K_s^+ = Ksp$ and $K_s^- = Ksm$ values. The circle blue line is the isothermal transformation of the gas, the black lines are the volume limits of the accumulator.

Where the K_{gas} is the constant gas function of the pre-charged pressure and V_G the total volume. To prevent an non-physical phenomenon, an artificial pressure is added. The fluid pressure is then calculated as $p_F = p_g + p_{HS}$, the sum of the gas pressure and the artificial term given in the relation 4.44.

$$P_{HS} = \begin{cases} K_s^+(V_F - V_C) + K_d^+ q_F^+(V_F - V_C), & ifV_F \geq V_C \\ K_s^- V_F + K_d^- q_F^- V_F, & ifV_F \leq V_C \\ 0, & otherwise \end{cases} \tag{4.44}$$

The p_{HS} term works only when the maximum or minimum volume is reached by the fluid. Moreover, it is a function of two parameters that tune the fluid pressure response in relation to the volume K_s and the flow rate K_d. An example of the accumulator response with different parametrization is given figure 4.4.

Chapter 5

Numerical Modelling of pressure waves

The systems of equations and the algorithms described in chapter 4 cannot be solved analytically. Therefore numerical techniques are required to approximate the solution. In addition, it is not guaranteed that the strategies to solve the equations are stable and converge to the correct solution. For these reasons, a careful analysis of the solution algorithms is required to prevent drawbacks and understand their limitations. In this section, the finite volume method is described as well as the ODE solver used to model the PD pump. Furthermore, the solution of the water hammer equation solved by the finite difference method is briefly introduced.

5.1 One dimensional Numerical modelling

The approach considered by the author consists of representing all parts of the pump using a one-dimensional analysis. Therefore, to simulate the pump behaviour using a one-dimensional analysis, it is required to represent the original pump geometries with a series of constant area ducts. Taking as reference the cross-section of a diaphragm pump reported in figure 2.2

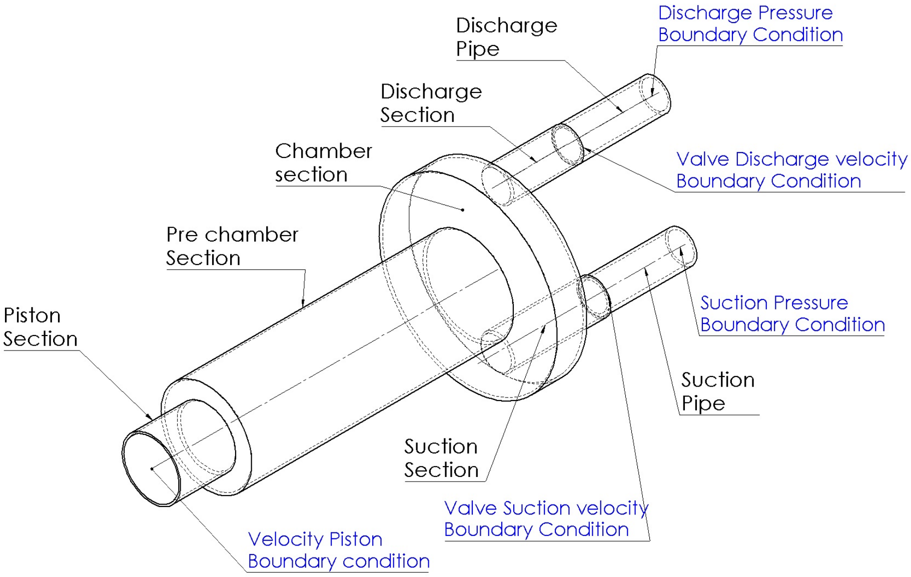

Discharge Pipe
Discharge Pressure Boundary Condition
Discharge Section
Chamber section
Valve Discharge velocity Boundary Condition
Pre chamber Section
Suction Pressure Boundary Condition
Piston Section
Suction Pipe
Suction Section
Valve Suction velocity Boundary Condition
Velocity Piston Boundary condition

Figure 5.1: Representation of the PD pump in one-dimensional model.

the transition into a series of pipes can be seen in figure 5.1.

This approach is possible because the positive displacement pump itself is composed of a series of pipes. Pipes that for the pump can be categorized into:

- piston section,
- pre-chamber section,
- chamber section,
- suction section,
- discharge section.

Therefore the pump is simulated as a hydraulic system where the suction and the discharge valves are the connecting boundaries with the external hydraulic components. In fact, as it will be shown in chapter 7, multi chamber pumps can also be modelled as a series of chambers interconnected

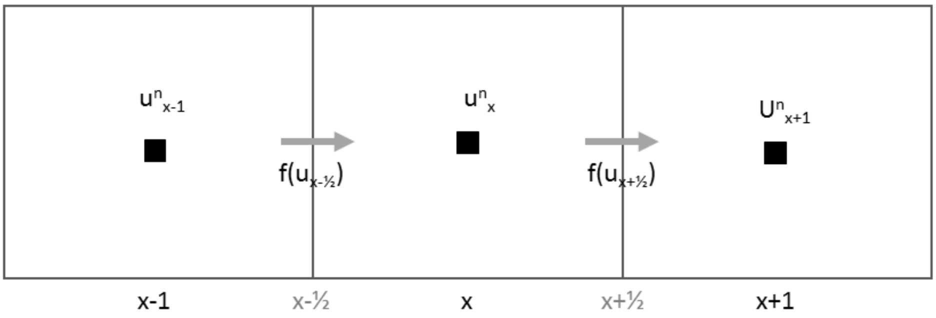


Figure 5.2: Finite volume grid.

with connecting pipes. The fluid transient phenomenon is simulated in each duct section and solved using the finite volume method, while the valve dynamics is solved from the solution of the ordinary differential equation ODE. In the next section, the numerical strategies to simulate the fluid response using a finite volume one-dimensional treatment along with the ODE solution methods.

5.2 Intro to Finite Volume method

The finite volume (FV) method is a numerical scheme used to solve partial differential equations in the form of equation 5.1, where the $s(u)$ is the source term.

$$\frac{\partial U}{\partial t} + \frac{\partial f(U)}{\partial x} = \boldsymbol{s}(U) \tag{5.1}$$

FV is commonly used for hyperbolic problems for its high reliability in discontinuous behaviour. The concept is to divide the entire domain into volumes to form a continuous grid of cells. As shown in figure 5.2, the discretization consists of a centre variable value U_x^n, and two boundary flux, $f(U_{x\pm\frac{1}{2}}^n)$.

Considering U_i^n as a variable spatially defined in the cell domain, the centre value can be considered as the average of U at the time t^n, expressed in the integral form 5.2.

$$U_i^n = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U(t,x)dx \tag{5.2}$$

The flux term is a function of the variable U, and the solution accuracy strongly depends on the method used to calculate the centre value and at the flux interface. The typical formulation for the finite volume strategy is shown in equation 5.3.

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}(f_{i+\frac{1}{2}}^n - f_{i-\frac{1}{2}}^n) + \Delta t\boldsymbol{s}(U_i) \tag{5.3}$$

Different approaches are available in the literature, with both advantages and disadvantages. For instance, the Lax-Wendroff, Lax-Friedrichs and upwind methods are easier to compute but are more dissipative that Godunov's method[25].

Therefore, the algorithms differ in the discretization of the flux function and in the calculation of the previous time step value. In this research, the MUSCL scheme method is used, due to its versatility and less diffusive behaviour when compared with other same order accurate methods[25, 26].

MUSCL Scheme

The Monotonic Upwind Scheme for a Conservative Law is a second order method in time and space. It is an extension of the Godunov first order upwind method where the constant piecewise formulation is replaced with a linear interpolation function. To compute the solution, three steps are required[25, 26]:

1. Data reconstruction

2. Evolution

3. Riemann solution

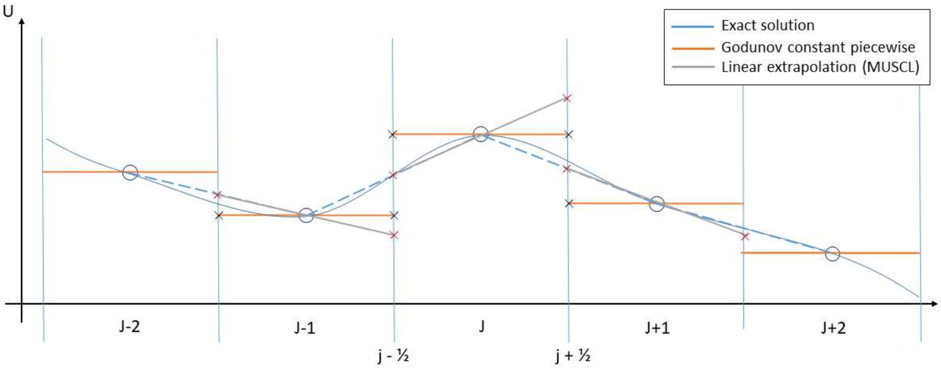


Figure 5.3: MUSCL grid.

Data reconstruction consists of evaluating the boundary values with the slope ratio function given in equation 5.4. A graphic representation of the linear extrapolation and the Godunov constant piecewise function is shown in figure 5.3.

$$U_i(x) = U_i + \frac{\Delta U}{\Delta x}(x - x_i), \epsilon \Big[x - \frac{1}{2}, x + \frac{1}{2}\Big] \tag{5.4}$$

The main difference is given by the variables point values used to compute the function. In the Godunov scheme the boundaries and the center cell have the same values. For the MUSCL scheme instead, two interface boundary values, different from the center cell are calculated as shown in figure 5.3. For each cell there are respectively U_i^L, U_i^R, where L is the left value and R the right.

At this stage, the evolution step takes place for a half time step. Where the evolved boundaries are calculated with equation 5.5.

$$\bar{U}_i^{L,R} = U_i^{L,R} + \frac{\Delta t}{2\Delta x}\Big[f(u_i^L) - f(u_i^R)\Big] \tag{5.5}$$

The presence of two distinct values at each boundary of the cell does not affect the conservative form of the algorithm. Schemes are defined conservative if the flux of the entire domain is equal to the sum of each adjacent control cell[64]. This property is essential for accuracy purposes.

The two calculated values at the interface are the initial condition of the Riemann problem. The Riemann problem, given in equation 5.6, is a spatial initial value problem useful to understand the propagation of a discontinuity. The analytical solution is achieved by computing the eigenvalues and the eigenvector of the Jacobian matrix[25].

$$U = \begin{cases} U_L, if x < x_{i-\frac{1}{2}} \\ U_R, if x > x_{i-\frac{1}{2}} \end{cases} \tag{5.6}$$

In that context, the initial value corresponds to $U_L = \bar{U}_i^R$ and $U_R = \bar{U}_{i+1}^L$. Using the characteristic formulation, the value of the pressure and the mass flow rate at the cell boundary can be calculated analytically by equation 5.7[25, 66].

$$[U]_{i+\frac{1}{2}} = [J]_i[M_L]_i[U]_i^R + [J]_{i+1}[M_R]_{i+1}[U]_{i+1}^L \tag{5.7}$$

In equation 5.7 the Jacobian matrix depends on the i cell, due to the speed of sound variation. Each cell can have different propagation speed due to cavity formation and pressure values. In addition, the value of the interface at the right side of the cell u_i^R is multiplied by the left matrix M_i^L due to the Riemann notation. The value of the characteristic propagation at the interfaces are defined by 5.8 and 5.9.

$$\Big[M_L\Big] = \begin{bmatrix} \frac{\Lambda_1}{\Lambda_1 - \Lambda_2} & \frac{c^2}{A(\Lambda_2 - \Lambda_1)} \\ \frac{A(\Lambda_1 \Lambda_2)}{c^2(\Lambda_1 - \Lambda_2)} & \frac{\lambda_2}{\Lambda_2 - \Lambda_1} \end{bmatrix} \tag{5.8}$$

$$\Big[M_R\Big] = \begin{bmatrix} -\frac{\Lambda_2}{\Lambda_1 - \Lambda_2} & -\frac{c^2}{A(\Lambda_2 - \Lambda_1)} \\ -\frac{A(\Lambda_1 \Lambda_2)}{c^2(\Lambda_1 - \Lambda_2)} & -\frac{\Lambda_2}{\Lambda_2 - \Lambda_1} \end{bmatrix} \tag{5.9}$$

In this condition, the internal boundaries are always explicitly calculated from the two adjacent cells. At the real boundaries, there is just one characteristic function. For that reason, it is required to define one of the two variables and evaluate the second. For example, in the case of piston motion, the mass flow rate is defined as: $\dot{m} = \rho A_P \dot{x_p}$ and the pressure value is computed by the characteristic function. In the case of the reservoir or an accumulator system, the pressure is set, and the mass flow rate is calculated. In the case of pipe connections, the boundary is calculated with function 5.7 where different pipe component values M_1^L and M_2^R are used. With this consideration, a multi-junction can also be defined. For instance, a T junction is calculated from a mass flow rate continuity equation with three different values of M^L and M^R.

Total Variation Diminishing algorithm

The Total Variation Diminishing scheme are algorithm that satisfy the inequality 5.10, where the quantity $TV(U) = \int_{-\infty}^{\infty} |\dot{U}(x)| dx$, .

$$TV(U^{n+1}) \leq TV(U^n) \tag{5.10}$$

To produce a TVD-MUSCL scheme the MUSCL scheme required a change to the linear extrapolation of the cell boundary defined in section 5.2. For the data reconstruction, a slope function of the variable U is calculated with the slope limiter function Ψ. The use of a slope limiter reduces noise and non-physical oscillations by computing the backward and forward information ratio r_i defined in equation 5.11.

$$r_i = \frac{U_i - U_{i-1}}{U_{i+1} - U_i}; \tag{5.11}$$

Equation 5.11 highlights the requirement for ghost cells to evaluate the first and the last real interface. The boundary interpolation scheme is used to ensure the same order of accuracy even through a boundary. In order to obtain a TVD algorithm, Sweby[72] provided the function limitation region

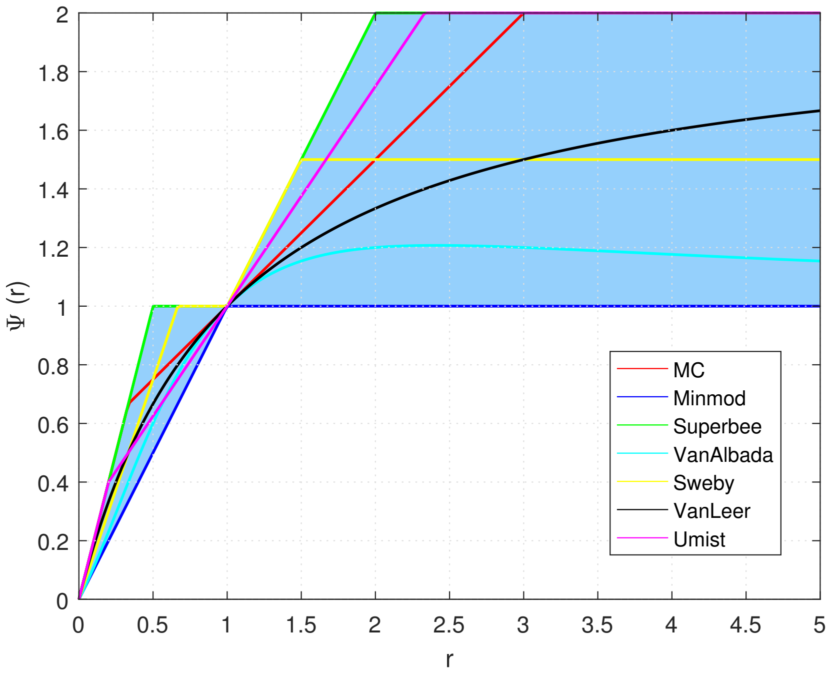


Figure 5.4: Sweby diagram for the slope limiter: MC, minmod, superbee, Van Albada, sweby, Van leer and Umist.

Ψ where the slope formulation must be bounded. Figure 5.4 shows the standard slope function and the TVD region in blue.

5.3 Ordinary Differential Equation

Ordinary differential equations are used in three main areas in this research. The first is related to the splitting method, in order to reintroduce the source term. The second is related to the valve dynamics to obtain the valve lift (section 5.6). The third is given to the Lumped Parameters Method (section 4.6). The ODEs can be solved with different strategies and a general view of the available approaches can be seen in figure 5.5[73].

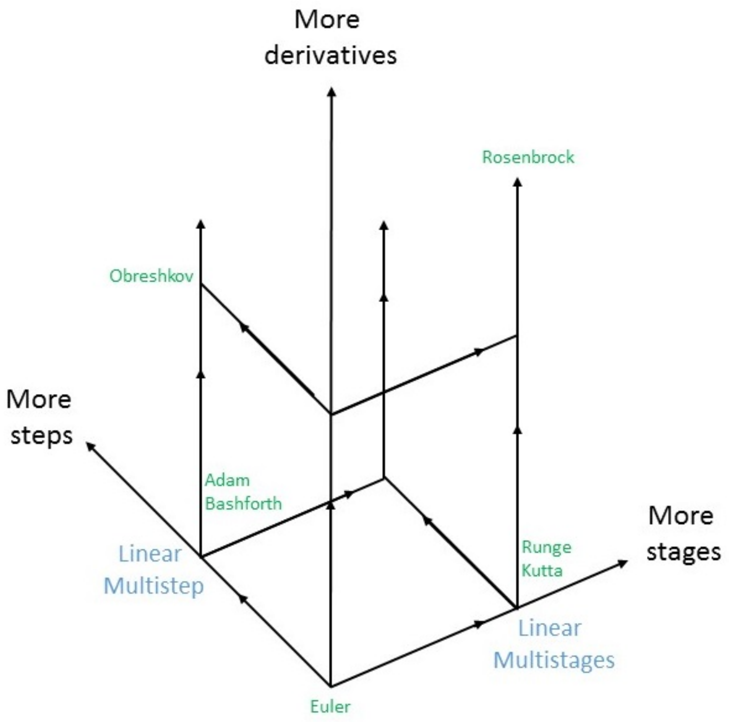


Figure 5.5: Scheme of ODE solver strategy.

As shown in the graph, there are three ways to categorize the ODE solver strategies.

- Steps, where previous time steps are considered.
- Stages, where more phases at the same time step are involved.
- Derivatives, where more terms of the Taylor series are used.

The first category includes the linear multistep algorithms where the solution is computed with a linear polynomial function of the previous time steps. To this category belong the Adam methods. The stages formulation, use a weight evolution formulation of the same time step. The derivative methods, use more terms of the Taylor derivative approximation. All of these strategies can be combined together to have a mix of algorithms. However, in this context, only the Runge-Kutta of the fourth-order (RK4), second-order Adam-Bashforth (AB) and Euler (EU) methods are analysed. The RK4 approach is used for its simplicity and computational performance.

$$\frac{\partial \tilde{U}}{\partial t} = \boldsymbol{s}(\tilde{U}) \tag{5.12}$$

The algorithm evaluates the slope derivative function in four consecutive evolution points.

$$U^{n+1} = U^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{5.13}$$

Where the term k_i is computed as:

- $k_1 = f(t^n, U^n)$, the derivative calculated from the slope at the previous time step t^n, equal to the Euler's method. This value is given by the solution of the homogeneous partial differential equation.
- $k_2 = f(t^n + \frac{\Delta t}{2}, U^n + \frac{k_1}{2})$, the derivative calculated from the slope at the midpoint, taking into account the increment given by half of k_1;
- $k_3 = f(t^n + \frac{\Delta t}{2}, U^n + \frac{k_2}{2})$, the derivative calculated from the slope at the midpoint, taking into account the increment given by half of k_2;
- $k_4 = f(t^n + \Delta t, U^n + k_3)$, the derivative calculated from the slope at the end of the time interval, taking into account the overall increment given by k_3.

To simulate the valve dynamics, Adam-Bashforth or the Euler method are used[73]. The RK4 could be used even in this case, however the required amount of coding is not trivial, although further research in that direction should be performed. The Euler method requires only one point of calculation and in the case of the explicit method has the form of equation 5.14.

$$U^n = U^{n-1} + \Delta t f(U^{n-1}) \tag{5.14}$$

Meanwhile, the explicit Adam-Bashforth method requires two previous time steps, and the formulation is given in equation 5.15.

$$U^n = U^{n-1} + \frac{3}{2}\Delta t f(U^{n-1}) - \frac{1}{2}\Delta t f(U^{n-2}) \tag{5.15}$$

5.4 Stability criteria

The Euler and Adam-Bashforth approaches are linear multistep methods for non-stiff problems. The common formulation of this method is given in equation 5.16.

$$U^n = \sum_{i}^{k} \alpha_i U^{n-i} + \Delta t \sum_{i}^{k} \Big(\beta_i f(U^{n-i})\Big) \tag{5.16}$$

The Euler method has $k = 1$ and $\beta_1 = 1$. While the Adam-Bashfort method has $k = 2$ and $\beta_1 = \frac{3}{2}$ and $\beta_2 = -\frac{1}{2}$. This notation allows the computation of a stability analysis in an efficient way. To evaluate the algorithm response, the characteristic polynomial of the methods λ, must be evaluated. To be stable the time step should satisfy the condition of $|\lambda| < 1$. The linear formulation of the AB method is reported in equation 5.17 while the characteristic polynomial is shown in 5.18.

$$U^n = U^{n-1} + \frac{3}{2}\Delta t q U^{n-1} - \frac{1}{2}\Delta t q U^{n-2} = \Big(1 + \frac{3}{2}z\Big)U^{n-1} - \frac{1}{2}z u^{n-2} \tag{5.17}$$

$$|\lambda| = |\omega^2 - \Big(1 + \frac{3}{2}z\Big)\omega - \frac{1}{2}z| \leq 1 \tag{5.18}$$

Solving the inequality formulation gives the region where the algorithm

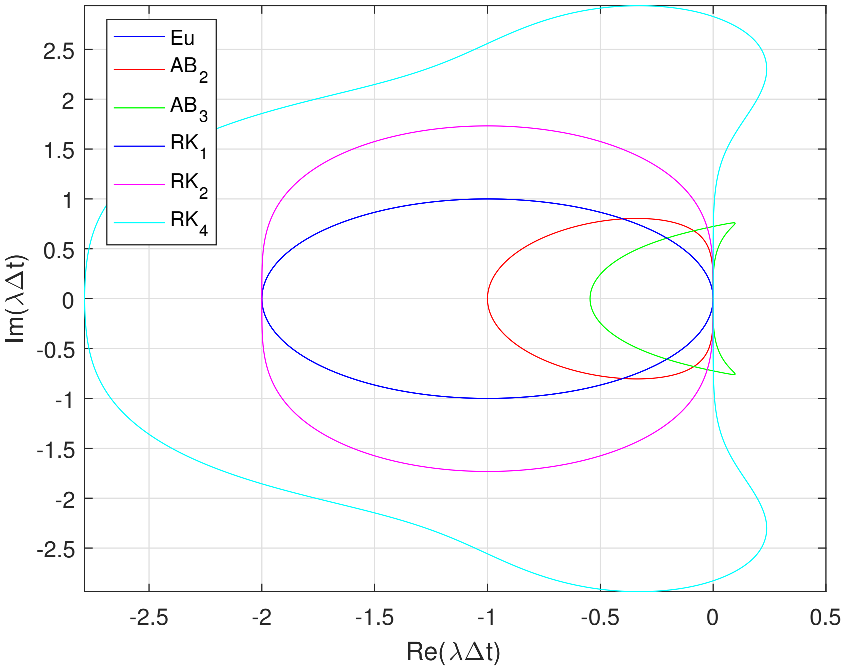


Figure 5.6: Stability Region.

is stable. Proceeding with the same method for RK4 and Euler algorithms, the stability region is calculated and plotted in figure 5.6.

The AB method is more accurate than the Euler method, however to be stable requires a smaller time step. This is not an issue, because the explicit MUSCL scheme requires the satisfaction of the Courant C_R condition that is smaller than for the other algorithms. In fact, the C_R condition, given in equation 5.19, is more restrictive than the other methods stability condition.

$$C_R = \frac{c\Delta t}{\Delta x} \leq 1 \tag{5.19}$$

The mathematical prove of the stability condition for MUSCL scheme can be found in[74] The value of the C_R number play an important role in the numerical dissipation phenomenon[25, 26].

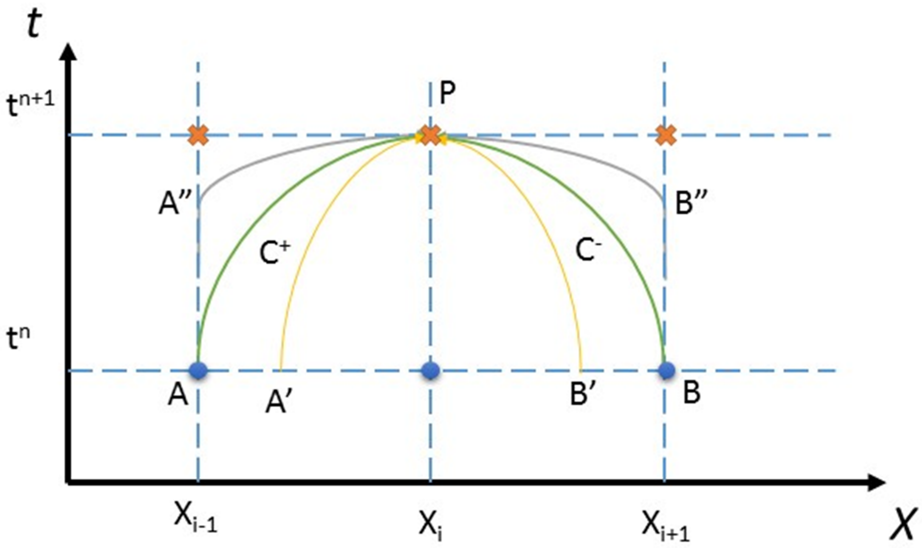


Figure 5.7: Characteristic grid.

5.5 Classic Water hammer formulation

Different, from the algorithm developed for the compressible FV algorithm, the water hammer formulation is commonly solved with the finite difference method. In this section only the theory of the solution is reported, while for a more complete description Wylie et al.[27] is cited. The Method Of Characteristic (MOC) is a mathematical technique to transform the partial differential equation into an ODE, by considering an external observer view. The water hammer system reported in equation 4.27 is then transformed into the system 5.20.

$$\begin{cases} \frac{dQ}{dt}\frac{c}{gA} + \frac{dH}{dt} + \frac{f}{2gDA^2}|\boldsymbol{Q}|\boldsymbol{Q} = 0, for\frac{dx}{dt} = c \\ \frac{dQ}{dt}\frac{c}{gA} - \frac{dH}{dt} + \frac{f}{2gDA^2}|\boldsymbol{Q}|\boldsymbol{Q} = 0, for\frac{dx}{dt} = -c \end{cases} \tag{5.20}$$

Referring to figure 5.7 and gathering together the known magnitude $B = \frac{c}{gA}$ and $R = \frac{f\Delta x}{2gDA^2}$, the solution by the finite difference method allows the solution of the linear system of equations 5.21.

$$\begin{cases} C^{+} : H_P = C_P - B_P Q_P \\ C^{-} : H_P = C_M - B_M Q_P \end{cases} \tag{5.21}$$

The four parameters are related to the previous time step by the relation 5.22.

$$\begin{cases} C_P = H_{i-1} + BQ_{i-1}, B_P = B + R|Q_{i-1}| \\ C_M = H_{i+1} + BQ_{i+1}, B_M = B + R|Q_{i-1}| \end{cases} \tag{5.22}$$

The power of this method is the computational calculation efficiency. In fact, the equations 5.22 show that each point is related only to the adjacent points, $i \pm 1$ and not to the previous i. This allows the algorithm to be further simplified with a staggered grid, namely considering only half of the fluid domain. However, as shown in figure 5.7 the information must be given from the points A and B. If this condition is not respected, interpolation schemes described in the literature review, section 3.2, must be introduced. This methods introduce higher numerical dissipation phenomena.

5.6 Valve: numerical solution

The fluid behaviour at the valve is modelled by taking into account the dynamic response of the valve itself. Therefore, to evaluate the valve lift and consequently the valve seat gap area , it is required to solve the Newton's second law. At this stage the forces acting on the valve are all explicitly calculated as equation 5.23.

$$m_v \ddot{x_v} = F_P - F_K - F_{Pre} - F_C - F_G \tag{5.23}$$

where:

- Pressure force, $F_P = \Psi A_V (p^n_{n+\frac{1}{2}Down} - p^n_{\frac{1}{2}Up})$,

- Spring force, $F_K = K_s x_v^n$, where K is the spring characteristic function.
- Preloaded force F_{Pre}
- Viscous damping force $F_C = C_d \dot{x}_v^n$
- Gravity force F_G

Where the superscript n indicates the previous time step. The pressure force coefficient, Ψ is calculated using equation 4.42. The three empirical value reported in equation are reported in appendix A.5. Numerical instability occurs for the case of closed valve due to the term $log(Re_{gap})$ in equation 4.42, that is overcome by considering a minimum value of $Re_{Gap_{min}} = 1e^{-6}$. The value of $(p^n_{n+\frac{1}{2}})_{Down}$ and $(p^n_{\frac{1}{2}})_{Up}$ refer to the boundary pressure values at the downstream and upstream pipe that are connected to the valve. Solving the valve dynamic equation is performed with a Leapfrog integration method[73] where each stage is calculated with the AB method. In detail, the valve velocity is obtained by equation 5.24 with the approximated form given in equation 5.17 is applied.

$$\dot{x}_V^{n+1} = \dot{x}_V^n + \frac{\sum F_i}{m} \tag{5.24}$$

While the valve lift is calculated with the implicit formulation of the velocity, equation 5.25.

$$x_V^{n+1} = x_V^n + f(\dot{x}_V^{n+1}) \tag{5.25}$$

Once the gap area is calculated from the valve lift, the mass flow rate is implicitly calculated. An iterative implicit solution of equation 4.40 is solved with the Newton-Rhapson method. The algorithm consists of:

- a guess value of the mass flow rate, using a random function, is calculated.

- calculate the function $f(\rho Q)$ (equation 4.40) and its derivative $\dot{f}(\rho Q)$
- evaluate the value of $\rho Q = \rho Q_{guess} - \frac{f(\rho Q)}{\dot{f}(\rho Q)}$
- calculate the relative error $err = |\rho Q - \rho Q_{guess}|$
- continue the iteration if the relative error has not reached the specified value.

The implicit iterative solution of the boundary value was required to avoid numerical instability.

5.7 Pipe fluid transient structure code

The fluid transient algorithm in a simple duct is the baseline code for the pump simulation. In addition, the developed code was used primarily to validate the result with the methodology available in the literature. The code structure is shown in figure 5.8 where the steps can be described as:

1. initialize the domain with the initial data;
2. calculate the optimal time step to satisfy the Courant condition from the speed of sound;
3. check if the time exceeds the final time;
4. calculate the boundary conditions in terms of pressure and mass flow rate;
5. apply the MUSCL scheme and the TVD limiter;
6. check the pressure value with the cavitation model;
7. update the variables: density, volume fraction of vapour and non-condensable gas, speed of sound and eigenvalues;

8. goto point 2

The strategy to reach the solution, passes to the cavitation algorithm, where an if clause is performed. The algorithm calculates the pressure at each half cell, and the following possibilities are considered:

- pressure higher than cavitation pressure;
- pressure higher than cavitation pressure but with cavity volume bigger than zero;
- pressure lower than cavitation pressure.

The cavitation pressure value is related with the temperature by data available in literature that for 20 degree gives a vapour pressure equal to 2338 Pa[75].

Therefore, the cavitation algorithm consists of three different output conditions. Where the pressure is found not to be less than the cavitation pressure and the cavitation is zero, the algorithm continues with the calculation of the fluid variables. When the pressure is lower than the cavitation pressure, the pressure is imposed at the vapour pressure value. When the pressure is higher than the cavitation pressure, but vapour bubbles are still present, the algorithm keeps the pressure at the vapour pressure level. When the pressure exceeds the vapour condition and all the formed bubbles collapse, the algorithms continue with the ideal strategy.

The cavitation algorithm calculates the amount of the vapour volume integrating the equation 4.31. The value is computed with the equation 5.26:

$$V_{cav} = \int_{t^n}^{t^{n+1}} \left(\frac{\rho Q_j^{n+1}}{\rho_{m_j}^{n+1}} - \frac{\rho Q_{j+1}^{n+1}}{\rho_{m_{j+1}}^{n+1}} \right) dt \tag{5.26}$$

Referring to figure 5.9, the algorithm allows cavitation to be formed, grow and collapse only in between the two half cells. Therefore, in the case

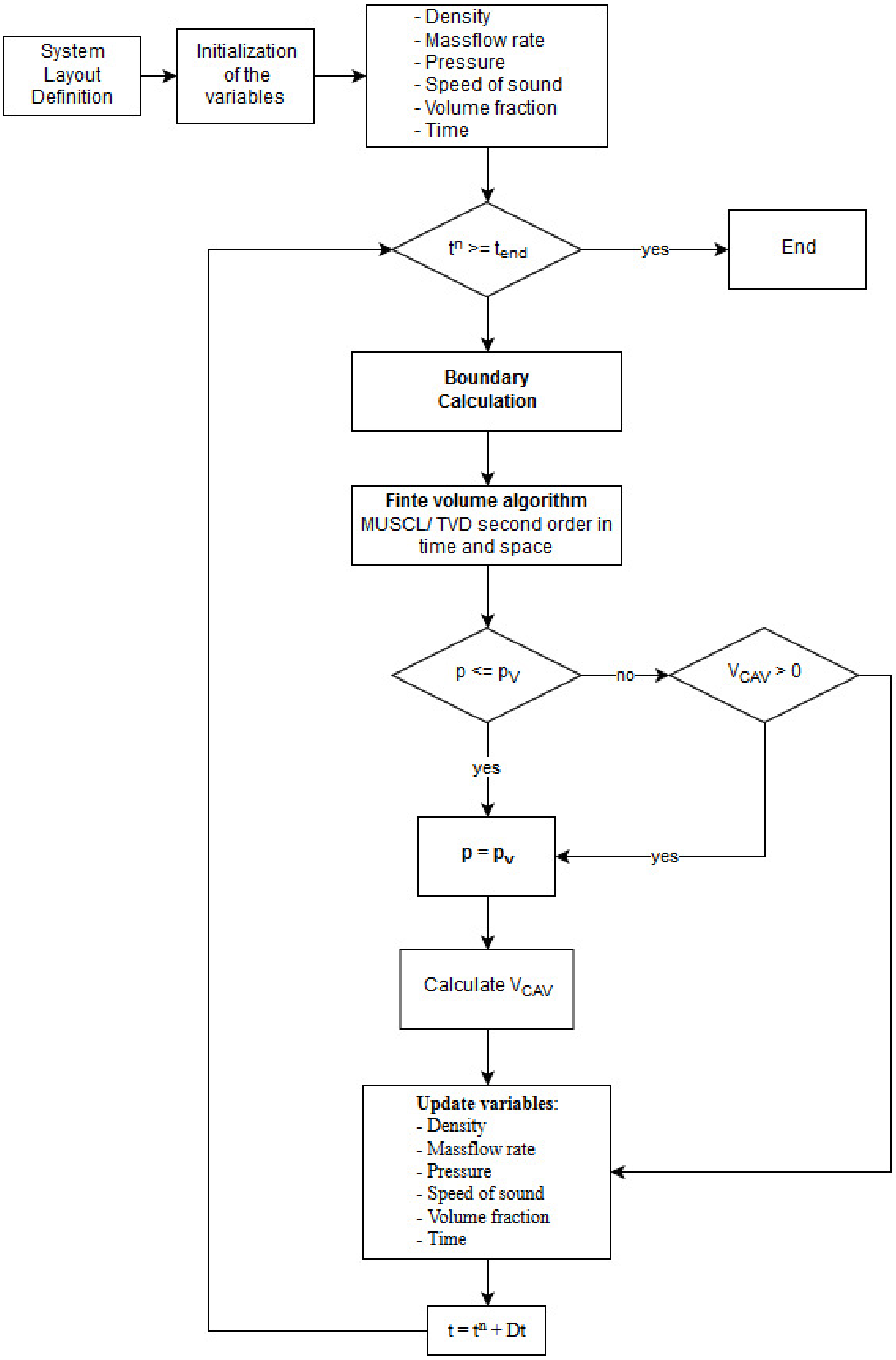


Figure 5.8: Code structure for water hammer simulation.

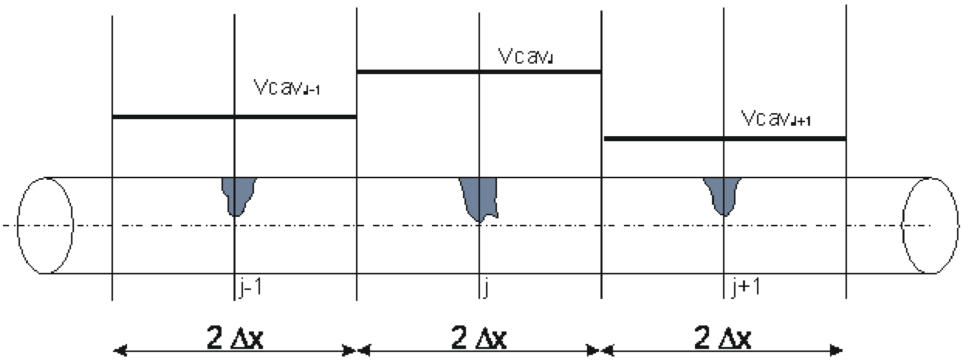


Figure 5.9: Cavity grid description.

of a pressure lower than the vapour pressure in one or in both cells, the pressure is set to the cavitation value in both.

Given the pressure, the vapour volume fraction α_i and the mass flow rate in the entire domain, the algorithm calculates the mixture density, with the homogeneous equation 5.27.

$$\rho_m = \rho_L\Big(1 - \sum_i^N \alpha_i\Big) + \sum_i^N \rho_{\alpha_i}\alpha_i \tag{5.27}$$

Each term of equation 5.27 is calculated with equations 5.28.

$$\begin{cases} \rho_L = \rho_{L_{Ref}} exp^{\frac{1}{K}(p-p_{ref})} \\ \rho_{\alpha_i} = \frac{p}{R_{\alpha_i}T_{amb}} \end{cases} \tag{5.28}$$

At this stage, the algorithm proceeds to calculate the speed of sound of the fluid mixture with the equation 5.29 given by Brennen[51]. This approach consider the overall behaviour of the homogeneous fluid and no different slip velocity between the phases is computed.

$$\frac{1}{c_m^2} = \rho_m\Big(\frac{\sum_i^N \alpha_i}{kp} + \frac{1 - \sum_i^N \alpha_i}{\rho_L c_L^2}\Big) \tag{5.29}$$

Figure 5.10 shows the effect that the second phase has on the speed of sound. The graph consists of a three-dimensional logarithmic plot wherein

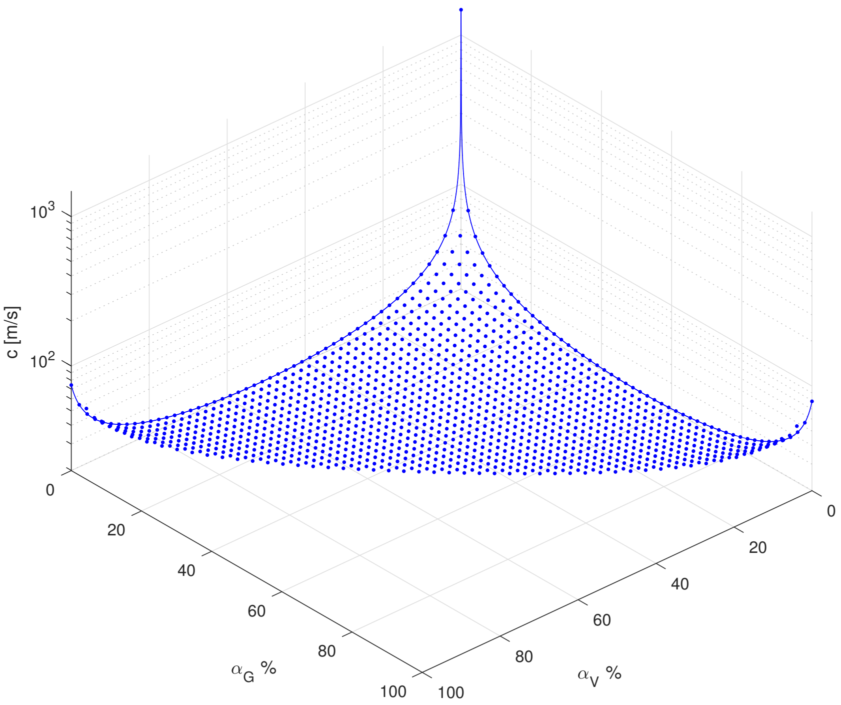


Figure 5.10: Variation of speed of sound.

the XY plane evaluates the volume fraction of the vapour and non-condensable gas phases. A small percentage of the second phase changes the propagation velocity drastically. Therefore this step is a crucial point in the algorithms to evaluate the effect of the cavitation phenomenon effectively.

Thanks to the simplicity of the system simulated, namely a straight pipe, a further improvement in the speed of calculation is performed. A Fluid-Structure Interaction (FSI) model is included to calculate the speed of sound for pure liquid and corrected with the effect of the pipe elasticity. The algorithms consists of the Korteweg formulation[76] shown in equation 5.30.

$$\begin{cases} \beta = \frac{2}{[(\phi+2s)^2-\phi^2]}((1-\nu)\phi^2+(1+\nu)(\phi+2s)^2) \\ c_L^* = \sqrt{\frac{B}{\rho_L}} \\ c_L^2 = \frac{{c_L^*}^2}{1+\frac{\rho_L \beta {c_L^*}^2}{E}} \end{cases} \tag{5.30}$$

where ν is the Poisson value, s the thickness, ϕ the diameter and B the bulk modulus. Two different approaches for the discrete cavity algorithm can be used: Discrete Vapour Cavity Model (DVCM) and Discrete Gas Cavity Model (DGCM). The two methods are similar except that the DGCM includes the non dissolved gas in the solution, a further phase. The concentration of Non dissolved gas is expressed in volume fraction α_{Gas0}. The DGCM uses the same algorithm developed by Zhou et al.[32] where a Non condensable gas parameter (NCG_p) is used to prevent numerical instability. This parameter has a significant effect on the numerical dampening of the scheme. When the NCG_p value is equal to one, the gas initially dissolved in the fluid is entirely ignored making the DGCM equal to DVCM, whereas when it is equal to zero, the cell assume that each cell has the gas pressure. The pressure equation is then calculated with the equation 5.31[32].

$$p_j^{n+1} = NCG_p \bullet p_j^{n+1} + (1-NCG_p) \bullet p_{gas_j} \tag{5.31}$$

5.8 Pump modelling approach

The structure of the code for evaluating the pump performance is slightly different from that for a pipe. First and foremost, the FSI for the pure liquid is not active in this context for two main reasons:

- the stiffness of the material in the pump is much higher than a simple pipe, therefore the variation is negligible.
- the thickness of the wall is not always constant, hence the Korteweg

equations are not easily applicable.

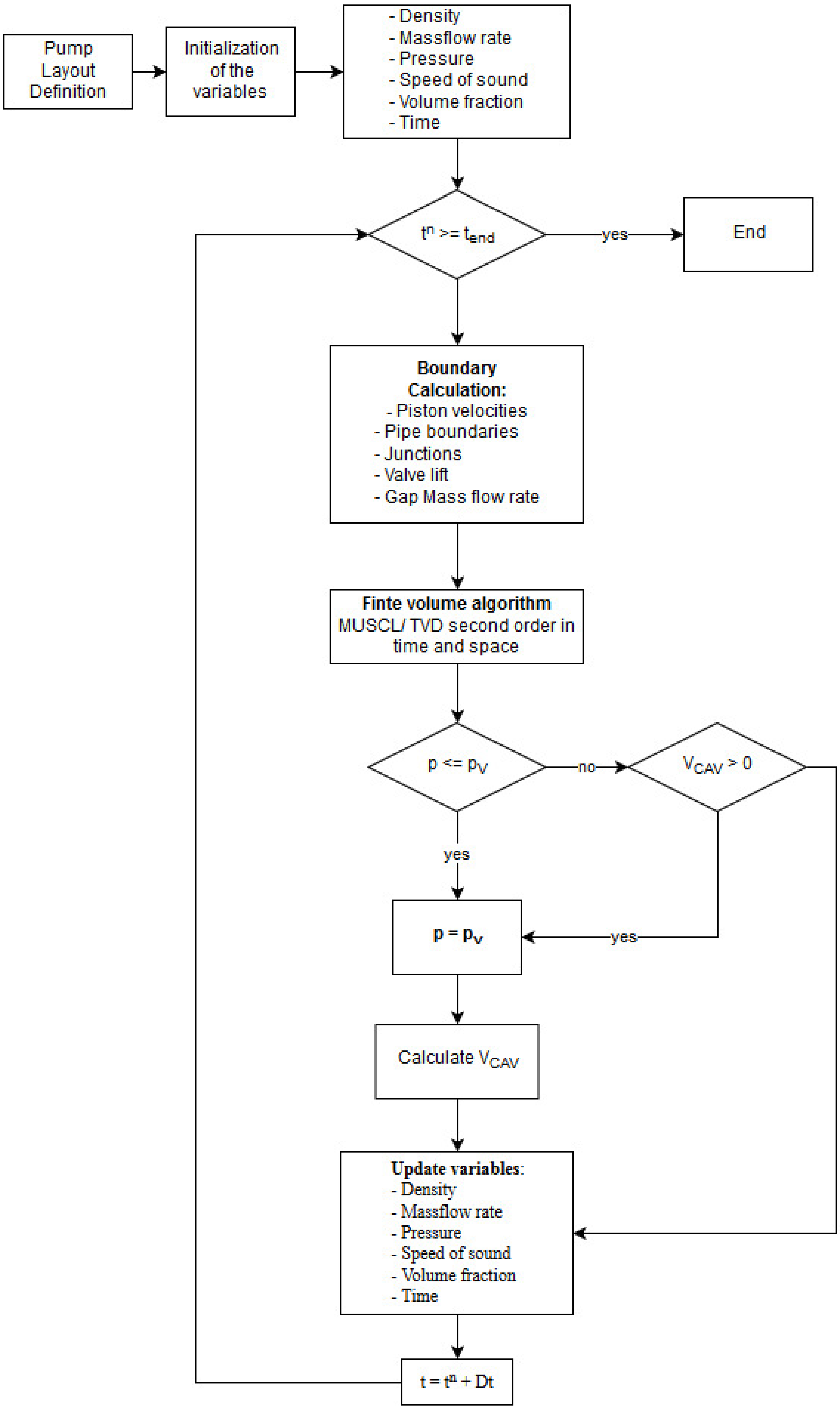


Figure 5.11: Code structure for pump simulation.

Therefore the bulk modulus speed of sound formulation is used for the liquid phase.

Secondly, the system network requires to model junctions, valves and hydraulic dissipation systems. These parts are considered as internal boundaries and solved at the beginning of each time step solution. The logical structure of the code is shown in figure 5.11.

5.9 Conclusion

In this section the numerical methods used to solve the system of equation developed in the previous section were explained. The section included a description of the MUSCL scheme with the TVD algorithm and the application to the ordinary differential equations used for solving the valve dynamics and other components. In addition the classical water hammer formulation was shown to give a broader understanding when methods are compared in the next section. The code structure for both pump and simple pump duct was given for an understanding of the procedure to reach the solution. This section is the baseline required to understand the validation of the method given in section 6 and the simulation of the pump analysed in section 7.

Chapter 6

Validation of pressure waves model

Although the aim of this project is to model a positive displacement pump including cavitation, the algorithm was initially validated for a single duct. Therefore, in this section, the comparison of the developed model (DM) by the author with data available from the literature is given.

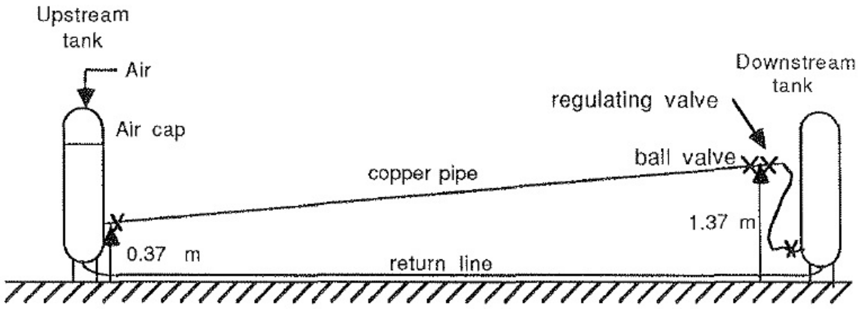


Figure 6.1: Test rig of Simpson experimental data[77].

Two different experimental rig layouts were considered, and in figure 6.1 is shown the layout of the Simpson experimental test[77].

- Simpson, with a 36 m copper pipeline, connecting tanks with a height difference of 1.37 m, and an inner diameter of 0.01905 m.

- Pezzinga and Cannizzaro, with a 37.2 m copper pipeline, connecting tanks with a height difference of 2.03 m, and inner diameter 0.0221m.

For both test rigs, the transient fluid phenomenon was created by a sudden closure of a ball valve mounted at one end of the pipe. However, Pezzinga and Cannizaro modelled the closure of the valve with a specific time of closure T_C, reported in table 6.1. Simpson used the exact fluid velocity profile at the valve extrapolated from the experimental observation.

A summary of the data sources and simulation characteristics are shown in table 6.1.

Table 6.1: Experiment cases

# Case	V_0 **[m/s]**	p_{ref} ***[Bar]***	T_c ***[sec]***	***D [m]***	***L [m]***	***c [m/s]***	***Source***
1	0.239	0.9918	-	0.01905	36	1280	Simpson[77]
2	0.332	0.9917	-	0.01905	36	1280	Simpson[77]
3	0.401	0.9939	-	0.01905	36	1280	Simpson[77]
4	0.466	0.9939	-	0.01905	36	1280	Simpson[77]
5	0.507	0.9938	-	0.01905	36	1280	Simpson[77]
6	0.596	0.9918	-	0.01905	36	1280	Simpson [77]
7	0.696	0.9918	-	0.01905	36	1280	Simpson[77]
8	0.938	0.9918	-	0.01905	36	1280	Simpson[77]
9	1.125	0.9918	-	0.01905	36	1280	Simpson[77]
10	0.300	2.1539	0.006	0.0221	37.2	1318	Pezzinga[37]
11	1.400	2.1539	0.006	0.0221	37.2	1318	Pezzinga[37]

The idea behind this two approaches is to evaluate the wave propagation in pipe and indirectly calculate the cavitation phenomenon effect. In fact, the amount of cavitation is extrapolated from the pressure history profile.

6.1 Experiment comparison

The solution of the partial differential equations by the finite volume method depends on several different factors. The behaviour of the solution for the developed model can change due to numerical or model parameters. Therefore the models performance is evaluated in relation to:

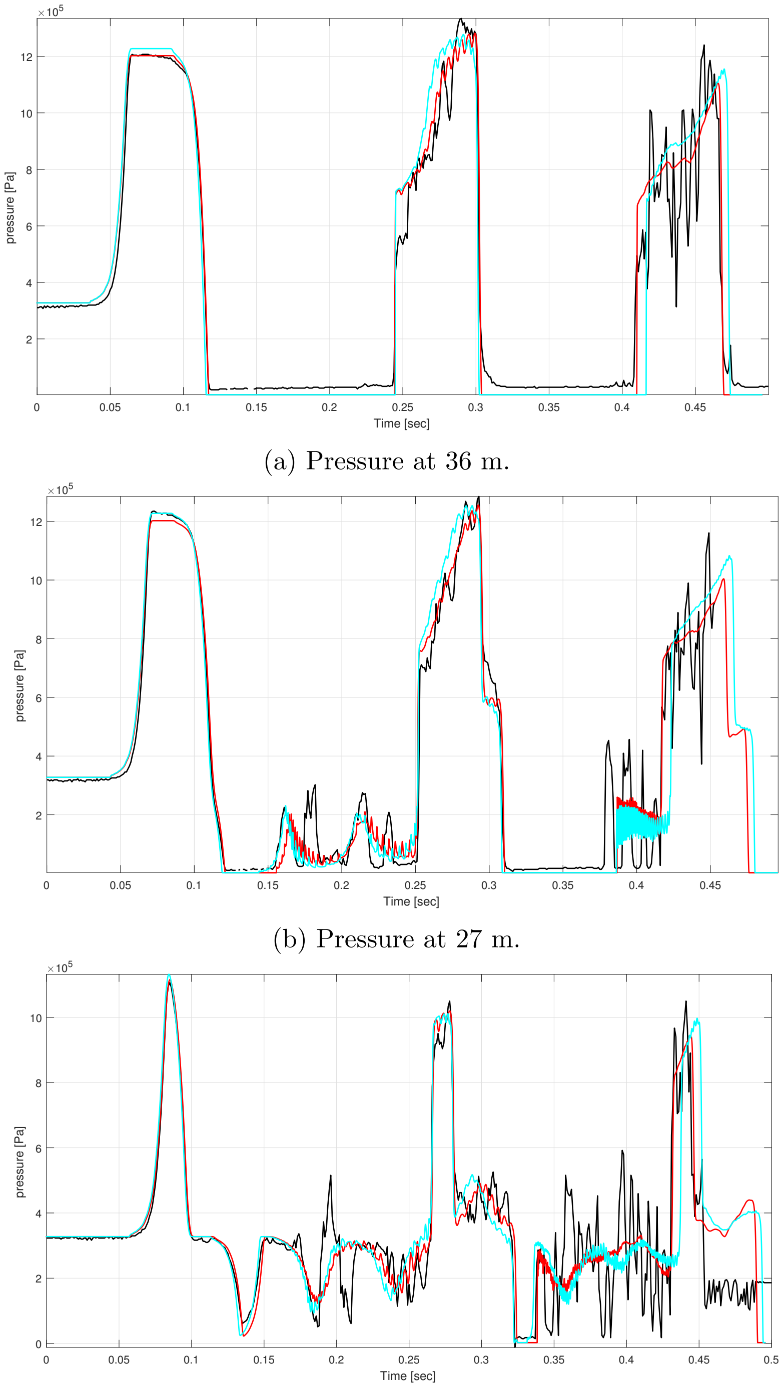
- *Compressibility*, where the compressible developed model is compared with the constant density developed model, in finite volume treatment.
- *DGCM/DVCM*, where the two multi-phases fluid flow models are compared.
- α_{Gas}, where the model response is studied as a function of the initial gas in solution.
- *Grid size*, where different grid size values are used.
- *Courant Number*, where the stability condition is reduced.
- *Advective Term*, where the effect on the model of neglecting the advective term is assessed.

6.1.1 Compressibility

Compressibility is the capacity of the fluid to change its volume in relation to the applied pressure. The developed algorithm consists of a homogeneous flow, and the fluid can be a mixture of gas and liquid. For this reason, the compressibility can play an essential role due to the different responses that the two phases have. To investigate these two different cases the model settings shown in table 6.2 were used.

Table 6.2: Simulation parameters for developed model vs incompressible DM

Simulation Variable	
C_R	*0.9*
Grid size	*0.14 m*
Algorithm	*DGCM*
Initial gas in solution α_{Gas0}	10^{-7}



(a) Pressure at 36 m.

(b) Pressure at 27 m.

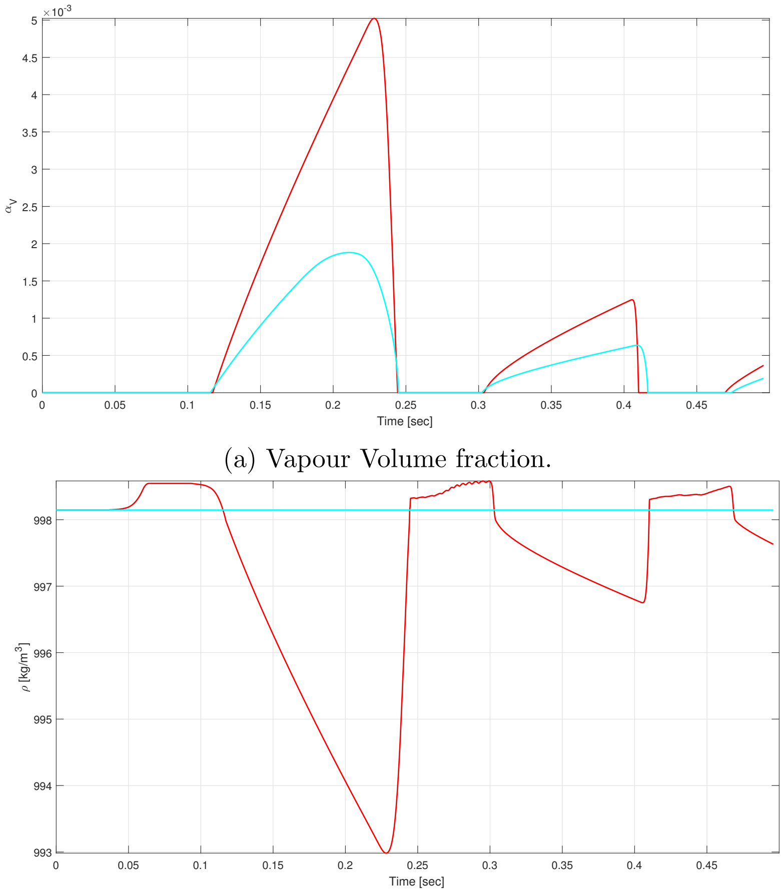
(c) Pressure at 9 m.

Figure 6.2: Evaluation of the compressibility effect for the developed model for case 7. Red line is DM, blue line is constant density developed method and the black line is the experimental data.

Figures 6.2 and 6.4 are respectively the pressure profile for the cases 5 and 7 in table 6.1. For Figures 6.2, a series of three wave reflections are captures. Figure 6.2a represent the pressure history at the valve, where the black line is the measured value captured with a pressure transducer.

Figure 6.2b and 6.2c are the pressure profiles respectively at 27 meter and 9 meter. The high pressure wave shown in the picturs are refered to the cavitation bubble collapse. In both cases analysed, the compressible algorithm (DM) matches the experimental results better than a constant density developed algorithm, where the later overestimates the maximum pressure value. In addition, the constant density developed strategy does not predict exactly the time of the third reflection wave for case 7, figure 6.2.

Differences are highlighted in the peak pressure where a lower value is reached by the compressible model. For the Joukowsky formulation, the variation in the pressure is given by the variation in velocity multiplied by the speed of sound and the density. The compressible model has two density variations, given by the presence of the second phase and the liquid density variation. Albeit, the later is commonly negligible for the experiment is pressure range. However, the second phase reduces the overall density, and the effect can be seen in figure 6.3 where both the density and the vapour volume fraction at the valve are given. The variation in the density is not high, for this velocities. The maximum pressure variation is 0.5% and reader can judge of the value of this approach. The algorithm is performed for describing the pump behaviour where the fluid velocity around the valve can reach dozens of meter per second and the variation due to cavitation can change reducing the overall density. However in that condition is more important the variation in the speed of sound related function of the density, as shown in figure 5.10.

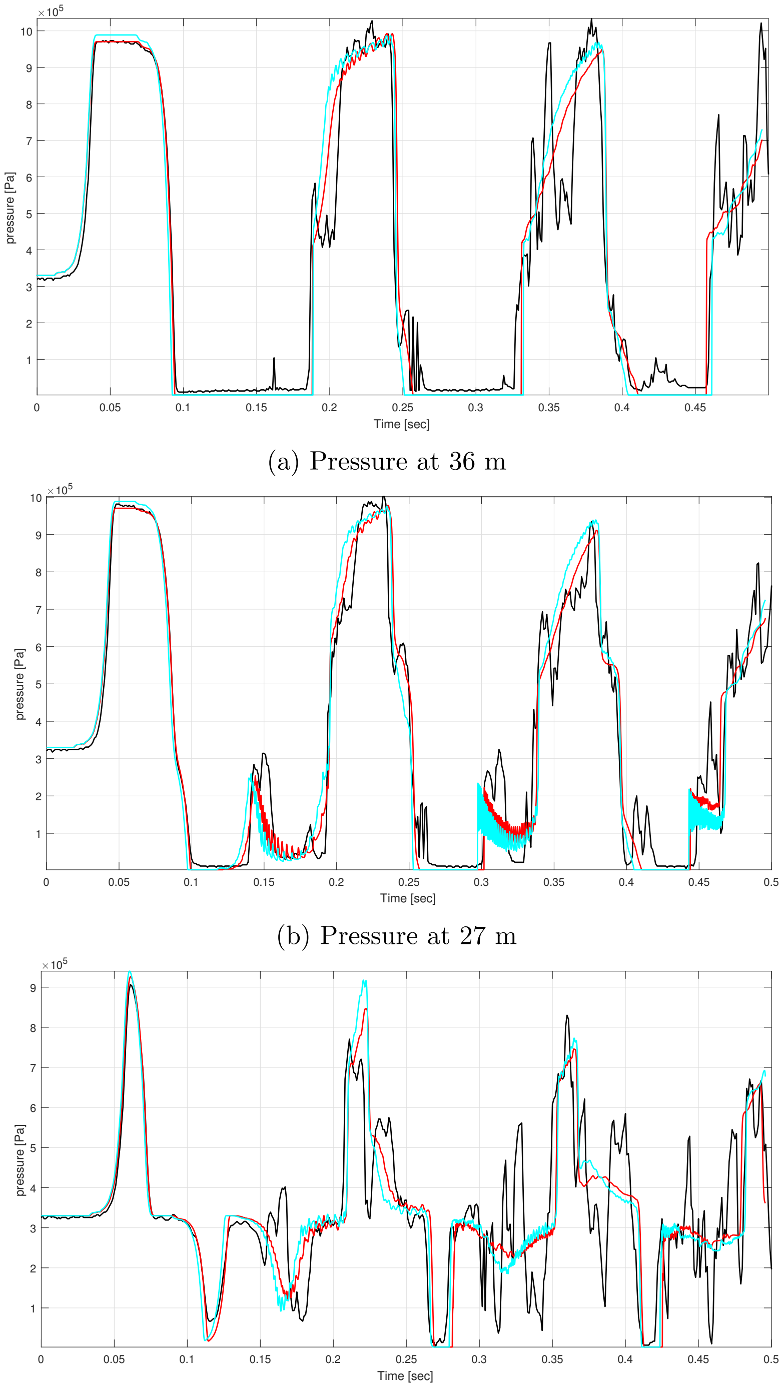


(a) Vapour Volume fraction.

(b) Density.

Figure 6.3: Density and vapour volume fraction profile for compressibility effect evaluation for case 7. Red line is the developed model and blue line is the constant density developed method.

In case 5, the differences are less evident. this time the simulated time allow to capture four wave reflection. The delay in the reflection wave is still present although minimal as shown in figure 6.4. More evident is the higher value of pressure reached for the constant density developed algorithm rather than the compressible method, especially for the first peak pressure at 9 m, figure 6.4c. In addition, the compressible method better matches the experimental result compared to the constant density developed algorithm in terms of time for the wave reflection. A high-frequency response is shown in all of the pressure history profiles. This phenomenon is also present in



(a) Pressure at 36 m

(b) Pressure at 27 m

(c) Pressure at 9 m

Figure 6.4: Evaluation of the compressibility effect for the developed model for case 5. Red line is DM, blue line is constant density developed method and the black line is the experimental data.

the algorithms of Daude et al.[48], Pezzinga and Cannizzaro[37], and Zhou et al.[31, 32].

6.1.2 Comparison DGCM and DVCM

In this section, a comparison of the results of the DM using the discrete gas cavity model and the discrete vapour cavity model is given. The simulations are shown respectively for case 2 and 8 in table 6.1, and the simulation parameters are reported in table 6.3.

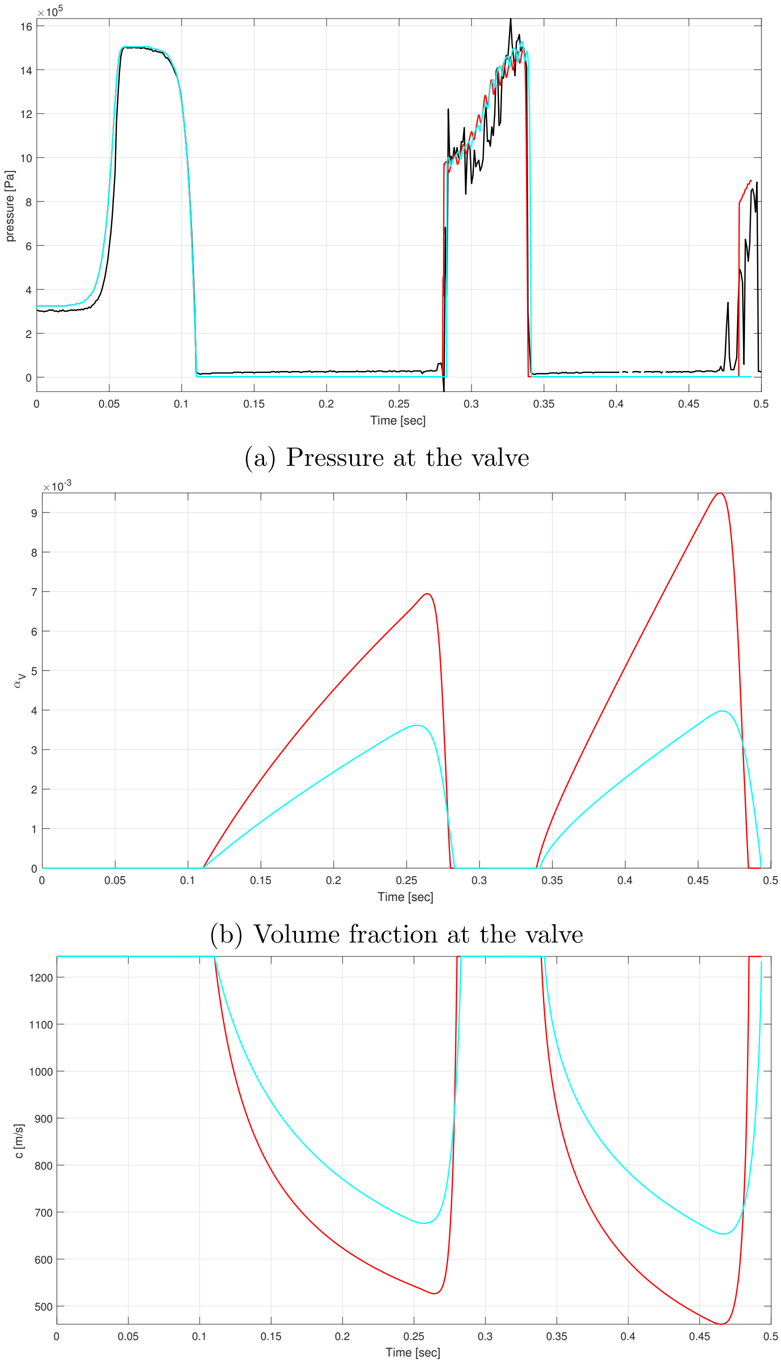
Table 6.3: Simulation parameter for DGCM and DVCM comparison

Simulation Variable	
C_R	*0.9*
Grid size	*0.14 m*
Initial gas in solution α_{Gas0}	10^{-7}

The response of the algorithms are very similar as shown in figure 6.5. The effect of the NCG is limited. However, the presence of the third fluid changes the response of the maximum pressure peak and in the time of the wave reflection. The pressure peak value is slightly lower for the DGCM method, due to the reduction in density and speed of sound.

However, considering case 8, the main difference is shown in the second wave reflection. In figure 6.5a, close to 0.5 seconds the DGCM produced the peak pressure that is not present in the DVCM. The reason for this behaviour is found in figure 6.5b. The cavity collapse time for the DGCM is different from the DCVM case. Considering equation 6.1, the density in the denominator is different for the two models. For the same mass flow rate, the variation of the cavity volume is more relevant for the DGCM algorithm due to the less density value.

$$V_{cav} = \int_{t^n}^{t^{n+1}} \left(\frac{\rho Q_j^{n+1}}{\rho_{m_j}^{n+1}} - \frac{\rho Q_{j+1}^{n+1}}{\rho_{m_{j+1}}^{n+1}} \right) dt \tag{6.1}$$

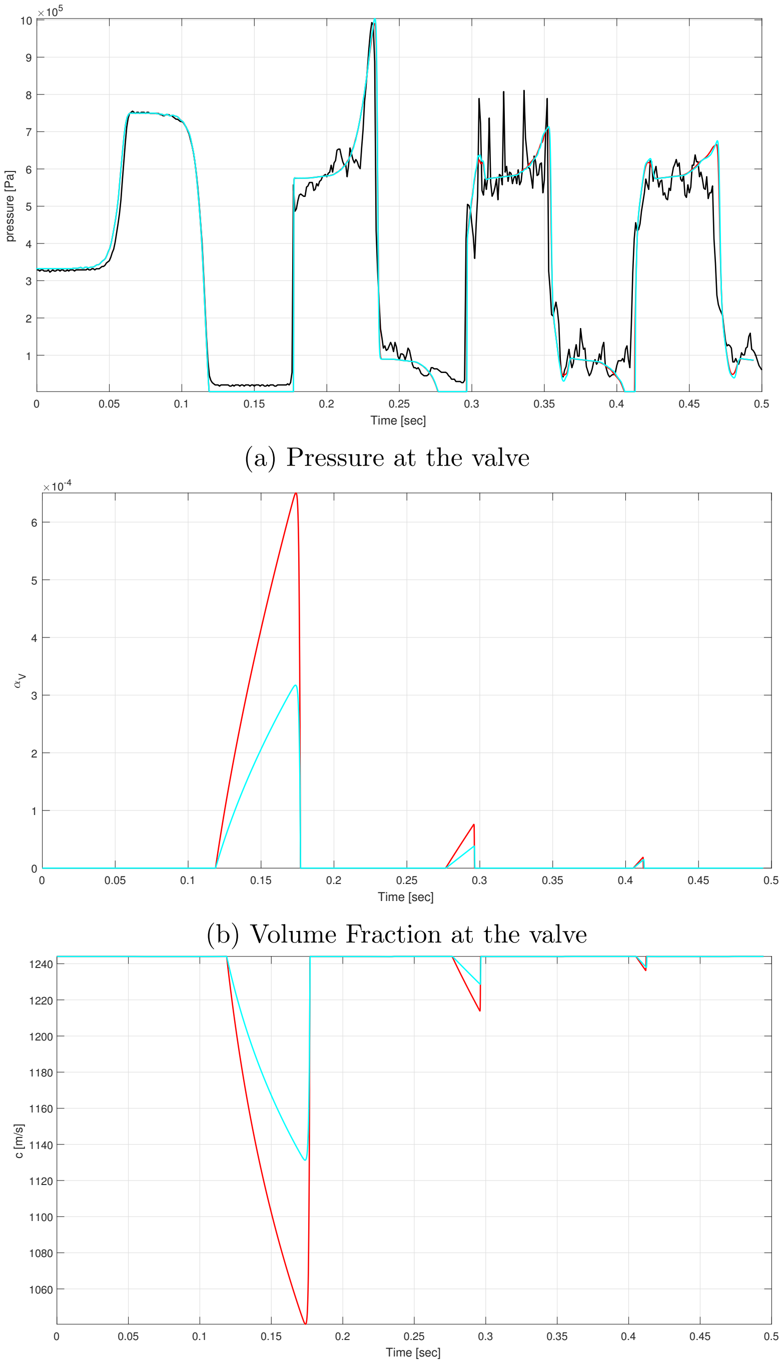


(a) Pressure at the valve

(b) Volume fraction at the valve

(c) Speed of Sound at the valve

Figure 6.5: Evaluation of the DGCM and the DVCM algorithms for the developed model for case 8. Red line is DM with DGCM, blue line is the DM with DVCM and the black line is the experimental data.



(a) Pressure at the valve

(b) Volume Fraction at the valve

(c) Speed of Sound at the valve

Figure 6.6: Evaluation of the DGCM and the DVCM algorithms for the developed model for case 2. Red line is DM with DGCM, blue line is the DM with DVCM and the black line is the experimental data.

The differences between the models becomes less evident when the speed is reduced to the value of the case 2, as shown in figure 6.6. Even in this case the amount of vapour affects the pressure value history, although barely seen in figure 6.6a.

Summarizing, the two algorithms performed in a similar way, although the presence of the third fluid improves the response of the model. The same trend of behaviour can be highlighted by comparing the DGCM and DVCM model performed by Zhou et al.[31, 32].

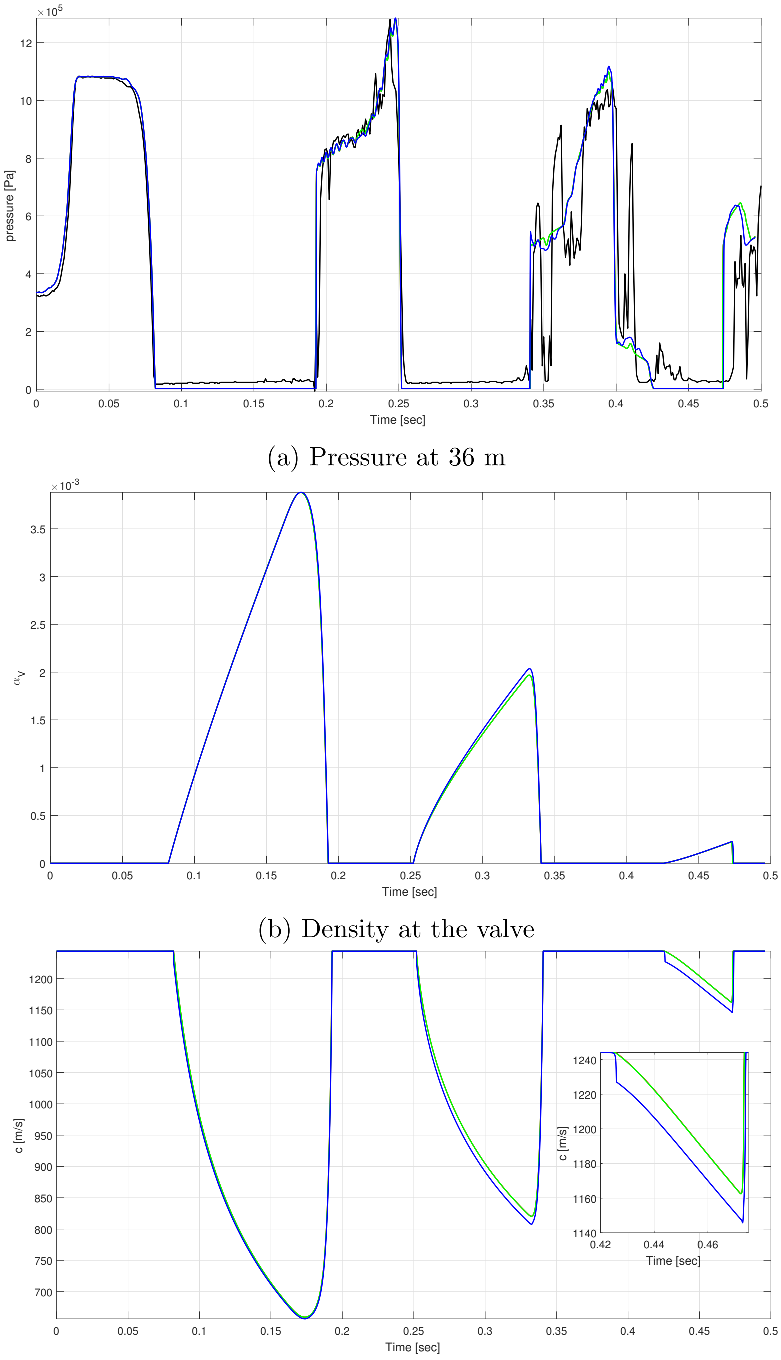
6.1.3 Influence of the initial dissolved gas value

In this section, the effect of non-condensable gas is studied. The simulated parameters are reported in table 6.4. The initial gas fraction values are $\alpha_{Gas0} = 1^{-10}, 1^{-7}, 1^{-5}$. The simulation is shown for cases 6 and 3 respectively.

Table 6.4: Simulation parameters of initial void fraction comparison

Simulation	***Variable***
C_R	*0.9*
Grid size	*0.14 m*
Algorithm	*DGCM*

The variation in the pressure profile is not really significant for the simulated range. The main differences are shown in cavity formation and the speed of sound. Both figure 6.8 and 6.7 show the same fluid behaviour, although differences are shown in the figure 6.7c and 6.8c. Higher content of NCG creates a higher speed of sound variation. Therefore, only a significant amount of NCG changes the algorithm response, agreeing with the results given by Wylie et al.[27].

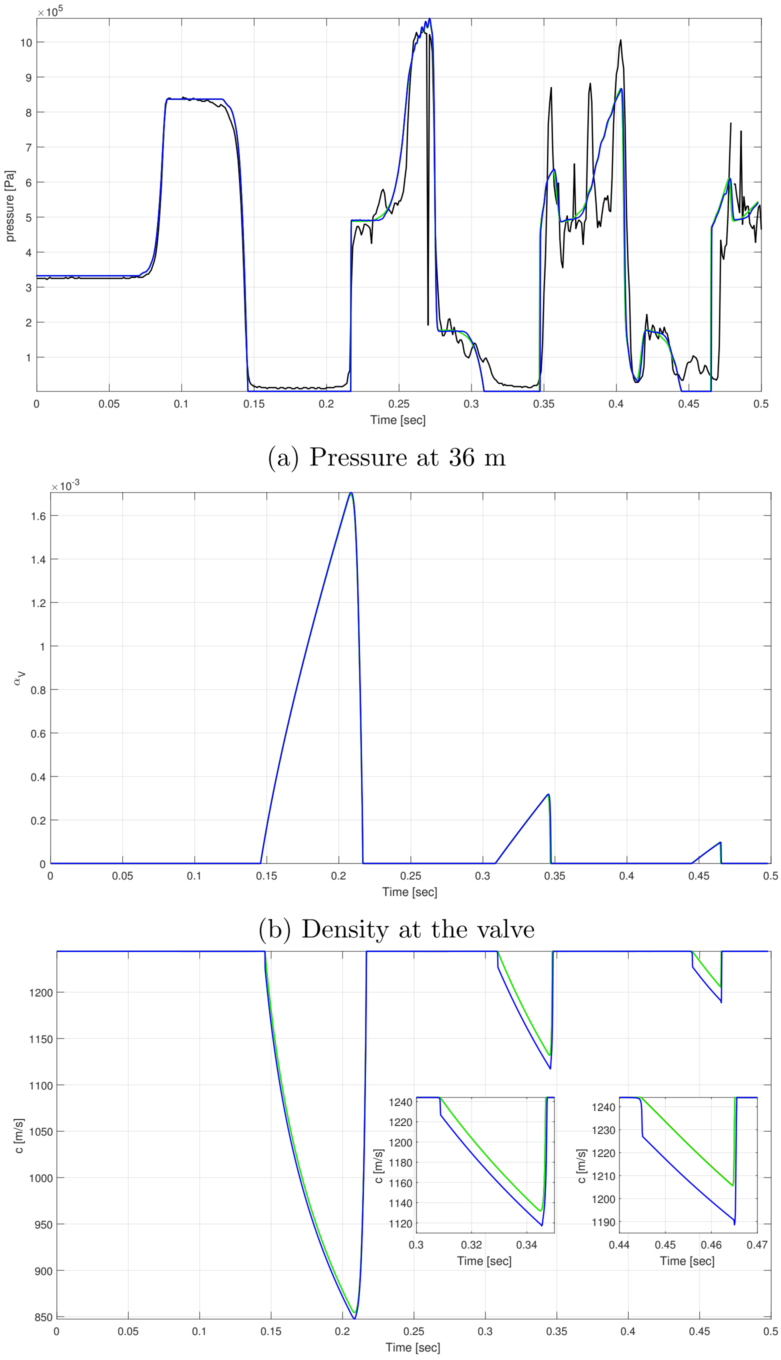


(a) Pressure at 36 m

(b) Density at the valve

(c) Speed of Sound at the valve

Figure 6.7: Evaluation of the void fraction effect for the developed model for case 6. red line is DM with $\alpha_G = 1^{-10}$, green line is the DM with $\alpha_G = 1^{-7}$, blue line is the DM with in $\alpha_G = 1^{-5}$ and black line is the experimental data.



(a) Pressure at 36 m

(b) Density at the valve

(c) Speed of Sound at the valve

Figure 6.8: Evaluation of the void fraction effect for the developed model for case 3. red line is DM with $\alpha_G = 1^{-10}$, green line is the DM with $\alpha_G = 1^{-7}$, blue line is the DM with in $\alpha_G = 1^{-5}$ and black line is the experimental data.

6.1.4 Influence of the grid size

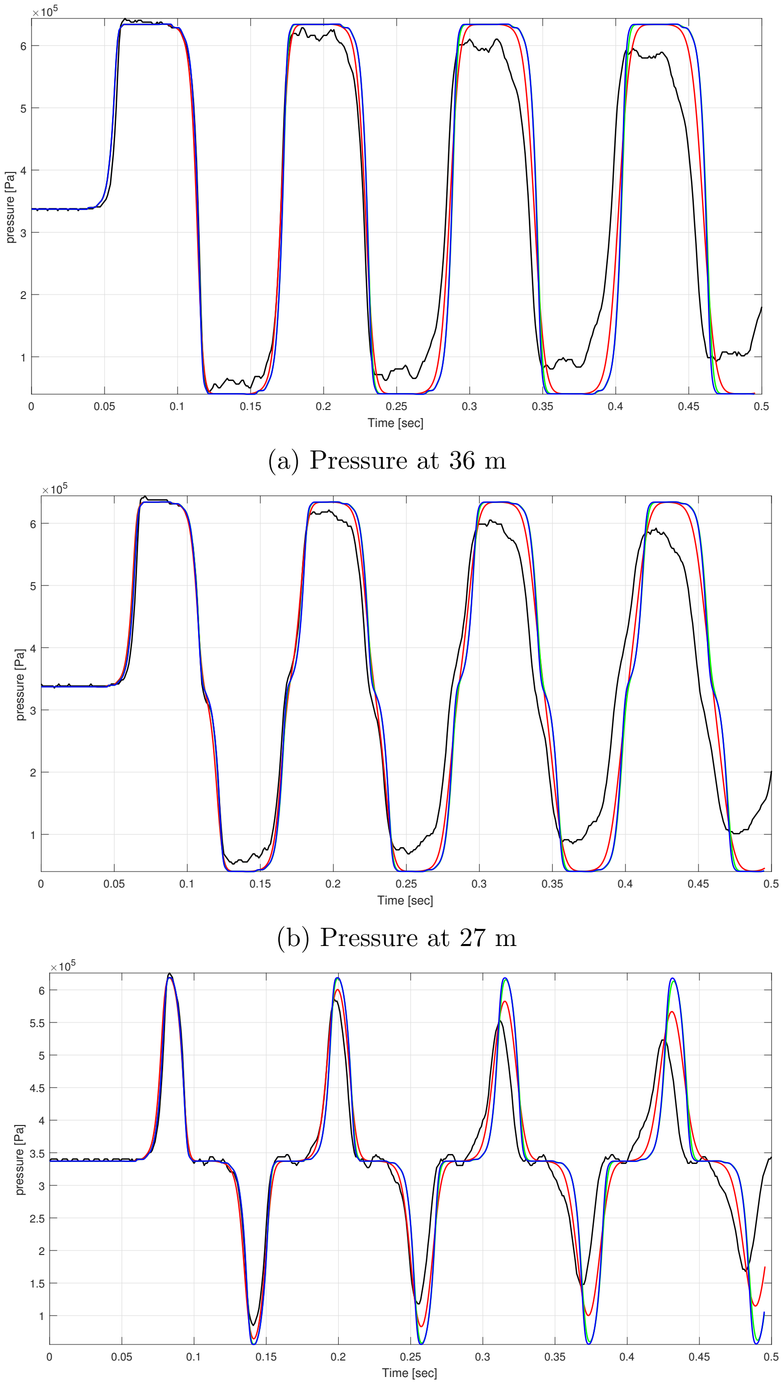
The grid size study is a crucial parameter for the model performance. The numerical method is *consistent* if by decreasing the mesh grid size, the truncation error given by the method itself decreases. This is a key factor that in combination with stability, is necessary but not sufficient to have a converged algorithm [25, p. 142]. The experimental results are given for the cases 1, 3 and 9 with the parameters in table 6.5:

Table 6.5: Simulation parameter for grid refine comparison

Simulation Variable	
C_R	*0.9*
Algorithm	*DGCM*
Initial gas in solution α_{Gas0}	10^{-7}

The variation of the grid points are for all the experiment 32, 256 and 1024 grid cells. In case 1, cavitation did not occur; therefore the experimental pressure profile is smeared out only by the dissipation term. In this condition, the algorithm does not simulate the real dissipation. The simulation results in an overestimate of the pressure profile compared with the experiment, figure 6.9. The reason can be found in the reseach conducted by Ghidaoui et al.[28]. During the water hammer formation, the friction term becomes extremely important when the non-dimensional friction term described in equation 3.6, has $\Gamma \geq 1$. This condition is often achieved when the simulation exceeds the second reflection cycle. In this condition, the algorithm shows its limitation, no matter the grid size. Indeed smaller grid sizes produce better result thanks to its higher dissipation effect.

When cavitation occurs, the algorithm performs better for smaller mesh grids as shown in figures 6.10 and 6.11. For the case of 32 grid cells, the pressure reflection is faster. In addition, the amount of cavity formation is consistently less than the case of 256 or 1024 cells.

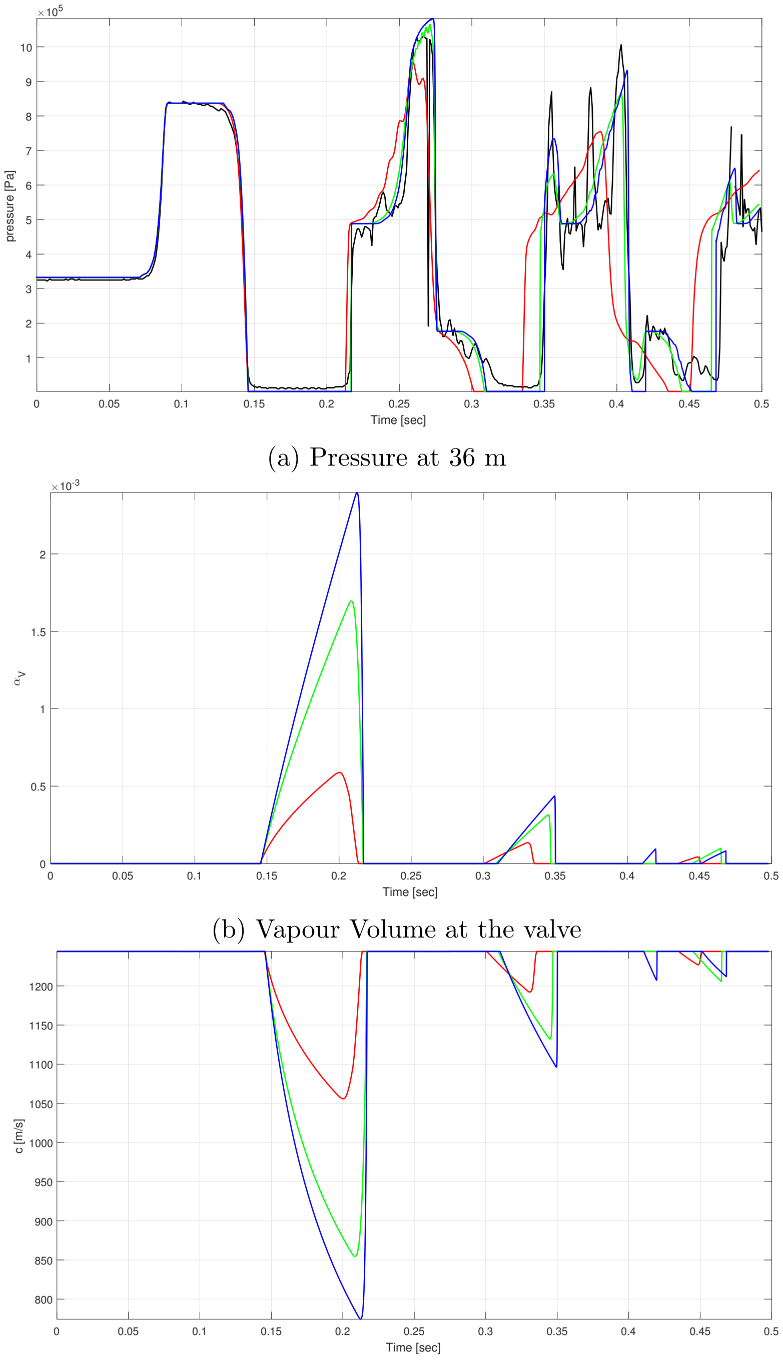


(a) Pressure at 36 m

(b) Pressure at 27 m

(c) Pressure at 9 m

Figure 6.9: Evaluation of the grid size for the developed model for case 1. red line is the DM with 1.125 m (32 cells), green line is the DM with 0.14 m (256 cells), blue line is the DM with 0.035 m (1024 cells), and the black line is the experimental data.

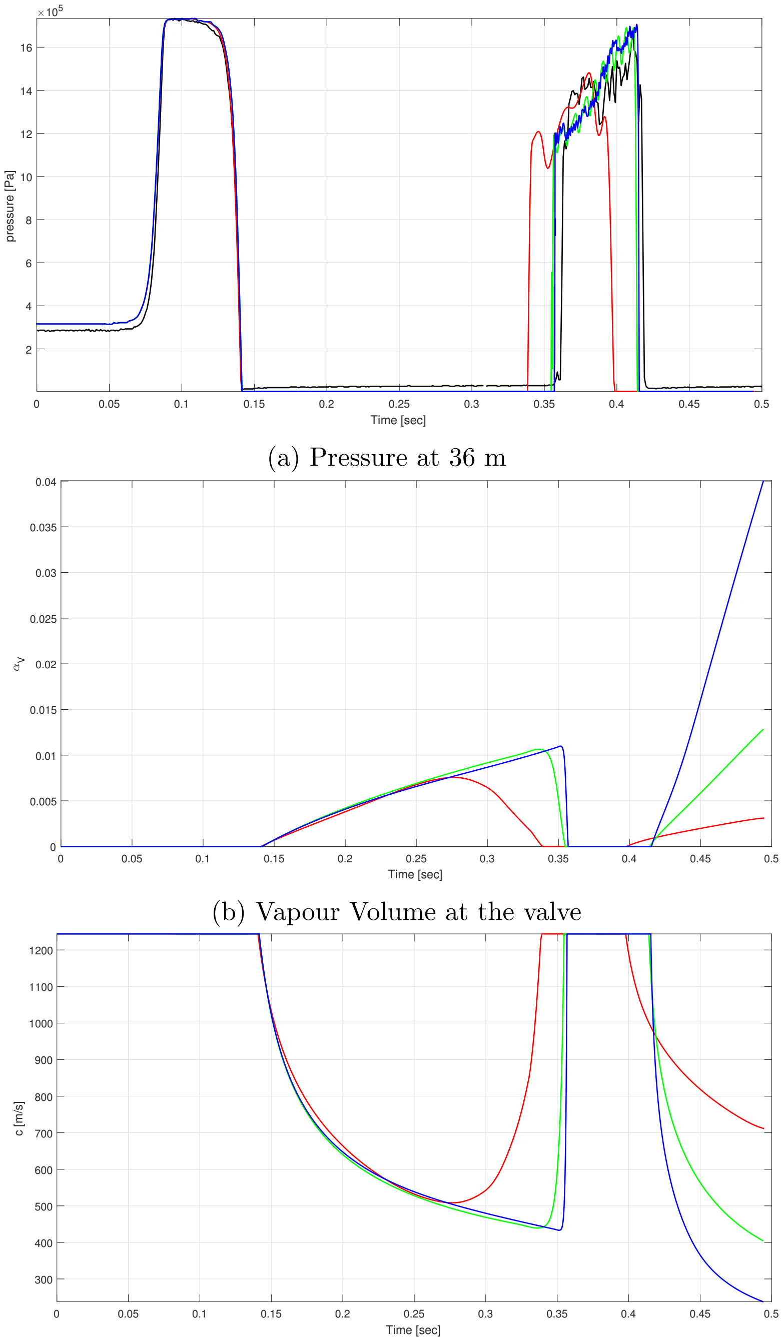


(a) Pressure at 36 m

(b) Vapour Volume at the valve

(c) Speed of sound at the valve

Figure 6.10: Evaluation of the grid size for the developed model for case 3. red line is the DM with 1.125 m (32 cells), green line is the DM with 0.14 m (256 cells), blue line is the DM with 0.035 m (1024 cells), and the black line is the experimental data.



(a) Pressure at 36 m

(b) Vapour Volume at the valve

(c) Speed of sound at the valve

Figure 6.11: Evaluation of the Grid size for the developed model for case 9. red line is the DM with 1.125 m (32 cells), green line is the DM with 0.14 m (256 cells), blue line is the DM with 0.035 m (1024 cells), and the black line is the experimental data.

Table 6.6: CPU time [sec] vs Grid Size

V_0 **[m/s]**	**Grid size**		
	1024	***256***	***32***
0.239	379.1	24.8	1.1
0.401	353.5	28.48	1.33
1.125	421.1	28.16	1.4

Refining the mesh to 1024 points, increases the high-frequency noise in the pressure profile, as a highlighted in figure 6.11a. However, the amplitude is drastically reduced compared with the case of 256 cells. Furthermore, for case 3, the reduction in the mesh has two beneficial effects. First and foremost the smoothing effect on the pressure profile and secondly by improving the response of the algorithm for third and fourth reflected waves.

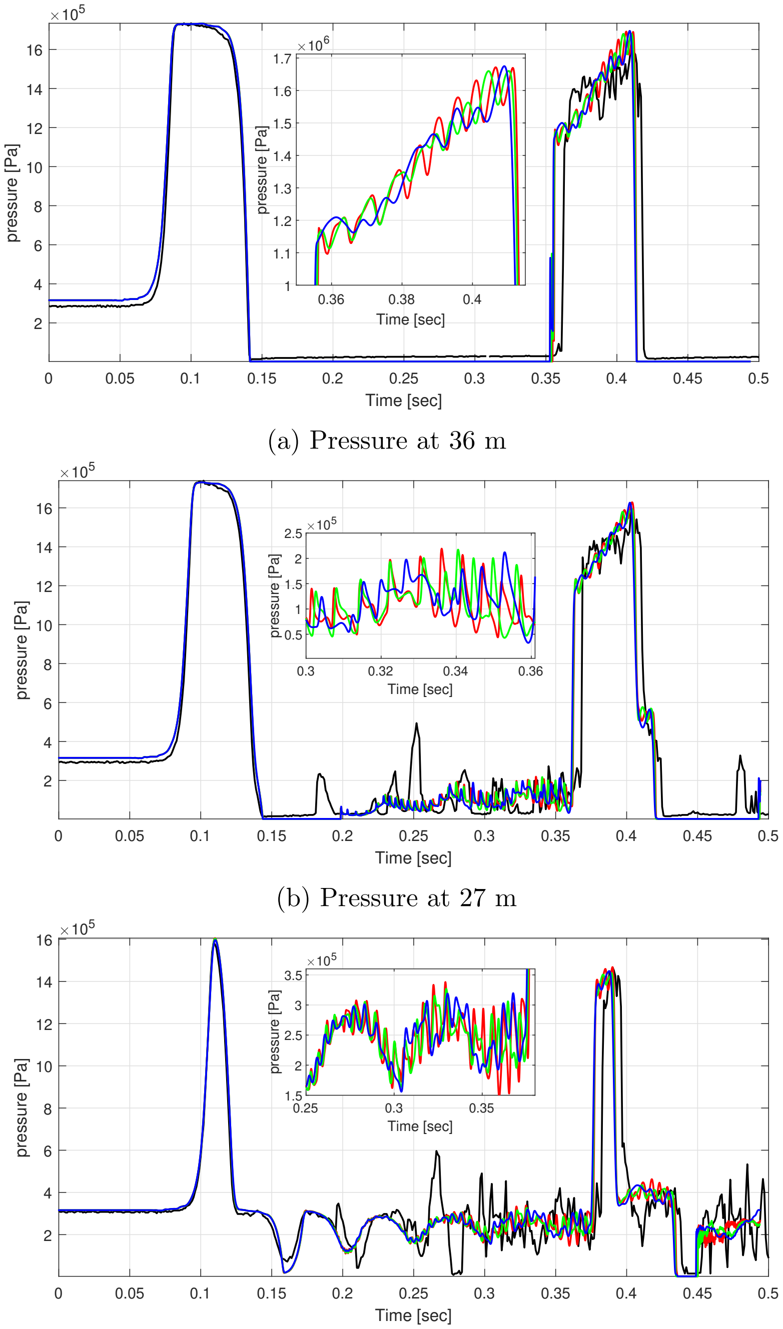
The improvement in the result given by the refined mesh is evident. However as reported in table 6.6, the CPU time required changes significantly. For higher speed, a variation in the grid size of 4 times increases the CPU time by 15 times.

6.1.5 Influence of the Courant number

Investigating the stability factor gives an understanding of the dissipation of the method as well as any behaviour that is not physically realistic. The Courant variation is shown only for case 9 with simulated parameters reported in table 6.7.

Table 6.7: Simulation parameters for Courant number simulation

Simulation Variable	
Grid size	*0.14 m*
Algorithm	*DGCM*
Initial gas in solution α_{Gas0}	10^{-7}



(a) Pressure at 36 m

(b) Pressure at 27 m

(c) Pressure at 9 m

Figure 6.12: Evaluation of the Courant number for the developed model for case 9. Red line has a $C_R = 0.9$, green line has a $C_R = 0.8$, blue line has a $C_R = 0.5$, and the black line is the experimental data.

Decreasing the Courant number increases the stability; however at the same time the effect of numerical dissipation occurs[25]. The overall response of the second-order compressible model does not seem to change with the Courant number significantly. Moreover, as shown in figure 6.12, the high-frequency pressure oscillations are drastically reduced. Considering cavitation formation, a decrease of the Courant number does not affect the overall cavitation result, figure 6.13.

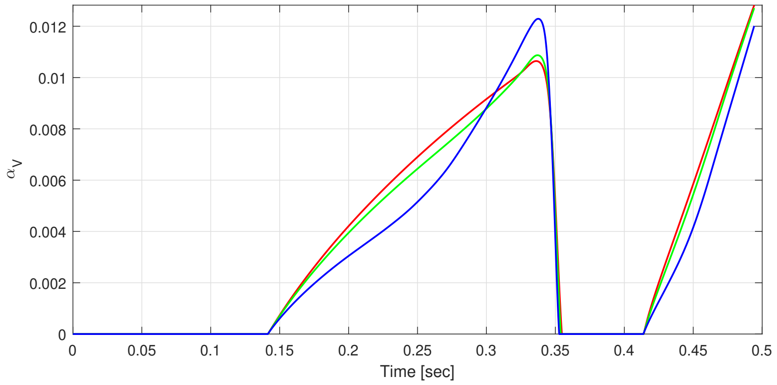


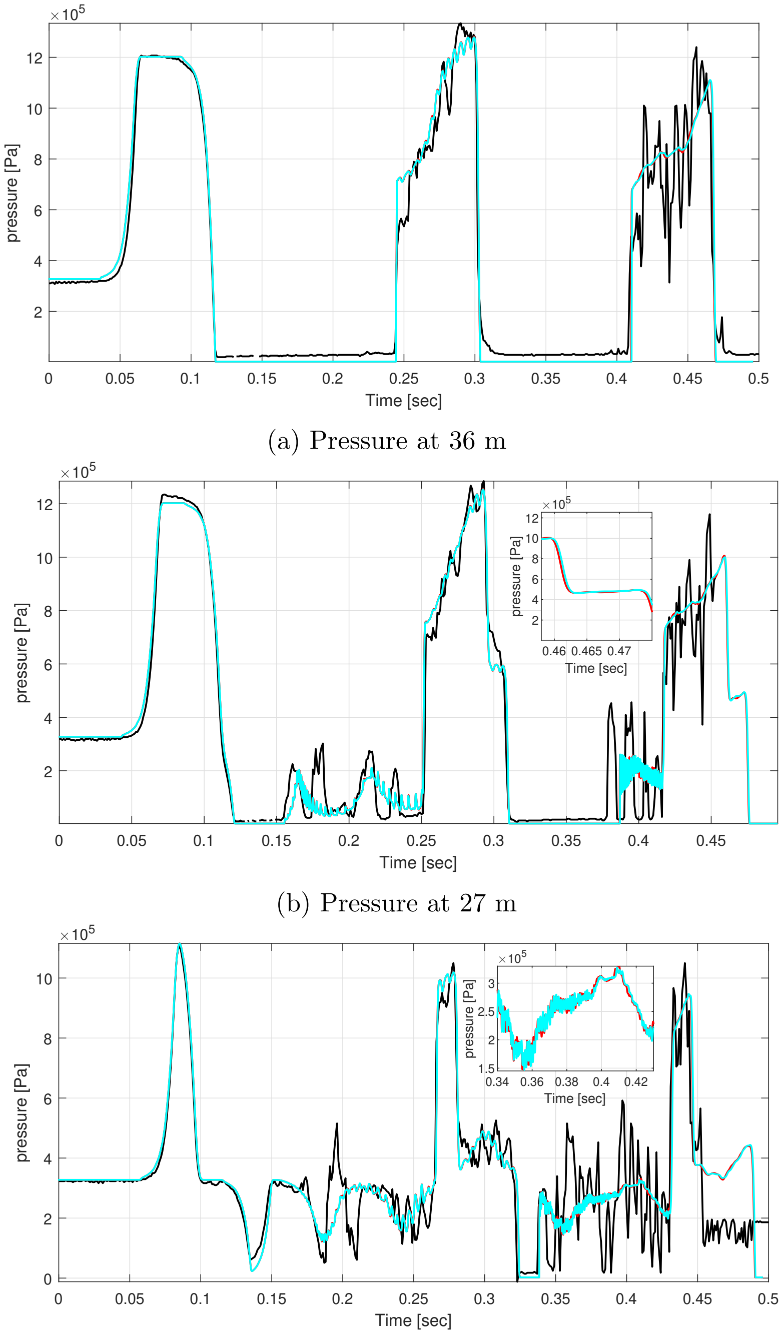
Figure 6.13: Evaluation of the cavity profile for the developed model for case 9. The values of the Courant factor are: red line has a $C_R = 0.9$, green line has a $C_R = 0.8$, blue line has a $C_R = 0.5$, and black line is the experimental data.

6.1.6 Influence of the advective term

As already pointed out in the literature[28, 66], the advective term is commonly neglected for water hammer description.

Table 6.8: Parameters of advective model formulation

Simulation	***Variable***
C_R	*0.9*
Grid size	*0.14 m*
Algorithm	*DGCM*
Initial gas in solution	10^{-7} *ppm*

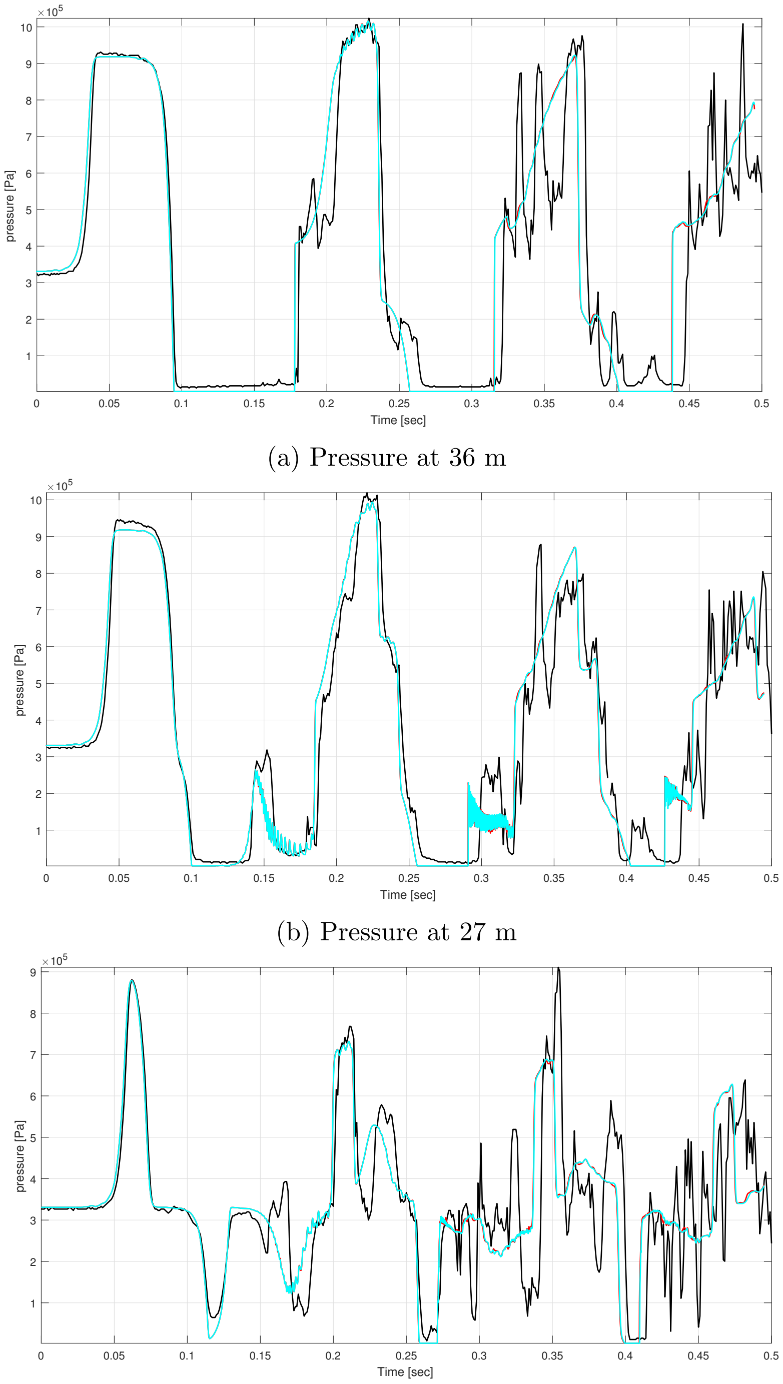


(a) Pressure at 36 m

(b) Pressure at 27 m

(c) Pressure at 9 m

Figure 6.14: Evaluation of the advective term for the developed model for case 7. Red line is the complete equation, blue line is without and the black line is the experimental data.



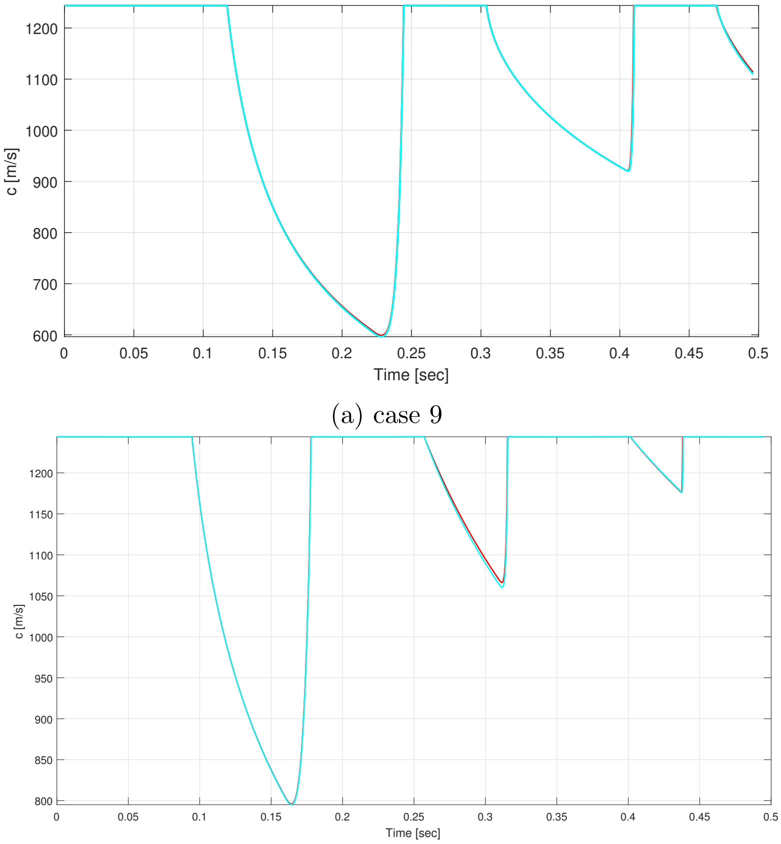
(a) Pressure at 36 m

(b) Pressure at 27 m

(c) Pressure at 9 m

Figure 6.15: Evaluation of the advective term for the developed model for case 8. Red line is the complete equation, blue line is without and the black line is the experimental data.

However, it becomes more important as soon as the speed of sound is of the same order of magnitude as the fluid velocity. The ratio between the speed of sound and fluid velocity, namely the Mach number, can change in the transient fluid phenomenon. The increase of the Mach number can occur not only due to the increase of the fluid velocity but also in relation to the reduction of the speed of sound.

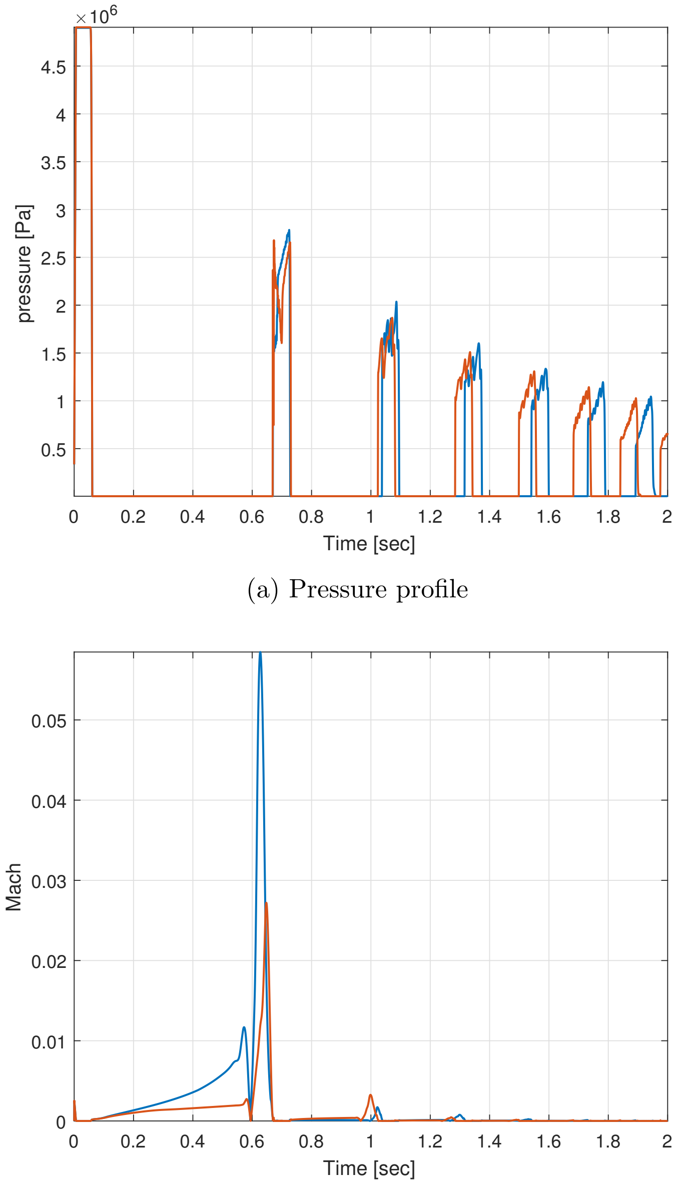


(a) case 9

(b) Case 4

Figure 6.16: Comparison of the speed of sound for the developed model at the valve. Red line is the complete equation, blue line is without the advective term and the black line is the experimental data.

The Mach number for fluid flow in a duct is typically less than 0.01[66]. However, during the cavity formation, the speed of sound can decrease, and the Mach number can reach a value of 0.1. The study of the effect of this term is given for cases 9 and 4 with simulated parameters given in table 6.8. In figure 6.14 and 6.15 are shown respectively the pressure profile for the higher and lower initial fluid speed. Meanwhile, in figure 6.16 the speed of sound at the valve is shown.



(a) Pressure profile

(b) Mach number

Figure 6.17: Example of pressure profile and Mach number at the valve considering the advective term blue line, and neglecting red line.

Although the speed of sound is drastically reduced, the effect of the advective term is still negligible. Even in case 9 where the speed of the fluid is $1.125m/s$ the Mach number only reaches a maximum value of 0.002. The advecting term does not change the accuracy of the algorithm since the Mach number is too small. Considering the result, it can be argued that the value of this term inside the algorithm is negligible. However, the main idea behind this term is related to the pump model. During valve cavitation in the pump, the fluid speed in the valve could reach a value of tens of meters per second. In the case of cavitation, the local Mach number can reach a value of 0.1. Therefore the advective term is important in the description of the dynamics of the fluid by adding more dissipation. A simulation of a fluid with an initial velocity of $5m/s$ was performed to consider the increased effect of the advective term. Figure 6.17 shows the importance in the delay and dissipation. The variation in the fluid response is related to the eigenvalues of the system. Neglecting the advective term the fluid response is faster, figure 6.17b. In conclusion, for the studied velocity range, the advective term is negligible. It is only required for higher Mach numbers to see its effect.

6.2 Comparison with other algorithms

In this section the comparison of the model developed by the author and the available algorithms in the literature is presented. Although, in the literature there are different approaches to solve the water hammer equations, this discussion is focused only on the following methods.

- Method of characteristics
- FV solution with constant density and speed of sound for DVCM model developed by Zhou et al.[31]
- FV solution with a compressible model developed by Daude et al.[48]

- FV solution for 2D and 1D models with a constant speed of sound and density[37]

The method of characteristics consists of solving the classical water hammer with finite difference methods reported in section 5.5. The finite volume method designed by Zhou et al.[31] is a variation of the Wylie[27] DVCM formulation where a finite volume method with second order MUSCL scheme is performed. The other two methods are completely different in approach and equations. Daude et al.[48] uses the system of equations shown in 6.2.

$$\begin{cases} \frac{\partial(\rho A)}{\partial t} + \frac{\partial(\rho \boldsymbol{u} A)}{\partial x} = 0 \\ \frac{\partial(\rho \boldsymbol{u} A)}{\partial t} + \frac{\partial(\rho \boldsymbol{u}^2 A + pA)}{\partial x} - p\frac{\partial A}{\partial x} = \pi d\tau_w - \rho A g sin\theta \\ \frac{\partial(\rho e A)}{\partial t} + \frac{\partial(\rho e \boldsymbol{u} A + puA)}{\partial x} + p\frac{\partial A}{\partial t} = -\boldsymbol{u}\rho A g sin\theta \end{cases} \tag{6.2}$$

The calculation of the vapour value is carried out by an iterative method and calculated indirectly from the density value. The solution scheme is based on the first order Godunov scheme[47]. The two-dimensional treatment of Pezzinga and Cannizzaro[37] consists of a cylindrical coordinate system. The pressure and the cavity volume fraction are calculated by inverting an auxiliary function. The overall equations are given in equation 6.3.

$$\begin{cases} \frac{\partial \phi}{\partial t} + \frac{c^2}{g}\frac{\partial \boldsymbol{u}}{\partial x} = 0 \\ \frac{\partial \boldsymbol{u}}{\partial t} + g\frac{\partial H}{\partial x} + \frac{1}{\rho r}\frac{\partial(r\tau)}{\partial r} = 0 \\ \phi = \frac{p}{\rho_L g} + \frac{c^2}{g} ln(1 - \alpha_V) \\ p = max(\rho_L g\phi, p_v) \\ \alpha_v = max\Big[0, 1 - exp\Big(\frac{\rho_L g\phi - p_v}{\rho_L c^2}\Big)\Big] \end{cases} \tag{6.3}$$

The solution scheme used was a MacCormack method with forward finite differences for the spatial derivative. Therefore, a comparison is performed with all the methods: MOC, Zhou, Daude and Pezzinga, to highlight the limitation and the opportunities of the one-dimensional compressible model.

6.2.1 Developed model comparison with Zhou model and MOC approaches

In this section, a comparison with the Zhou and the MOC methods is shown. The data evaluation considers case 3 of Simpson's data. In addition, the simulation took place for two different grid sizes, 32 and 256 cells. In figure 6.18 is shown the comparison with Simpson's data, the developed model, the MOC and the Zhou algorithm for 32 grid points. As expected, the classical method of characteristics captures only the main pressure behaviour. The MOC result overestimates the pressure peak and incurs a time delay in relation to the experiment. The algorithm developed by Zhou, captures the pressure behaviour with more accuracy. In contrast with the rest of the models, the Zhou model computes high-frequency oscillations that are not present in the developed model and MOC algorithms.

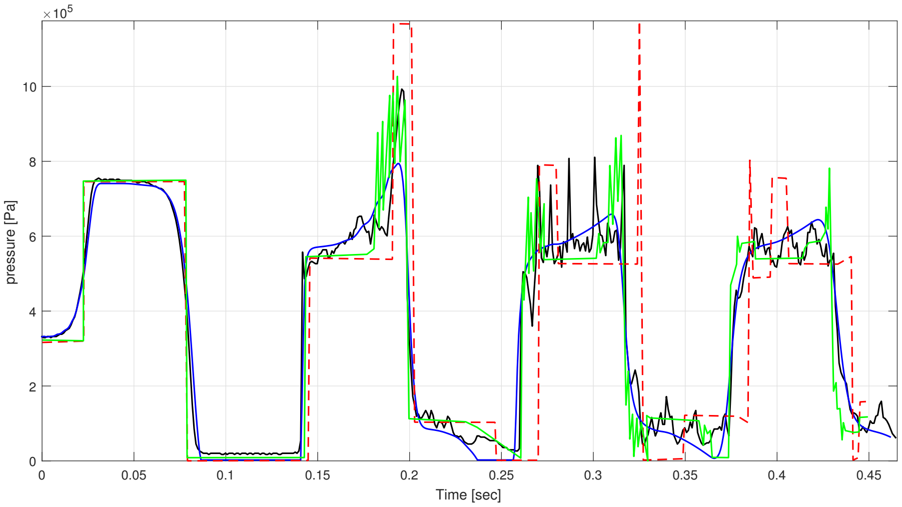


Figure 6.18: Comparison of the pressure history for case 3 and 32 grid points. The blue line is the developed model, green line is the Zhou's model, dash red line is the MOC and the black line is the experimental data.

The same behaviour is shown in figure 6.19 where the number of grid points is increased to 256. The developed model does not produce the non-physical spikes that are shown for the MOC. In addition, Zhou's algorithm increases the high-frequency oscillation, whereas the compressible model improves the quality of the result.

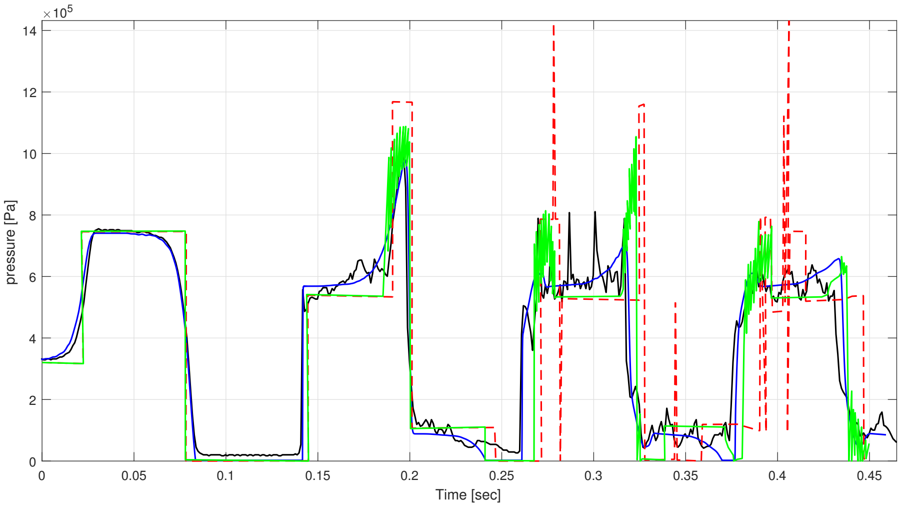


Figure 6.19: Comparison of the pressure history for case 3 and 256 grid points. The blue line is the developed model, green line is the Zhou's model, dash red line is the MOC and the black line is the experimental data.

The same behaviour was reported for the comparison data in figure 6.10a, where a smoother pressure response and a better agreement was reported for a refined mesh grid.

6.2.2 Developed model comparison with Pezzinga approach

The idea behind this comparison is to evaluate the response of one-dimensional analysis with a multi-dimension model. Although there is research where a three-dimensional analysis is performed,[36], the simulation time required was extremely high. Therefore the comparison was conducted with a two dimensional treatment given by Pezzinga and Cannizaro[37]. Two simula-

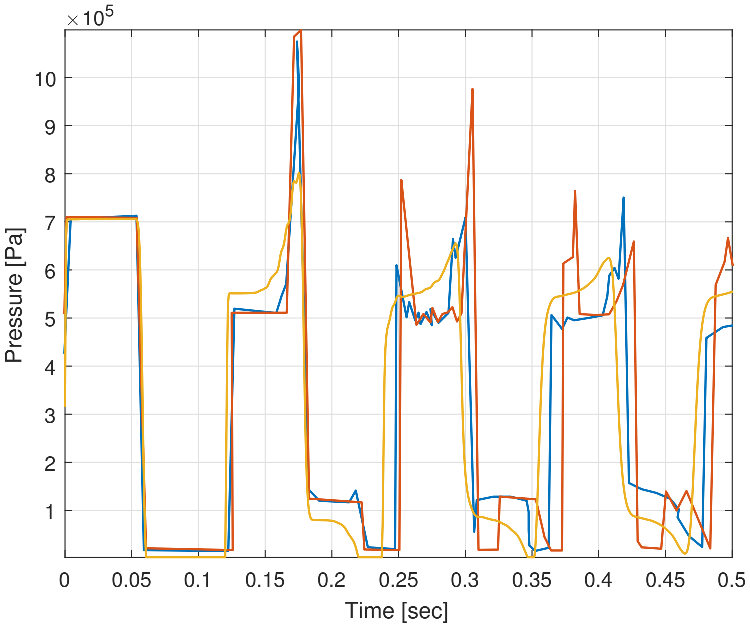


Figure 6.20: Comparison of pressure profile for case 10, for DM compressible model orange line, one dimension Pezzinga's model red line, and two dimension Pezzinga's model blue line.

tions were performed for this purpose: case 10 an 11 in table 6.1. In figure 6.20 the differences are shown between the developed model with 32 grid nodes and a one and two-dimensional model developed by Pezzinga and Cannizaro. The developed model adds more damping than the Pezzinga and Cannizaro one-dimensional simulation but less than the Pezzinga and Cannizaro two-dimensional model.

Figure 6.21 shows the performance of the developed compressible model and the Pezzinga's models in the presence of cavitation. Both methods anticipate the pressure reflection wave. Although Pezzinga and Cannizaro explained the phenomenon suggesting the advective terms as a crucial factor, this should not be the case. The compressible equation considers the advective terms, and the role that these terms play is minimal as long as the Mach number is small. No precise explanation could be found; therefore further investigation should be carried out in this context.

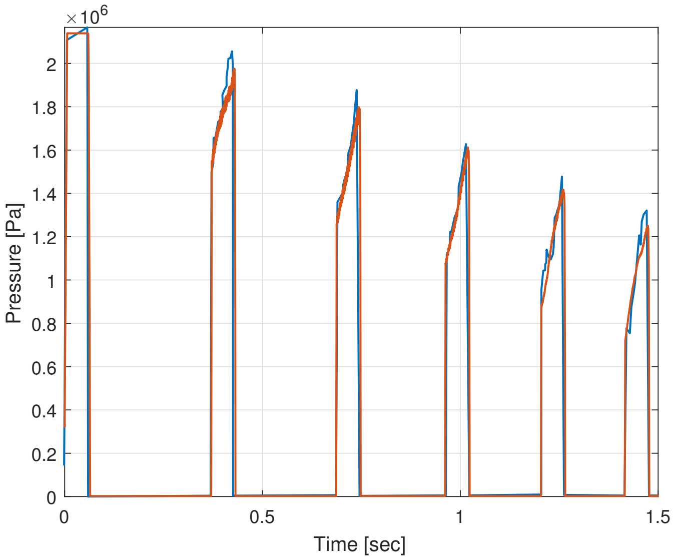


Figure 6.21: Comparison of pressure profile for case 11 for the developed model red line, and two dimension Pezzinga's model, blue line.

Comparison of the volume fractions in figure 6.22 show a better agreement with Pezzinga and Cannizzaro than with Daude et al.[48], as is shown in the next section.

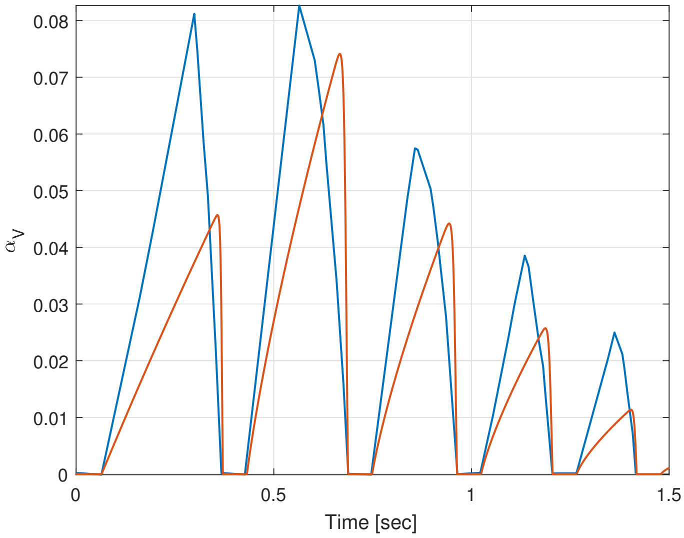


Figure 6.22: Comparison of void fraction profile for case 11, for developed model red line, and two dimension Pezzinga's model, blue line.

6.2.3 Developed model comparison with Daude approach

In the research carried out by Daude et al.[48] a compressible algorithm was compared with the same Simpson data. Therefore, in this section, the comparison for case 3 of table 6.1 is shown. The characteristic parameters for the comparison are reported in table 6.9.

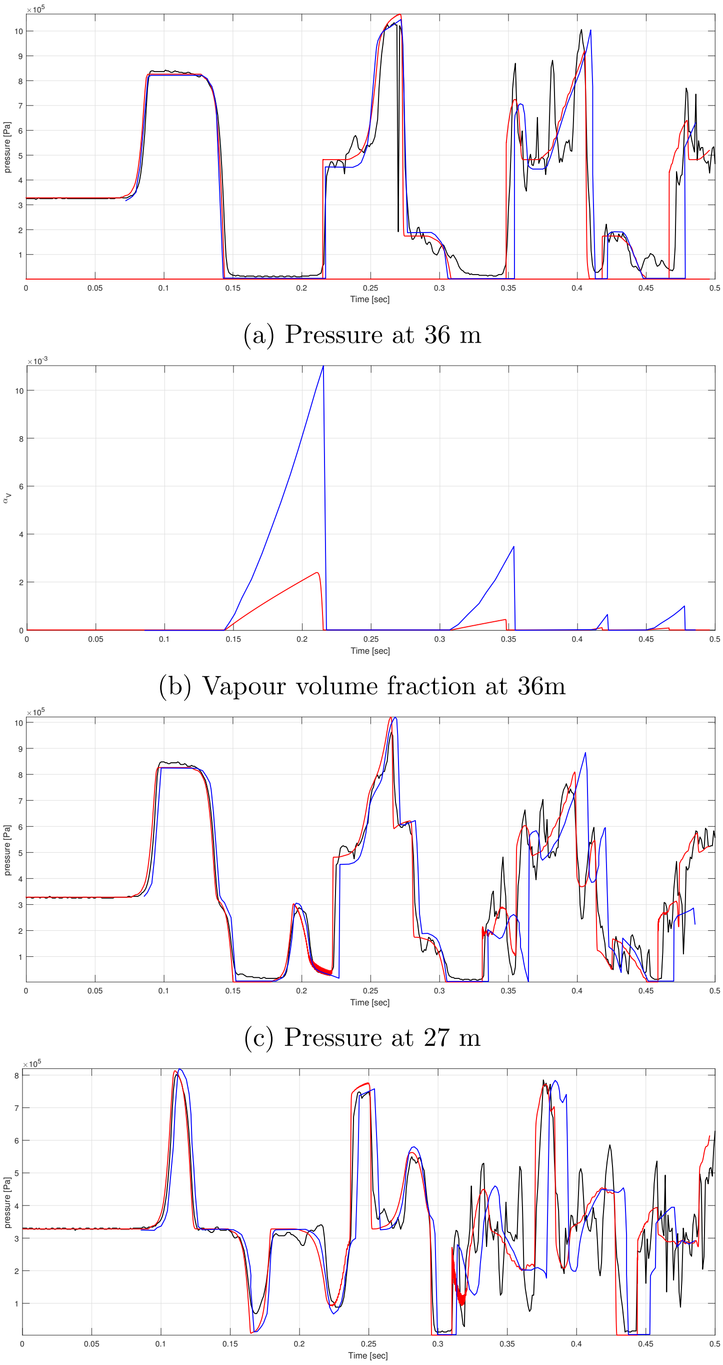
Table 6.9: Simulation parameter for compressible vs incompressible DM

Simulation Variable	
C_R	*0.8*
Grid points	*1000*
Algorithm	*DGCM*
Initial gas in solution α_{Gas0}	10^{-7}

From figure 6.23 it is possible to observe the differences between the two models and the Simpson data. Although the two algorithms performed equally well, the main difference is in the volume fraction value. The two algorithms show the same time for the cavity to grow and collapse, but the amount of vapour produced is higher in Daude's model. It could be possible that the Daude's algorithms overestimate the cavity amount, due to the delay in the pressure reflections.

The compressible model with the energy equation included by Daude et al.[47] is an extremely powerful tool. The algorithm predicts the transient flow with high agreement however two points were highlighted:

- The use of the energy equation for the one-dimensional analysis of the transient fluid is not required. The temperature of the phenomenon is simulated as almost constant as reported in the Daude's research[48].
- In the same research the difficulty of multi-junction simulation came to light and required a more careful treatment than a continuity equation formulation[48].



(a) Pressure at 36 m

(b) Vapour volume fraction at 36m

(c) Pressure at 27 m

(d) Pressure at 9 m

Figure 6.23: Comparison of the pressure profile and speed of sound at the valve for the developed model red line and Daude's model blue line.

6.3 Summary

The developed compressible model produces high fidelity results when compared to the experiment. In both analysed cases, Simpson and Pezzinga, the algorithm performed better than the MOC and comparable with the algorithms from the literature. In addition, the relatively simple implementation of the the code allow its extension to more complex systems such as those involving positive displacement pumps. The algorithms; thanks to the description in characteristic form, can be combined to simulate multi-junction systems in a straightforward manner. In conclusion, the developed model shows its potential benefit also for simulating cavitation effects. In the next section the application of the developed model to a positive displacement pump is given.

Chapter 7

Pump Simulation

The primary object of this research was to develop an experimentally validated numerical model for predicting the operation of a positive displacement pump. The main idea was to predict cavitation formation and identify the parameters that could mitigate its formation. To verify the successful development of the model two different pump layouts were simulated and compared with experimental data. The first test was performed in a closed loop rig with a one-chamber diaphragm pump. Data from a three-chamber pump were collected directly from an operational pump, where the pump was part of a wider network. In addition, comparison with the LPM and Opitz algorithms was performed to highlight the limitations and the advantages of the finite volume algorithm developed by the author in comparison with other algorithms.

7.1 One chamber pump description

The experiment for the one-chamber pump was conducted in the Weir Mineral's test facility (Venlo, the Netherlands). The overall pump component can be subdivided into four categories:

- Pump

- Suction manifold and inlet dampeners
- Discharge manifold and outlet dampeners
- Choke station to create the pressure load.

Figure 7.1 shows the pump detail of the rig test.

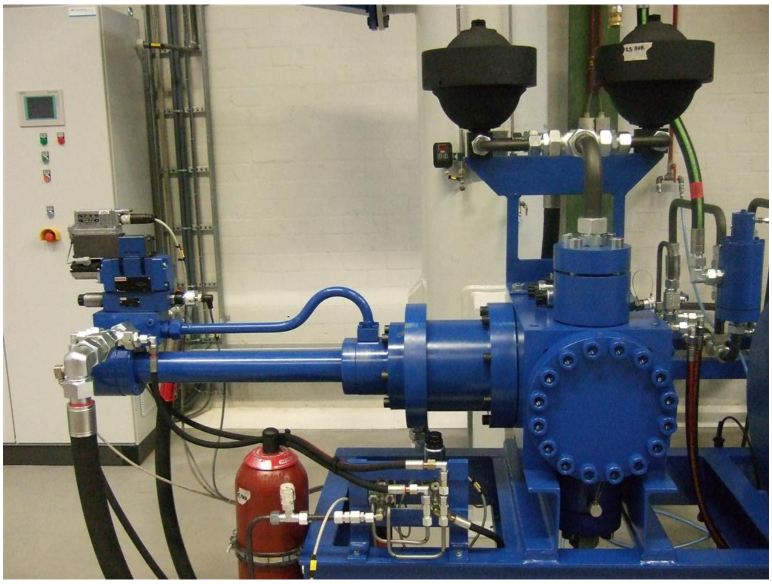


Figure 7.1: Weir Minerals's test rig in Venlo, the Netherlands.

A schematic description of the system is shown in figure 7.2. However, the simulation were not performed for the entire system due to the computational effort required. In fact, the simulation considered a structured grid where the grid size is governed by the smallest part size. Therefore a truncated domain is simulated and the representation is given in figure 7.3.

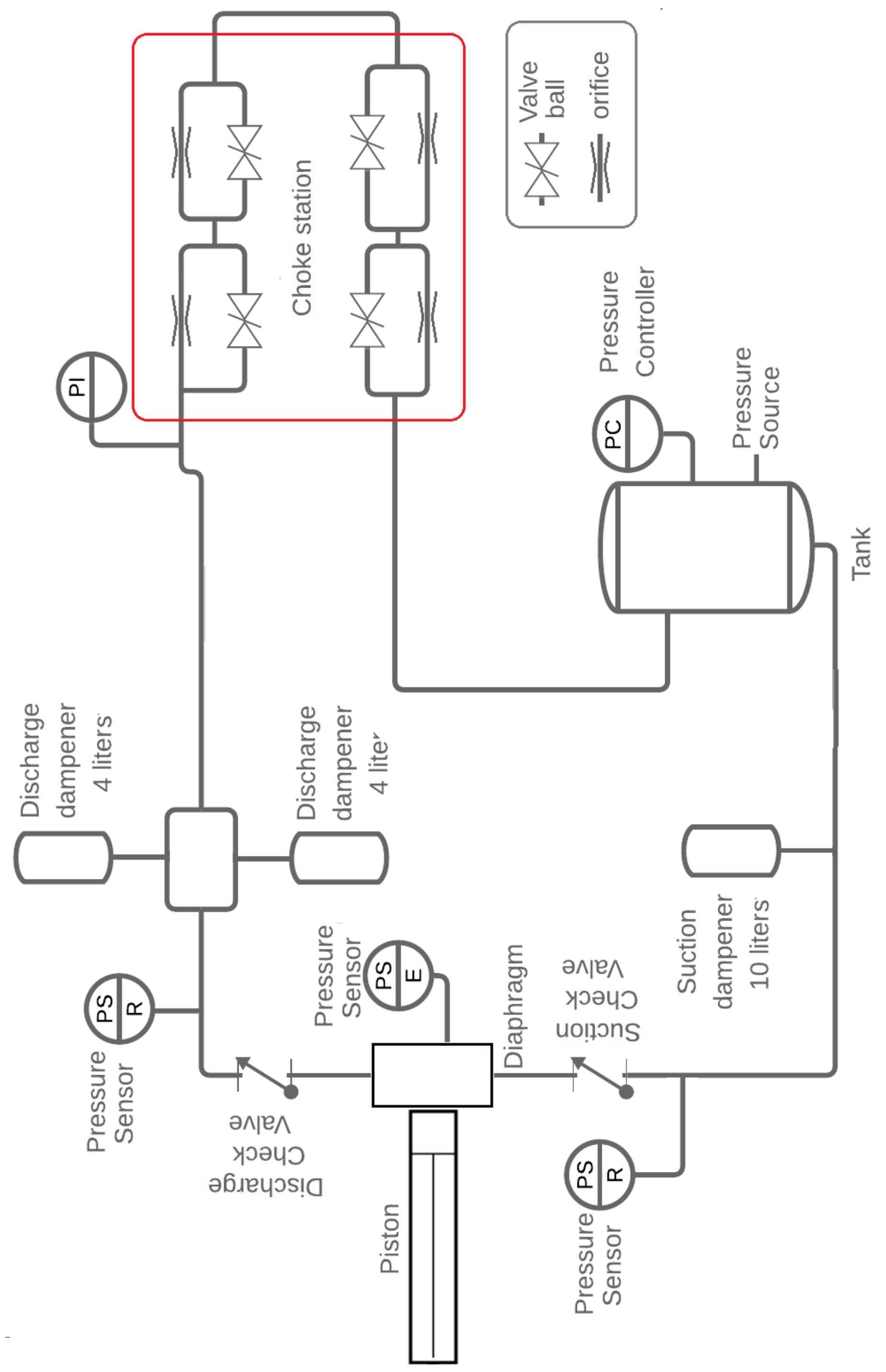


Figure 7.2: One chamber test rig layout.

The simulated rig does not consider the choke station in the discharge line (figure 7.4) which provide the load pump pressure and the tank in the suction line.

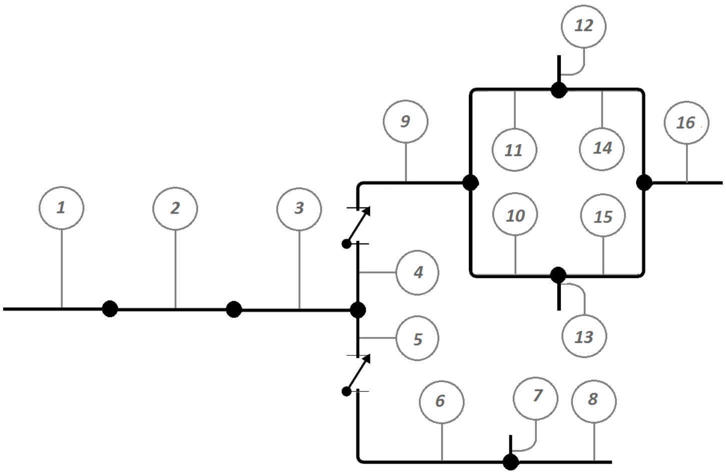


Figure 7.3: Hydraulic simulated scheme of Weir Minerals's test rig (see table 7.1 for components name).

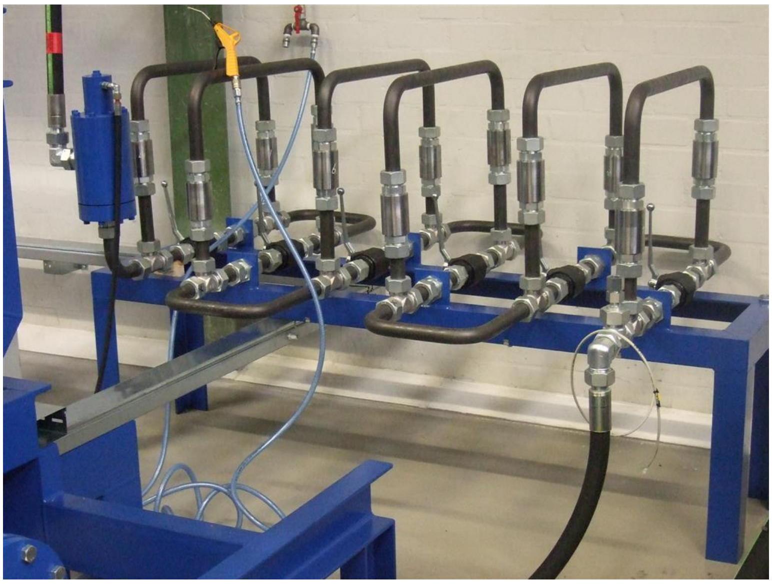


Figure 7.4: Choke station for producing the pressure chamber load.

The dimensions of the simulated component are reported in table 7.1.

Figure 7.5 provides the CAD drawing of the simulated component.

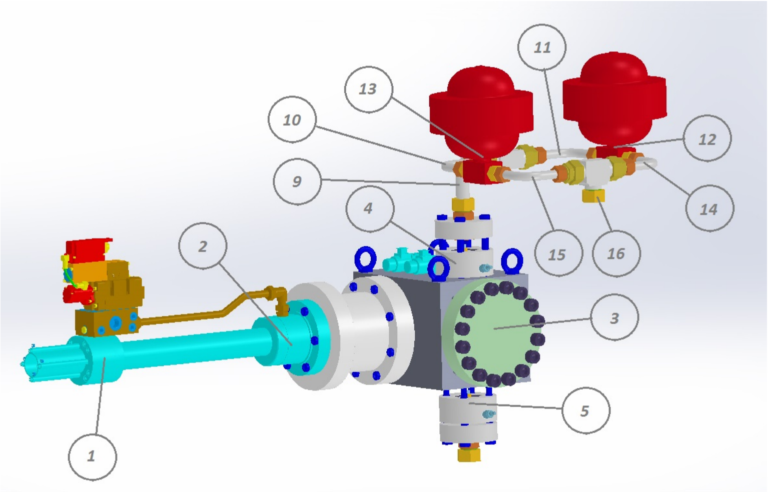


Figure 7.5: Cad drawing of the simulated component.

The pump section consists of a piston component, hydraulically moved, a pre-chamber section, the diaphragm housing, and the discharge and suction valve sections. The piston motion was achieved by an external hydraulic system governed by a high-frequency spool valve. The schematic of the piston control system is shown in figure 7.6.

Weir Minerals chose this control method for the piston motion for its versatility. Conveniently, it is possible to vary the piston stroke and velocity without any mechanical variation. However, this approach was not reliable in terms of piston velocity prediction, as shown later in this section.

Table 7.1: Dimension of the simulated component of the pump

#	**Name**	**Length** [m]	ϕ [m]
1	Piston section	0.063	0.063
2	PreChamber section	0.3	0.1
3	Chamber section	0.03	0.2
4	Discharge section	0.08	0.033
5	Suction section	0.1	0.026
6	Suction pipe	0.4	0.03
7	Suction accumulator	0.1	0.03
8	Suction inlet pipe	1.35	0.03
9	Discharge pipe	0.442	0.03
10	Discharge pipe left	0.5	0.019
11	Discharge pipe right	0.5	0.019
12	Discharge accumulator left	0.1	0.019
13	Discharge accumulator right	0.1	0.019
14	Discharge pipe left - 2	0.49	0.019
15	Discharge pipe right - 2	0.49	0.019
16	Discharge Outlet pipe	4	0.03

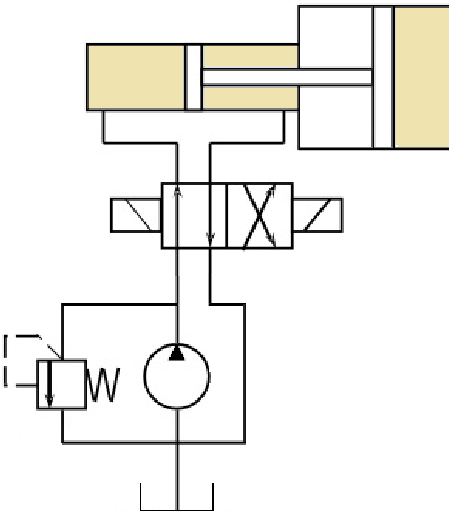


Figure 7.6: Piston control motion.

The suction section consists of:

- a tank of 80 litres as a reservoir where the suction and the discharge pipes are connected;
- one suction dampener of 10 litres, (SB330-10A1 \112U-330A manufactured by Hydac);
- flexible pipe;
- a pressure controller to regulate the vessel pressure, connected to an air compressor.

The layout for the discharge section is more complex. It consists of:

- two discharge dampeners, each of 4 litres (series 400 manufactured by Hydac);
- choke station, with a series of orifices and ball valves;
- rigid steel pipes;
- high-pressure flexible pipes.

In order to record the pressure history during the operating cycles, different pressure sensors were used:

- two Sensortec A-105 piezo resistant sensors mounted before the suction and after the discharge valve;
- one Kistler 6005 piezoelectric sensor with a 5011B amplifier mounted in the diaphragm chamber;
- pressure indicator connected before the choke station to check the pressure load.

Table 7.2: One-chamber pump simulation

		Suction pressure			
		1 bar	**2 bar**	**3 bar**	**4 bar**
SPM	**100**	1	2	3	4
	150	5	6	7	8
	175	9	10	11	12
	200	13	14	15	16

The data acquisition was performed by a Supervision Control and Data Acquisition (SCADA) tool. This system was directly connected to a PC where the software, Catman, was used to collect the following parameters:

- time;
- suction pressure;
- discharge pressure;
- chamber pressure;
- piston position;
- membrane position.

The acquisition frequency, in order to collect the cavitation dynamics, was set to 9600 Hz. The maximum pressure operating for the test rig is 250 bar with a maximum speed of 200 SPM, with the piston length of 0.063 m. For further details and a description of the test rig, the research of Rijswick is mentioned[1].

In order to have a wide range of pump behaviour, sixteen experiments were performed, as shown in table 7.2.

Although the hydraulic system for driving the piston permits a wide range of possibilities (stroke velocity and piston length), figure 7.6, this type of system had a significant disadvantage in the fluid response. When high pressure variation and speed are required, the response of the control fluid in the spool valve becomes relevant. In figures 7.7, 7.8, 7.9, 7.10 the position of the piston as a function of theoretical crackshaft angle is given. In the same figures a comparison with a mechanical driven piston are made. The SCADA system does not provide the velocity of the piston but only its displacement. Therefore the piston velocity is computed numerically from a time derivative. Equation 7.1 is applied to calculate the velocity in relation to the virtual crankshaft angle.

$$\frac{\partial x}{\partial t} = \frac{\partial x}{\partial \theta}\frac{\partial \theta}{\partial t} = \frac{\partial x}{\partial \theta}\omega \approx \frac{\Delta x}{\Delta \theta}\omega \tag{7.1}$$

Once the velocity is calculated, a filter function is applied to smooth the numerical noise. For low pump velocity, the profile of the piston displacement and its velocity can be described by the equations 2.1 and 2.2. In this condition, the fluid inside the hydraulic system behaves like a rigid body. For high velocity rate, figures 7.9 and 7.10, the profiles are shifted in phase and exhibit a multi-peaks behaviour as shown. The inertia of the fluid and the rheology response of the mineral oil inside the system are affected by the pump system. A comparison between the velocity and the piston displacement with the theoretical profile shows significant differences.

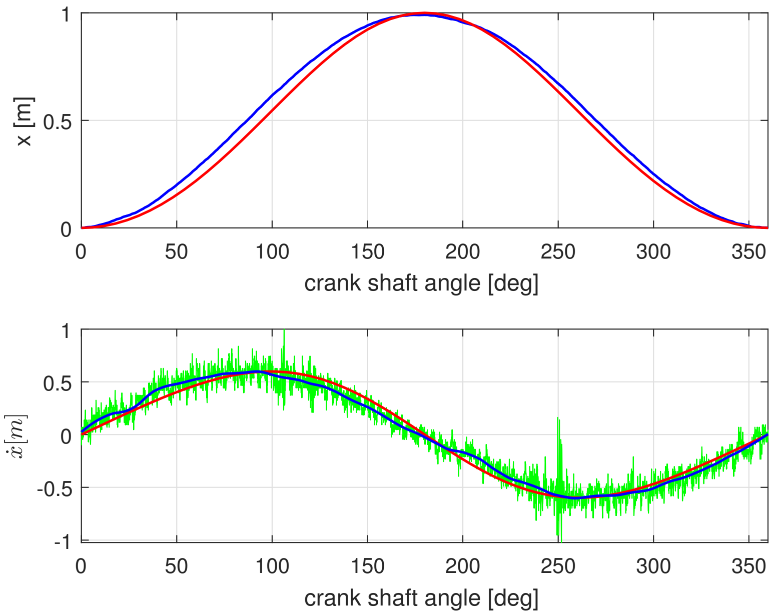


Figure 7.7: Displacement and Velocity of the piston for case 1. Red line are the response of a mechanical driven simulation, the blue line are the response from experiment data, and green line is the raw data.

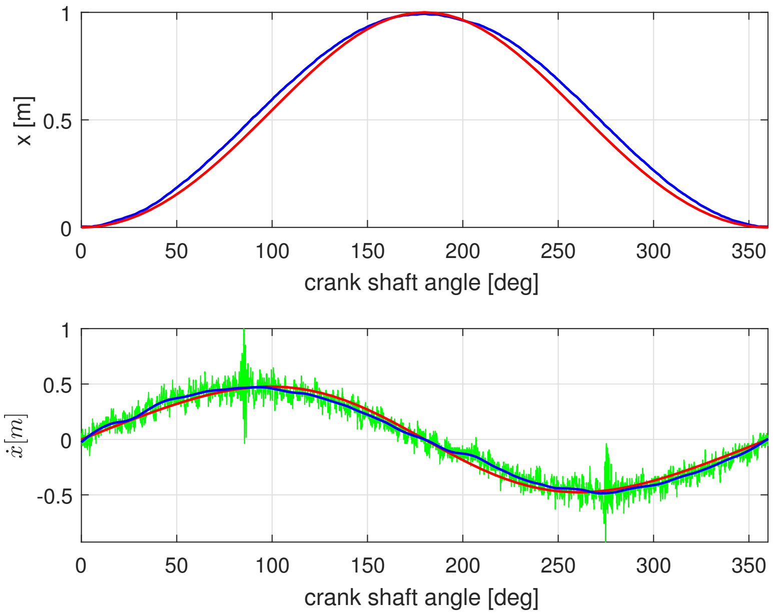


Figure 7.8: Displacement and Velocity of the piston for case 4. Red line are the response of a mechanical driven simulation, the blue line are the response from experiment data, and green line is the raw data.

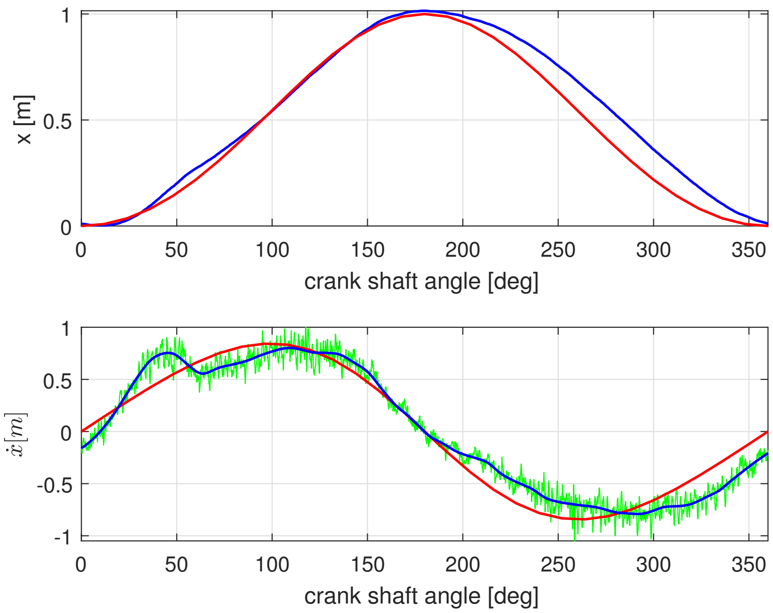


Figure 7.9: Displacement and Velocity of the piston for case 13. Red line are the response of a mechanical driven simulation, the blue line are the response from experiment data, and green line is the raw data.

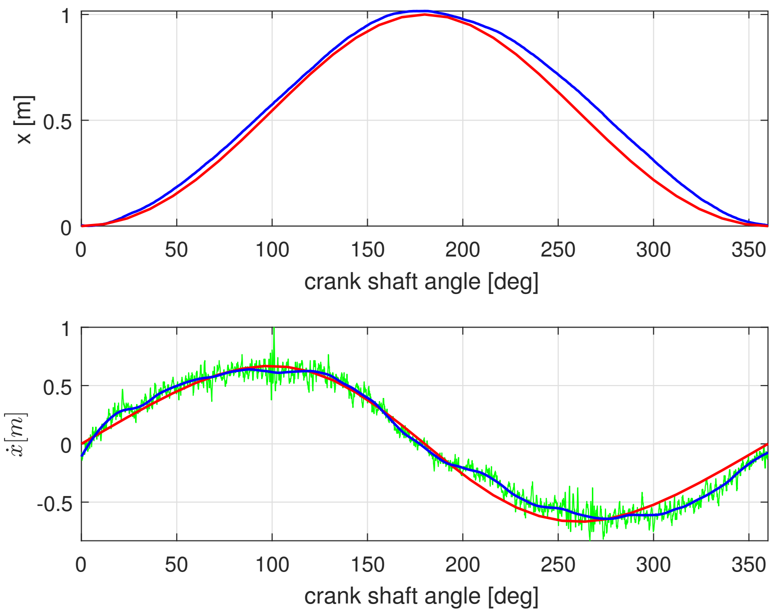


Figure 7.10: Displacement and Velocity of the piston for case 16. Red line are the response of a mechanical driven simulation, the blue line are the response from experiment data, and green line is the raw data.

7.2 Multi chamber

Data used for the three-chamber pump were collected in a measurement campaign in an iron ore mineral facility. The system description can be found in the publicly available Weir report[2]. The plant is designed to provide an average of 56 million tonnes per year (MTPA) of mineral slurry. The configuration of the pump system consists of two distinct stations of eight and ten pumps. Although the two stations are different in terms of pressure performance, the model of the pump used in both is the same, namely the TZPM2000. Station one, close to the mine, has a capacity of 301 m^3/h at a rated pressure of 183 bar with a maximum speed of 63 SPM. Station two is located between the harbour and the mine site, and its maximum capacity is 263 m^3/h at 206 bar with a maximum speed of 63 SPM. The pumps are equipped with a 2100 kW electric motor, and have a diaphragm chamber of 100 litres. The collection of the data was performed on the first pump in station one, and the same instrumentation as described for the single chamber rig was used. However, due to the inability to measure the crankshaft angle, a piston proximity sensor was used. In figure 7.11 the setup of the sensors and the layout of the pump is shown. The pump data were collected for iron ore suspended in water as a slurry fluid. The slurry had a density of approximately 2300 kg/m^3. The fluid has a high slurry concentration, however the exact value of the particle concentration is not available, due to mining company restrictions. However, for validation of the model, only measurements with water are considered. The data with the slurry were used to increase the data set for the pump described in section 7.3.7. In figure 7.12 the comparison of the normalized pressure values for slurry and water in similar conditions are shown for each pumps chamber. The increase of the density and the presence of the particles changes the overall pump response. The pressure shows a different cavitation behaviour as well as differences in the pressure pulsation frequency.

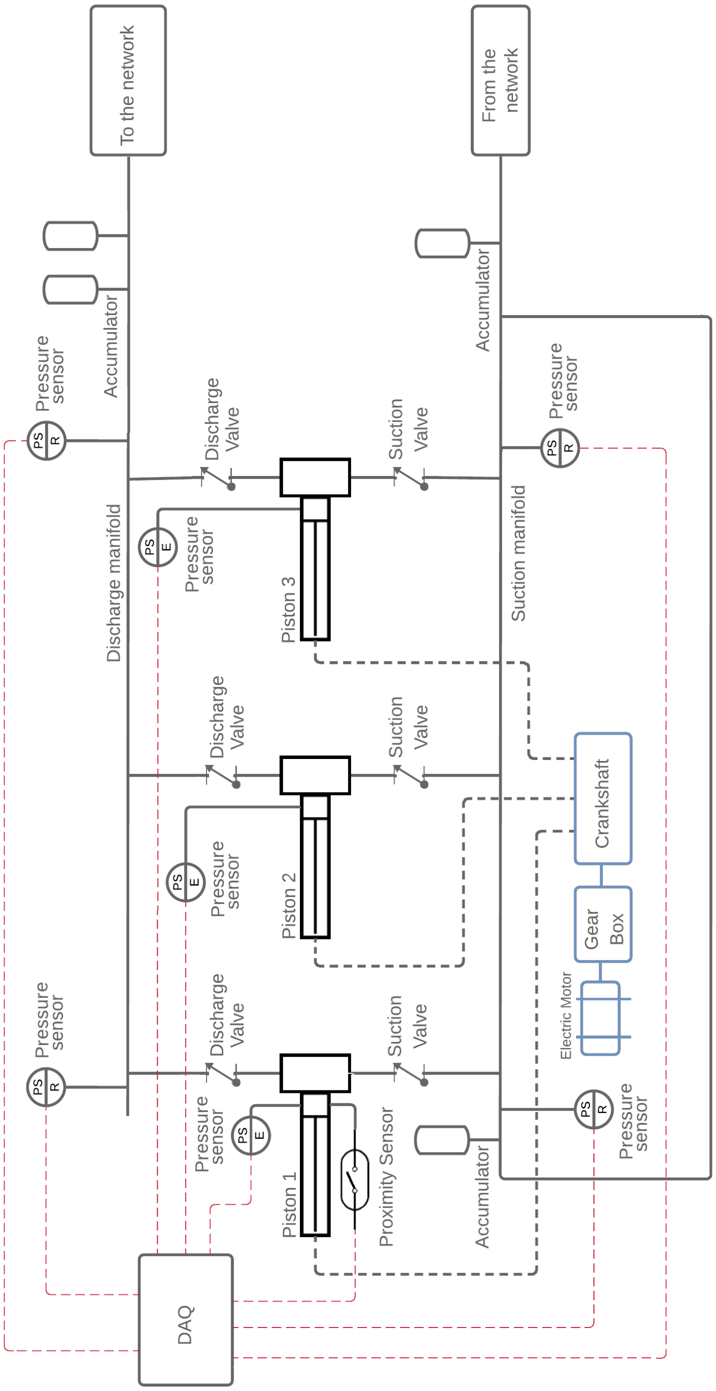


Figure 7.11: Three chamber pump layout.

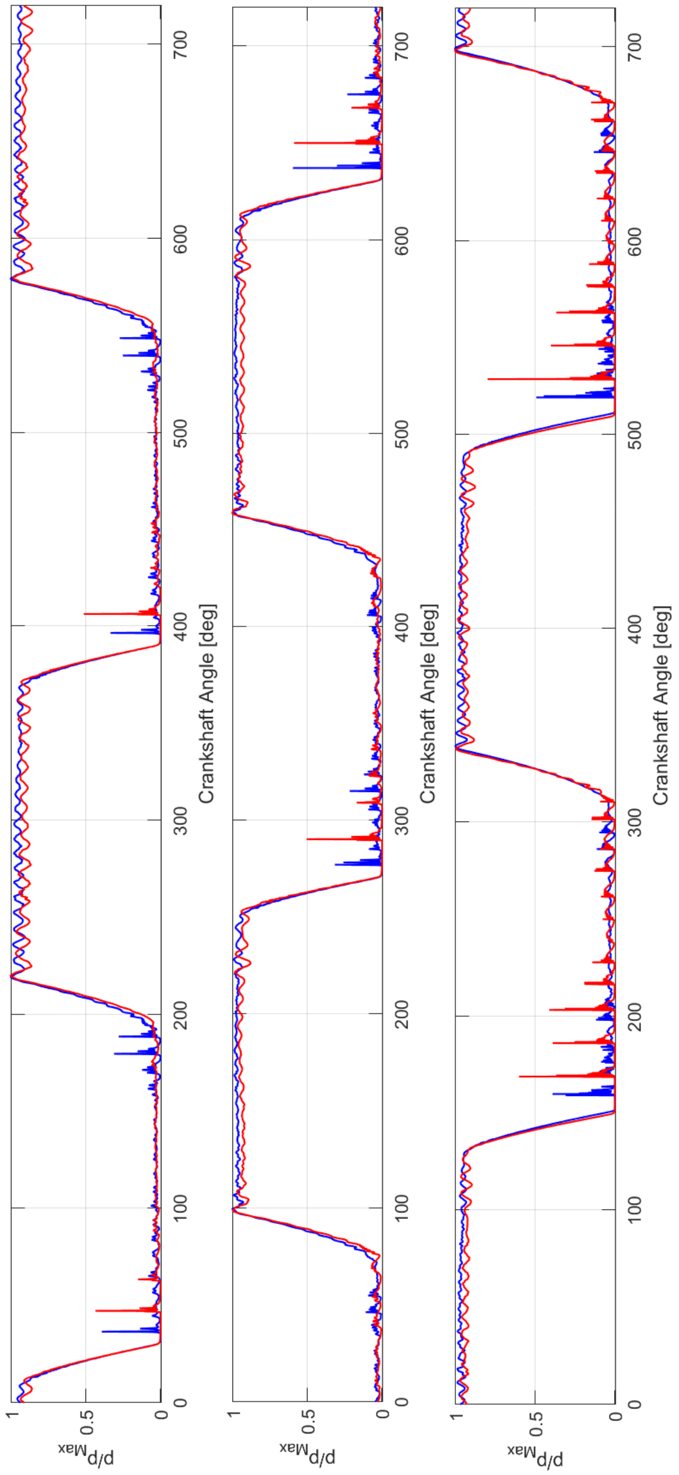


Figure 7.12: Comparison of the each pump chambers pressure for slurry red line and water blue line, for 4 bar of suction pressure and 63 SPM.

The characteristics of the TZPM200 pumps for two distinct velocities are reported in the table 7.3.

Table 7.3: Performance value of three chamber pump

SPM	**Pressure**	**Theoretical Piston Displac.**	**Theoretical Flow**	**Effective Flow**	**Volumetric Efficiency**
	[bar]	$[m^3]$	$[m^3/h]$	$[m^3/h]$	%
50	210	24.94	224	200	89.3
70	189	24.94	314	283	90.1

For the measurement campaign performed at the extraction site, a summary of speed, suction and discharge pressure is reported in table 7.4. The pressure $p_{Discharge}$ and $p_{Suction}$ are the average pressure measured during the entire data collection time. The stroke speed is defined as the percentage of the maximum speed of 63 SPM.

Table 7.4: Three-chamber pump running value for water fluid

Low pressure				**High pressure**			
#	**SPM**	$p_{Discharge}$	$p_{Suction}$	#	**SPM**	$p_{Discharge}$	$p_{Suction}$
	%	[bar]	[bar]		%	[bar]	[bar]
1	100	147.406	4.105	8	100	180.487	6.908
2	90	148.428	4.096	9	90	163.027	6.970
3	80	148.343	4.093	10	80	161.243	6.604
4	70	148.167	4.091	11	70	162.651	6.604
5	60	148.399	4.0.97	12	60	162.651	6.625
6	40	153.930	4.078	13	40	164.122	6.630
7	20	159.005	4.085	14	20	162.954	6.991

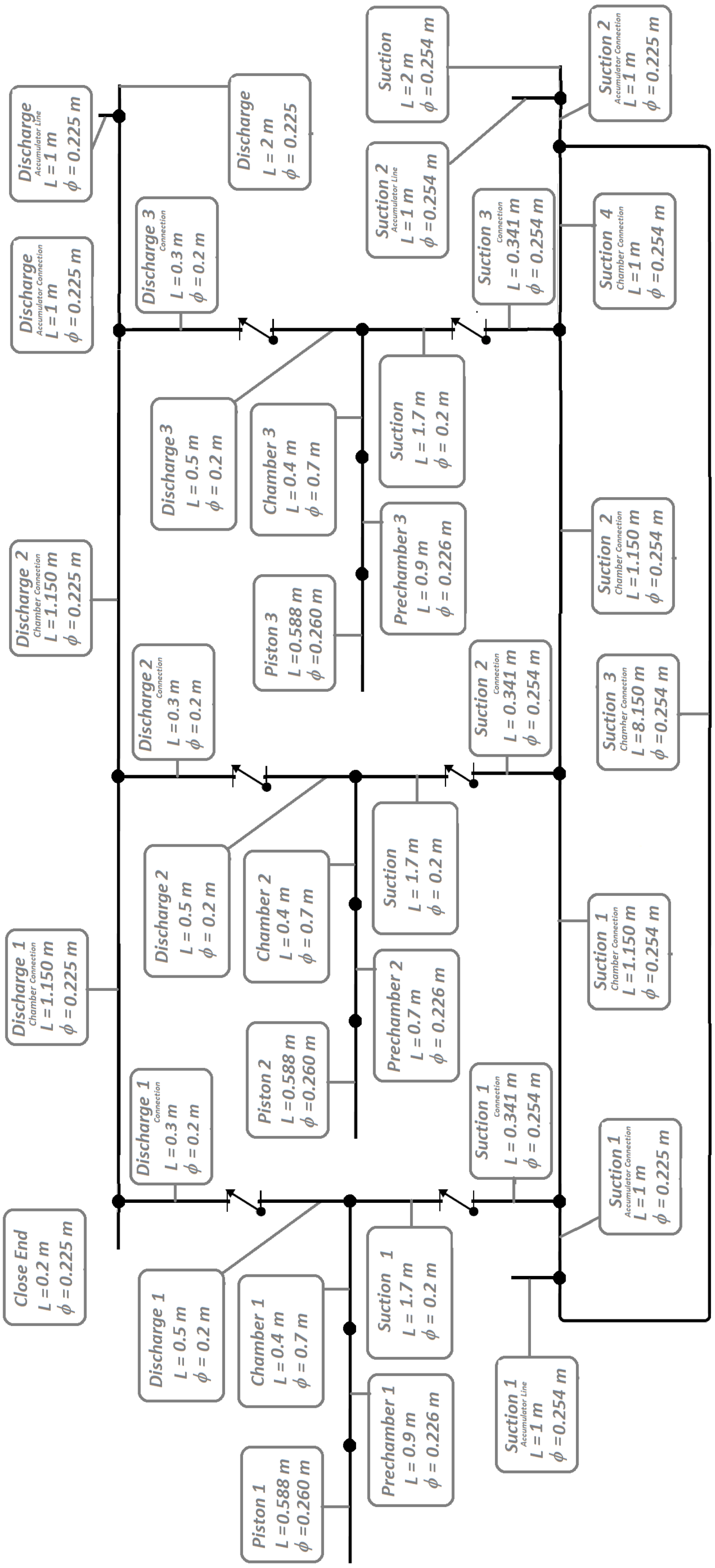

Discharge Accumulator Line L = 1 m ϕ = 0.225 m
Discharge Accumulator Connection L = 1 m ϕ = 0.225 m
Discharge L = 2 m ϕ = 0.225
Discharge 3 Connection L = 0.3 m ϕ = 0.2 m
Discharge 3 L = 0.5 m ϕ = 0.2 m
Chamber 3 L = 0.4 m ϕ = 0.7 m
Suction L = 1.7 m ϕ = 0.2 m
Prechamber 3 L = 0.9 m ϕ = 0.226 m
Piston 3 L = 0.588 m ϕ = 0.260 m
Discharge 2 Chamber Connection L = 1.150 m ϕ = 0.225 m
Discharge 2 Connection L = 0.3 m ϕ = 0.2 m
Discharge 2 L = 0.5 m ϕ = 0.2 m
Chamber 2 L = 0.4 m ϕ = 0.7 m
Suction L = 1.7 m ϕ = 0.2 m
Prechamber 2 L = 0.7 m ϕ = 0.226 m
Piston 2 L = 0.588 m ϕ = 0.260 m
Discharge 1 Chamber Connection L = 1.150 m ϕ = 0.225 m
Discharge 1 Connection L = 0.3 m ϕ = 0.2 m
Discharge 1 L = 0.5 m ϕ = 0.2 m
Chamber 1 L = 0.4 m ϕ = 0.7 m
Suction 1 L = 1.7 m ϕ = 0.2 m
Prechamber 1 L = 0.9 m ϕ = 0.226 m
Piston 1 L = 0.588 m ϕ = 0.260 m
Close End L = 0.2 m ϕ = 0.225 m
Suction L = 2 m ϕ = 0.254 m
Suction 2 Accumulator Line L = 1 m ϕ = 0.254 m
Suction 2 Accumulator Connection L = 1 m ϕ = 0.225 m
Suction 3 Connection L = 0.341 m ϕ = 0.254 m
Suction 4 Chamber Connection L = 1 m ϕ = 0.254 m
Suction 2 Chamber Connection L = 1.150 m ϕ = 0.254 m
Suction 2 Connection L = 0.341 m ϕ = 0.254 m
Suction 3 Chamber Connection L = 8.150 m ϕ = 0.254 m
Suction 1 Chamber Connection L = 1.150 m ϕ = 0.254 m
Suction 1 Connection L = 0.341 m ϕ = 0.254 m
Suction 1 Accumulator Connection L = 1 m ϕ = 0.225 m
Suction 1 Accumulator Line L = 1 m ϕ = 0.254 m

Figure 7.13: Three chamber pump dimension.

7.3 Simulation

In the literature, there are several different approaches to simulating positive displacement pumps, even when evaluating cavitation. In this research, the three-dimensional RANS simulation was not taken into consideration due to computational limitations. Diaphragm pump simulation requires the motion of the membrane to model the fluid velocity, and this approach requires a fluid-structure interaction that is beyond the scope of this research. Therefore only the simple methods like the Optiz algorithm and the lumped parameter are considered for comparison. The calculation of the cavity amount is made with an indirect process. The approach consists of calculating the pressure range where the cavity pressure is reached. Only the time of vapour creation and destruction can be extimated with this approach while the exact amount of vapour formation is not available. To obtain this value, a direct method must be applied. Methodology that was not available at the facilities where the tests were performed. Therefore only a pressure pulsation comparison are available in this research.

7.3.1 Lumped Parameter model

In a wide range of research, the description of the pump was simulated by the lumped parameter method, briefly described in section 4.6. The model and the equation are given in section 4.6. This method collapses the dimensions and the dynamics of the components to a single point. The discharge and the suction fluid are typically given from a common node to all the chambers. In this particular case, the simulations with the LPM were performed for a three-chamber pump. Analysing the pressure history for case 1 of table 7.4, it is possible to highlight that the LPM captures only the initial fluid response. This behaviour is shown in figure 7.14 where the pressure pulsation is over dampened in both suction and discharge phases.

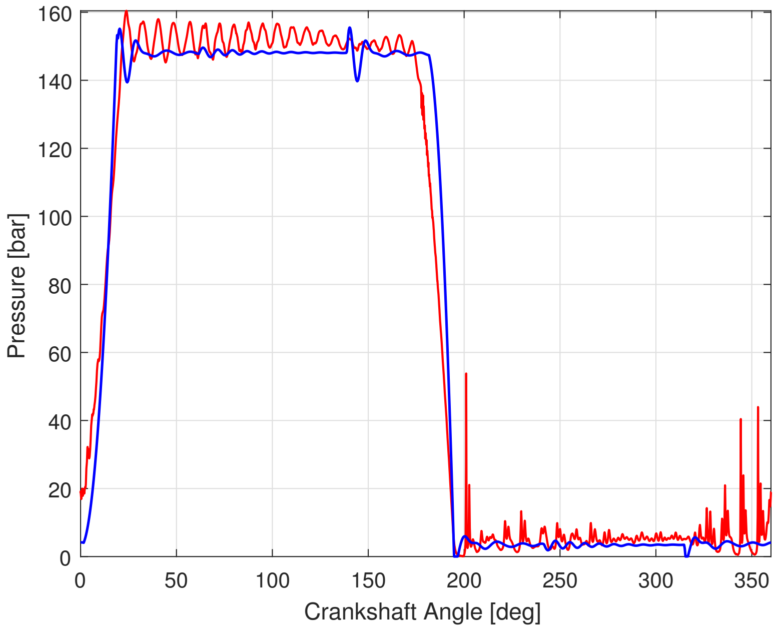


Figure 7.14: Pressure comparison of experimental data red line with LPM blue line for case 1, with discharge pressure at 147 bar, suction pressure 4.1 bar and 100% of the pump speed.

The LPM does not model the reflection wave in the system. Therefore, only the system natural frequency is modelled. Analysing the amount of cavitation, the experimental data shows a vapour dynamics time (formation and collapse) of 5 crankshaft degrees. A detailed comparison of with the LPM shown in figure 7.15, shows an under estimation of this value. No flow meter was installed during the experiments, and no real comparison of this can be reported. However, the flow rate response shown in figure 7.16 can be used to evaluate the pump performance. The main flow behaviour is depicted, and also the mutual interaction between the chambers.

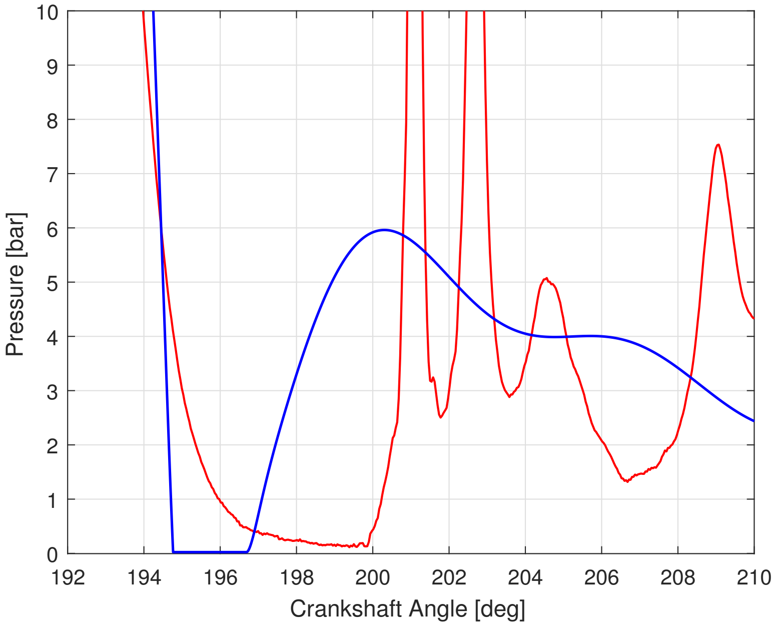


Figure 7.15: Pressure detail for the cavitation condition for experimental data red line and LPM blue line for case 1, with discharge pressure at 147 bar, suction pressure 4.1 bar and 100% of the pump speed.

Considering the volumetric efficiency, comparison between 50 and 70 SPM reported in table 7.3 shows a value of 96.8% and 97.1% respectively. These data are calculated as ratio between the volumetric flow rate at the discharge and the piston displacement volume. These data are much higher than the numbers given in table 7.3. The difference is primarily due to the simplicity of the algorithms where compressibility and volume flow reduction are not considered. Only the pressure loss given by friction is introduced. In addition, the valve dynamics consider the chamber pressure to be the same as the suction and discharge pipe pressures which is not realistic.

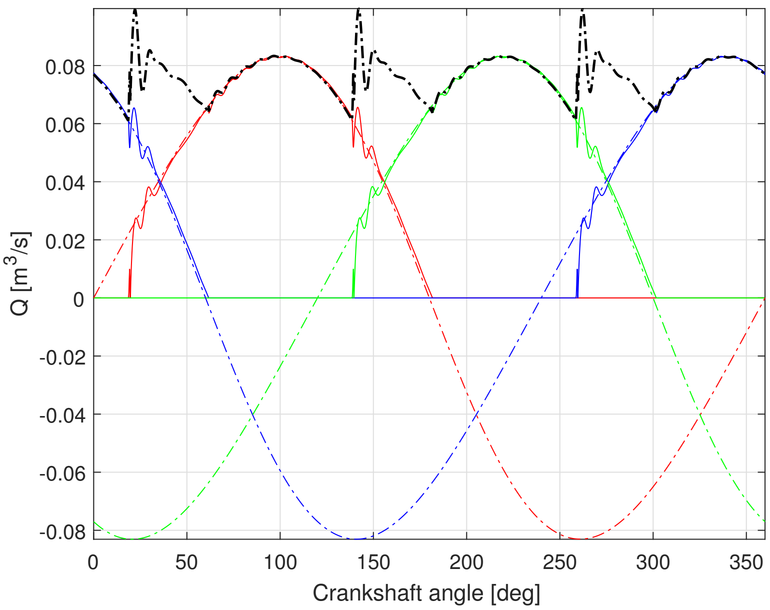


Figure 7.16: LPM Flow rate for case 1, with discharge pressure at 147 bar, suction pressure 4.1 bar and 100% of the pump speed. Red line is the discharged fluid volume by the chamber one while the dash line is the first piston displacement volume. Blue line is the discharged fluid volume by chamber two while the dash line is the second piston displacement volume. Green line is the discharged fluid volume by chamber three while the dash line is the third piston displacement volume, and the black line is the overall discharged fluid volume.

The same computational behaviour is reported for case 5, in figure 7.17. For the same case, the pressure cavity detail and the cavitation condition is reported in figure 7.18. Summarizing, the LPM model can predict the main pressure profile; however, it does not simulate the entire cycle correctly. Limitations of this algorithm are evident, although the computational effort makes this algorithm acceptable as a first approximation to the pump prediction.

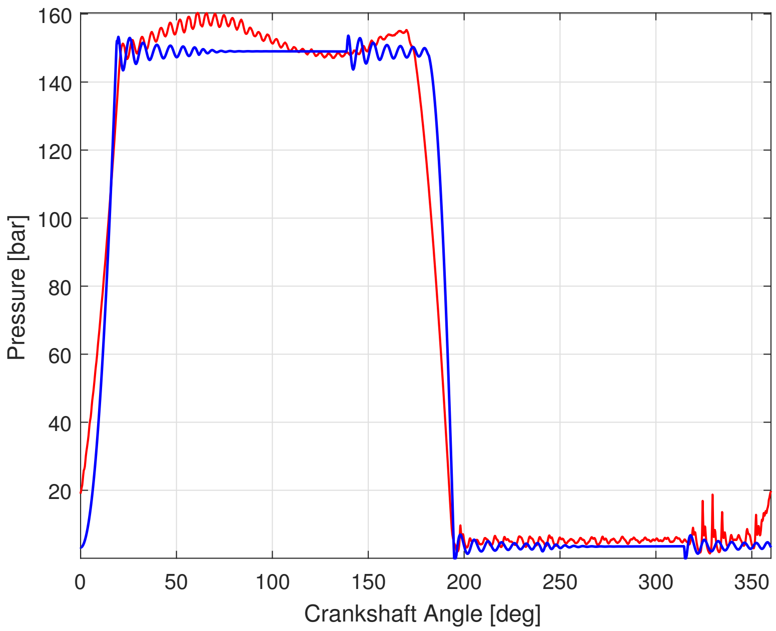


Figure 7.17: Pressure comparison of experimental data red line with LPM blue line for case 5, with discharge pressure at 148 bar, suction pressure 4.1 bar and 60% of the pump speed.

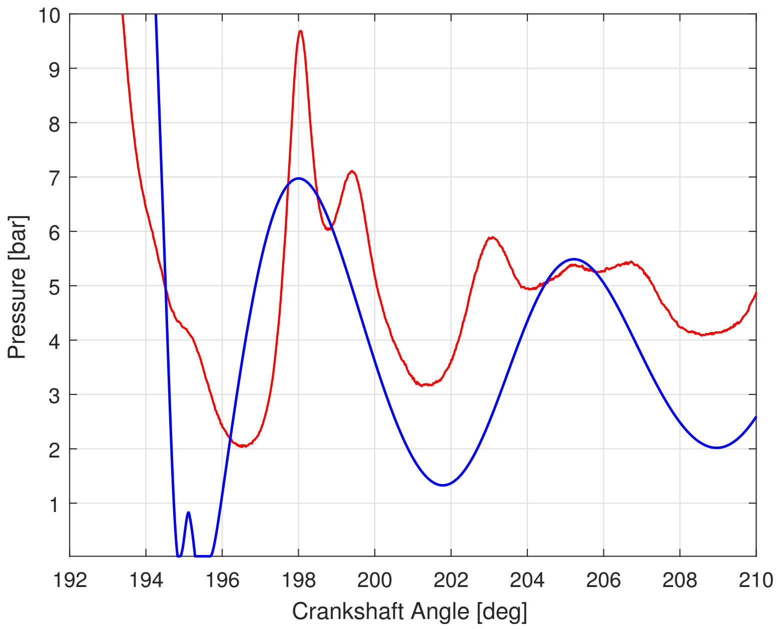


Figure 7.18: Pressure detail for the cavitation condition for experimental data red line and LPM blue line for case 5, with discharge pressure at 148 bar, suction pressure 4.1 bar and 60% of the pump speed.

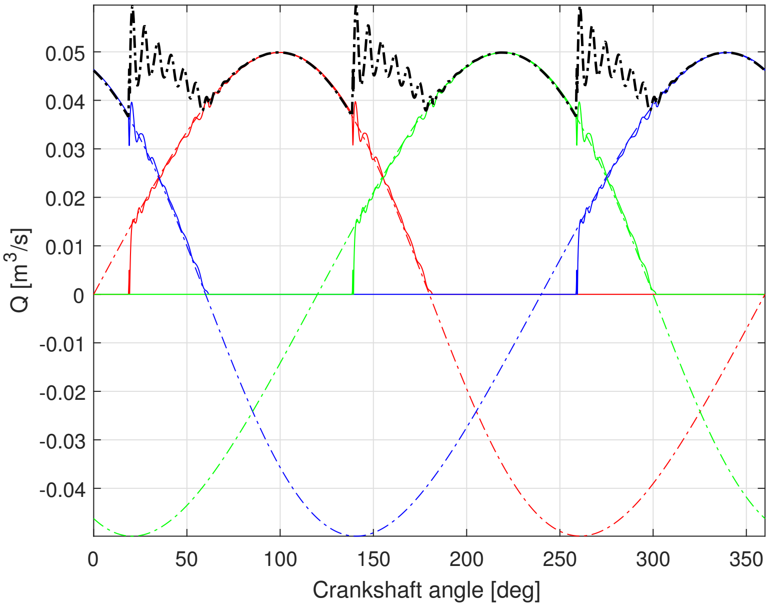


Figure 7.19: LPM Flow rate for case 5, with discharge pressure at 148 bar, suction pressure 4.1 bar and 60% of the speed. Red line is the discharged fluid volume by chamber one while the dash line is the first piston displacement volume. Blue line is the discharged fluid volume by chamber two while the dash line is the second piston displacement volume. Green line is the discharged fluid volume by chamber three while the dash line is the third piston displacement volume, and the black line is the overall discharged fluid volume.

7.3.2 Opitz Method

Opitz's method, briefly described in chapter 3 is a powerful method to predict the cavitation duration. However, it has limitations and a lack of performance for complex or multi-chamber pumps. The method performed well in a simple pump layout[11]. However, as is shown in section 7.3.7, limitations constrain this method. The simulation given in figure 7.20 shows the valve gap fluid velocity comparison for the Opitz method and the finite volume developed method.

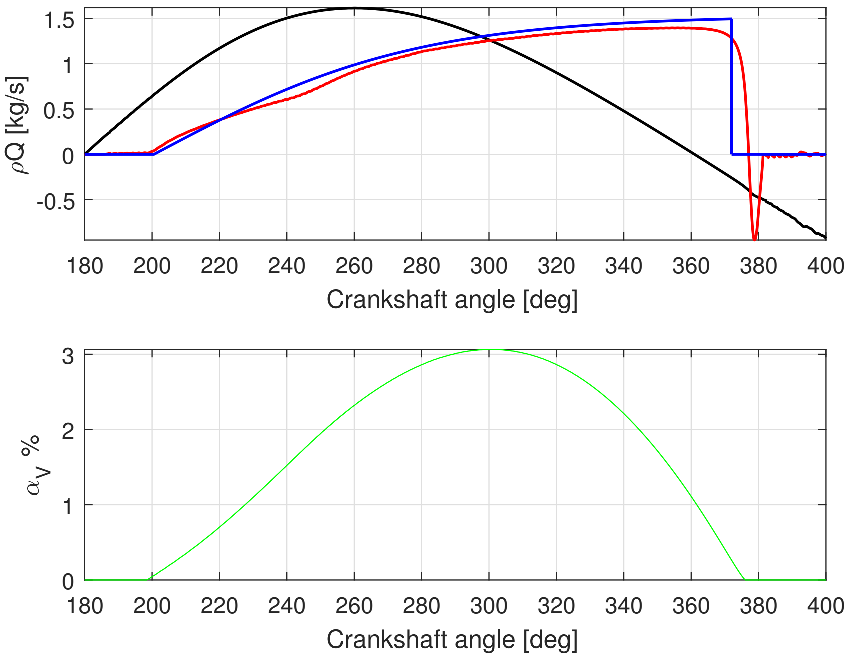


Figure 7.20: Opitz model comparison with the FV developed algorithm, where the black line is the piston displacement, the red line is the FV valve mass flow rate, and the blue line is the Opitz model mass flow rate. In the figure below the FV model of the vapour volume fraction is given.

The comparison shows a similar profile although differences are highlighted in the valve closure and the backflow formation. The reason for the similar behaviour is given by the formulation of the valve velocity. The valve algorithm developed in this research used a variation of the equation developed by Opitz[11]. The behaviour of the valve in the case of closure is not depicted by the Opitz model. In addition, the compressible developed algorithm was simulated first to obtain the valve opening delay to use in the Opitz method.

7.3.3 One-chamber pump developed model

As already described in the validation chapter for the water hammer phenomenon, the response of the algorithm developed in this research is promis-

Table 7.5: Water and Propelling Liquid characteristics

Water		***Propelling Liquid***	
ρ	998 kg/m^3	ρ	810 kg/m^3
E	$2.15e^9 Pa$	E	$1.37e^9 Pa$
ν	$8.94e^{-4}$ Pa	ν	$8.94e^{-4} Pa$
p_{Vapour}	2338 Pa	p_{Vapour}	2338 Pa
α_{Gas_0}	$1e^{-11}$	α_{Gas_0}	$1e^{-10}$

ing. In this section, the description of the fluid dynamics for a one chamber pump is analysed. The data comparison are given for the result of the test rig in Weir, the Netherlands facility and the FV developed algorithm. All the simulations that are shown in this section were performed with the fluid parameters reported in table 7.5.

The parameters used in the FV simulation, are:

- Limiter, the slope limiter function used to produce a TVD scheme.
- Cavitation algorithm, where the choice can be either DVCM or DGCM. The DGCM was seen performing better in the validation chapter, for this reason all simulations are computed with DGCM.
- Pressure force, the methodology to describe the pressure function across the valve, where Thiel methodology described in section 4.7.1.
- CFL, the Courant number condition.
- NCG_p, the number to avoid numerical noise in the DGCM algorithm[32].
- Cavitation limits, namely the maximum value allowed in each cell in term of cavity volume fraction.
- Advective term, if it is used.
- Liquid density, in relation to this function it is also connected to the

Table 7.6: Simulation parameters

Solver

Limiter	Minmod
Cavitation	DGCM
Pressure Force	Thiel
CFL	0.9
NCG_p	0.9
Cavity Limits	0.85
Advective Term	Yes
Liquid density	Bulk
Spring force	Linear
dx	0.003 m

speed of sound. The bulk modulus method does not use the Fluid Structure Interaction algorithm.

- Spring force is dependent on the typology of the spring used. The algorithm can allow the use of polynomian sping force coefficient for non-linear spring response.

- dx is the space grid value.

A summury of this are re reported in table 7.6.

The choice of these parameters for all the simulations were found the best combination for all the simulations.

Referring to table 7.2 the first analysis is shown for case 2, where the suction pressure was set at 2 bar and the pump speed at 100 SPM. Figure 7.21 shows the second cycle of the pump pressure and the pressure profile matches the experimental data with a high degree of accuracy. The second cycle of the simulation was used to avoid error due to the initialization process. At the initial condition, the algorithm has valve lift set to zero. Therefore the suction valve is closed and no backflow is considered. In the real case at zero degree of crankshaft angle, the suction valve is open due to a physical delay.

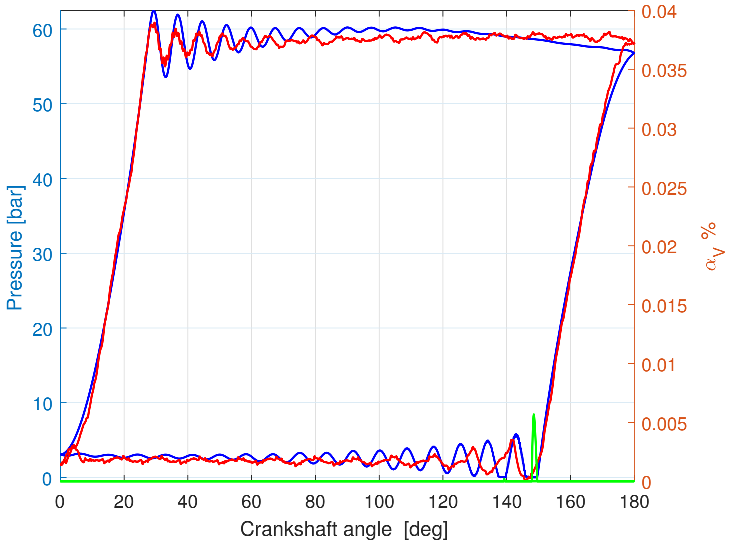


Figure 7.21: Chamber pressure comparison of experimental data, red line and FV model developed by the author blue line for case 2, with discharge pressure at 60 bar, suction pressure 2 bar and 100 SPM. The green line is the vapour volume fraction.

The pressure profile duration is accurate, matching the slope of the compression and decompression phases. However, concerning the pressure pulsation, the simulation overestimates the fluctuation in both the suction and the discharge sections. However, the frequency is in agreement with the experiment. The cavitation duration, extrapolated from the pressure profile, is of the same order, although in the simulation a second period of cavitation is produced at 140 degrees, albeit very small. This discrepancy is related to the overestimation of the reflected pressure wave. The pressure profile in the experiment is smeared out immediately after the cavitation formation, something that does not occur in the simulation.

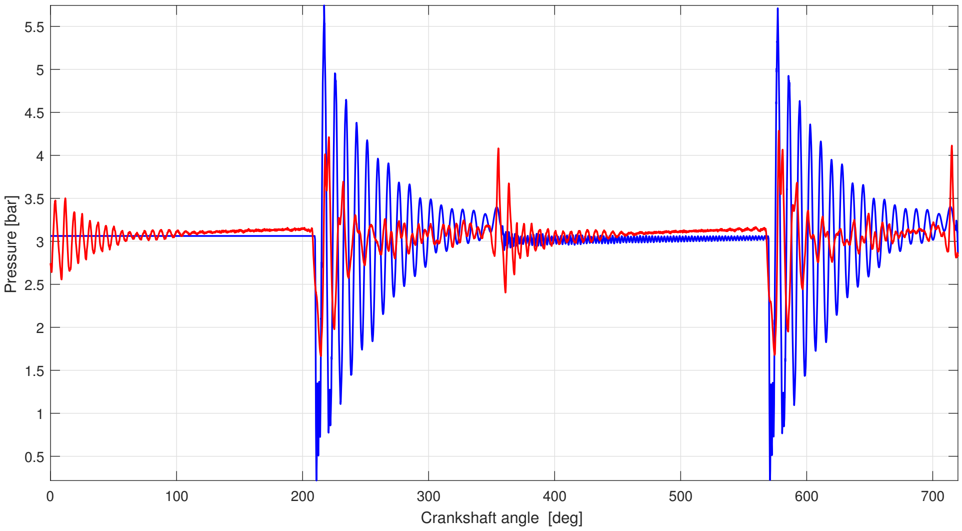


Figure 7.22: Suction line pressure comparison of experimental data red line with FV model developed by the author blue line for case 2, with discharge line pressure at 60 bar, suction pressure 2 bar and 100 SPM.

The main reason could be attributed to the dissipation term. The same issues were also found in the water hammer simulation in the absence of vapour formation. The exact same behaviour is given for the pressure in the suction manifold line, before the suction valve. In figure 7.22 is shown the suction pressure for both two cycles simulated. As soon as the valve opens the pressure drops due to the increase of the fluid velocity across the valve. This behaviour is well described from the simulation, although the value reached is lower than the experiment. The reason can be attributed to the valve velocity model. This model uses empirical parameters available in the literature, that require the be better matched to the valve.

What has been described for the pressure profile can be seen in the Fourier analysis shown in figure 7.23. The frequencies for the pressure pulsation in the chamber coincide with the experimental data and displays only small differences.

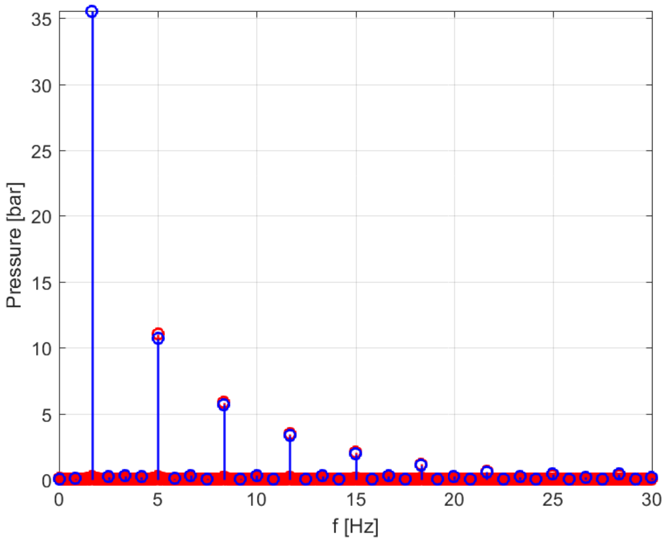


Figure 7.23: Frequency analysis of the chamber pressure for experimental data red line and FV model developed by the author blue line for case 2, with discharge pressure at 60 bar, suction pressure 2 bar and 100 SPM.

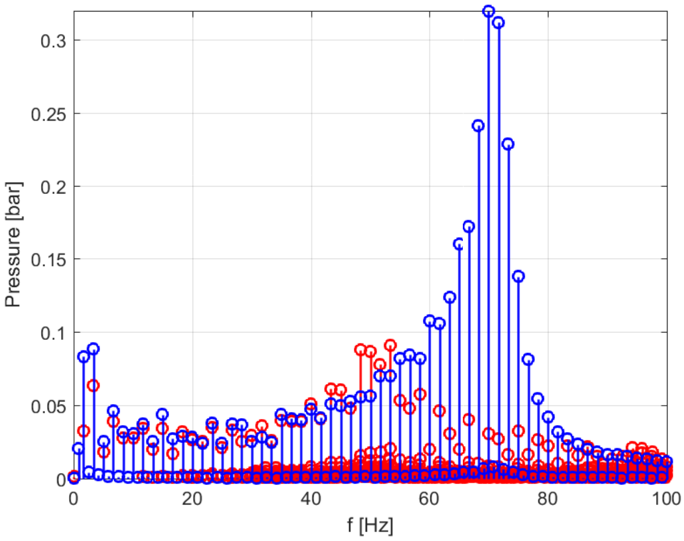


Figure 7.24: Frequency analysis of the suction pressure for experimental data red line with FV model developed by the author blue line for case 2, with discharge pressure at 60 bar, suction pressure 2 bar and 100 SPM.

In figure 7.24 the suction manifold frequency spectrum shows a good agreement for the first 50 frequencies. For the high-frequency values, the amount of energy is higher for the simulation and is also reflected in the higher peak pressures in figure 7.22. The reason can be found in the less dissipative behaviour of the simulated pump compar with the experimental data.

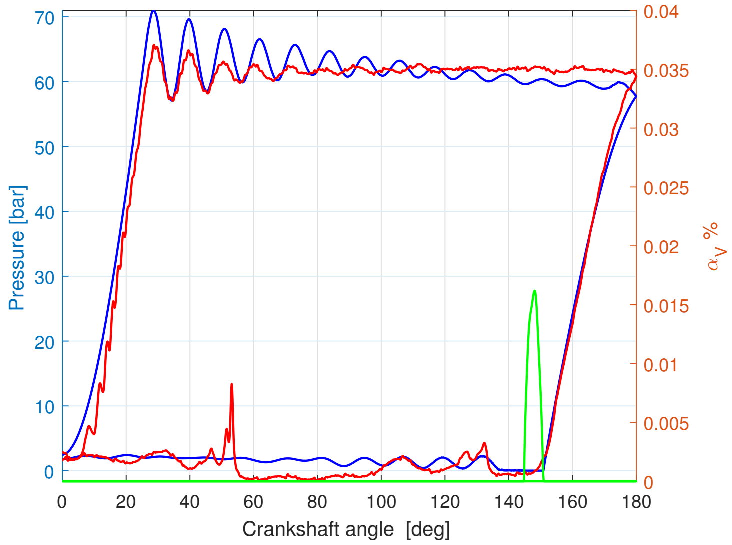


Figure 7.25: Chamber pressure comparison of experimental data red line and FV blue line for case 6, with discharge pressure at 60 bar, suction pressure 2 bar and 150 SPM. The green line is the vapour volume fraction.

Increasing the pump speed by 50%, the fluid response slightly changes. Once more the overall pump behaviour is well described. However, in the suction phase the comparison is not perfect. The experimental data shows a pressure profile that is not depicted in the simulation, figure 7.25. The first peak pressure as well as the duration of the cavitation, are well described by the model. However, after the first peak, the experimental data constantly decreases its value. In addition, close to the end of the suction phase, at 50 degree of crankshaft rotation, the experimental data shows a peak pressure due to the cavitation collapse or a wave reflection.

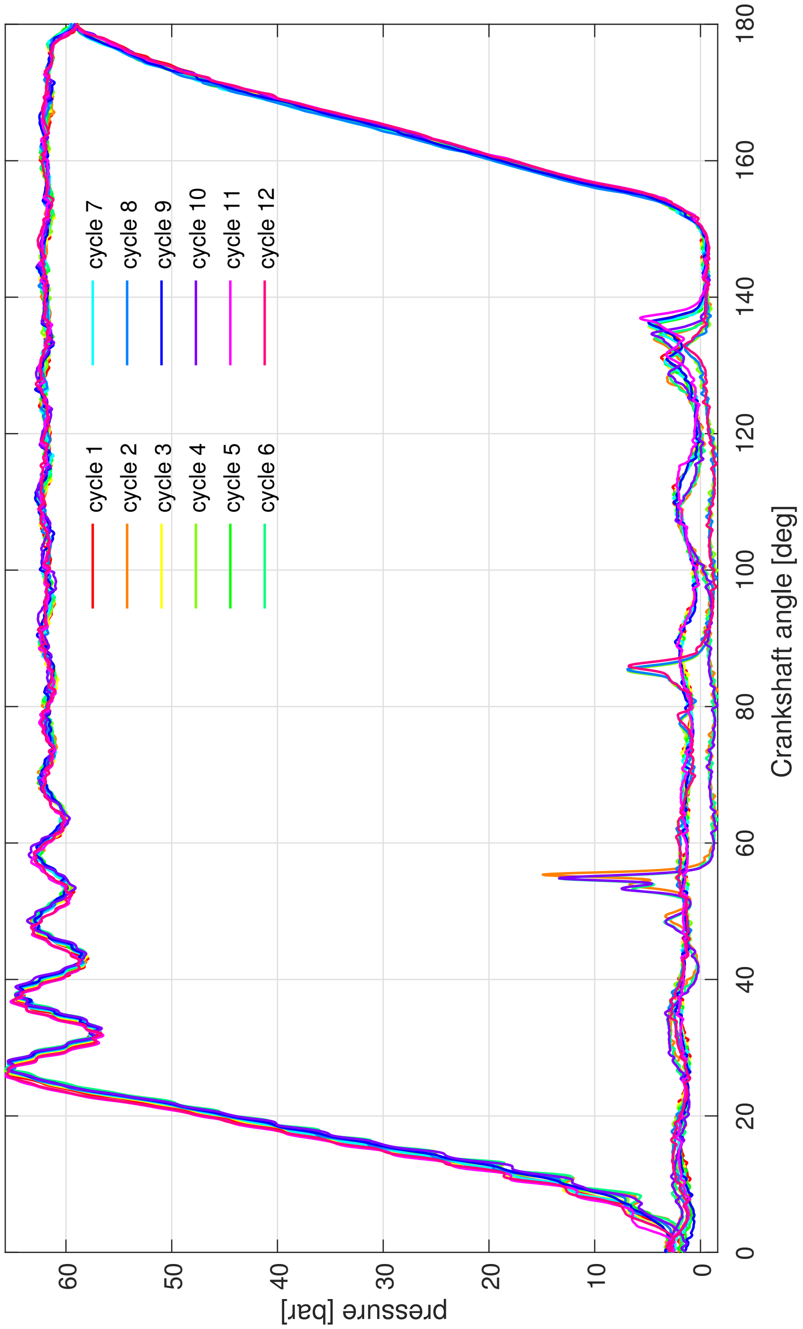


Figure 7.26: Experimental chamber pressure comparison for 12 consecutive pump cycles for case 6, with discharge pressure at 60 bar, suction pressure 2 bar and 150 SPM.

The differences are due to the limited number of cycles simulated. In figure 7.26, 8 consecutive cycles of the experimental results are plotted. It is possible to see that the test rig presents a superimposed cycle equal to 4 pump revolutions. This is the frequency response of the suction tank that is not modelled in the system. Surprisingly, this behaviour was only depicted for cases 10 and 6. No definitive conclusion can be made. However, it is possible that the working points excite the entire system at the tank frequency. The pressure pulsation spectrum in figure 7.27 shows good agreement between the experimental data and the simulation for the higher running speed.

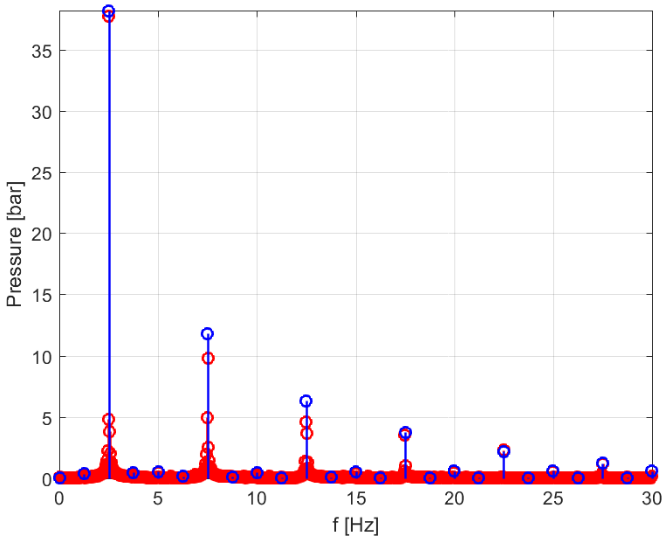


Figure 7.27: Frequency analysis of the chamber pressure for experimental data red line with FV blue line for case 6, with discharge pressure at 60 bar, suction pressure 2 bar and 150 SPM.

However, the pressure response of the simulation in the suction manifold is different from the experimental as shown in figure 7.28. The same behaviour described in the chamber is amplified in the suction manifold.

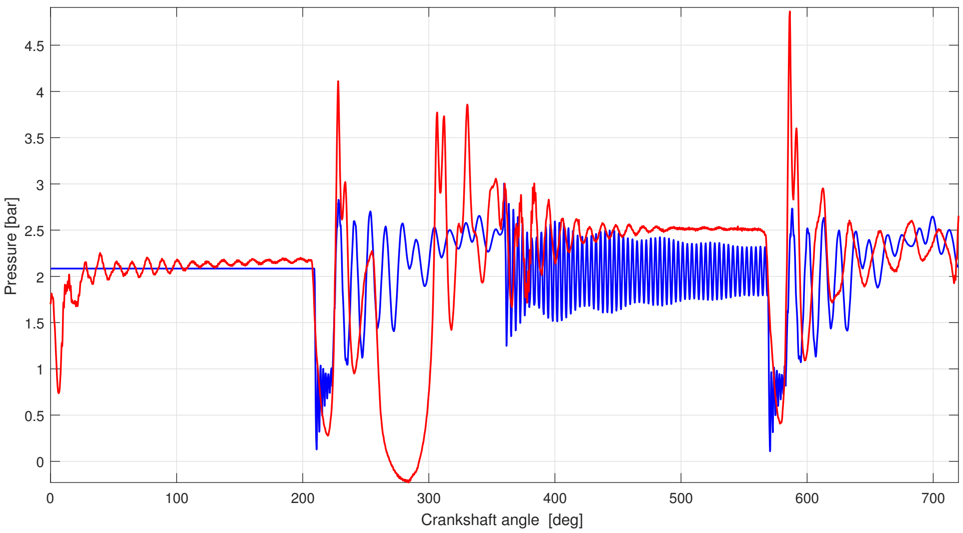


Figure 7.28: Suction pressure comparison of experimental data red line with LPM blue line for case 6, with discharge pressure at 60 bar, suction pressure 2 bar and 150 SPM.

Although the trough at 210 and 570 degrees is perfectly matched as soon as the valve opens, the experimental pressure history is not well described by the model. Therefore the first limitation due to the lack of a complete model of the system (tank missing) is evident. In addition, when the valve closes, at approximatively 380 crankshaft degrees, the high frequency pulsation is not smeared out as fast as the experimental data. Once more this is related to the domain restriction and the dissipation model. For this case, the differences can also be seen in the frequency spectrum of the pressure in the suction manifold, figure 7.29. The simulation captures only the principal frequencies, but with different amplitude.

For further increases of the pump speed, the simulation performance is still in good agreement. The pressure pulsation shows good agreement for 175 and 200 SPM, as indicated in figure 7.30 and 7.31 respectively.

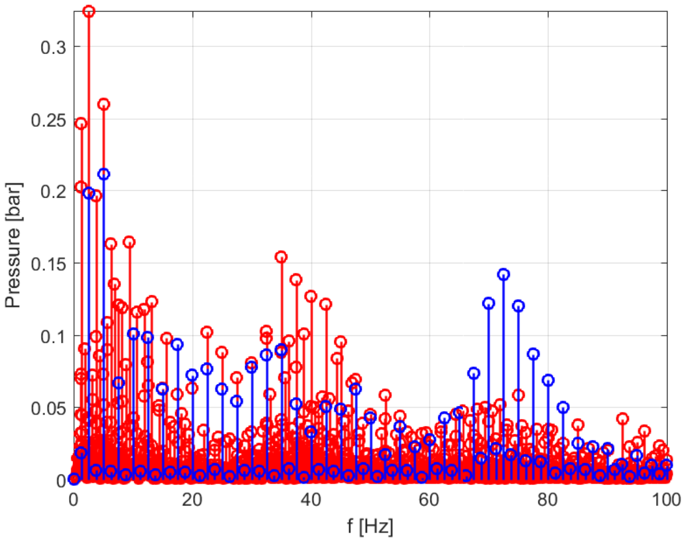


Figure 7.29: Frequency analysis of the suction pressure for experimental data red line with FV blue line for case 6, with discharge pressure at 60 bar, suction pressure 2 bar and 150 SPM.

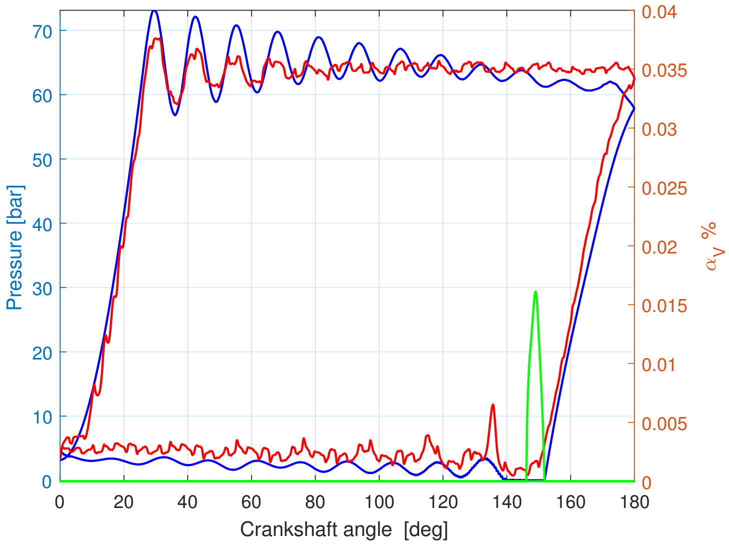


Figure 7.30: Chamber pressure comparison of experimental data red line and FV blue line for case 11, with discharge pressure at 60 bar, suction pressure 3 bar and 175 SPM. The green line is the vapour volume fraction for the developed FV model.

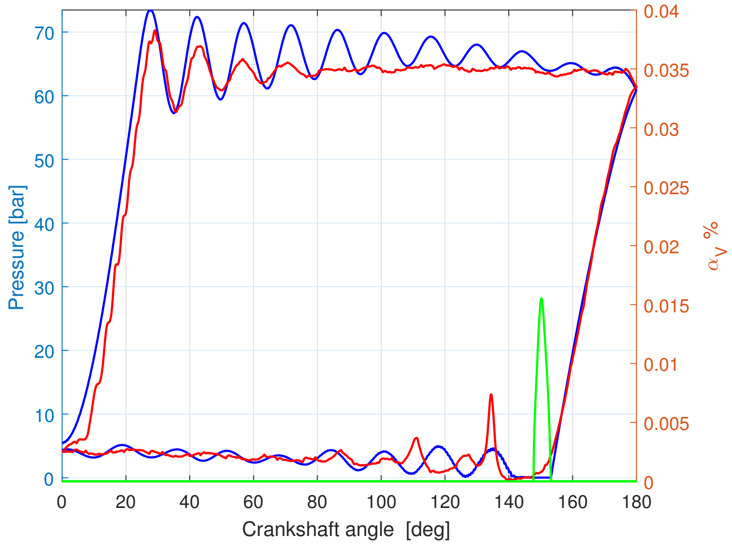


Figure 7.31: Chamber pressure comparison of experimental data red line and FV blu line for case 15, with discharge pressure at 60 bar, suction pressure 3 bar and 200 SPM. The green line is the vapour volume fraction for the developed FV model.

When low suction pressure at high velocity was tested, large differences between the simulation and experimental results started to appear. As shown in figure 7.32, the model is not able to simulate and evaluate the pressure profile in fully developed cavitation conditions[11].

It is likely that the amount of cavitation produced in the algorithm is much lower than the experimental data, and the algorithm is not able to predict the pump response correctly. The proof of this assumption can be seen in the pressure pulsation for the compression phase, at the beginning of the figure 7.32. In the compression phase the piston must destroy the formed bubbles before compressing the liquid, causing the multi-peak behaviour. As a further drawback, the produced cycle is shorter than the cycles with less cavitation, producing a drastic decrease in volumetric efficiency. More research in this extended cavity condition must be performed to understand the real limit of this algorithm, although no pumps should reach this condition because it can be seriously damaged.

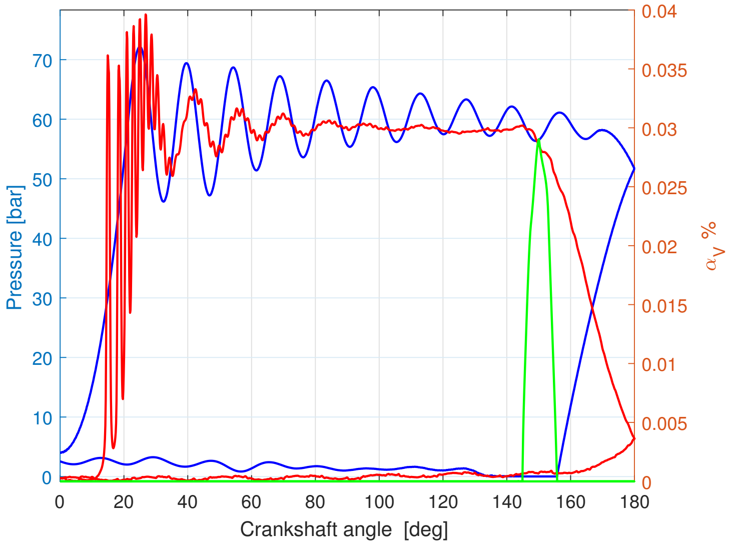


Figure 7.32: Chamber pressure comparison of experimental data red line and FV blu line for case 13, with discharge pressure at 60 bar, suction pressure 1 bar and 200 SPM. The green line is the vapour volume fraction for the developed FV model.

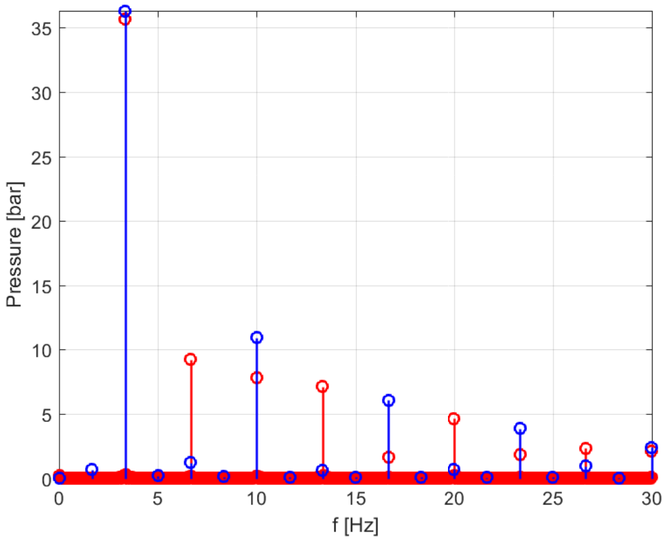


Figure 7.33: Frequency analysis of the camber pressure for experimental data red line with FV blue line for case 11, with discharge pressure at 60 bar, suction pressure 1 bar and 200 SPM.

Even the pump frequency response is no longer the same as the exper-

Table 7.7: Correlation factor r_{xy} for one chamber pump

SPM	$p_{Suction}$			
	1	*2*	*3*	*4*
100	0.998	0.999	0.9972	0.9984
150	0.9981	0.9975	0.9969	0.9975
175	0.9966	0.9963	0.9965	0.9015
200	0.9967	0.9964	0.9963	0.9129

iment as highlighted in 7.33. The frequencies have a completely different energy content. In conclusion, the algorithm works well in all the cases analysed except for the high cavitation regime. The limitation is given principally by the system simulated, namely the absence of the tank and all pipelines. The frequency response of the simulation data is in accord with the experimental data in most of the cases, albeit with some differences. The energy content is higher for the developed compressible model, especially in the suction manifold at high frequencies. In addition, the dissipation term does not always perform well when compared to the physical pump test. However, considering Pearson's correlation coefficient, r_{xy}, defined in the equation 7.2, to evaluate the pressure response in the chamber, the results are reasonably accurate. In table 7.7 are reported the values of the r_{xy} correlation with the experiment.

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - \left(\sum x_i\right)^2}\sqrt{n\sum y_i^2 - \left(\sum y_i\right)^2}} \tag{7.2}$$

7.3.4 Parametric study

After evaluating the performance and the response of the model by comparison with experimental data, in this section, a parametric study is implemented. Different pump characteristics are considered in order to analyse their impact on the pump performance, especially for cavitation formation and volumetric efficiency. The volumetric efficiency of the test rig pump is low, using formula 2.4, the value calculated without backflow is 84%. This value is affected by the dead volume present in this pump. In fact, the simulated volumetric efficiency, taken as reference is 82.2%, which is in agreement with Tackett's formulation[4]. In table 7.8 are given the simulation parameter values and the reference experiment. Two values of each following parameters were assessed:

- suction valve weight,
- Discharge valve weight,
- Valve seat angle θ,
- Suction spring force,
- Discharge spring force,
- Suction preload force,
- Discharge preload force,
- ϕ_{Dia} membrane diameter,
- propelling liquid density.

The reference data for these parameters are given in the first line of the table 7.8.

The first analysis was performed for the suction valve weight. The sensitivity analysis of this parameter shows a small impact on overall pump behaviour. However, as expected, the reduction of the inertia of the valve decreases the amount of cavitation, although by only a small amount, figure 7.34.

Table 7.8: Parameters value

#	**SPM**	**Suction Pressure [Pa]**	**Discharge Pressure [Pa]**	**Valve Suc [kg]**	**Valve Dis [kg]**	θ [deg]	**Suc Spring [N/m]**	**Dis Spring [N/m]**	**Suc PreL [N/m]**	**Dis PreL [N/m]**	ϕ_{Dia} [m]	**Propelling density [kg/m3]**
Ref	200	4	60	0.1	0.1	45	750	750	10	10	0.2	875
1	200	4	60	0.05	0.1	45	750	750	10	10	0.2	875
2	200	4	60	0.2	0.1	45	750	750	10	10	0.2	875
3	200	4	60	0.1	0.05	45	750	750	10	10	0.2	875
4	200	4	60	0.1	0.2	45	750	750	10	10	0.2	875
5	200	4	60	0.1	0.1	75	750	750	10	10	0.2	875
6	200	4	60	0.1	0.1	15	750	750	10	10	0.2	875
7	200	4	60	0.1	0.1	45	450	750	10	10	0.2	875
8	200	4	60	0.1	0.1	45	1050	750	10	10	0.2	875
9	200	4	60	0.1	0.1	45	750	450	10	10	0.2	875
10	200	4	60	0.1	0.1	45	750	1050	10	10	0.2	875
11	200	4	60	0.1	0.1	45	750	750	20	10	0.2	875
12	200	4	60	0.1	0.1	45	750	750	5	10	0.2	875
13	200	4	60	0.1	0.1	45	750	750	10	20	0.2	875
14	200	4	60	0.1	0.1	45	750	750	10	5	0.2	875
15	200	4	60	0.1	0.1	45	750	750	10	10	0.4	875
16	200	4	60	0.1	0.1	45	750	750	10	10	0.1	875
17	200	4	60	0.1	0.1	45	750	750	10	10	0.2	775
18	200	4	60	0.1	0.1	45	750	750	10	10	0.2	975

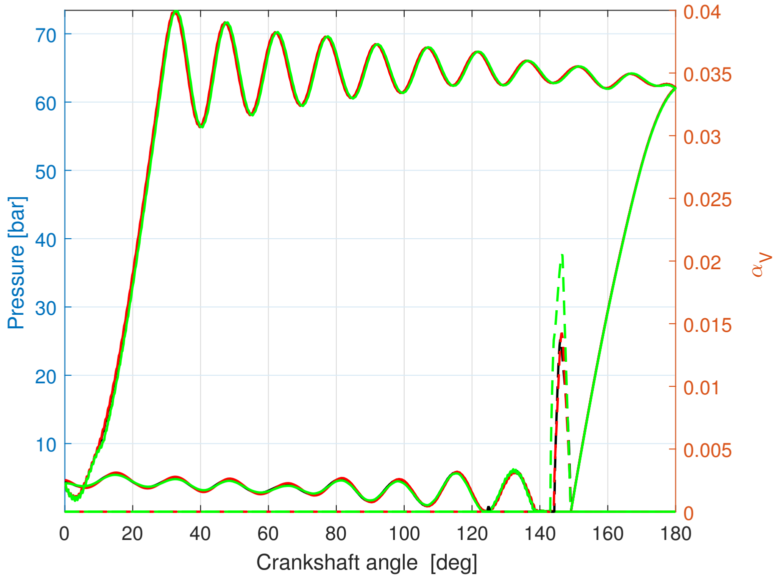


Figure 7.34: Pressure history comparison for the suction valve variation, red line is case 1, lighter valve; green line is case 2, heavier valve; and the black line is the reference. The dash line refers to the vapour volume fraction.

Therefore, the suction valve should be as light as possible to reduce the opening delay and the consequent cavitation. However, mechanically speaking, the valve requires a structure capable of supporting the stresses that could limit the reduction in mass. In addition, during the closure phase, the valve should close as fast as possible to reduce the closure delay and backflow. This behaviour is commonly achievable with an increase of the suction valve weight which is in contradiction with the previous goal.

In figure 7.35 the opening valve dynamics are shown. No significant difference is formed although a lower valve lift is reached with the heavier valve.

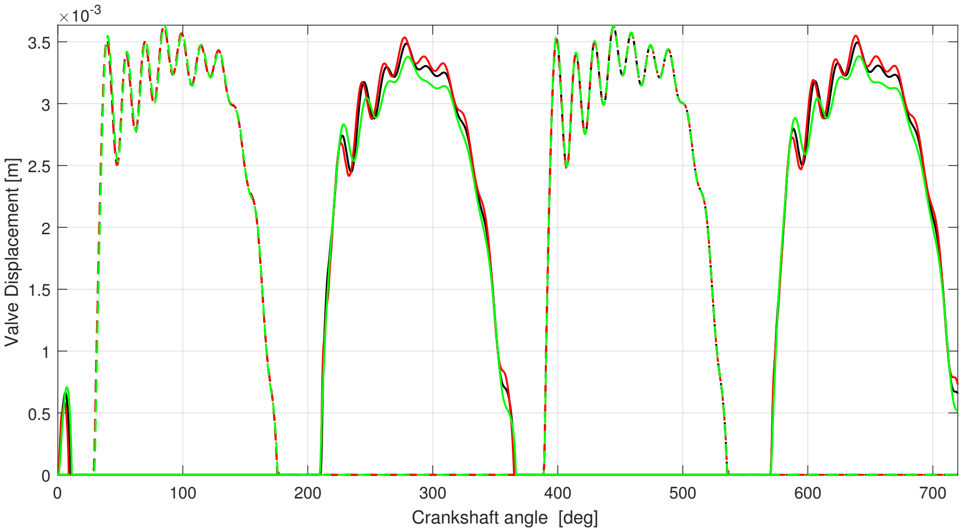


Figure 7.35: Valve model comparison for the suction and discharge (dash line) valves, red line is case 1, lighter suction valve; green line is case 2, heavier suction valve, and the black line is the reference.

No significant variation in pressure and cavitation are shown for the case of discharge valve variation. The pressure pulsation is slightly higher for the heavier valve at the beginning of the discharge phase. While less oscillation is shown at the end of the cycle. The reason is the higher inertia that increases the first pulsation and the breakdown pressure, figure 7.36 .

The effect of the valve seat angle is interesting as shown in figure 7.37. The reference valve seat angle is 45 degree, which is the test rig valve seat angle. The variation of this angle changes the gap area for the same valve lift condition. For this reason, the algorithm performs better for the middle to high seat angle range. However, a limitation of this value must be considered. Iannetti[3] in his experimental results defined a threshold for the seat angle to be around $35 - 55$ degrees, which is in contradiction with this result. However, the algorithm developed by the author is only one-dimensional. Therefore the radial direction and the wall interaction are not considered in this simulation.

The sensitivity analysis of the spring preload variation does not change

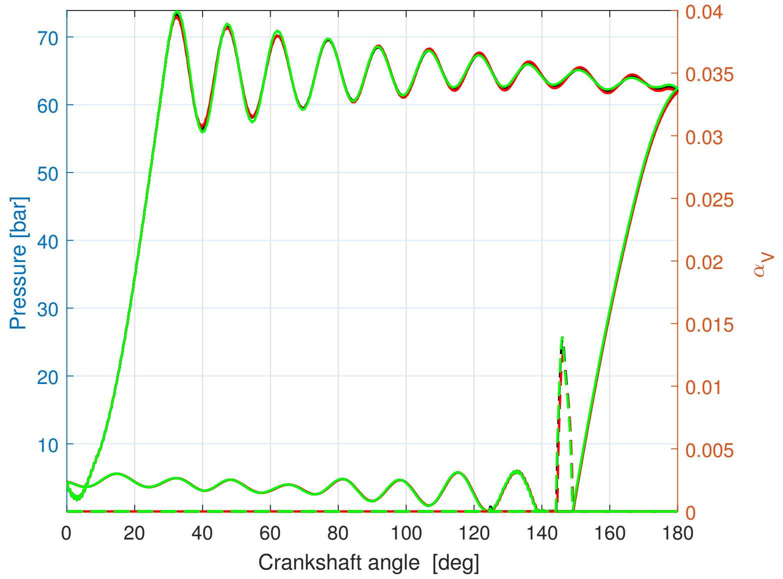


Figure 7.36: Pressure history comparison for the discharge valve variation, red line is case 3, lighter valve; green line is case 4, heavier valve; and the black line is the reference. The dash line refers to the vapour volume fraction.

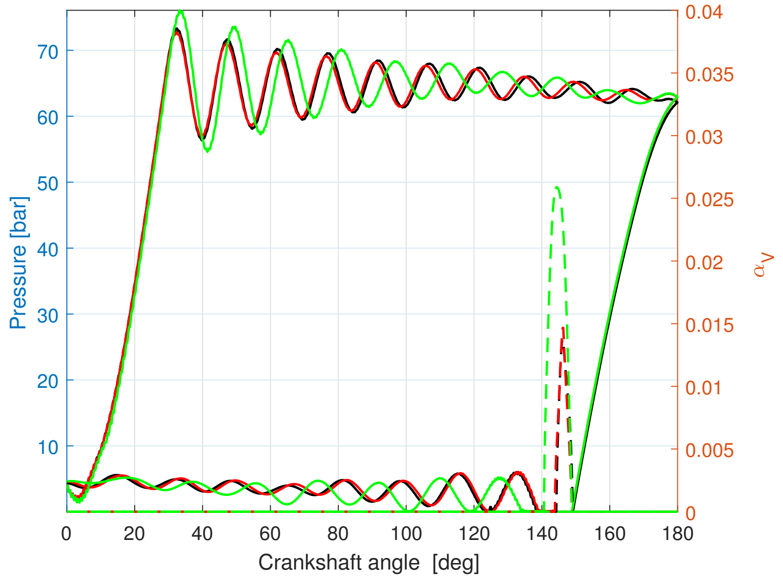


Figure 7.37: Pressure history comparison for the θ angle of the valve, red line is case 5, $\theta = 75$, green line is case 6, $\theta = 15$, and the black line is the reference. The dash line refers to the vapour volume fraction.

the pump behaviour. Hence no results for this are shown. Of more importance is the pump diaphragm size. The volume of the chamber is typically much larger than the piston discharge volume. Figure 7.38 shows the significant impact this has on the pump performance. From an engineering point of view, reducing the pump chamber size is critical to be optimising pump performance.

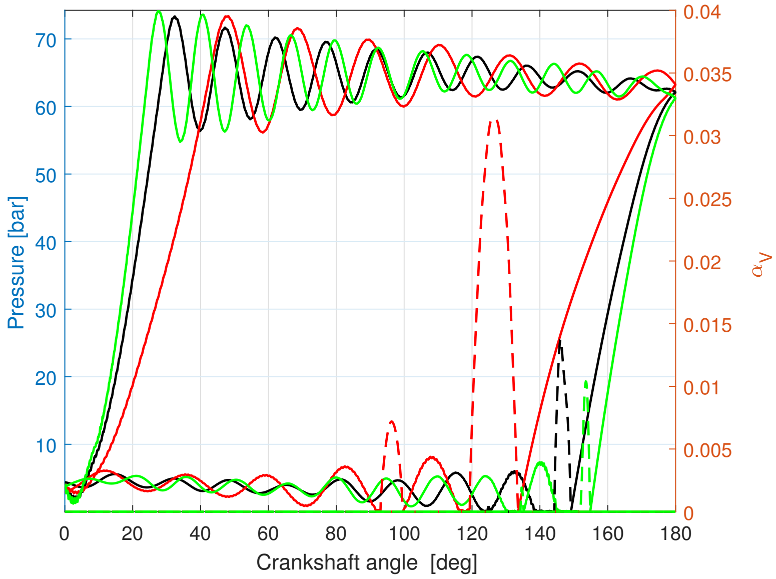


Figure 7.38: Pressure history comparison for the diaphragm size variation, red line is the case 15, bigger chamber; green line is the case 16, smaller chamber; and the black line is the reference. The dash line are refereed to the vapour volume fraction.

Performance of the pump will benefit from chamber volume reduction. Cavitation will be reduced due to the reduction of the fluid inertia and the higher response rate. Moreover, flow pulsations will be reduced, although the oscillation frequency increased as shown in figure 7.39.

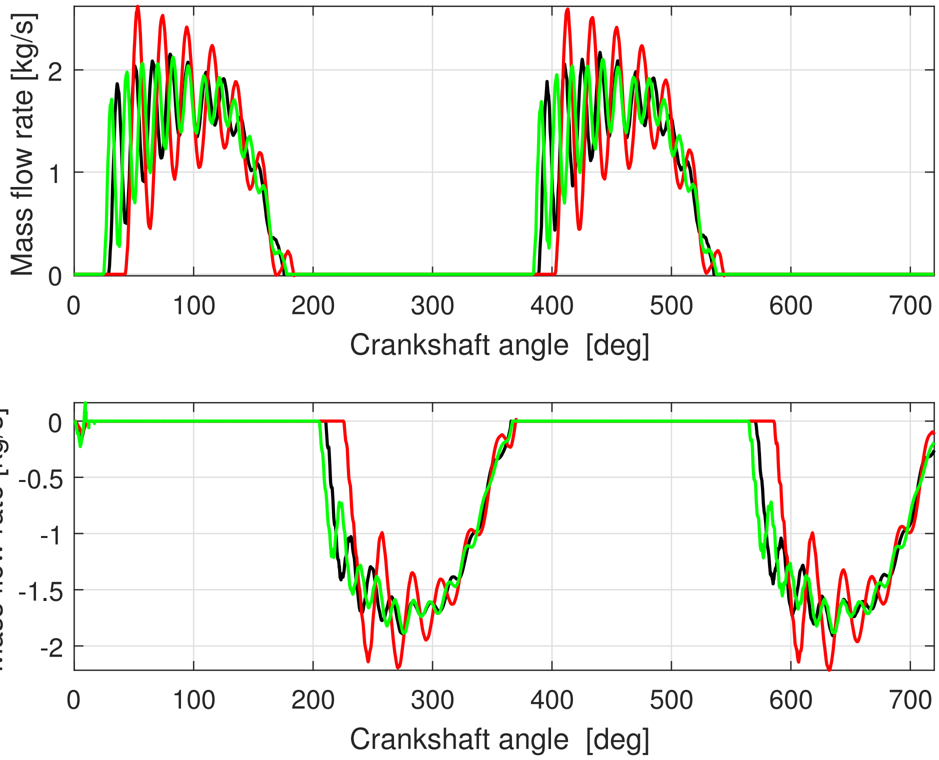


Figure 7.39: Mass flow rate history comparison for the diaphragm size variation, red line is case 15, bigger chamber; green line is case 16, smaller chamber; and the black line is the reference.

Furthermore, the frequency response of the fluid will be improved. The frequency spectrum shows a shift of the frequency for both suction and discharge manifold pressure to a higher value by decreasing the diaphragm size, figure 7.40.

The flow rate response in the suction and in the discharge manifold will have a beneficial improvement in the case of diaphragm size reduction as shown in figure 7.39. However, a reduction in the membrane area will increase the local stresses and the stretch of the rubber, which could lead to possible structural damage. Therefore, this model gives a good idea of possible pump improvement, but the result must be carefully considered.

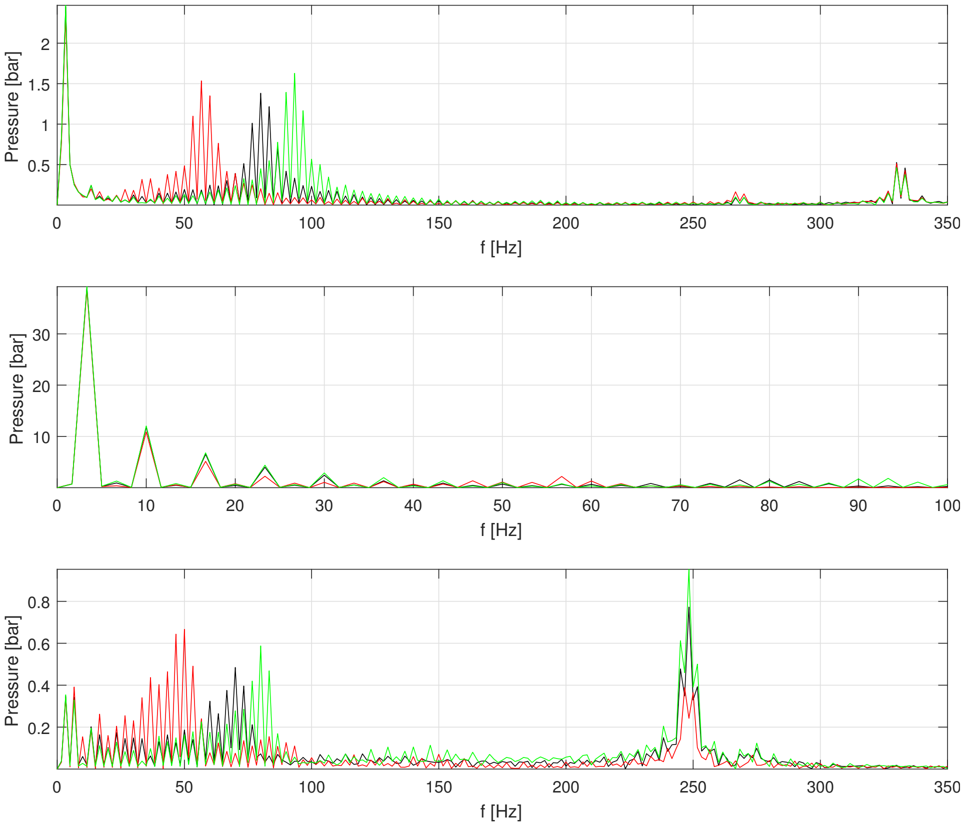


Figure 7.40: Frequency spectrum analysis for the diaphragm pump size variation, red line is the case 15, bigger chamber; green line is the case 16, smaller chamber; and the black line is the reference.

The last parameter analysed is the propelling fluid's property. A variation in the density of the propelling's fluid slightly changed the behaviour of the pump. From figure 7.41 it is possible to see the effect in terms of cavitation where, for higher density, the cavitation marginally increases.

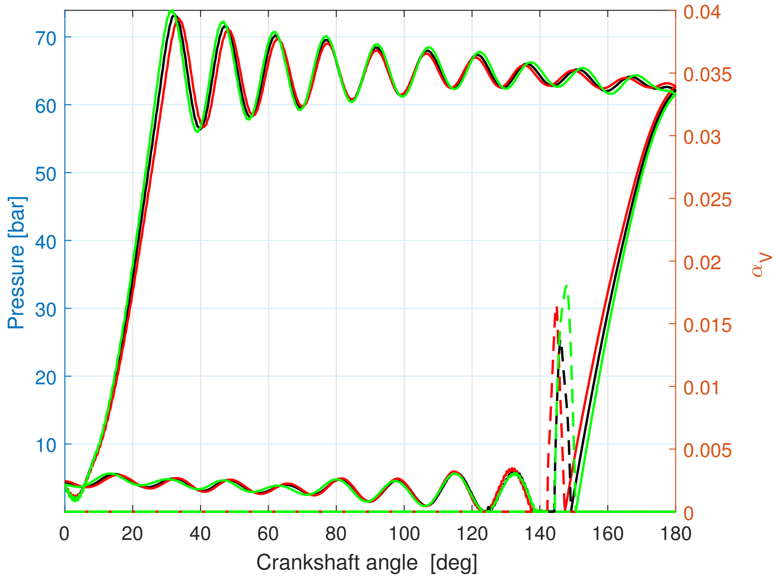


Figure 7.41: Pressure history comparison for the propelling fluid variation, red line is the case 17, less dense green line is the case 18, more dense, and the black line is the reference.

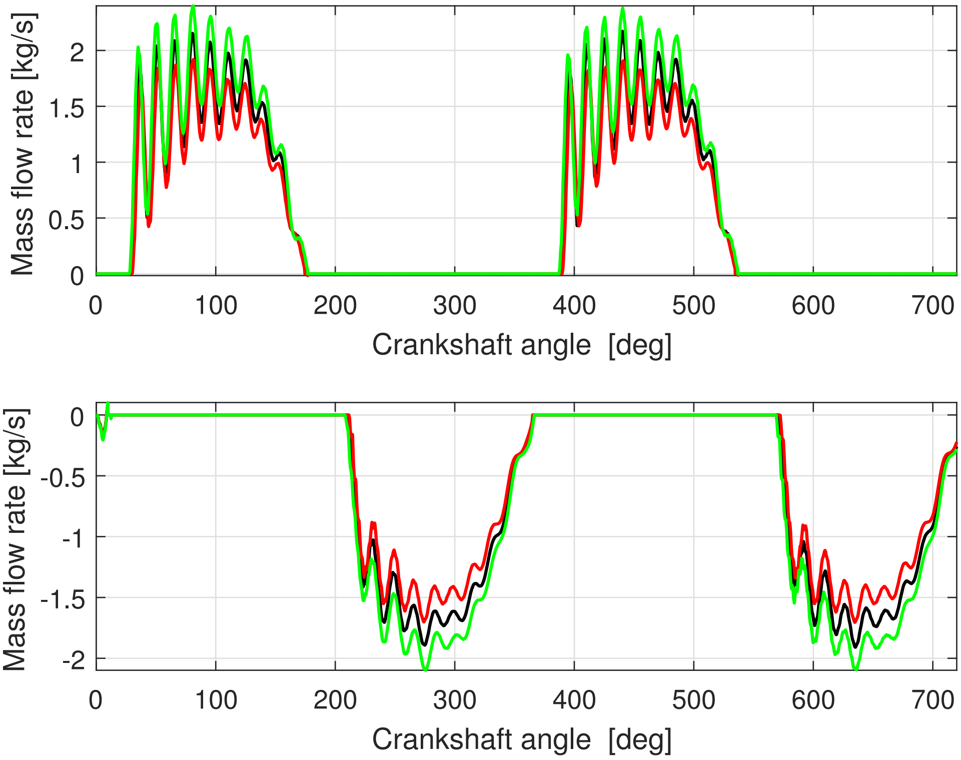


Figure 7.42: Pressure history comparison for the propelling fluid variation, red line is the case 17, less dense green line is the case 18, more dense, and the black line is the reference.

In conclusion, where cavitation increase's, figure 7.44, the volumetric efficiency decrease consequently, figure 7.43. This is true for all analysed cases except for the study of the propelling fluid density. The increase of the propelling liquid density creates a faster response of the system. The system follows the piston much faster than the lower density fluid case. Therefore the effect is an increase in the flow rate figure 7.42, and therefore an increase in the volumetric efficiency. However, cavitation increases due to the increase in the velocity of the fluid at the valve.

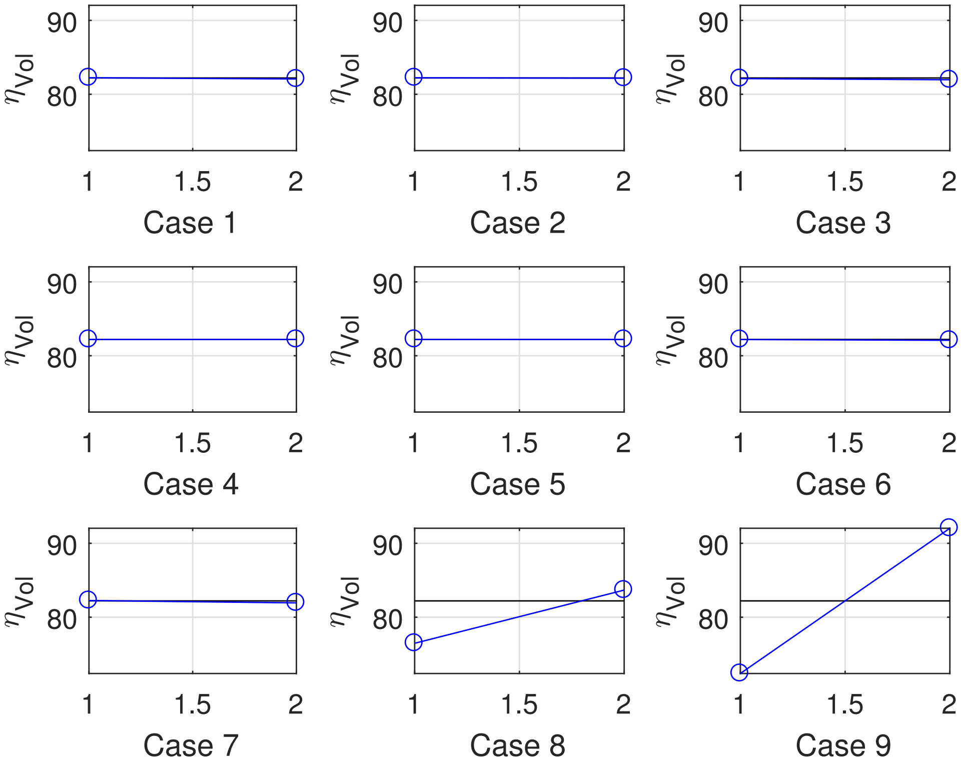


Figure 7.43: Comparison of the volumetric efficiency η_{Vol} for the all the study parameters.

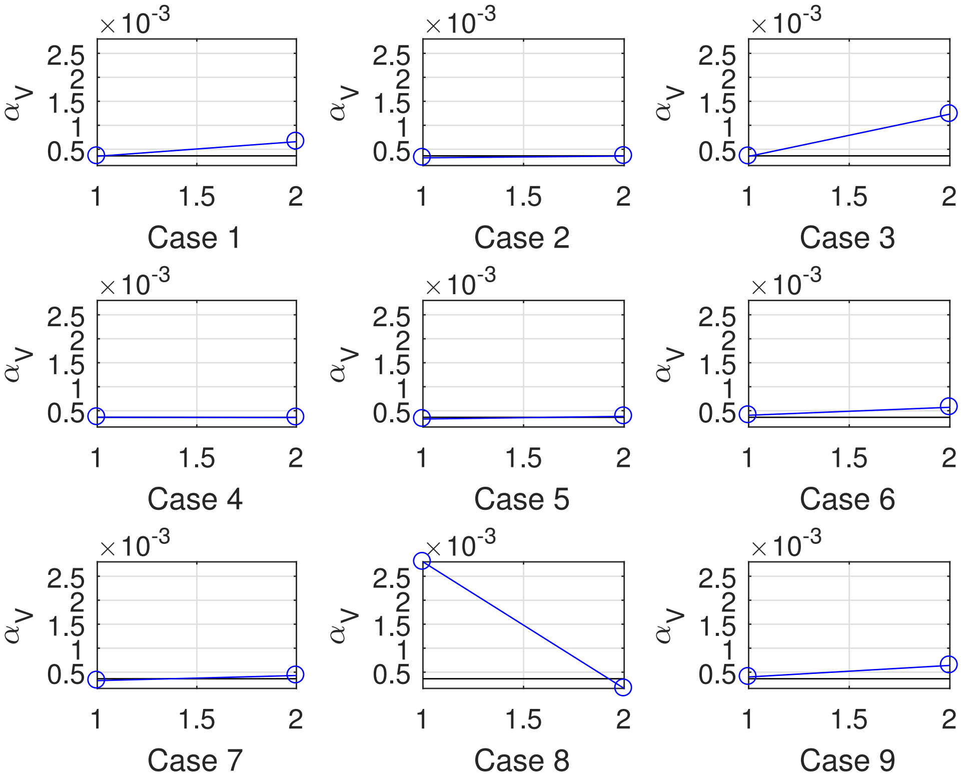


Figure 7.44: Comparison of the vapour volume fraction α for all the study parameters.

7.3.5 Three-chamber pump analysis

In this section, the three chamber pump is analysed. The simulated parameters and the fluid's characteristics are given in table 7.9 and 7.10 respectively.

Table 7.9: Slurry and Propelling Liquid characteristics for three chamber pump

Water		***Propelling Liquid***	
ρ	998 kg/m^3	ρ	810 kg/m^3
E	$2.15e^9 Pa$	E	$1.37e^9 Pa$
ν	$8.94e^{-4}$ Pa	ν	$8.94e^{-4} Pa$
p_{Vapour}	2338 Pa	p_{Vapour}	2338 Pa
α_{Gas_0}	$1e^{-11}$	α_{Gas_0}	$1e^{-10}$

Table 7.10: Simulation parameters for three chamber pump

Solver	
Limiter	Minmod
Cavitation	DGCM
Pressure Force	Thiel
CFL	0.9
NCG_p	0.9
Cavity Limits	0.85
Advective Term	Yes
Liquid density	Bulk
Spring force	Linear
dx	0.031 m

At the start of the simulation an initialization problem occurred. Unlike the one chamber pump; the initialization parameters of the multi-chamber pump can potentially produce computational issues. The main problem appears in the piston velocity boundary condition, due to the different phase angles between the chambers.

Considering an initial case of steady velocity for the entire domain, as soon as the algorithm solves the piston boundary condition, a variation in speed profile arises. The velocity in the chamber that has a phase angle equal to zero will increase the speed gradually. While, the other chambers with a shifted crankshaft angle will produce a higher step in velocity. According to the Joukowsky formula, the step in velocity produces an increase in pressure which in the algorithm, is computed with equation 7.3.

$$p_i^{n+1} = p_i^n + \rho_i^n c_i^n (v_i^{n+1} - v_i^n) \tag{7.3}$$

Therefore, every chamber should have the correct value of velocity from the previous time step, in order to prevent non-physical waves. A solution can be to initialize all the parts with the exact initial value as a function

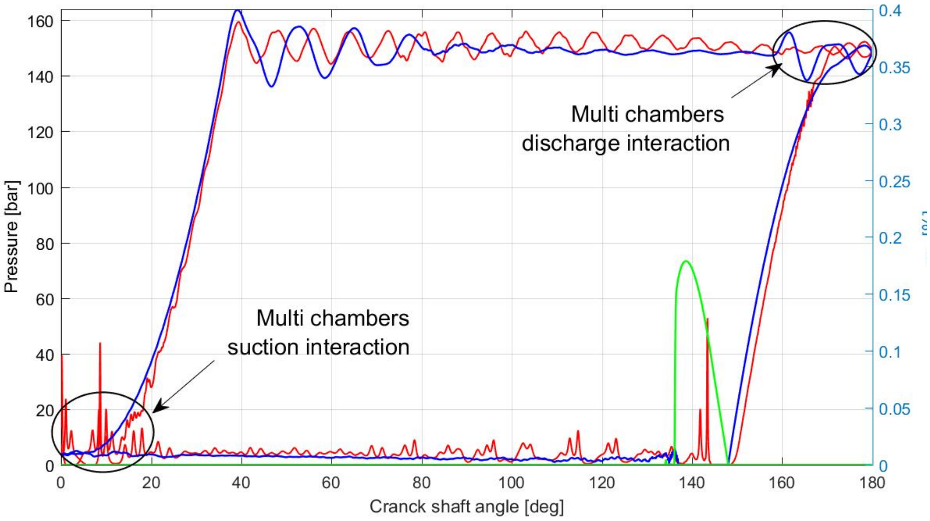


Figure 7.45: Chamber pressure comparison of the experimental data, red line and the FV developed model blue line for the case 1, with discharge pressure at 147 bar, suction at 4 bar and 63 SPM. The green line is the vapour volume fraction for the developed FV model.

of the crankshaft angle. This method implies that all pump components, valves, accumulator, fluid characteristics for the entire domain, must initialize correctly. This approach is impossible to apply. Another approach is to simulate a ramp-up of the pump. In this way the initialization of the pump is equal everywhere but the speed of the pump is not set at the steady condition. The acceleration of the pump gives the model time to adjust over all of the domain. Usually, the real pump is accelerated in less than a minute, namely in the order of 10 cycles. This is impossible to perform in a reasonable computational time. Therefore the pump acceleration took place in only one single cycle. Mathematically, the approach consists of applying a gradual increase in pump speed with a velocity profile shown in figure 7.46.

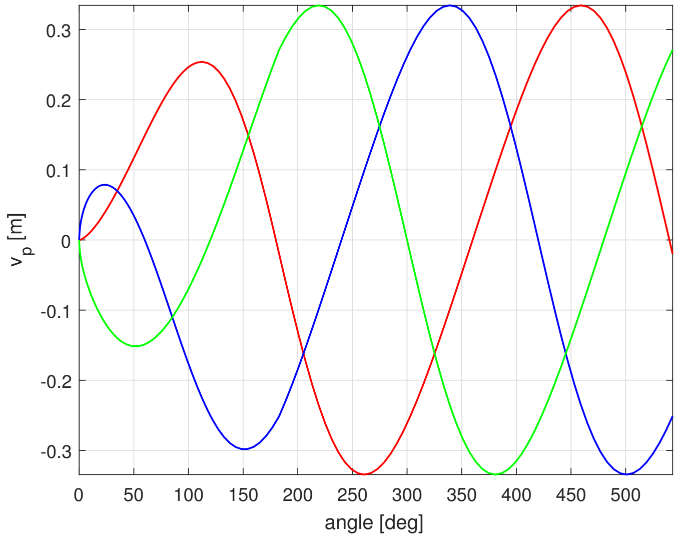


Figure 7.46: Acceleration time for three chambers pump.

This procedure was applied for all the simulations, and the results are only shown the second cycles.

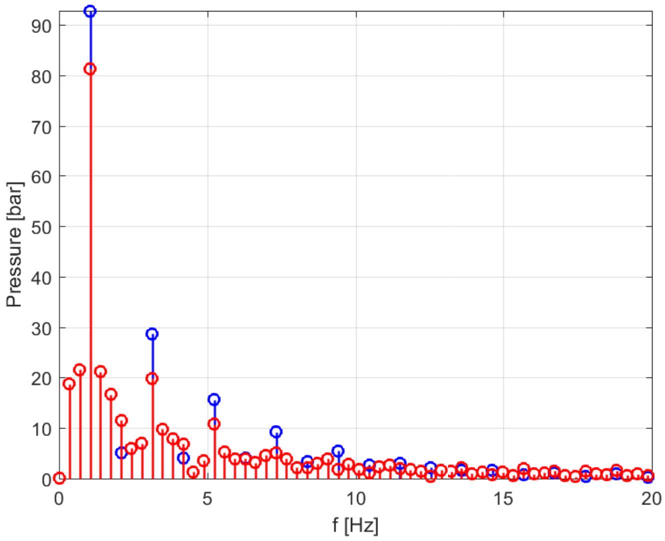


Figure 7.47: Frequency analysis of the chamber pressure for the experimental data, red line and the FV developed model blue line for the case 1, with discharge pressure at 147 bar, suction at 4 bar and 63 SPM.

In addition, the simulated conditions did not take into account the whole network of the pump. In other words, the real pump is affected by the entire connected system, that is the pipelines and the other nine pumps. The simulated domain is truncated after the second hydraulic dampener due to the computational effort required. This approach will affect the results in terms of the pressure pulsation, interference and flow rate response.

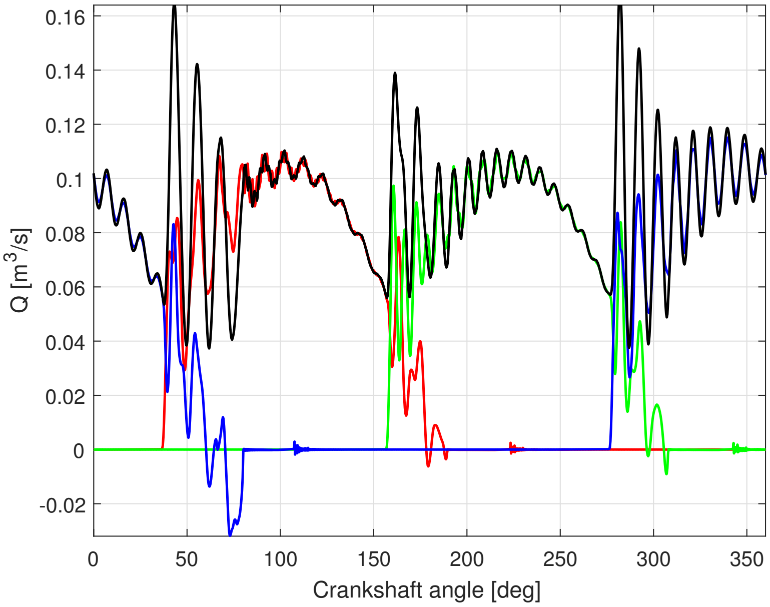


Figure 7.48: Discharge flow rate for the FV developed model for case 1, with discharge pressure at 147 bar, suction at 4 bar and 63 SPM. Blue line is the chamber one, red line is the chamber two, green line is the chamber three and black line is the overall simulated discharged volume flow rate.

The pressure history of the three chamber pump for the maximum speed and the minimum pressure, namely case 1 in table 7.4, is shown in figure 7.45. The pressure pulsation is comparable although discrepancies are highlighted. The interaction with the other chambers is overestimated in the discharge phase and underestimated in the suction phase, as highlight in the boxes of figure 7.45. Even the cavitation extent is overestimated for this simulation case. The peak pressure due to the bubble collapse is earlier in the experimental data than the developed model.

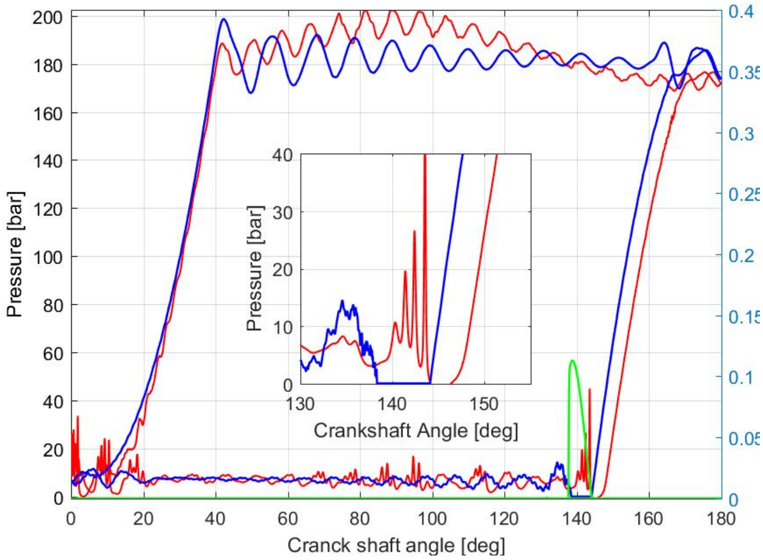


Figure 7.49: Chamber pressure comparison of the experimental data, red line and the FV developed model blue line for the case 8, with discharge pressure at 180 bar, suction at 7 bar and 63 SPM. The green line is the vapour volume fraction for the developed FV model.

The frequency response is comparable only for the dominant frequency as shown in figure 7.47.

The simulated volumetric efficiency, compared with the experimental data available in table 7.3 gives a better agreement than the LPM. In fact, the efficiency is calculated as 88% from the mass flow rate profile in the figure 7.48, while the actual value is equal to 89.3% reported in table 7.3. In addition, comparison with the LPM results, figure 7.16 shows higher variation in the flow rate profile and backflows that are not depicted by the LPM.

Increasing the suction pressure and simulating case 8, the results improve in performance. Figure 7.49 shows a better agreement than the previous case. In this condition the amount of cavitation predicted is in agreement with the experimental data as shown by the peak to peak value reported in the box in figure 7.49. A slight delay in the valve closure is simulated by the algorithms, although the pressure slope, $\frac{\partial p}{\partial x}$ is comparable.

The frequency analysis showed a higher content of energy for the simulated pump, although the harmonic response is in agreement, figure 7.51.

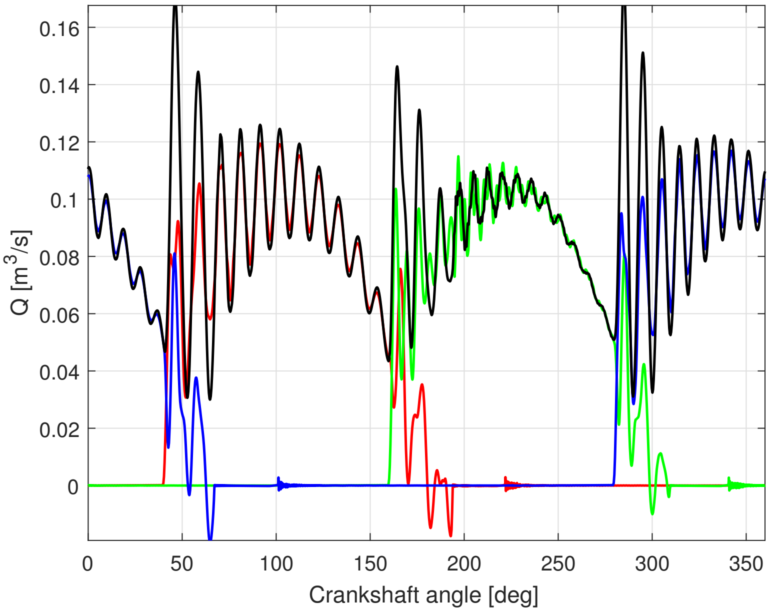


Figure 7.50: Discharge flow rate for the FV developed model for case 8, with discharge pressure at 180 bar, suction at 7 bar and 63 SPM. Blue line is the chamber one, red line is the chamber two, green line is the chamber three and black line is the overall discharged volume flow rate.

The same behaviour reported for the flow rate in case 1, is depicted in figure 7.50. Backflow occurs for a value of 20 crankshaft degrees.

In the case of slow pump speed, the performance of the algorithm is in accord with the experiment, although the hydraulic dampener in the discharge phase is not simulated properly as shown in 7.52.

The overall description of the pump is still in agreement with the experiment. However, differences in the pressure profile are highlighted and are higher than for a one-chamber pump. In this condition, the interaction with the system and the other pumps play an essential role that is not simulated correctly by the algorithm. This drawback is the main difference due to the lack of complete system simulation, although the correlation factor for the

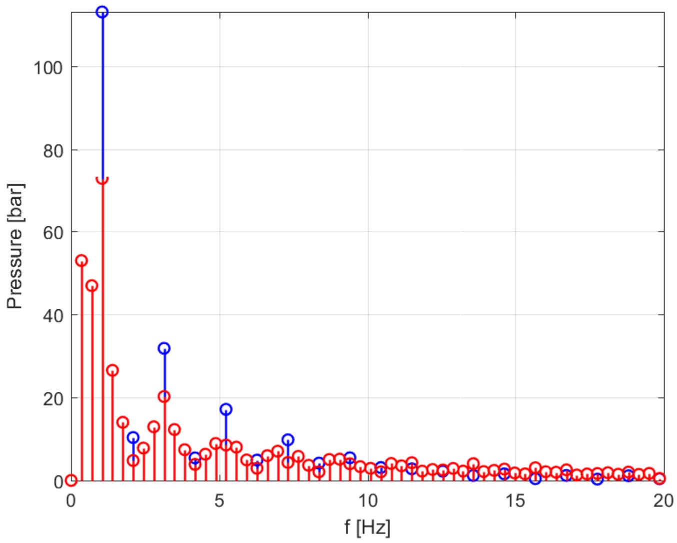


Figure 7.51: Frequency analysis of the chamber pressure for the experimental data, red line and the FV developed model blue line for the case 8, with discharge pressure at 180 bar, suction at 7 bar and 63 SPM.

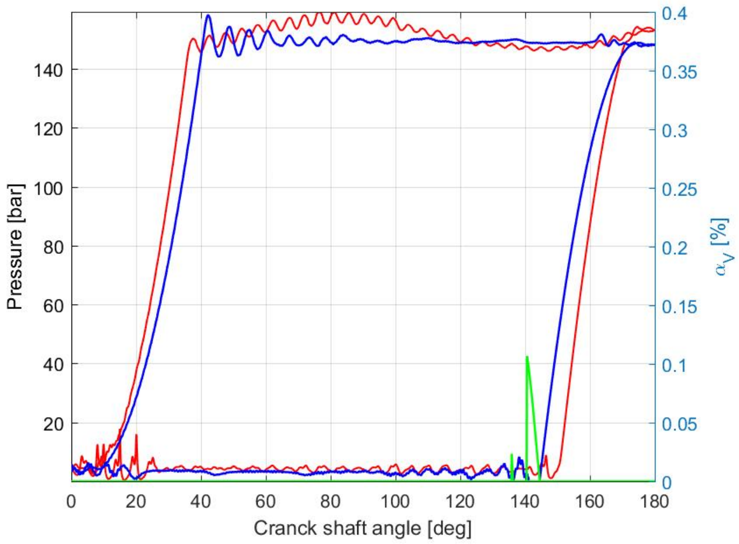


Figure 7.52: Chamber pressure comparison of the experimental data, red line and the FV developed model blue line for the case 5, with discharge pressure at 148 bar, suction at 4 bar and 40 SPM. The green line is the vapour volume fraction for the developed FV model.

pressure profile is higher than 0.99 for all the simulations. In addition, in the flow rate profile, the algorithm shows a back flow which is not described in the LPM.

7.3.6 Pump Network effects

To consider how the FV algorithm developed by the author responds in a network environment of two, three chamber pump interaction analysis was performed. Each pump has three chambers with the same characteristics as the TZPM2000. The configurations analysed are:

- symmetric in layout, where the network is a mirrored layout, namely the pumps have precisely the same characteristic, and
- an asymmetric pump layout, where the second pump has a longer suction and discharge line of one meter.

A picture of the layout of the network is reported in figure 7.53 where reference pump for the simulation is reported as a left pump. While the second pump is refered as the right pump. The simulation conditions consist of a suction pipeline with a constant inlet pressure of 8 bar, while the discharge condition is set to 60 bar, and the pump speed is set to 60 SPM. No accumulators are simulated for this comparison. The idea behind this approach is to evaluate the algorithm's potential and understand if there is a specific pump configuration that provides better network performance. The idea is not new; Weir Group owns the “shift control“ patent to avoid beat pulsation in the discharge pipeline. As described in the Weir Minerals report[2] the phase shift controller makes all the pumps communicate with a PLC virtual pump machine. The strategy compares the pump behaviour with a numerical LPM model used as a virtual master pump controller. With this function on, the pumps in the system follows the same performance with feedback control, mitigating or cancelling the different frequency response at the root of the beat effect.

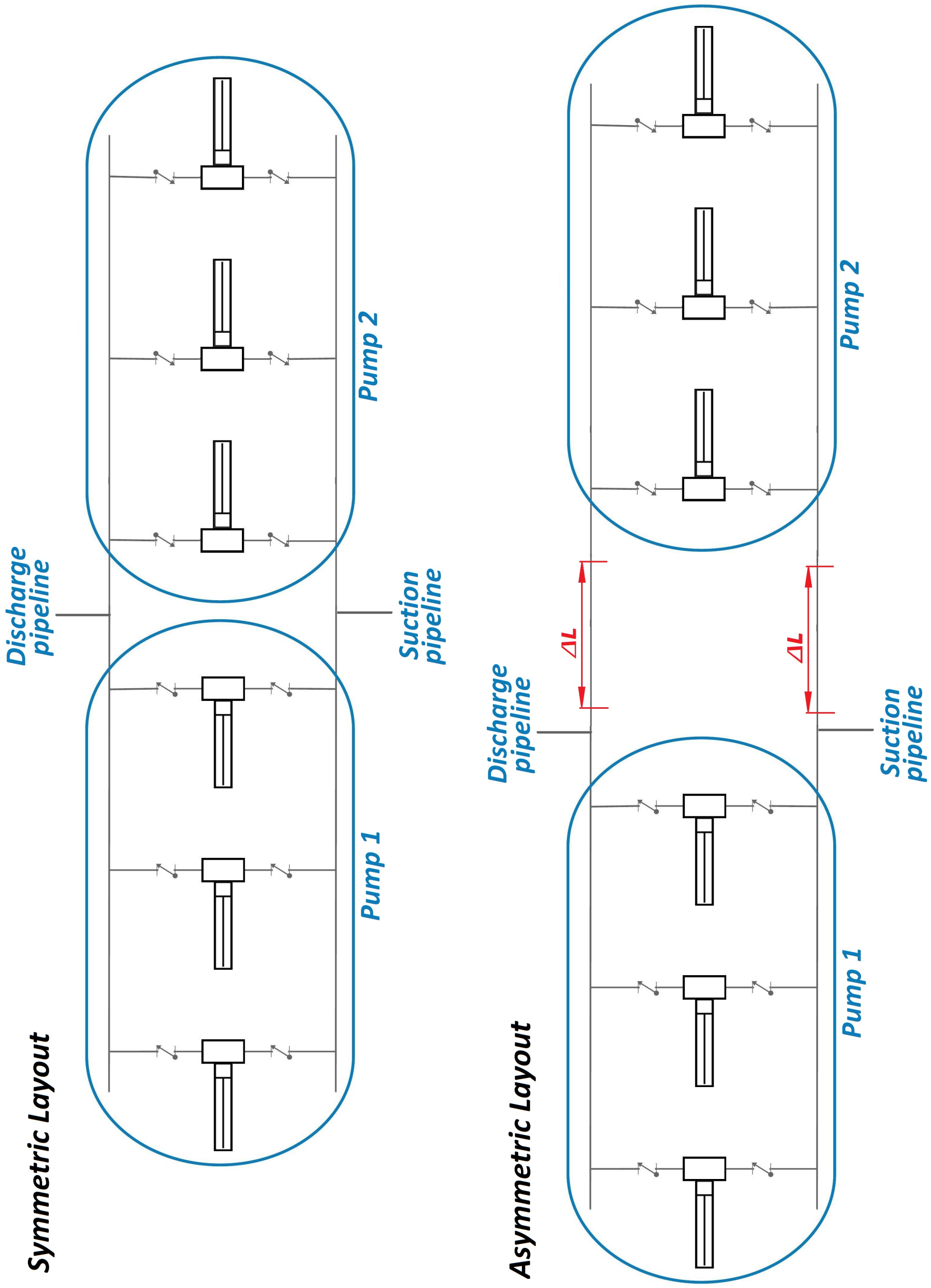


Figure 7.53: Scheme of the two network layouts.

The process described by Weir Minerals, however, presents the following limitations:

- it is valid only for the low-frequency region up to the 9^{th} harmonic.
- The pump is modelled as an ideal incompressible flow.
- No acoustic reflections are included.
- No interaction between the individual pumps.
- No interaction between the connected systems.

Therefore as a drawback, the model is not able to capture high-frequency pressure pulsations that appear when the phase shift control is activated[2]. Other companies apply a different system; for instance, the master pump is a real pump. This procedure creates a problem in the case of master pump failure. Therefore the Weir strategy is the most rubust to be taken into consideration due to the numerical approach. Cavitation is not easy to avoid and the solution to reduce its formation is simple:

- increasing the suction pressure, and
- decrease the pump speed.

Both strategies are not convenient from a customer's point of view. In a world where high productivity is required the reduction in production can be fatal for the market. Therefore, currently, optimization and re-design strategies are taking place to increase pump efficiency. In this case, the idea is to mitigate cavitation and pressure pulsation using the pump system itself.

The idea behind the symmetric layout is to evaluate the algorithm in a complex system where the analytical solution is known. Therefore, it is expected to exhibit a higher pressure fluctuation and cavitation formation

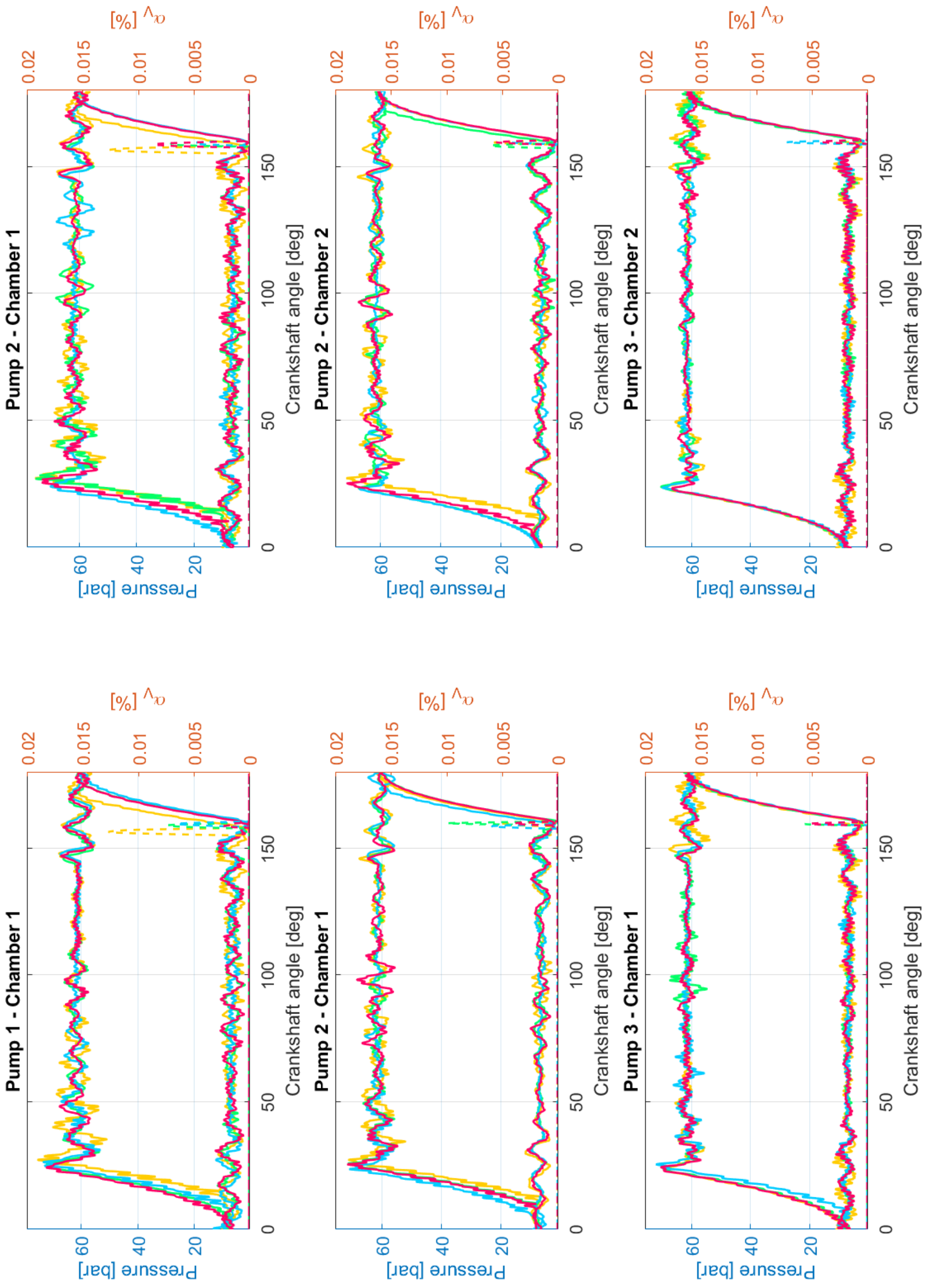


Figure 7.54: Comparison of the pressure history in the chamber and the cavitation (dash line) formation for different crankangle phase shift in symmetric layout. Yellow line is the 0 degree, green line is the 60 degree, blue line is the 90 degree, and red line is the 180 degree.

when there is the same phase angle between the two pumps. In this condition, the pumps are working in phase and sucking and discharging at the same time. To reduce the pulsation, the two pumps should be in anti-phase.

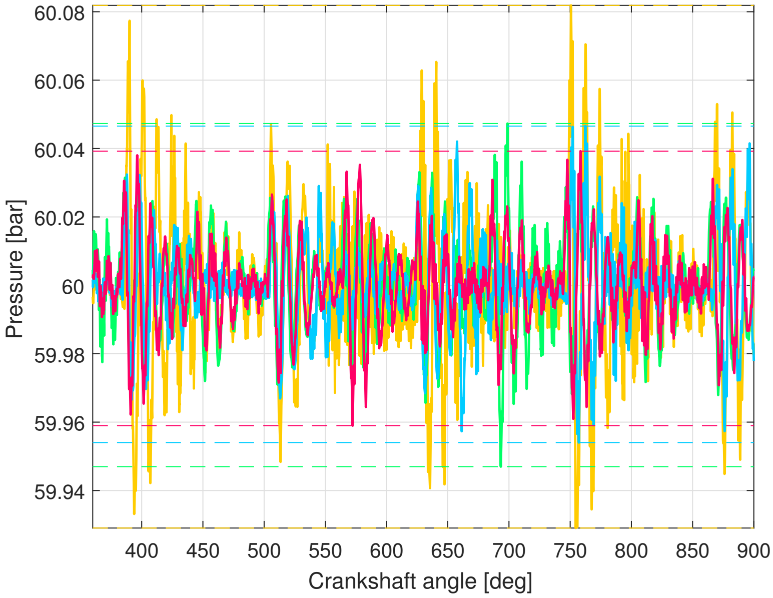


Figure 7.55: Comparison of the pressure history in the discharge line, where Yellow line is the 0 degree pump shift, green line is the 60 degree pump shift, blue line is the 90 degree pump shift, and red line is the 180 degree pump shift.

Figure 7.54 shows the pressure history and the cavitation formation for shift angles of 0, 60, 90, 180 crankshaft degree of difference. The graphs are overlaid to highlight the differences, although the pressure profile for the second pump should be shifted. Differences are highlighted in the pressure profile where the highest fluctuation is shown for the 0 degree shift, as expected. Cavitation formation is also shown in figure 7.54 (dash lines), where once more the 0 degree shift shows its drawback. In figure 7.55 the pressure pulsation at the far end of the discharge line is shown where the mean pressure was set to 60 bar. In the same graph, the boundary values are reported. As expected, the higher pressure pulsation is simulated for a in-phase pump layout.

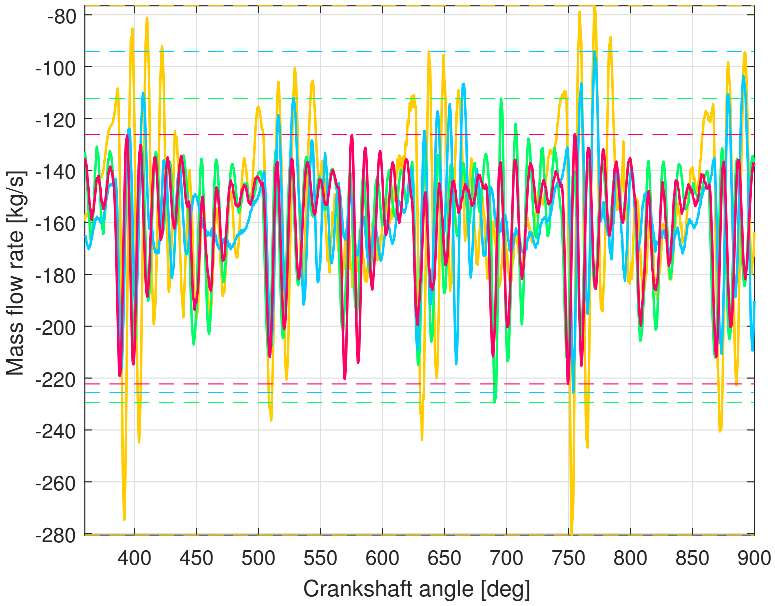


Figure 7.56: Comparison of the mass flow rate history in the discharge manifold for symmetric layout. Yellow line is the 0 degree pump shift, green line is the 60 degree pump shift, blue line is the 90 degree pump shift, and red line is the 180 degree pump shift.

The differences in the pressure values can reach 50% more for the in-phase condition compared with the anti-phase layout. The differences are also evident in the mass flow rate depicted in figure 7.56. Improving the pressure pulsation response improve at the same time the flow rate fluctuation inside the system. An interesting result can be seen in the frequency analysis of the discharged mass flow rate in figure 7.57 and in the pressure response in figure 7.58. The anti-phase configuration annihilates the multiple of the third harmonics presenting only the multiple of the 6^{th} harmonic. Differently, the configurations of in-phase condition (no phase shift), quadrature (90 degrees of shift) and 60 degrees of phase shift, present multiples of the third harmonics. Therefore, the anti-phase condition is more suitable for reducing the pressure pulsation in the discharge pipeline.

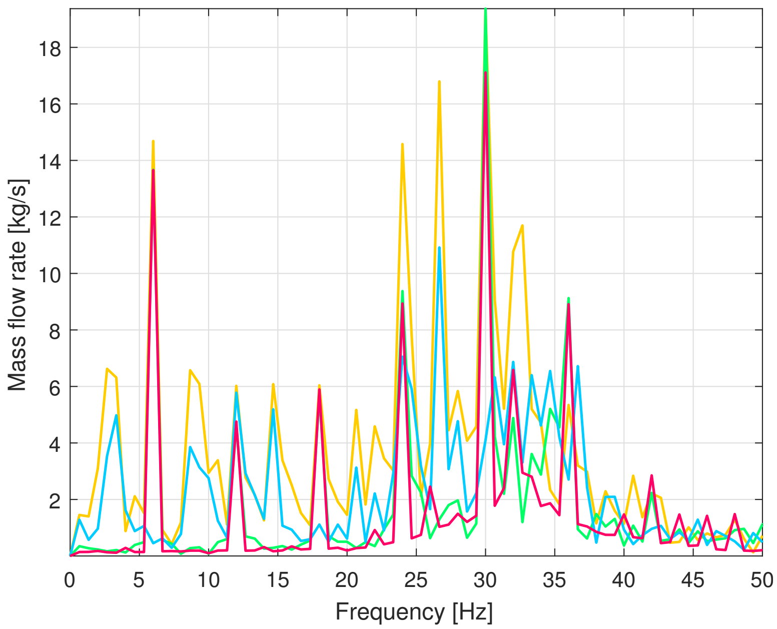


Figure 7.57: Comparison of the frequency spectrum in the discharge manifold for symmetric layout. Yellow line is the 0 degree pump shift, green line is the 60 degreepump shift, blue line is the 90 degree pump shift, and red line is the 180 degree pump shift.

Similar behaviour to that observed in the discharge manifold is also shown in the suction manifold. The reduction in the pressure pulsation, shown in figure 7.59, can give a general improvement for a cavitating condition but may not be the best way. In fact, the reduction in the pressure fluctuation creates a more constant value at the suction valve thereby avoiding the risk of a low-pressure condition. However, further improvement could be theoretically made by creating a higher pressure condition at the instant of valve opening.

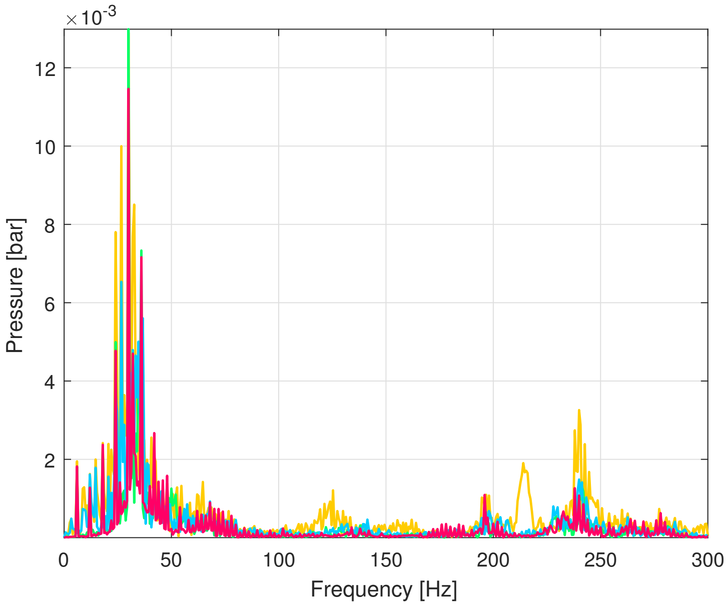


Figure 7.58: Comparison of the frequency spectrum anlysis for pressure in the discharge manifold for symmetric layout. Yellow line is the 0 degree pump shift, green line is the 60 degree pump shift, blue line is the 90 degree pump shift, and red line is the 180 degree pump shift.

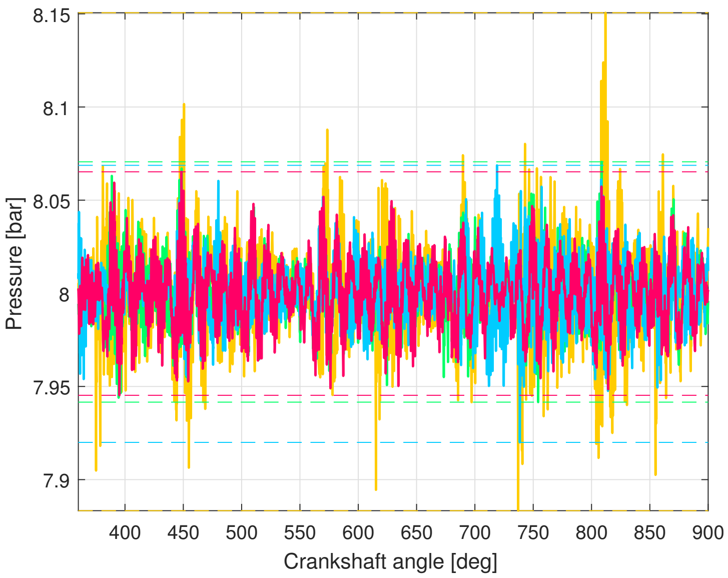


Figure 7.59: Comparison of the the pressure history in the suction manifold for symmetric layout. Yellow line is the 0 degree pump shift, green line is the 60 degree pump shift, blue line is the 90 degree pump shift, and red line is the 180 degree pump shift.

The reduction in the harmonics for the anti-phase layout is shown for the pressure spectrum, in figure 7.60 and the mass flow rate in figure 7.61. The algorithm seems to predict the variation in the frequency up to 250 Hz. This result is important when compared with the limitations given by the LPM. A summary of the pressure and the mass flow variation for both suction and discharge line are given in figure 7.62. All the parameters have the minimum value around 180 degrees. However, in figure 7.63 the variation in the cavitation formation is shown for each pump in relation to the phase shift value. The result shows that 180 degrees is not the best value for reducing cavitation, although it is one of the best results. In fact, a phase shift of 60 degrees seems to be the best option for reducing cavitation formation. This means that cavitation is reduced when the pump phase shift has the same variation for each chambers.

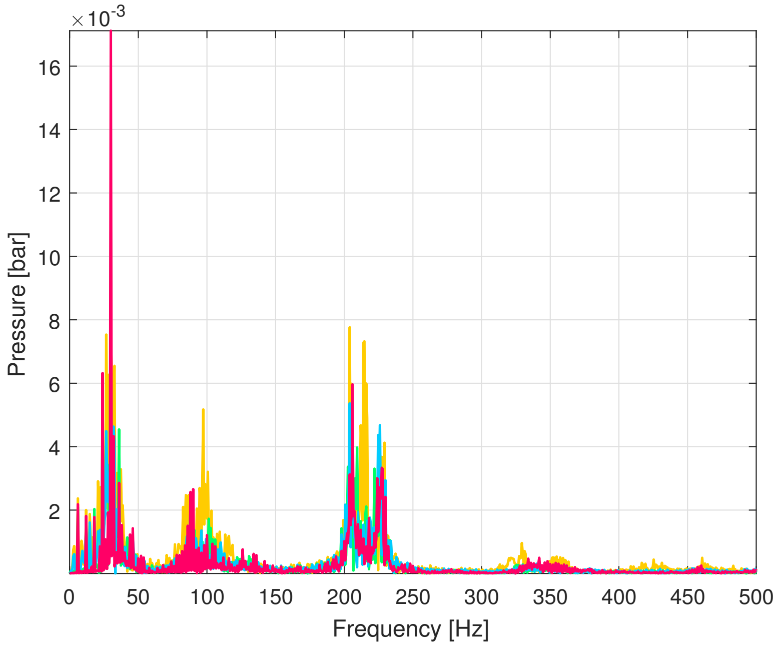


Figure 7.60: Comparison of the frequency spectrum in the suction manifold for symmetric layout. Yellow line is the 0 degree pump shift, green line is the 60 degree pump shift, blue line is the 90 degree pump shift, and red line is the 180 degree pump shift.

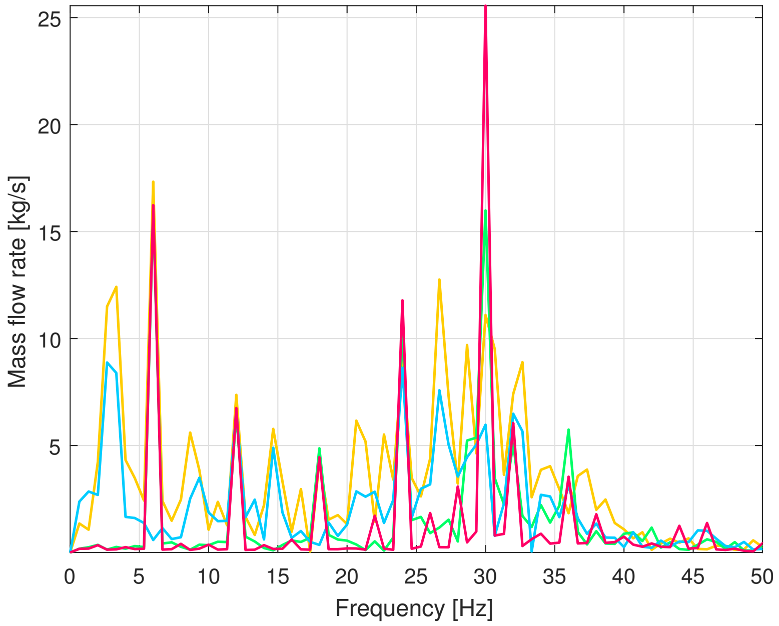


Figure 7.61: Comparison of the frequency spectrum in the suction manifold for symmetric layout. Yellow line is the 0 degree pump shift, green line is the 60 degree pump shift, blue line is the 90 degree pump shift, and red line is the 180 degree pump shift.

The symmetric configuration had the scope to evaluate and validate the algorithm for a known behaviour. In a real system configuration, it is difficult to design the network as a symmetric layout. The disadvantage of an asymmetric layout is "virtual shift" caused by the distance between the pumps. The same simulations performed for the symmetric condition were conducted for the asymmetric condition. To create the asymmetry, the right pump had an additional distance of one meter from the suction and discharge line. The analysis is given only for the 0 and 180 degree phase shift.

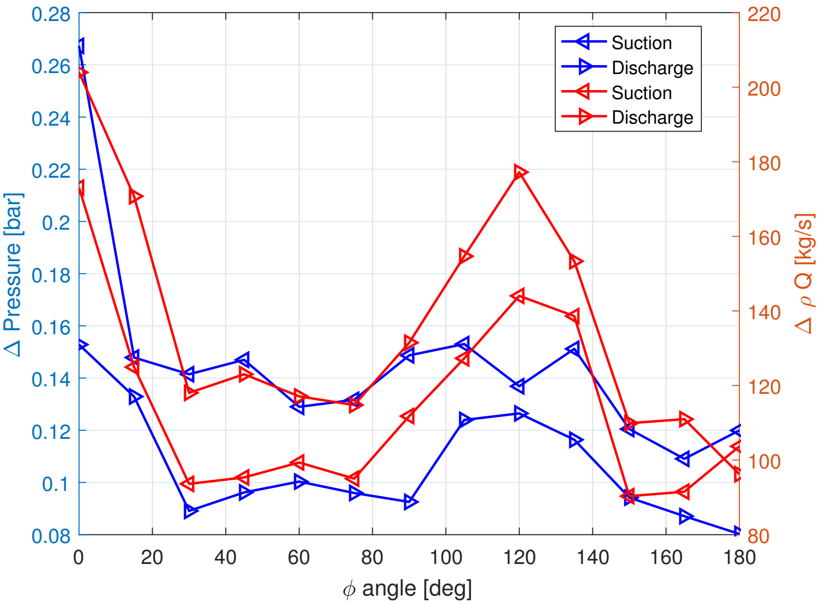


Figure 7.62: Variation of the bounded inlet and discharged pressure in blue, and mass flow rate in red.

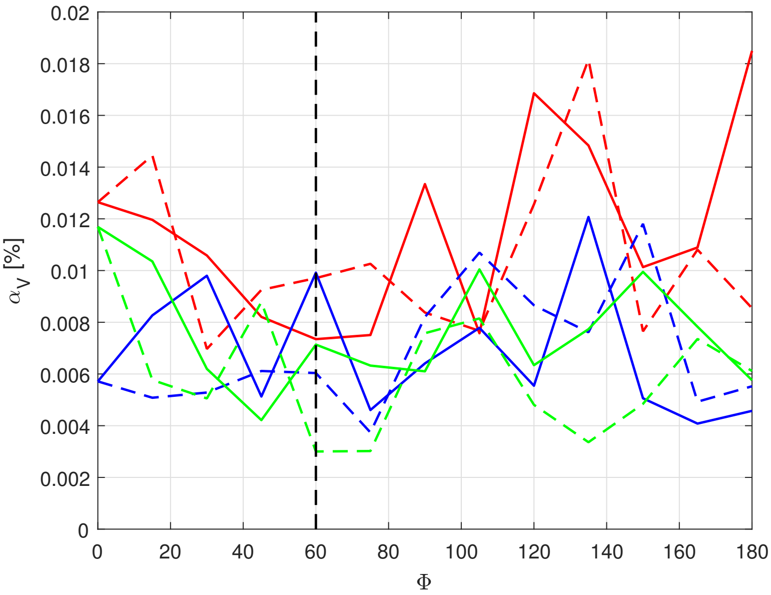


Figure 7.63: Variation of the maximum cavity formation in relation on the shift angle. The dash line are related to the second pump and red line is the chamber three, green line is the chamber two and blue line is the chamber one.

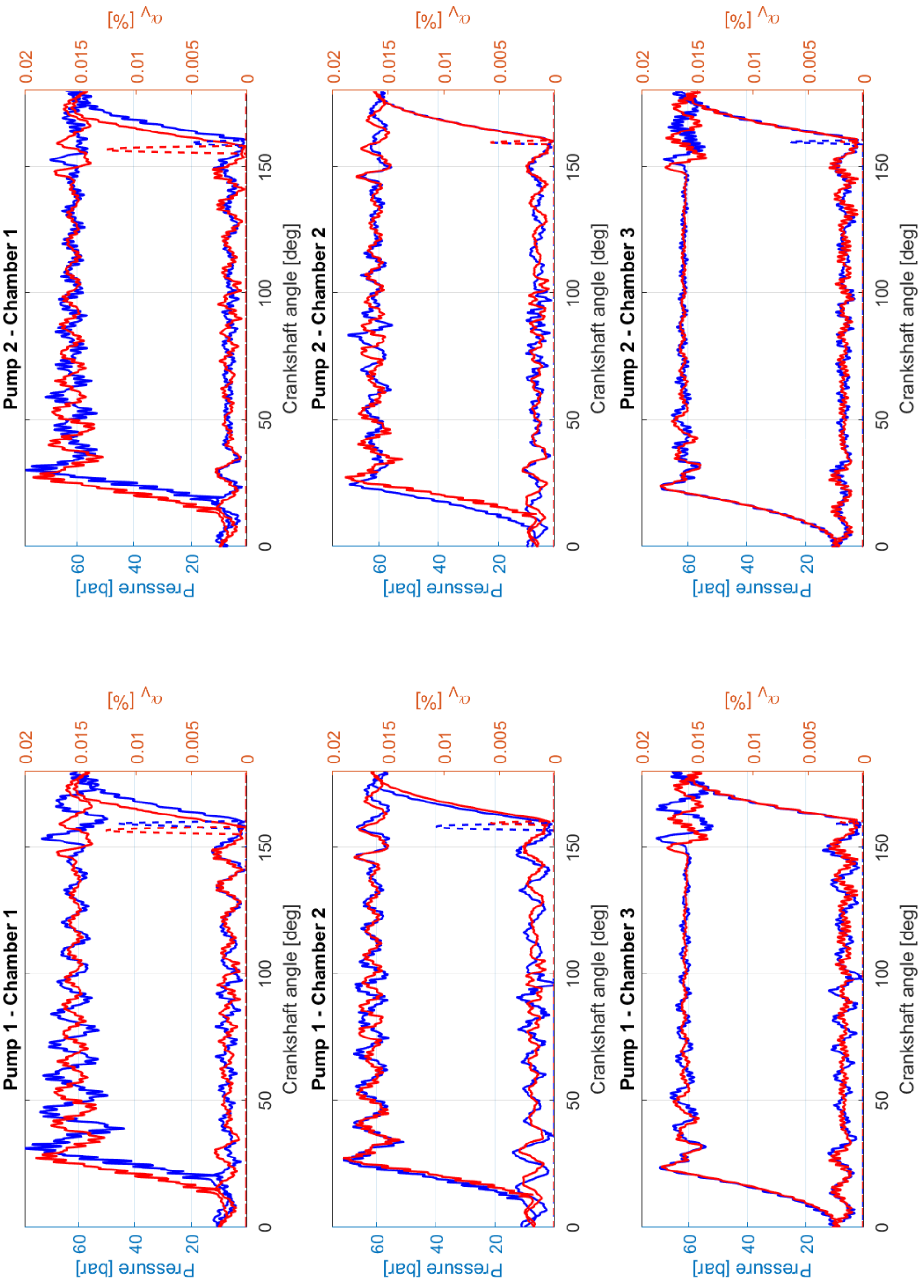


Figure 7.64: Comparison of the pressure history in the chamber and the cavity formation for 0 degree shift of symmetric layout red line and different pipe length blue line.

Figure 7.64 shows the comparison between the chambers response for the symmetric and asymmetric layout. The response is difference in terms of cavitation and pressure pulsation. The symmetric layout performed with less pressure fluctuation in all pump chambers.

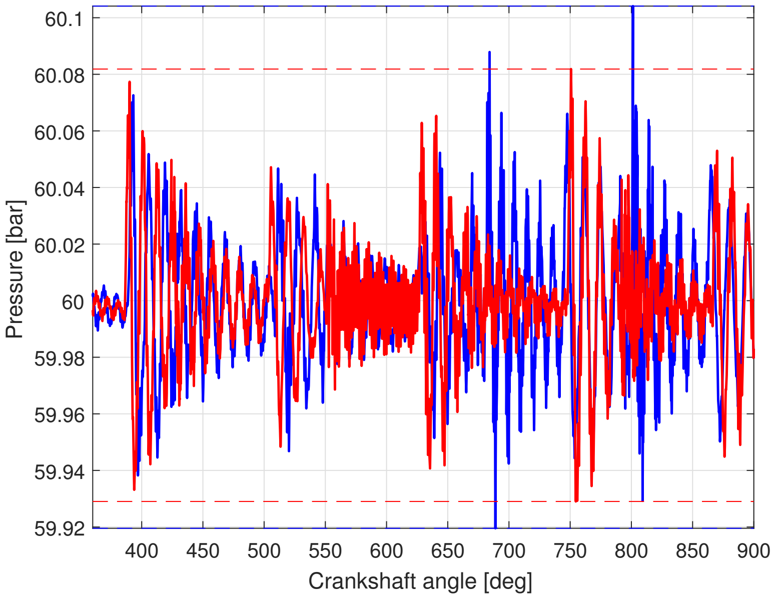


Figure 7.65: Comparison of the pressure history in the discharge manifold for 0 degree shift of symmetric layout red line and different pipe length blue line.

The same result is presented for the discharge manifold pressure in figure 7.65 and the mass flow rate, in figure 7.66.

The reason for a higher pressure pulsation in the discharge manifold is explained in the frequency analysis, shown in figure 7.67. The asymmetric layout presents high energy content in the same harmonics. A different behaviour is displayed in the suction manifold. In this case the symmetric layout produce a greater fluctuation in the pressure, figure 7.68, and in mass flow rate, figure 7.69.

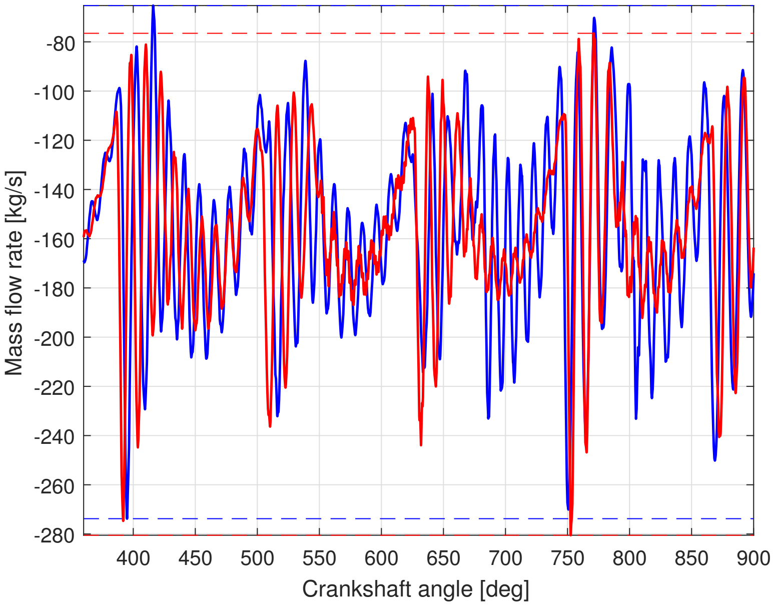


Figure 7.66: Comparison of the mass flow rate history in the discharge manifold for 0 degree shift of symmetric layout red line and different pipe length blue line

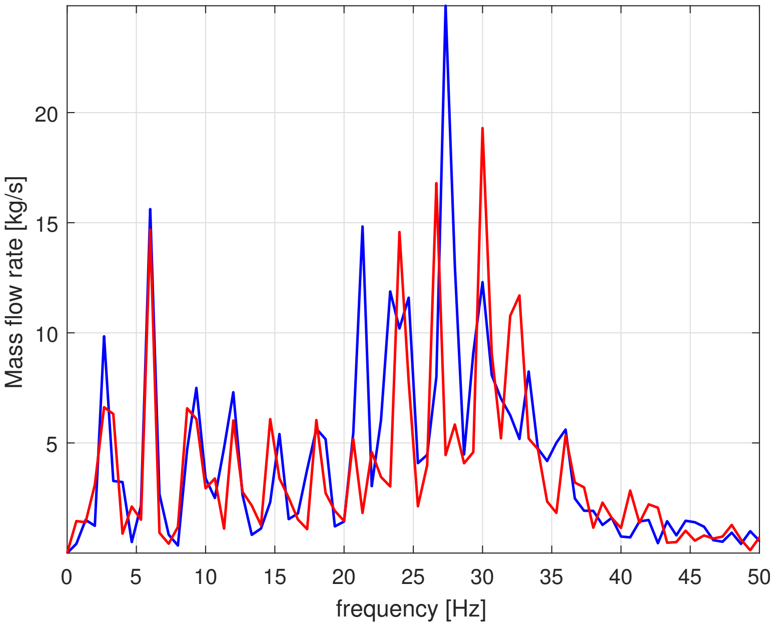


Figure 7.67: Comparison of the frequency analysis for the mass flow rate in the discharge manifold for 0 degree shift of symmetric layout red line and different pipe length blue line.

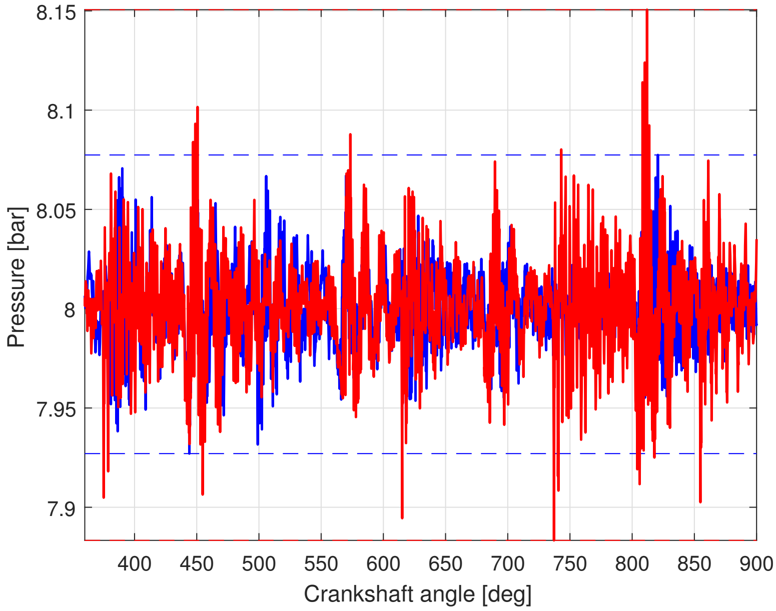


Figure 7.68: Comparison of the pressure history in the suction manifold for 0 degree shift of symmetric layout red line and different pipe length blue line.

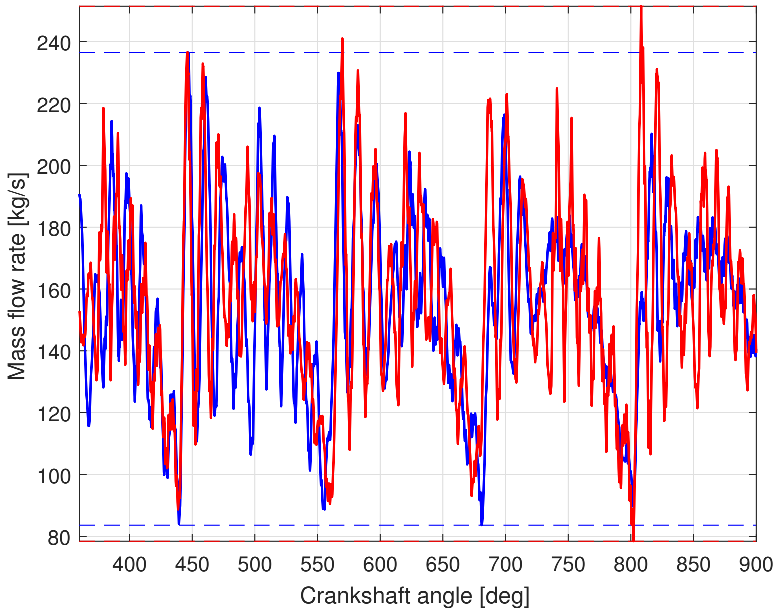


Figure 7.69: Comparison of the mass flow rate history in the suction manifold for 0 degree shift of symmetric layout different red line and different pipe length blue line.

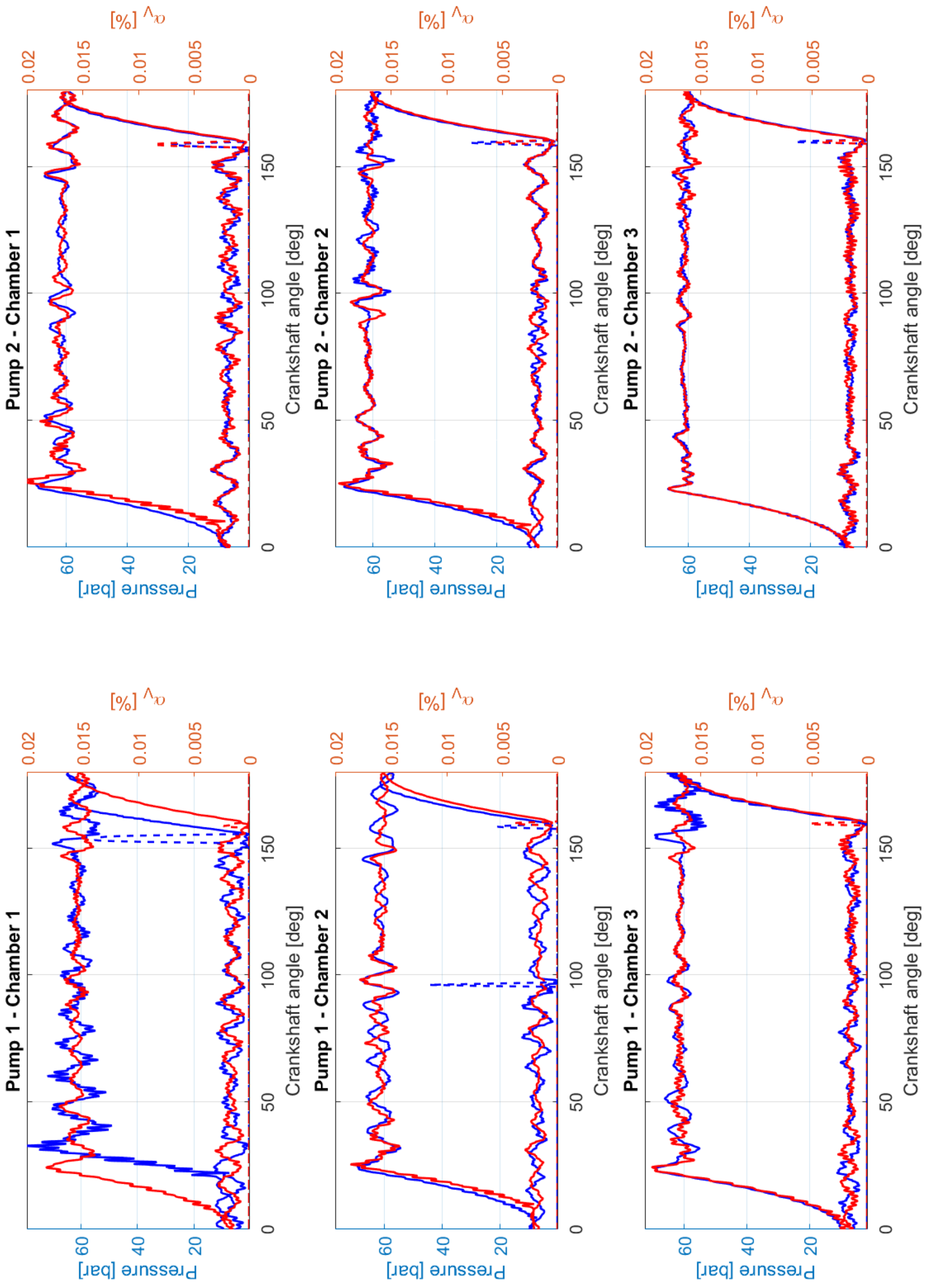


Figure 7.70: Comparison of the pressure history in the chamber and the cavity formation for 180 degree shift of symmetric layout red line and different pipe length blue line.

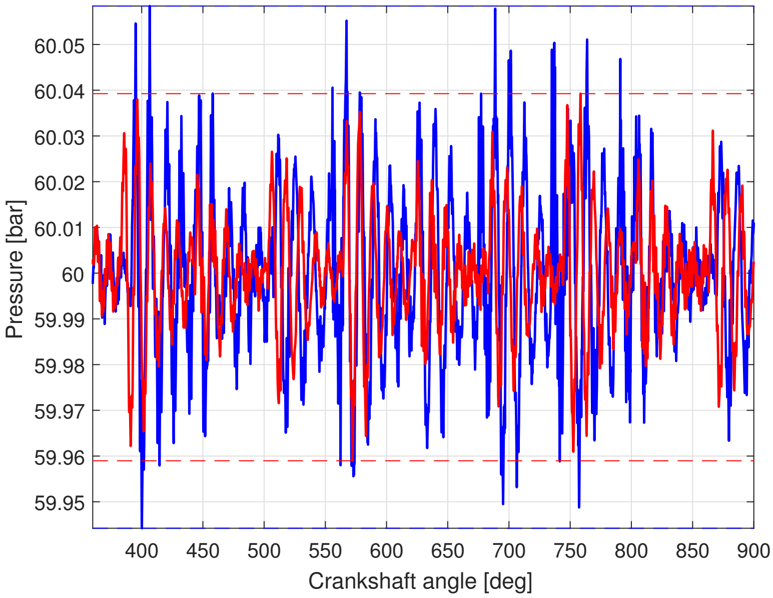


Figure 7.71: Comparison of the pressure history in the discharge manifold for 180 degree shift of symmetric layout red line and different pipe length blue line.

The energy content in the suction frequency shows similar behaviour for the first harmonics. However, the symmetric configuration has higher energy for frequencies above 15 Hz. Even the cavitation formation is different depending on the chamber response and can vary as shown in figure 7.64. In the case of 180 degree phase shift, the behaviour of the pressure and the mass flow is even more evident. Pressure pulsation and mass flow rate in the discharge are shown in figure 7.71 and 7.72 respectively and show the importance of the symmetric layout. Higher pressure pulsations for both suction and discharge manifold are shown, figure 7.71 and figure 7.74. The same trend is shown for the mass flow rate indicated in figure 7.72 and figure 7.75.

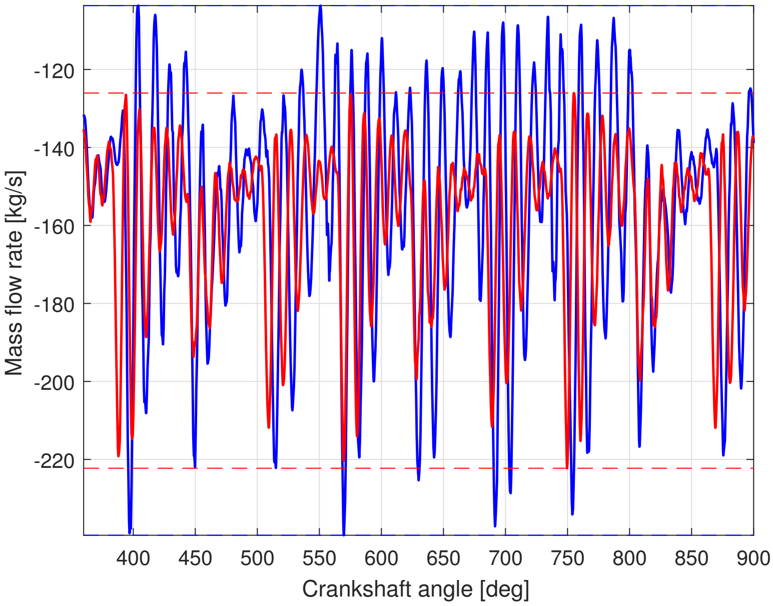


Figure 7.72: Comparison of the mass flow rate history in the discharge manifold for 180 degree shift of symmetric layout red line and different pipe length blue line.

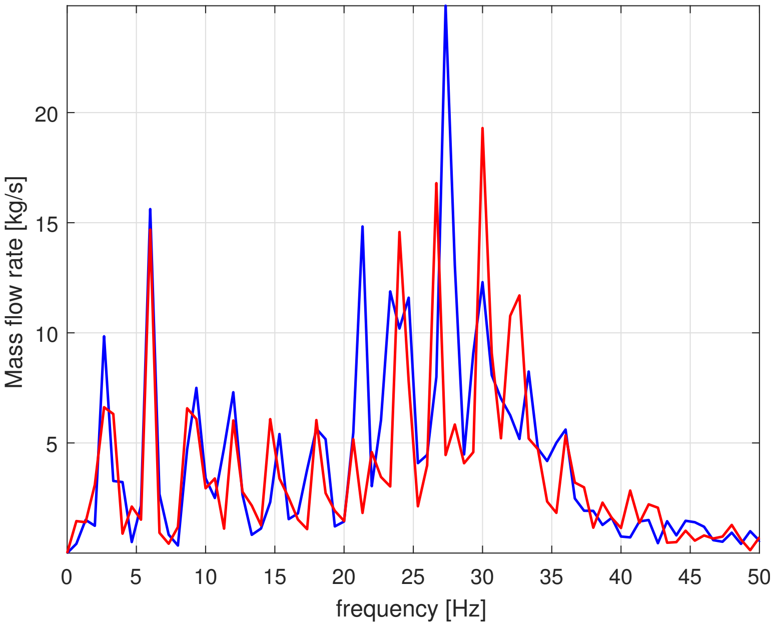


Figure 7.73: Comparison of the frequency analysis for the mass flow rate in the discharge manifold for 0 degree shift of symmetric layout red line and different pipe length blue line.

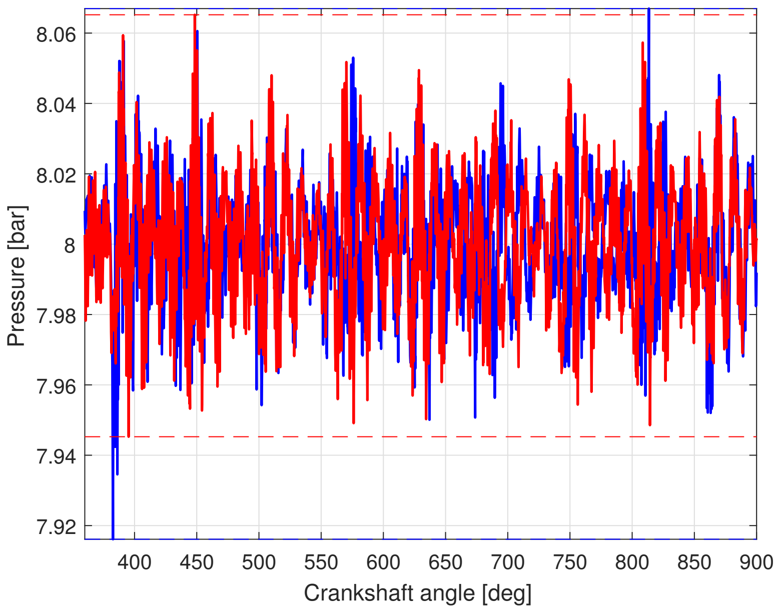


Figure 7.74: Comparison of the pressure history in the suction manifold for 180 degree shift of symmetric layout red line and different pipe length blue line.

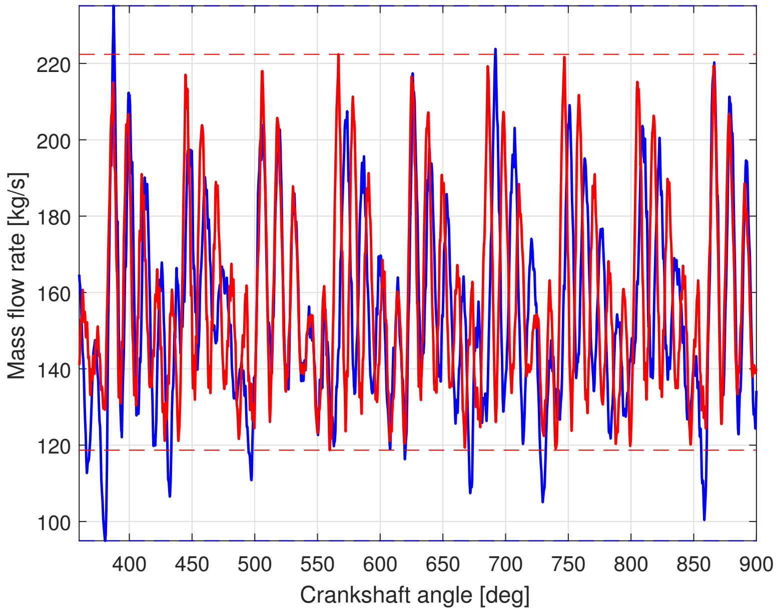


Figure 7.75: Comparison of the Mass flow rate for the pressure in the suction manifold for 180 degree shift of symmetric layout red line and different pipe length blue line.

The asymmetric condition show the multiple of the third harmonic that are not present in the symmetric condition. In fact, the frequency analysis in figure 7.76 shows a completly different energy content in the spectrum.

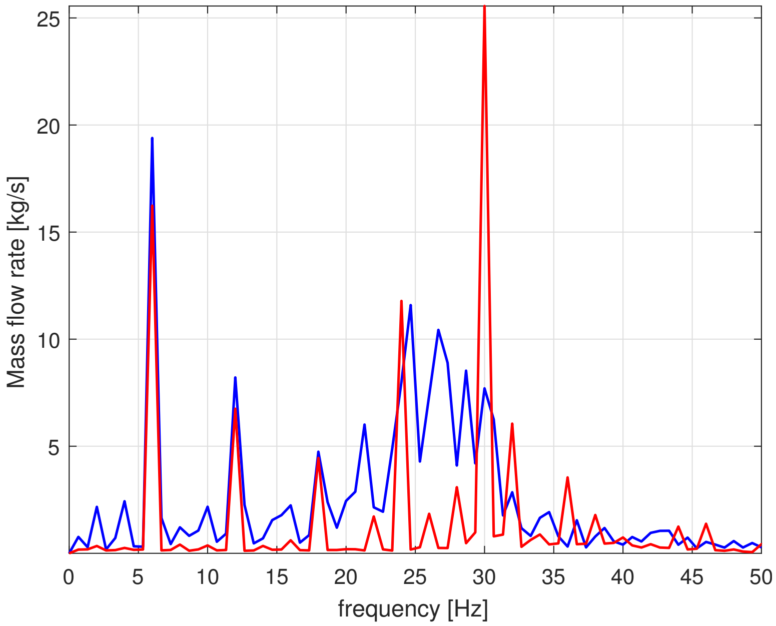


Figure 7.76: Comparison of the frequency analysis for the pressure in the suction manifold for 180 degree shift of symmetric layout red line and different pipe length blue line.

The analysis of the cavitation formation shows a completely different behaviour than the symmetric configuration. Figure 7.77 shows a shift in the optimal cavitation performance at 40 degrees. This result considers the additional time for the wave to interact with the pumps. Although these studies were performed to evaluate the potential of the algorithm, the results are in line with theoretical result.

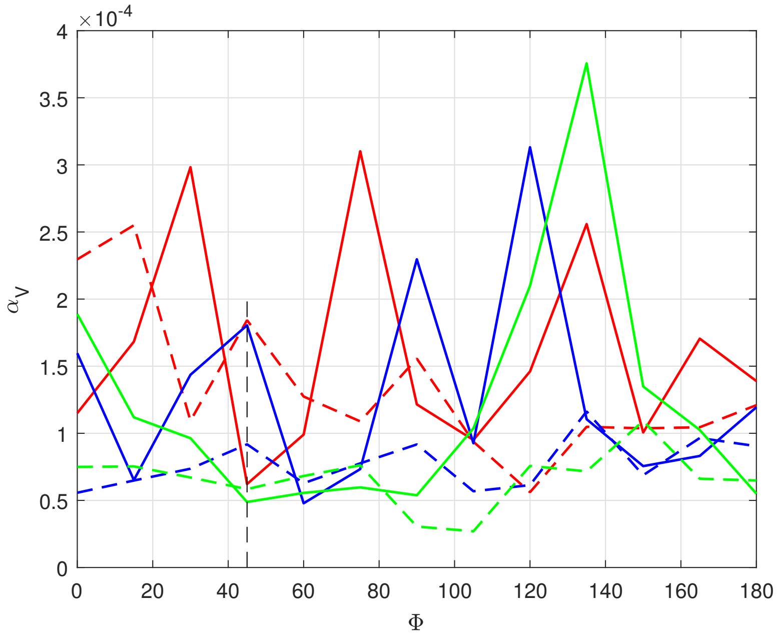


Figure 7.77: Variation of the maximum cavity formation in relation on the shift angle. The dash line are related to the second pump and red line is the chamber three, green line is the chamber two and blue line is the chamber one.

Therefore, this research has permitted the evaluation of the hypothetical, optimal pump phase shift condition for the two network layouts with two pumps interacting together. This approach highlights the possibility to create a virtual boost and reduce the cavitation formation. The main achievement of this simulation is the application of the algorithm to a complex network configuration. In addition, extrapolation from this work can also be carried out for diaphragm pump design. A three-chamber pump layout has commonly different pre-chamber pipe lengths as shown in figure 7.78. The outer pre-chamber pipe is longer than the central. This discrepancy creates a virtual phase shift between the chambers in the same pump. Typically the three chamber pump has a 120 degree of shift phase angle between each piston. As is shown, it is possible that this configuration is not the best solution but requires small adjustment. Therefore the algorithms can be used to study and optimize the pump performance with the best phase shift angle between the pistons in the same pump. However these considerations must face the structural issues that arise due to possible vibration to the crankshaft.

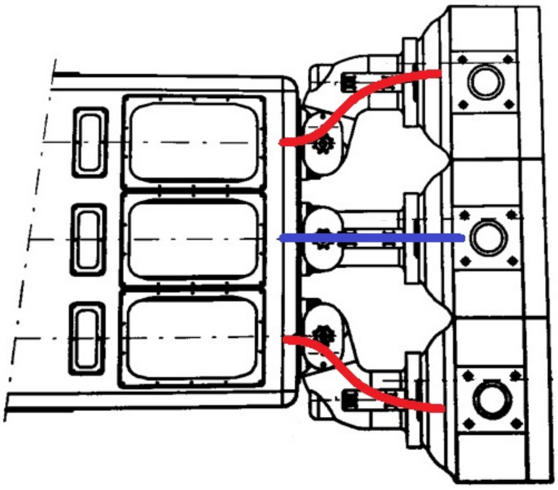


Figure 7.78: Blue print of the diaphragm pump.

7.3.7 Design Rules

In this section, another application of this algorithm is performed. The cavitation pump map behaviour is developed. Using non-dimensional analysis[1], the cavitation and crankshaft angle can be related to the following parameters:

- characteristics of the fluid, principally ρ density,
- the fluid velocity,
- suction pressure.

Considering the formulation of the flow rate in the lumped parameter model, described in section 4.6 and equal to $\dot{Q} = \frac{A}{\rho l} p$, is possible to highlight that:

- The time derivative of the volume flow rate is a function of the derivative of the fluid velocity multiplied by the area, namely $\dot{Q} = \dot{\boldsymbol{u}}A$. Moreover, the fluid velocity is proportional to the piston velocity and the stroke rate $\dot{\boldsymbol{u}} = \ddot{x_p} \approx r\omega^2$.
- the pipe area is negligible.

From these assumptions a non dimensional formulation can be written in the form of equation 7.4. More details in appendix B.

$$\frac{p}{SPM^2 \rho l} = 1 \tag{7.4}$$

All the experiments and the simulation with different suction pressure, density and speed can be compared using equation 7.4. The cavitation value is calculated at a crankshaft angle θ_{Cav}. The value identified the crankshaft rotation where cavitation occurs. This treatment permits the identification of a correlation for the cavitaton experiments. In figure 7.79, 7.80 and 7.81 three sets of data are shown for each chamber. The first is the experimental cavitation data (square points). The second is the simulation data with the finite volume algorithm (the cross point) for each chamber. The third set of data is given by the Opitz algorithm. The last algorithm is subdivided into, black circles calculated with a guessed opening valve angle of 15 degrees and the cyan circles with the correct valve delay opening time for each chamber given by the FV developed code. A power regression method of the form $y = Ax^B$ is used to interpolate the points for the treatment using a logarithmic scale. Analysing the Opitz algorithm, its limitations are highlighted. Although the Opitz's method is compelling, the guessed value of the valve opening requires good pump behaviour knowledge. In addition, the three chambers behave in the same way although the suction manifold layout has a different length for each chamber. The finite volume method instead, presents a response that is in a good agreement with the experimental data. Only when the valve dynamics of the finite volume simulation are used in the Opitz model the results of the Opitz method improve. Therefore, the developed algorithm permits the evaluation of the pump cavitation behaviour in terms of pressure and stroke rate with reasonable confidence. The approach can be used to simulate the pump behaviour, and map the pump cavitation characteristics.

Table 7.11 reports the correlation factor between the experimental data and the conducted simulation. The developed FV algorithm is much more accurate that the Opitz methodology for calculating the cavitation extent, with the exception of chamber two where the FV algorithm is slightly lower than the corrected Opitz model. This section has shown the potential of the code to be employed for cavitation map design. Figure 7.79, 7.80 and 7.81 describe the cavitation behaviour for different condition: suction pressure, SPM and suction pipe length, in one single graph for each chambers.

Table 7.11: Correlation factor between the power regression interpolation and the experimental data.

	FV developed model	**Opitz**	**Opitz Corrected**
Chamber 1	0.9998	0.9981	0.9969
Chamber 2	0.9876	0.997	0.9981
Chamber 3	0.999	0.983	0.9974

7.3.8 NPSH

The Net Positive Suction Head, NPSH, is an important parameter which the manufacturing pump company must provide. It is the minimum pressure head value required at the suction for no cavitation to occur. The creation of this information can be expensive due to the number of real tests required. The API regulation restricts the value of NPSH to 3%, $NPSH_3$. This value is referring to the maximum mass flow rate loss allowed during the pump cycle. In theory, this regulation is applied to centrifugal pumps. However

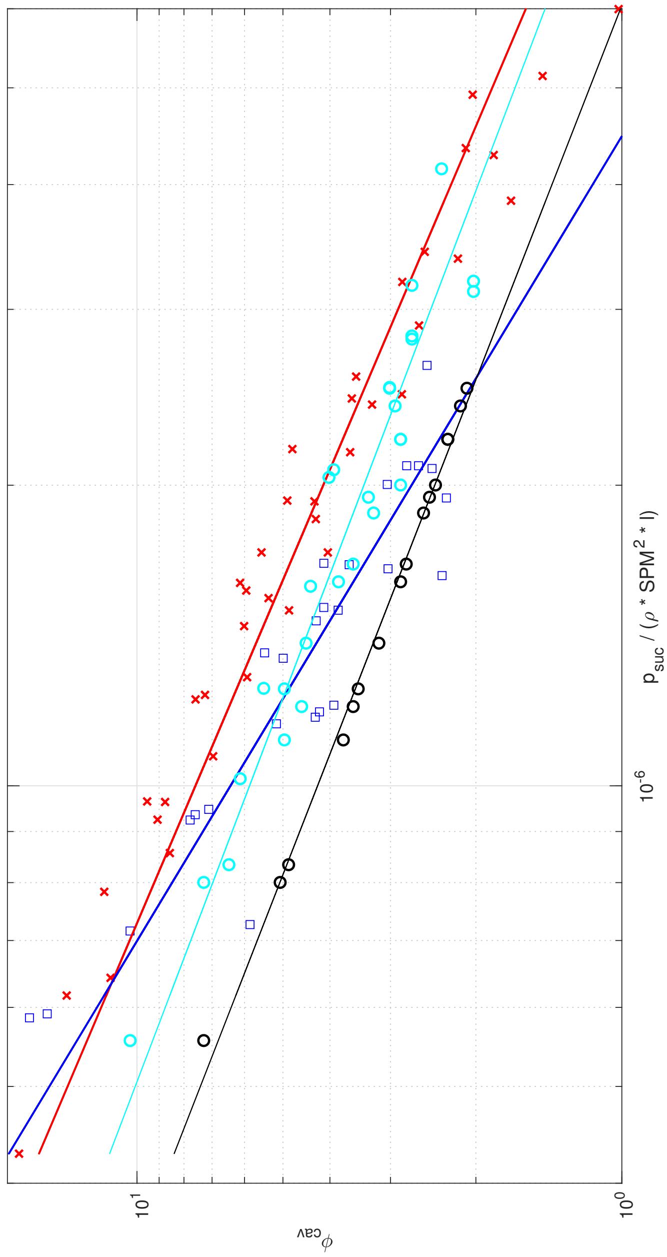


Figure 7.79: Cavitation evaluation in θ_{cav} and non-dimensional value for chamber 1, where the square data and blue color line are the experimental data, the cross data and red line is the simulation data with the developed FV model, the circle data and black line are the value given by the Opitz's model with a opening valve value of 15 degree, and the circle data and cyan line are the value given by the Opitz's model adjusted with the developed FV opening valve delay.

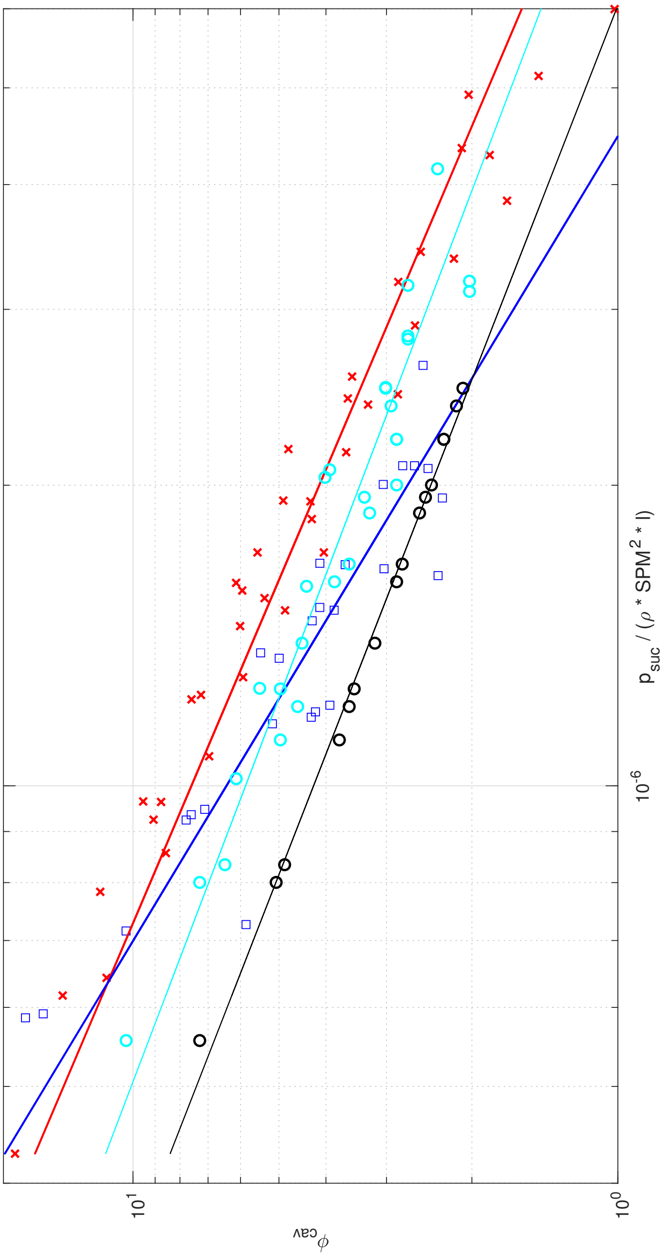


Figure 7.80: Cavitation evaluation in θ_{cav} and non-dimensional value for chamber 2, where the square data and blue color line are the experimental data, the cross data and red line is the simulation data with the developed FV model, the circle data and black line are the value given by the Opitz's model with a opening valve value of 15 degree, and the circle data and cyan line are the value given by the Opitz's model adjusted with the developed FV opening valve delay.

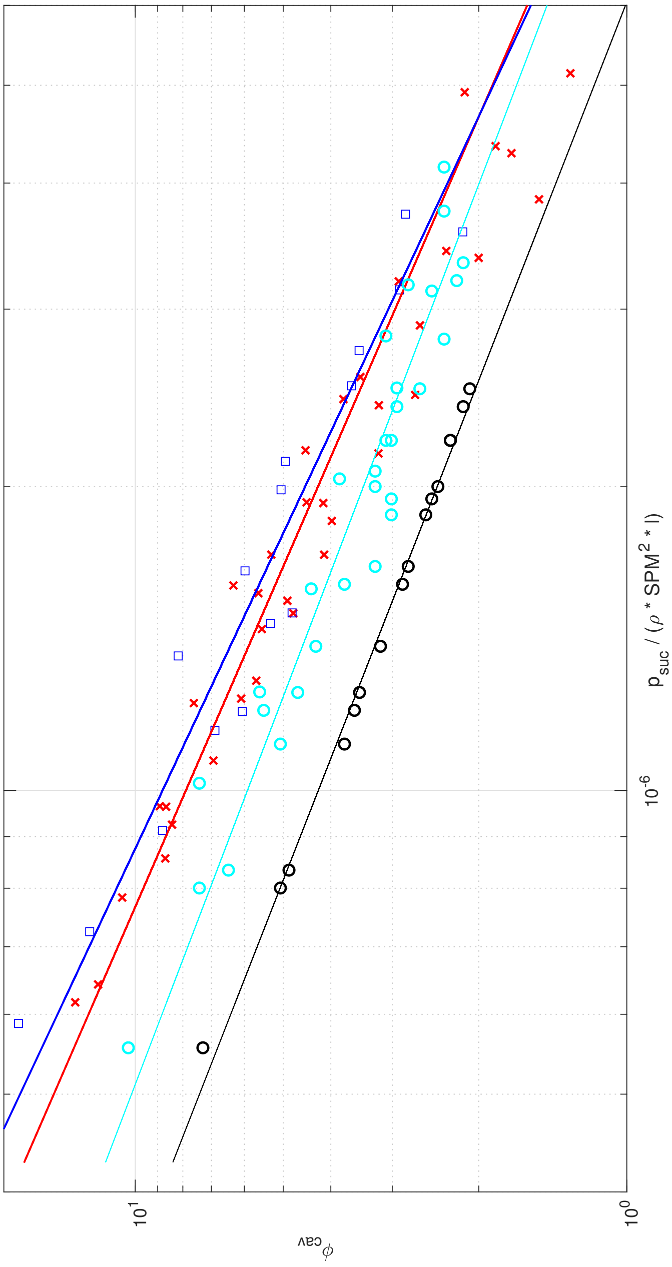


Figure 7.81: Cavitation evaluation in θ_{cav} and non-dimensional value for chamber 3, where the square data and blue color line are the experimental data, the cross data and red line is the simulation data with the developed FV model, the circle data and black line are the value given by the Opitz's model with a opening valve value of 15 degree, and the circle data and cyan line are the value given by the Opitz's model adjusted with the developed FV opening valve delay.

the same regulation applied to PD pumps did not allow cavitation formation in all working cases. As it is shown, it is impossible to satisfy this condition, and companies use the parameter $NPSH_3$ also for PD pumps. To map the behaviour of a pump the company must test the pump with different speeds and slowly decrease the suction pressure until the $NPSH_3$ is achieved. This value is then speed dependent. In addition, the test must be conducted under steady state conditions, namely avoiding any transient effects that occur in the testing system. This means that to obtain a single point in the map pressure head-η_{vol} the testing may take a long time. For the company this is translated into significant effort in terms of cost and time.

Figure 7.82 shows an example of the NPSH calculation carried out using the FV algorithm for the TZPM2000 for 63 SPM. The $NPSH_R$ reported in the graph, 30 m (3 bar) is in agreement with the company specification. The graph shows an interpolation curve and the $NPSH_3$ value calculated from that. The calculation time to determine this value required less than 6 hours. Therefore, the FV algorithms developed by the author can be used to simulated the NPSH condition of the pump. Thereby emulating the factory experimental test and reducing drastically the cost. It is also possible to simulate the real network where the pump is going to be used. This approach should permit the evaluation of the pump performance in the real working environment rather than the company test loop, thereby giving more confidence and reliability. However, more experimental data must be used to validate the algorithms for these circumstances.

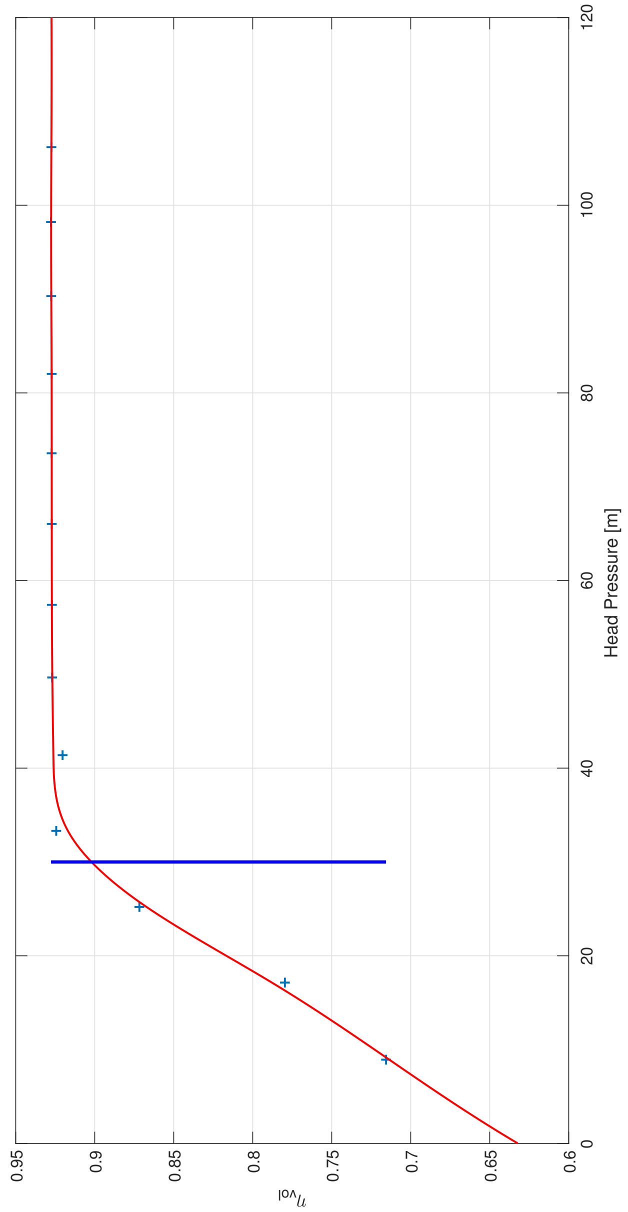


Figure 7.82: NPSH for the TZPM2000 at 63 SPM, where the blue line is the $NPSH_3$ value.

7.4 Summary

In this section validation of the pump model in cavitating conditions was performed. The results have shown good agreement with experiments even with high levels of cavitation. However, with full cavitation the DGCM is not able to capture the pump dynamics. In addition, the limits of the dissipation term were highlighted, especially in the absence of cavitation formation. The performance of the model for one and for multi-chamber pumps was proven, showing good agreement with the experimental data. Furthermore, a parametric analysis was performed for a macro view of the pump response, as well as multi-pump network configurations and a NPSH test.

Chapter 8

Conclusion

This thesis has presented the development of a model code for the dynamic pressure analysis of positive displacement pumps. The available commercial codes are mainly focused on two aspects: a complete three dimensional description or a simple model. Therefore, correct simulations allow a complete fluid description with cavitation, or a simple model restricted to the Lumped parameter method. These extremes have gap has a significant impact on industry. In fact, the complete three dimensional analysis can only be applied to a specific part of the pump, if the required simulation time is to be kept within a reasonable limit. Therefore, to describe the interaction that occurs in a complex system, industry tends to use a simpler and less accurate models. No commercial software to simulate a pump response accurately in a reasonable time is available. Therefore, a new approach for pump simulation was required. The model developed in this thesis focused on a one-dimension transient model with vapour formulation and compressibility. The algorithm is based on the wave transport equation, with advective formulation and a different discretization of the classic discrete gas cavity model. The numerical technique developed, to evaluate the pump performance, was the splitting method that allows the decoupling of the equations into a homogeneous PDE and a standard ODE. This strategy allowed the use of the most suitable methods for the solution of each set of equations. The PDE was solved with the MUSCL scheme, while the ODE was solved with RK4. The MUSCL scheme is second order in time and space with a slope limiter function to produce a TVD method. The splitting strategy is only first order. The solution of the equations was first validated with well-known algorithms from different researchers. The results from the model were in line with, if not better than published data, especially regarding the computational time.

It was shown that the new model performed in an equivalent manner to the MOC for frictionless simulation. In addition, for the Courant condition equal to one, it converged to the analytical solution. However, as soon as friction and dissipation are included in the system, the MOC and other available software showed their limitations. The new algorithm performed better for these issues and showed high versatility and reliability. Cavitation in positive displacement pumps and column separation in straight pipes were simulated and gave good results. Limitations were shown when the amount of the vapour/gas fluid phase was high. Simulation of a single diaphragm pump and a three chamber pump were performed, for both a controlled loop rig and an on-site field network. The response of the algorithm is promising for giving a better understanding of the pump's performance in terms of cavitation, pressure pulsation and flow rate. Once the model had been validated, the focus moved to a parametric study of the pump's performance.

- *Parametric studies* were performed for a one chamber pump where the code was used to analyse the volumetric efficiency and the cavitation formation.

- *Network performance.* The main drawback of the previous algorithms was the inability to model pumps interactions. The use of this algorithm allowed the prediction of the pump behaviour with different pump network configurations. The solution was created in a reasonable time without the loss of accuracy. In addition, use of the new model allowed cavitation and pressure pulsation studies in a complex system to be performed as part of an optimization strategy.

- *NPSH testing.* To evaluate this parameter, a significant amount of experimental testing is required. Data are collected for different pump speed and suction pressure. This methodology is expensive in both time and cost, especially because the performance has to be analysed at steady state conditions. The algorithm developed in this research was able to perform the NPSH analysis in an efficient way. Therefore, this code could be used to provide additional knowledge of NPSH conditions.

- *Pump cavitation map*, thanks to the non-dimensional analysis of the results of the pump simulation, they can be used to create a cavitation map. The vapour formation and collapse time, with respect to crankshaft angle can be related to the slurry properties, the speed of the pump and the suction layout. This method allows a broader view of the pump cavitation behaviour.

Although the experimental data and the simulation were conducted for a diaphragm pump, the methodology can be extended to all PD pump configurations. A variety of suction manifold layouts can be simulated and studied before a more detailed CFD analysis., could be carried out if required. In addition, more accurate power requirement calculations can be performed. Different network characteristics such as pump shut down or leakage can also be examined. Therefore, the new model developed in this thesis has the benefit of cost reduction and an operational understanding of pump behaviour and system response.

8.1 Future work

The quality of the algorithm allows the evaluation of a wide variety of pump system simulations. However limitations have been brought to light during the studies, therefore further improvement is possible.

- *The friction model.* For both water hammer phenomenon and pump performances evaluation, the dissipation was not evaluated adequately. Therefore, although there will be an increase in the computational effort, transient friction model for more accurate dissipation behaviour should be introduced.

- *Valve model.* The valve model is based on algorithms from the published literature. However better analysis should be performed to improve the reliability of the model for the API valve standard. In addition, a more accurate function for dampening effects as well as the viscous regime must be introduced.

- *Cavitation model.* Although the cavitation models with transport equation factors is more computationally expensive, they could allow more flexibility in the research. The transport equation, with bubble interaction studies, could improve the required results.

- *Accumulator model.* An improvement in the accumulator description both mathematically and numerically could be introduced.

- *The code structure.* The logic of the algorithm was not structured in an optimized way. The logic of the code is recursive, and functions are not called efficiently. Therefore an optimization study for the code could be performed to reduce tun time.

- *The code language.* The developed code was run in Matlab. This platform allowed flexibility in the development of the algorithms. However, its performance is slow when compared with other coding languages. Therefore a Python or C language could be more suitable to allow faster solution.

- *Slurry fluid.* The algorithm was developed to consider only homogeneous liquid without particulates. However the behaviour of slurries is different from pure water. Therefore, particle interaction should be introduced for a better fluid description.
- *Multi Slurry fluid.* Positive displacement pumps are normally used with different working fluids. A discharged fluid batch can be periodically be changed from water to slurry due to industrial requirements. Therefore at the same time, the pipeline can be filled with different fluid properties. Therefore, the introduction of this aspect could open potentially new mineral and Oil & Gas research fields.
- *Pump network.* More simulation and validation data should be obtained in a multi-pump network. As an example pump failure or a mix of pumps operating at different speeds could be analysed.

References

[1] R. Van Rijswick, "Fluid structure interaction in piston diaphragm pumps," TU Delft Offshore and Dredging Engineering, Mar. 2017, iSBN: 978-9-49-183715-9.

[2] R. van Rijswick and E. Vlot, "Pump phase shift control in world's largest iron ore concentrate pipeline." [Online]. Available: https://www.global.weir/assets/files/newsroom/Hydrotransport2017-PumpPhaseshiftcontroller.pdf

[3] A. Iannetti, "A numerical and experimental study on cavitation in positive displacement pumps and its application in valve design optimization," Ph.D. dissertation, 2015, iSNI: 0000 0004 5370 8715. [Online]. Available: http://digitool.lib.strath.ac.uk/R/?func=dbin-jump-full&object_id=26006

[4] H. H. Tackett, "Positive displacement reciprocating pump fundamentals - power and direct acting types," *Turbomachinery and Pump Symposia*, 2008. [Online]. Available: https://oaktrust.library.tamu.edu/bitstream/handle/1969.1/163923/06-tackett.pdf?sequence=1&isAllowed=y

[5] Y. W. Shin and W. L. Chen, "Numerical fluid-hammer analysis by the method of characteristics in complex piping networks," *Nuclear Engineering and Design*, vol. 33, no. 3, pp. 357 – 369, 1975. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0029549375900059

[6] A. Josifovic, M. Stickland, A. Iannetti, and J. Corney, "Engineering procedure for positive displacement pump performance analysus based on 1d and 3d commercial codes," in *Volume 1: 37th Computers and Information in Engineering Conference*. ASME, aug 2017.

[7] G. Vetter and F. Schweinfurther, "Pressure pulsations in the piping of reciprocating pumps," *Chemical Engineering & Technology - CET*, vol. 10, no. 1, pp. 262–271, 1987.

[8] R. V. Rijswick, "Pressure pulsation analysis of crankshaft driven positive displacement pumps," 2007. [Online]. Available: https://www.researchgate.net/publication/290610315_Pressure_pulsation_analysis_of_crankshaft_driven_positive_displacement_pumps

[9] N. D. Manring, "The discharge flow ripple of an axial-piston swash-plate type hydrostatic pump," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, no. 2, p. 263, 2000.

[10] P. Singh and J. Madavan, "Complete analysis and simulation of reciprocating pumps including system piping," *fourth international pump symposium*, 1987, dOI: 10.21423/r16x26.

[11] K. M. Opitz, S. R. Engel, A. F. Koegler, A. Leipertz, and E. Schlücker, "High-speed particle image velocimetry measurements of turbulent pipe flows for verification of a fluid-dynamic cavitation model," *Chemical Engineering & Technology*, vol. 35, no. 11, pp. 2035–2044, oct 2012.

[12] J. Pei, C. He, M. Lv, X. Huang, K. Shen, and K. Bi, "The valve motion characteristics of a reciprocating pump," *Mechanical Systems and Signal Processing*, vol. 66-67, pp. 657 – 664, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327015003015

[13] J. K. Lee, J. K. Jung, J.-B. Chai, and J. W. Lee, "Mathematical modeling of reciprocating pump," *Journal of Mechanical Science and Technology*, vol. 29, no. 8, pp. 3141–3151, aug 2015.

[14] K. A. Edge, O. P. Boston, K. C. S. Xiao, K. C. M. J. Longvill, and K. C. C. R. Burrows, "Pressure pulsations in reciprocating pump piping systems part 2: Experimental investigations and model validation," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 211, no. 3, pp. 239–250, may 1997.

[15] A. Josifovic, J. J. Roberts, J. Corney, B. Davies, and Z. K. Shipton, "Reducing the environmental impact of hydraulic fracturing through design optimisation of positive displacement pumps," *Energy*, vol. 115, pp. 1216 – 1233, nov 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544216312579

[16] R. van Rijswick, A. Talmon, and C. van Rhee, "Fluid structure interaction (FSI) in piston diaphragm pumps," *The Canadian Journal of Chemical Engineering*, vol. 94, no. 6, pp. 1116–1126, may 2016.

[17] M. B. Alberto, F. O. J. Manuel, and M.-F. Andrés, "Numerical methodology for the CFD simulation of diaphragm volumetric pumps," *International Journal of Mechanical Sciences*, vol. 150, pp. 322–336, jan 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002074031830242X

[18] D. Knežević, A. Milašinoviü, Z. Milovanovic, and V. Saviü, "Analysis of changes of bulk modulus of mineral oil – effects on the dynamic behaviour of hydraulic actuators," 04 2011.

[19] A. Iannetti, M. Stickland, and W. Dempster, "An investigation of the performance of a positive displacement reciprocating pump at low pressure npsh incorporating a three phase cavitation model," in *11th World Congress on Computational Mechanics, WCCMXI*, July 2014. [Online]. Available: https://strathprints.strath.ac.uk/48988/

[20] D. N. Johnston, "Numerical modelling of reciprocating pumps with self-acting valves," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 205, no. 2, pp. 87–96, may 1991.

[21] A. Iannetti, M. T. Stickland, and W. M. Dempster, "A computational fluid dynamics model to evaluate the inlet stroke performance of a positive displacement reciprocating plunger pump," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 228, no. 5, pp. 574–584, 2014. [Online]. Available: https://doi.org/10.1177/0957650914530295

[22] D. N. Johnston, K. A. Edge, and N. D. Vaughan, "Experimental investigation of flow and force characteristics of hydraulic poppet and disc valves," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 205, no. 3, pp. 161–171, aug 1991.

[23] D. N. Johnston and K. A. Edge, "The impedance characteristics of fluid power components: Restrictor and flow control valves," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 205, no. 1, pp. 3–10, feb 1991.

[24] S. I. Bernad and R. Susan-Resiga, "Numerical model for cavitational flow in hydraulic poppet valves," *Modelling and Simulation in Engineering*, vol. 2012, pp. 1–10, 2012. [Online]. Available: http://dx.doi.org/10.1155/2012/742162

[25] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002, iSBN: 9780511791253.

[26] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd ed. Springer Berlin Heidelberg, 2009, iSBN: 9783540498346.

[27] E. Wylie, V. Streeter, and L. Suo, *Fluid transients in systems.* Prentice Hall, 1993, iSBN: 9780133221732. [Online]. Available: https://books.google.co.uk/books?id=Ep9RAAAAMAAJ

[28] D. H. Axworthy, M. S. Ghidaoui, and D. A. McInnis, "Extended thermodynamics derivation of energy dissipation in unsteady pipe flow," *Journal of Hydraulic Engineering*, vol. 126, no. 4, pp. 276–287, apr 2000.

[29] J. T. Karam, Jr., "An efficient method for simulating frequency-dependent friction in transient liquid flow," *Journal of Fluids Engineering*, vol. 97, no. 1, p. 97, 1975. [Online]. Available: http://dx.doi.org/10.1115/1.3448149

[30] V. Guinot, "Numerical simulation of two-phase flow in pipes using godunov method," *International Journal for Numerical Methods in Engineering*, vol. 50, no. 5, pp. 1169–1189, 2001.

[31] L. Zhou, H. Wang, D. Liu, J. Ma, P. Wang, and L. Xia, "A second-order finite volume method for pipe flow with water column separation," *Journal*

of Hydro-environment Research, vol. 17, no. Supplement C, pp. 47 – 55, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570644316300107

[32] L. Zhou, H. Wang, A. Bergant, A. S. Tijsseling, D. Liu, and S. Guo, “Godunov-type solutions with discrete gas cavity model for transient cavitating pipe flow,” *Journal of Hydraulic Engineering*, vol. 144, no. 5, p. 04018017, may 2018.

[33] M. Zhao and M. S. Ghidaoui, “Godunov-type solutions for water hammer flows,” *Journal of Hydraulic Engineering*, vol. 130, no. 4, pp. 341–348, apr 2004.

[34] J. C. P. Liou, “Numerical properties of the discrete gas cavity model for transients,” *Journal of Fluids Engineering*, vol. 122, no. 3, p. 636, 09 2000.

[35] V. Guinot, “Riemann solvers for water hammer simulations by godunov method,” *International Journal for Numerical Methods in Engineering*, vol. 49, no. 7, pp. 851–870, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0207%2820001110%2949%3A7%3C851%3A%3AAID-NME978%3E3.0.CO%3B2-%23

[36] S. SaemI, M. Raisee, M. J. Cervantes, and A. Nourbakhsh, “Computation of two- and three-dimensional water hammer flows,” *Journal of Hydraulic Research*, pp. 1–19, jul 2018. [Online]. Available: https://doi.org/10.1080/00221686.2018.1459892

[37] G. Pezzinga and D. Cannizzaro, “Analysis of transient vaporous cavitation in pipes by a distributed 2d model,” *Journal of Hydraulic Engineering*, vol. 140, no. 6, p. 04014019, jun 2014.

[38] A. Bergant, A. Simpson, and J. Vítkovský, “Review of unsteady friction models in transient pipe flow,” 09 1999.

[39] B. Brunone, B. W. Karney, M. Mecarelli, and M. Ferrante, “Velocity profiles and unsteady pipe friction in transient flow,” *Journal of Water Resources Planning and Management*, vol. 126, no. 4, pp. 236–244, jul 2000.

[40] B. S. Arris S.Tijseeling, Qingzhi Hou and A. Bergant, “Acoustic resonance in a reservoir pipeline orifice system,” *Proceedings of the ASME 2010 Pressure Vessels & Piping division / K-PVP Conference*, 2010.

[41] A. Seck, M. Fuamba, and R. Kahawita, “Finite-volume solutions to the water-hammer equations in conservation form incorporating dynamic friction using the godunov scheme,” *Journal of Hydraulic Engineering*, vol. 143, no. 9, p. 04017029, 2017.

[42] J.-J. Shu, C. R. Burrows, and K. A. Edge, “Pressure pulsations in reciprocating pump piping systems part 1,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 211, no. 3, pp. 229–235, may 1997.

[43] W. Zielke, “Frequency-dependent friction in transient pipe flow,” *Journal of Basic Engineering*, vol. 90, no. 1, p. 109, 1968. [Online]. Available: http://dx.doi.org/10.1115/1.3605049

[44] Z. Z. Kamil Urbanowicz and S. Kunzma, “Universal weighting function in modeling transient cavitating pipe flow,” pp. 889–902, 10 2012. [Online]. Available: https://www.researchgate.net/profile/Kamil_Urbanowicz/publication/263441361_Universal_weighting_function_in_modeling_transient_cavitating_pipe_flow/

[45] Y.-H. Hwang and N.-M. Chung, “A fast godunov method for the water-hammer problem,” *International Journal for Numerical Methods in Fluids*, vol. 40, no. 6, pp. 799–819, 2002.

[46] A. S. León, M. S. Ghidaoui, A. R. Schmidt, and M. H. García, “Efficient second-order accurate shock-capturing scheme for modeling one- and two-phase water hammer flows,” *Journal of Hydraulic Engineering*, vol. 134, no. 7, pp. 970–983, jul 2008.

[47] F. Daude and P. Galon, “A finite-volume approach for compressible single- and two-phase flows in flexible pipelines with fluid-structure interaction,” *Journal of Computational Physics*, vol. 362, pp. 375–408, jun 2018.

[48] F. Daude, A. Tijsseling, and P. Galon, "Numerical investigations of water-hammer with column-separation induced by vaporous cavitation using a one-dimensional finite-volume approach," *Journal of Fluids and Structures*, vol. 83, pp. 91–118, nov 2018.

[49] T. Wang, J. Jiang, and G. Lan, "Research on accumulator for water hammer protection of long-distance slurry transportation pipelines," in *2014 ISFMFE - 6th International Symposium on Fluid Machinery and Fluid Engineering*. Institution of Engineering and Technology, Oct 2014, pp. 1–6.

[50] B. Bbosa, E. DelleCase, M. Volk, and E. Ozbayoglu, "A comprehensive deposition velocity model for slurry transport in horizontal pipelines," *Journal of Petroleum Exploration and Production Technology*, vol. 7, no. 1, pp. 303–310, jun 2016.

[51] C. E. Brennen, *Fundamentals of Multiphase Flow.* Cambridge University Press, 2005, iSBN: 9780511807169.

[52] Y. Zhao, G. Wang, and B. Huang, "A cavitation model for computations of unsteady cavitating flows," *Acta Mechanica Sinica*, vol. 32, no. 2, pp. 273–283, aug 2015.

[53] A. Murrone and H. Guillard, "A five equation reduced model for compressible two phase flow problems," *Journal of Computational Physics*, vol. 202, no. 2, pp. 664–698, jan 2005.

[54] D. Liuzzi, "Two-phase cavitation modelling," PhD, 2012. [Online]. Available: https://web.uniroma1.it/aerophd/sites/default/files/Thesis-06-06-2012.pdf

[55] P. J. Zwart, A. G. Gerber, and T. Belamri, "A Two Phase Flow Model for Predicting Cavitation Dynamics," in *ICMF 2004 International Conference on Multiphase Flow*, yokohama, Japan, May 30 – June 3, 2004.

[56] A. K. Singhal, M. M. Athavale, H. Li, and Y. Jiang, "Mathematical basis and validation of the full cavitation model," *Journal of Fluids Engineering*, vol. 124, no. 3, p. 617, 2002. [Online]. Available: http://dx.doi.org/10.1115/1.1486223

[57] K. S. Sumam, S. G. Thampi, and N. Sajikumar, "An alternate approach for modelling of transient vaporous cavitation," *International Journal for Numerical Methods in Fluids*, 2009. [Online]. Available: http://dx.doi.org/10.1002/fld.2093

[58] E. Goncalves and R. F. Patella, "Numerical simulation of cavitating flows with homogeneous models," *Computers & Fluids*, vol. 38, no. 9, pp. 1682–1696, oct 2009.

[59] E. Goncalves and B. Charriere, "Modelling for isothermal cavitation with a four-equation model," *International Journal of Multiphase Flow*, vol. 59, pp. 54–72, feb 2014. [Online]. Available: http://dx.doi.org/10.1016/j.ijmultiphaseflow.2013.10.015

[60] J. G. Zheng, B. C. Khoo, and Z. M. Hu, "Simulation of wave-flow-cavitation interaction using a compressible homogenous flow method," *Communications in Computational Physics*, vol. 14, no. 02, pp. 328–354, aug 2013.

[61] K. Opitz, "Cavitation in reciprocating positive displacement pumps," 2011.

[62] D. B. Mahamadou Adama Maiga, Olivier Coutier-Delgosha, "Cavitation in a hydraulic system: The influence of the distributor geometry on cavitation inception and study of the interactions between bubbles," 2016.

[63] E.-A. Brujan, "Cavitation bubble dynamics in non-newtonian fluids," *Polymer Engineering & Science*, vol. 49, no. 3, pp. 419–431, mar 2009. [Online]. Available: http://dx.doi.org/10.1002/pen.21292

[64] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics*, 2007.

[65] P. S. Arya, *Introduction to Micrometeorology.* ACADEMIC PR INC, 2011. [Online]. Available: https://www.ebook.de/de/product/3249463/paul_s_arya_introduction_to_micrometeorology.html

[66] V. Guinot, *Wave Propagation in Fluids: Models and Numerical Techniques*, W. iSTE, Ed. ISTE LTD, 2008. [Online].

Available: https://www.ebook.de/de/product/8483902/vincent_guinot_wave_propagation_in_fluids_models_and_numerical_techniques.html

[67] S. L. Wood, “Modeling of pipeline transients: Modified method of characteristics,” Ph.D. dissertation, 2011. [Online]. Available: https://digitalcommons.fiu.edu/etd/456/

[68] C. F. Colebrook, “Turbulent flow in pipes with particular reference to the transition region between the smooth and rough pipe laws,” *Journal of the Institution of Civil Engineers*, vol. 12, no. 8, pp. 393–422, oct 1939.

[69] E. Thiel, “Kinematik und druckverlust selbsttatiger ventile oszillierender verdrangerpumpen,” University of Erlangen, 1990. [Online]. Available: http://d-nb.info/910344442

[70] P. Puddu and M. Paderi, “Hydro-pneumatic accumulators for vehicles kinetic energy storage: Influence of gas compressibility and thermal losses on storage capability,” *Energy*, vol. 57, pp. 326–335, aug 2013.

[71] A. Pourmovahed and D. R. Otis, “An experimental thermal time-constant correlation for hydraulic accumulators,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 112, no. 1, p. 116, 1990.

[72] P. K. Sweby, “High resolution schemes using flux limiters for hyperbolic conservation laws,” *SIAM Journal on Numerical Analysis, Vol. 21, No. 5. (Oct., 1984), pp. 995-1011.*, 1985.

[73] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 3rd ed. Wiley, 2016, ISBN: 9781119121503. [Online]. Available: http://gen.lib.rus.ec/book/index.php?md5=43A7A457B95E0443C75D23DC1B46FEE7

[74] C. Berthon, “Stability of the muscl schemes for the euler equations,” *Commun. Math. Sci.*, vol. 3, no. 2, pp. 133–157, 06 2005. [Online]. Available: https://projecteuclid.org:443/euclid.cms/1118778272

[75] O. A. Alduchov and R. E. Eskridge, “Improved magnus form approximation of saturation vapor pressure,” *Journal of Applied Meteorology*, vol. 35, no. 4, pp. 601–609, apr 1996.

[76] A. Tijsseling, "Water hammer with fluid–structure interaction in thick-walled pipes," *Computers & Structures*, vol. 85, no. 11-14, pp. 844–851, jun 2007.

[77] A. R. Simpson, "Large water-hammer pressures for column separation in pipelines," pp. 1310–1316, oct 1986. [Online]. Available: https://www.researchgate.net/publication/34228001_LARGE_WATER_HAMMER_PRESSURES_DUE_TO_COLUMN_SEPARATION_IN_SLOPING_PIPES_PhD_Thesis_University_of_Michigan

[78] M. S. Ghidaoui, M. Zhao, D. A. McInnis, and D. H. Axworthy, "A review of water hammer theory and practice," *Applied Mechanics Reviews*, vol. 58, no. 1, p. 49, 2005. [Online]. Available: http://dx.doi.org/10.1115/1.1828050

Appendix A

Matlab Code

A.1 Main Structure

```
%% Main code for three chamber diaphragm pump with accumulator in the suction and discharge
close all
clear
clc

%% IMPUT FILE
SimulationData;      % Simulation Data Description
SuctionLayout;       % Suction Design layout
SuctionAccumulator;  % Suction Accumulator
ChamberLayout;       % Chamber algorithms
DischargeLayout;     % Discharge Design layout
DischargeAccumulator; % Dicharge Accumulator
ValveData;           % Valve definition , Suction and discharge
FluidSlurry;         % Slurry FLuid specification properties
```

```
FluidPropelling;     % Propelling FLuid specification properties
SolverDefinition;    % Solver algorithms
InitializationData;  % Initialization value

%% Initialization
%Preallocating the computer memeory
% SUCTION
for idxSuc = 1:SuctionNum
    Name = char(SuctionName(idxSuc,:));
    eval(['[NodeX.',Name,', NodeI.',Name,',Dx.',Name,',X.',Name,', XI.',Name,'] = Staggrid(0,Dimension.',Name,'.L,dx);'])
    eval(['[rho.',Name,', p.',Name,', rhoQ.',Name,', c.',Name,', alphaV.',Name,', alphaG.',Name,', u.',Name,', FV.',Name,']',...
        '= Memory(NodeX.',Name,', NodeT, rhoIn, rhoQIn, pIn, cIn, alphaVIn, uIn, alphaGIn, Solver);']);
end
% CHAMBER
for idxCha = SuctionNum+1:ChamberNum
    Name = char(ChamberName(idxCha,:));
    switch Name
        case 'Pis1'
            eval(['[NodeX.',Name,', NodeI.',Name,',Dx.',Name,',X.',Name,', XI.',Name,'] = Staggrid(0,Dimension.',Name,'.L,dx);'])
            eval(['[rho.',Name,', p.',Name,', rhoQ.',
                Name,', c.',Name,', alphaV.',Name,',
                alphaG.',Name,', u.',Name,', FV.',Name,
                '] = Memory(NodeX.',Name,', NodeT,
                rhoChaPropelling, rhoQChaPropelling,
                pChaPropelling, cChaPropelling,
                alphaVChaPropelling, uChaPropelling,
                alphaGChaPropelling, Solver);']);
        case 'Pis2'
            eval(['[NodeX.',Name,', NodeI.',Name,',Dx.
                ',Name,',X.',Name,', XI.',Name,'] =
                Staggrid(0,Dimension.',Name,'.L,dx);'])
            eval(['[rho.',Name,', p.',Name,', rhoQ.',
                Name,', c.',Name,', alphaV.',Name,',
                alphaG.',Name,', u.',Name,', FV.',Name,
                ']= Memory(NodeX.',Name,', NodeT,
                rhoChaPropelling, rhoQChaPropelling,
                pChaPropelling, cChaPropelling,
                alphaVChaPropelling, uChaPropelling,
                alphaGChaPropelling, Solver);']);
        case 'Pis3'
            eval(['[NodeX.',Name,', NodeI.',Name,',Dx.
                ',Name,',X.',Name,', XI.',Name,'] =
                Staggrid(0,Dimension.',Name,'.L,dx);'])
            eval(['[rho.',Name,', p.',Name,', rhoQ.',
                Name,', c.',Name,', alphaV.',Name,',
                alphaG.',Name,', u.',Name,', FV.',Name,
                ']= Memory(NodeX.',Name,', NodeT,
                rhoChaPropelling, rhoQChaPropelling,
                pChaPropelling, cChaPropelling,
                alphaVChaPropelling, uChaPropelling,
                alphaGChaPropelling, Solver);']);
        case 'Pre1'
            eval(['[NodeX.',Name,', NodeI.',Name,',Dx.
                ',Name,',X.',Name,', XI.',Name,'] =
                Staggrid(0,Dimension.',Name,'.L,dx);'])
            eval(['[rho.',Name,', p.',Name,', rhoQ.',
                Name,', c.',Name,', alphaV.',Name,',
                alphaG.',Name,', u.',Name,', FV.',Name,
                ']= Memory(NodeX.',Name,', NodeT,
                rhoChaPropelling, rhoQChaPropelling,
                pChaPropelling, cChaPropelling,
                alphaVChaPropelling, uChaPropelling,
                alphaGChaPropelling, Solver);']);
        case 'Pre2'
            eval(['[NodeX.',Name,', NodeI.',Name,',Dx.
                ',Name,',X.',Name,', XI.',Name,'] =
                Staggrid(0,Dimension.',Name,'.L,dx);'])
            eval(['[rho.',Name,', p.',Name,', rhoQ.',
                Name,', c.',Name,', alphaV.',Name,',
                alphaG.',Name,', u.',Name,', FV.',Name,
                ']= Memory(NodeX.',Name,', NodeT,
                rhoChaPropelling, rhoQChaPropelling,
                pChaPropelling, cChaPropelling,
                alphaVChaPropelling, uChaPropelling,
                alphaGChaPropelling, Solver);']);
        case 'Pre3'
            eval(['[NodeX.',Name,', NodeI.',Name,',Dx.
                ',Name,',X.',Name,', XI.',Name,'] =
                Staggrid(0,Dimension.',Name,'.L,dx);'])
            eval(['[rho.',Name,', p.',Name,', rhoQ.',
                Name,', c.',Name,', alphaV.',Name,',
                alphaG.',Name,', u.',Name,', FV.',Name,
                ']= Memory(NodeX.',Name,', NodeT,
                rhoChaPropelling, rhoQChaPropelling,
                pChaPropelling, cChaPropelling,
                alphaVChaPropelling, uChaPropelling,
                alphaGChaPropelling, Solver);']);
        otherwise
            eval(['[NodeX.',Name,', NodeI.',Name,',Dx.
                ',Name,',X.',Name,', XI.',Name,'] =
                Staggrid(0,Dimension.',Name,'.L,dx);'])
            eval(['[rho.',Name,', p.',Name,', rhoQ.',
                Name,', c.',Name,', alphaV.',Name,',
                alphaG.',Name,', u.',Name,', FV.',Name,
                ']= Memory(NodeX.',Name,', NodeT,
                rhoChaSlurry, rhoQChaSlurry, pChaSlurry
                , cChaSlurry, alphaVChaSlurry,
                uChaSlurry, alphaGChaSlurry, Solver);'
                ]);
    end
end
% DISCHARGE
for idxDis = ChamberNum+1:DischargeNum
    Name = char(DischargeName(idxDis,:));

    eval(['[NodeX.',Name,', NodeI.',Name,',Dx.',Name,'
        ,X.',Name,', XI.',Name,'] = Staggrid(0,
Dimension.',Name,'.L,dx);'])
    eval(['[rho.',Name,', p.',Name,', rhoQ.',Name,', c.',Name,', alphaV.',Name,', alphaG.',Name,', u.',Name,', FV.',Name,']= Memory(NodeX.',Name,', NodeT, rhoOut, rhoQOut, pOut, cOut, alphaVOut, uOut, alphaGOut, Solver);']);
end
% Preallocating the memory for the valves characteristics and the Piston
% position and velocity
[xd, xdd, NetForced, Agapd] = PreallocatedMemoryValve(NodeT,NumOfPumps); % Preallocated the Discharge valves
[xs, xds, NetForces, Agaps] = PreallocatedMemoryValve(NodeT,NumOfPumps); % Preallocated the Suction valves
[xp, vp, ]                  = PreallocatedMemoryValve(NodeT,NumOfPumps); % Preallocated the piston value

VFs         = VFs * ones(NodeT,1); % Preallocate the Volume fluid in the suction accumulator
VFd         = VFd * ones(NodeT,1); % Preallocate the Volume fluid in the discharge accumulator
Angle      = zeros(NodeT,1);       % Preallocate the angle of the crank shaft for each chamber
Time       = zeros(NodeT,1);       % Preallocate the simulation time
wReal      = zeros(NodeT,1);       % Preallocate the
```

```
speed of ratation
TimeIndex = 1;                      % Time index setting
Step      = 100;                    % scale of the progress index
Vgaps     = zeros(3,1);             % Preallocate the velocity at the valve gap for suction valves
Vgapd     = zeros(3,1);             % Preallocate the velocity at the valve gap for discharge valves
rhoQgaps  = zeros(3,1);             % Preallocate the mass flow rate at the valve gap for suction valves
rhoQgapd  = zeros(3,1);             % Preallocate the mass flow rate at the valve gap for discharge valves
%% Directory
% Create the directory folder to save the simulation data
Name  = ['SPM_',num2str(SPM),'-pIn_',num2str(pressure(pIn,'Pa','bar')),'-pOut_',num2str(pressure(pOut,'Pa','bar'))];
NameDirectory                   = [Folder,'\',Name]; % Location and name folder
[Status, Message, Messageid] = rmdir(NameDirectory);
mkdir(NameDirectory); % create the folder
%% FV
tic
t         = 1;
SaveIdx   = 0;
w         = SPM2rads(SPM);
Solver.dt = CFL * dx / max([cIn cChaPropelling
   cChaSlurry cOut]); %Define the first time step
   value using as speed of sound the maximum value
   calculated from:
% - Inlet condition
% - Outlet condition
% - propelling fluid condition
% - Water or slurry in the chamber condition
% The pressure at the inlet, chamber and discharge can
    be and normally are different.
% Although the initial pressure and velocity in the
   chamber are equal for the
% propelling liquid on one diaphragm side and the
   water on the other, fluid
% can have different characteristics.
while Time(t) <  TF
    t = t+1;
    Time(t) = Time(t-1) + Solver.dt;
    %% Junction
    % SUCTION Junction
    % resolving the node junction in the suction
       section
    pJSBC    = Junction(FV.SBC,    FV.In,   Dimension.SBC
       .A,   Dimension.In.A);
    pTJIn    = TJunction(FV.In,    FV.L,    FV.R,
       Dimension.In.A,     Dimension.L.A,     Dimension.R.
       A, 'N11');
    pTJAccS  = TJunction(FV.AccS,   FV.L,    FV.AccSC,
       Dimension.AccS.A,   Dimension.L.A,     Dimension.
       AccSC.A, '1NN');
    pTJSC1   = TJunction(FV.AccSC, FV.S13, FV.SC1, Dimension.AccSC.A, Dimension.S13.A, Dimension.SC1.A, '1NN');
    pTJSC2   = TJunction(FV.S23,  FV.R,    FV.SC2, Dimension.S23.A,  Dimension.R.A,    Dimension.SC2.A, '1NN');
    pTJSC3   = TJunction(FV.S13,  FV.S23, FV.SC3, Dimension.S13.A,  Dimension.S23.A, Dimension.SC3.A, '1NN');
    % Discharge
    % resolving the node junction in the discharge section
    pJDBC    = Junction(FV.DBC, FV.Out, Dimension.DBC.A, Dimension.Out.A);
    pTJAccD  = TJunction(FV.Out,  FV.AccD,    FV.AccDC, Dimension.Out.A,  Dimension.AccD.A,    Dimension.AccDC.A, 'N11');
    pTJDC1   = TJunction(FV.End, FV.D13, FV.DC1, Dimension.End.A, Dimension.D13.A, Dimension.DC1.A, '1NN');
    pTJDC2   = TJunction(FV.D23, FV.AccDC, FV.DC2, Dimension.D23.A, Dimension.AccDC.A, Dimension.DC2.A, '1NN');
    pTJDC3   = TJunction(FV.D13, FV.D23, FV.DC3, Dimension.D13.A, Dimension.D23.A, Dimension.DC3.A, '1NN');
    % Chamber
    % resolving the node junction in the discharge section
```

```
for i = 1:NumOfPumps
    eval(['pJPis',num2str(i),' = Junction(FV.Pis',num2str(i),',FV.Pre',num2str(i),',Dimension.Pis',num2str(i),'.A, Dimension.Pre',num2str(i),'.A);']);
    eval(['pJPre',num2str(i),' = Junction(FV.Pre',num2str(i),',FV.Dia',num2str(i),',Dimension.Pre',num2str(i),'.A, Dimension.Dia',num2str(i),'.A);']);
    eval(['pTJC',num2str(i),' = TJunction(FV.Dia',num2str(i),', FV.Suc',num2str(i),', FV.Dis',num2str(i),', Dimension.Dia',num2str(i),'.A, Dimension.Suc',num2str(i),'.A, Dimension.Dis',num2str(i),'.A, ',char(39),'N11',char(39),');']);
end
%% Valve and Data
% Linear increase of the speed of the pump from w(0) = 0 and
% w(TimeRump) = w
wReal(t) = Time(t) ./ (TimeRump + eps) * w * (Time(t) < TimeRump) + w * (Time(t) >= TimeRump); %Speed rum of the pump
for i = 1:NumOfPumps
    %% VALVE Suction
    eval(['rhoGaps(i) = rho.SC',num2str(i),'(t-1,1);']);
    eval(['[xs(t,i), xds(t,i), NetForces(t,i), Agaps(t,i), CFs] =',...
        'ValveModel(p.SC',num2str(i),'(t-1,1),p.Suc',num2str(i),'(t-1,end),rhoGaps(i),xs(t-1,i),xds(t-1,i),Vgaps(i),SuctionValve,Solver,NetForces(t-1,i),t);']);
    eval(['[rhoQgaps(i) Vgaps(i)]= GapFlowRev2(Agaps(t,i), rhoGaps(i),  FV.Suc',num2str(i),', Dimension.Suc',num2str(i),'.A, FV.SC',num2str(i),', Dimension.SC',num2str(i),'.A, rhoQgaps(i), Solver.dt, toll,SuctionValve);']);
    %% VALVE Discharge
    eval(['rhoGapd(i) = rho.Dis',num2str(i),'(t-1,end);']);
    eval(['[xd(t,i), xdd(t,i), NetForced(t,i), Agapd(t,i), CFd] =',...
        'ValveModel(p.Dis',num2str(i),'(t-1,end),p.DC',num2str(i),'(t-1,1),rhoGapd(i),xd(t-1,i),xdd(t-1,i),Vgapd(i),DischargeValve,Solver,NetForced(t-1,i),t);']);
    eval(['[rhoQgapd(i) Vgapd(i)]= GapFlowRev2(Agapd(t,i), rhoGapd(i),  FV.Dis',num2str(i),', Dimension.Dis',num2str(i),'.A,FV.DC',num2str(i),', Dimension.DC',num2str(i),'.A, rhoQgapd(i), Solver.dt, toll, DischargeValve);']);
    %% Piston
    Angle(t) = Angle(t-1) + Solver.dt * wReal(t);
```

```
            % Update the crank shaft angle
        eval(['xp(t,i) = Dimension.Pis',num2str(i),'.L / 2 * (1-cos(Angle(t))+ deg2rad(Angle0(i))) - lambda/2*(sin(Angle(t) + deg2rad(Angle0(i)))).^2;']);
        eval(['vp(t,i) = Dimension.Pis',num2str(i),'.L / 2 * wReal(t) *(sin(Angle(t)+ deg2rad(Angle0(i))) - (lambda/2 * sin(2* Angle(t) + 2* deg2rad(Angle0(i)))));']);

        eval(['rhoPiston(i) = extrapolation(rho.Pis',num2str(i),'(t-1,1:2), XI.Pis',num2str(i),'(1), X.Pis',num2str(i),'(1:2),',char(39),'inlet',char(39),',',char(39),'yes',char(39),');']);
        eval(['rhoQpis(i) = rhoPiston(i) * Dimension.Pis',num2str(i),'.A * vp(t,i);']);
    end
    %% Solver
    % The number of data request to the algorithms is the same number of
    % the parts. The name of the part, Check for example SuctionLayout, is
    % taken after the "Dimension.", for all of those, the following data
    % must be specify:
    % - BC.Name.Val(1) = the falue of the pressure or of the flowrate at
    % i=1/2 namely the first interface
    % - BC.Name.Val(2) = the falue of the pressure or of the flowrate at
    % i= N + 1/2 namely the last interface
    % - BC.Name.In  = specify if the first interface is pressure of flow
    % rate considering the 'p' or 'fr' as data.
    % - BC.Name.Out = specify if the first interface is pressure of flow
    % rate considering the 'p' or 'fr' as data.
    Partname = fieldnames(X);
    PartNum   = length(Partname);
    for i = 1 : PartNum
        %% Boundary condition <--------
        Name = char(Partname(i,:));
        switch Name
            % Suction
            case 'SBC'
                eval(['BC.',Name,'.Val(1) = pIn;'])
                eval(['BC.',Name,'.Val(2) = pJSBC;'])
                eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
                Fluid = Slurry; %Definition of what kind of fluid section is
            case 'In'
                eval(['BC.',Name,'.Val(1) = pJSBC;'])
                eval(['BC.',Name,'.Val(2) = pTJIn;'])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p
                    ',char(39),';']);
                Fluid = Slurry;
            case 'L'
                eval(['BC.',Name,'.Val(1) = pTJIn;'])
                eval(['BC.',Name,'.Val(2) = pTJAccS;'
                    ])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p
                    ',char(39),';']);
                Fluid = Slurry;
            case 'AccS'
               % for more details check the
                    AccumulatorData
                [pFs, VFs(t)] = Accumulator(VTs, VFs(t
                    -1), V0s, FV.AccS, gamma, Solver.dt
                    , Dimension.AccS.A, pPreS, pMaxS,
                    Fluid, KDs, KSs, fixS);
                eval(['BC.',Name,'.Val(1) = pTJAccS;'
                    ])
                eval(['BC.',Name,'.Val(2) = pFs;'])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p
                    ',char(39),';']);
                Fluid = Slurry;
            case 'AccSC'
```

```
            eval(['BC.',Name,'.Val(1) = pTJSC1;'])
            eval(['BC.',Name,'.Val(2) = pTJAccS;'
               ])
            eval(['BC.',Name,'.In  = ',char(39),'p
               ',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p
               ',char(39),';']);
            Fluid = Slurry;
        case 'R'
            eval(['BC.',Name,'.Val(1) = pTJIn;'])
            eval(['BC.',Name,'.Val(2) = pTJSC2;'])
            eval(['BC.',Name,'.In  = ',char(39),'p
               ',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p
               ',char(39),';']);
            Fluid = Slurry;
        case 'S23'
            eval(['BC.',Name,'.Val(1) = pTJSC2;'])
            eval(['BC.',Name,'.Val(2) = pTJSC3;'])
            eval(['BC.',Name,'.In  = ',char(39),'p
               ',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p
               ',char(39),';']);
            Fluid = Slurry;
        case 'S13'
            eval(['BC.',Name,'.Val(1) = pTJSC3;'])
            eval(['BC.',Name,'.Val(2) = pTJSC1;'])
            eval(['BC.',Name,'.In  = ',char(39),'p
               ',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'SC1'
            eval(['BC.',Name,'.Val(1) = rhoQgaps(1);'])
            eval(['BC.',Name,'.Val(2) = pTJSC1;'])
            eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'SC2'
            eval(['BC.',Name,'.Val(1) = rhoQgaps(2);'])
            eval(['BC.',Name,'.Val(2) = pTJSC2;'])
            eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'SC3'
            eval(['BC.',Name,'.Val(1) = rhoQgaps(3);'])
            eval(['BC.',Name,'.Val(2) = pTJSC3;'])
            eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
            % Discharge
        case 'DBC'
            eval(['BC.',Name,'.Val(1) = pOut;'])
            eval(['BC.',Name,'.Val(2) = pJDBC;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'Out'
            eval(['BC.',Name,'.Val(1) = pJDBC;'])
            eval(['BC.',Name,'.Val(2) = pTJAccD;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'AccD'
            [pFd, VFd(t)] = Accumulator(VTd, VFd(t-1), V0d, FV.AccD, gamma, Solver.dt, Dimension.AccD.A, pPreD, pMaxD, Fluid, KDd, KSd, fixD);
            eval(['BC.',Name,'.Val(1) = pTJAccD;'])
            eval(['BC.',Name,'.Val(2) = pFd;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'AccDC'
            eval(['BC.',Name,'.Val(1) = pTJAccD;'])
            eval(['BC.',Name,'.Val(2) = pTJDC2;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'D23'
            eval(['BC.',Name,'.Val(1) = pTJDC2;'])
            eval(['BC.',Name,'.Val(2) = pTJDC3;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'D13'
            eval(['BC.',Name,'.Val(1) = pTJDC3;'])
            eval(['BC.',Name,'.Val(2) = pTJDC1;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'End'
```

```
            eval(['BC.',Name,'.Val(1) = pTJDC1;'])
            eval(['BC.',Name,'.Val(2) = 0;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'fr',char(39),';']);
            Fluid = Slurry;
        case 'DC1'
            eval(['BC.',Name,'.Val(1) = rhoQgapd(1);'])
            eval(['BC.',Name,'.Val(2) = pTJDC1;'])
            eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'DC2'
            eval(['BC.',Name,'.Val(1) = rhoQgapd(2);'])
            eval(['BC.',Name,'.Val(2) = pTJDC2;'])
            eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
        case 'DC3'
            eval(['BC.',Name,'.Val(1) = rhoQgapd(3);'])
            eval(['BC.',Name,'.Val(2) = pTJDC3;'])
            eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Slurry;
            % Chamber 1
        case 'Pis1'
            eval(['BC.',Name,'.Val(1) = rhoQpis(1);'])
            eval(['BC.',Name,'.Val(2) = pJPis1;'])
            eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Propelling;
        case 'Pre1'
            eval(['BC.',Name,'.Val(1) = pJPis1;'])
            eval(['BC.',Name,'.Val(2) = pJPre1;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
            Fluid = Propelling;
        case 'Dia1'
            eval(['BC.',Name,'.Val(1) = pJPre1;'])
            eval(['BC.',Name,'.Val(2) = pTJC1;'])
            eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p
                ',char(39),';']);
            Fluid = Slurry;
        case 'Suc1'
            eval(['BC.',Name,'.Val(1) = pTJC1;'])
            eval(['BC.',Name,'.Val(2) = rhoQgaps
                (1);'])
            eval(['BC.',Name,'.In  = ',char(39),'p
                ',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'
                fr',char(39),';']);
            Fluid = Slurry;
        case 'Dis1'
            eval(['BC.',Name,'.Val(1) = pTJC1;'])
            eval(['BC.',Name,'.Val(2) = rhoQgapd
                (1);'])
            eval(['BC.',Name,'.In  = ',char(39),'p
                ',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'
                fr',char(39),';']);
            Fluid = Slurry;
            % Chamber 2
        case 'Pis2'
            eval(['BC.',Name,'.Val(1) = rhoQpis(2)
                ;'])
            eval(['BC.',Name,'.Val(2) = pJPis2;'])
            eval(['BC.',Name,'.In  = ',char(39),'
                fr',char(39),';']);
            eval(['BC.',Name,'.Out = ',char(39),'p
                ',char(39),';']);
```

```
                Fluid = Propelling;
            case 'Pre2'
                eval(['BC.',Name,'.Val(1) = pJPis2;'])
                eval(['BC.',Name,'.Val(2) = pJPre2;'])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p
                    ',char(39),';']);
                Fluid = Propelling;
            case 'Dia2'
                eval(['BC.',Name,'.Val(1) = pJPre2;'])
                eval(['BC.',Name,'.Val(2) = pTJC2;'])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p
                    ',char(39),';']);
                Fluid = Slurry;
            case 'Suc2'
                eval(['BC.',Name,'.Val(1) = pTJC2;'])
                eval(['BC.',Name,'.Val(2) = rhoQgaps
                    (2);'])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'
                    fr',char(39),';']);
                Fluid = Slurry;
            case 'Dis2'
                eval(['BC.',Name,'.Val(1) = pTJC2;'])
                eval(['BC.',Name,'.Val(2) = rhoQgapd
                (2);'])
                eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'fr',char(39),';']);
                Fluid = Slurry;
                % Chamber 3
            case 'Pis3'
                eval(['BC.',Name,'.Val(1) = rhoQpis(3);'])
                eval(['BC.',Name,'.Val(2) = pJPis3;'])
                eval(['BC.',Name,'.In  = ',char(39),'fr',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
                Fluid = Propelling;
            case 'Pre3'
                eval(['BC.',Name,'.Val(1) = pJPis3;'])
                eval(['BC.',Name,'.Val(2) = pJPre3;'])
                eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p',char(39),';']);
                Fluid = Propelling;
            case 'Dia3'
                eval(['BC.',Name,'.Val(1) = pJPre3;'])
                eval(['BC.',Name,'.Val(2) = pTJC3;'])
                eval(['BC.',Name,'.In  = ',char(39),'p',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'p
                    ',char(39),';']);
                Fluid = Slurry;
            case 'Suc3'
                eval(['BC.',Name,'.Val(1) = pTJC3;'])
                eval(['BC.',Name,'.Val(2) = rhoQgaps
                    (3);'])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'
                    fr',char(39),';']);
                Fluid = Slurry;
            case 'Dis3'
                eval(['BC.',Name,'.Val(1) = pTJC3;'])
                eval(['BC.',Name,'.Val(2) = rhoQgapd
                    (3);'])
                eval(['BC.',Name,'.In  = ',char(39),'p
                    ',char(39),';']);
                eval(['BC.',Name,'.Out = ',char(39),'
                    fr',char(39),';']);
                Fluid = Slurry;
            otherwise
        end
        %% Solver
        eval(['[',Name,',FV.',Name,']= MUSCL(Dimension
            .',Name,', X.',Name,', XI.',Name,', BC.',
            Name,'.Val,BC.',Name,'.In,BC.',Name,'.Out,
            Solver,Fluid, FV.',Name,');']);
        % Update the value of the portion ”Name”
        eval(['p.',Name,'(t,:)    =',Name,'(1,:);']);   % pressure
        eval(['rhoQ.',Name,'(t,:) =',Name,'(2,:);']);   % massflow rate
        eval(['alphaV.',Name,'(t,:) =',Name,'(3,:);']) ; % Vapour volume Cavity
        eval(['alphaG.',Name,'(t,:) =',Name,'(4,:);']) ; % Gas volume fraction
        eval(['rho.',Name,'(t,:) =',Name,'(5,:);']);    % Density
        eval(['c.',Name,'(t,:) =',Name,'(6,:);']);      % Speed of sound
        eval(['u.',Name,'(t,:) =',Name,'(7,:);']);      % Velocity of the fluid
    end
    %% MISCELANIUS
    % plot the progress of the solution
    StepTime  =  TF  / Step;
    if Time(t) > (StepTime * TimeIndex)
        TimeIndex = TimeIndex + 1;
        fprintf([num2str(floor(Time(t)/ TF * Step)),'/',num2str(Step),'\n'])
    end
    if or(t== NodeT, Time(t) >= TF)
        % Prevent the overflow data and save if reach the simulation Time
        % or the maximum preallocated data.
        SaveIdx = SaveIdx + 1;
        p      = SaveVariable(p,'p',t,NameDirectory,
            SaveIdx);
        rhoQ   = SaveVariable(rhoQ,'rhoQ',t,NameDirectory,SaveIdx);
        alphaV = SaveVariable(alphaV,'alphaV',t,NameDirectory,SaveIdx);
        alphaG = SaveVariable(alphaG,'alphaG',t,NameDirectory,SaveIdx);
        rho    = SaveVariable(rho,'rho',t,NameDirectory,SaveIdx);
        c      = SaveVariable(c,'c',t,NameDirectory,SaveIdx);
        u      = SaveVariable(u,'u',t,NameDirectory,SaveIdx);
        Time   = Time(1:t);
        wReal  = wReal(1:t);
        save([NameDirectory,'\x_',num2str(SaveIdx)],'xd','xs');
        save([NameDirectory,'\VF_',num2str(SaveIdx)],'VFd','VFs');
        save([NameDirectory,'\Angle_',num2str(SaveIdx)],'Angle');
        save([NameDirectory,'\Time_',num2str(SaveIdx)],'Time');
        save([NameDirectory,'\wReal_',num2str(SaveIdx)],'wReal');
        save([NameDirectory,'\SimulationFile'],'SaveIdx','SPM','pIn','pOut','X','Angle0','lambda','pIn','pOut','Solver')

        % Reseting the data
        Time(1)        = Time(t);
        Angle(1,:)     = Angle(t,:);
        % Accumulator
        VFd(1)         = VFd(t);
        VFs(1)         = VFs(t);
        % Suction Valve
        xs(1,:)        = xs(t,:);
        xds(1,:)       = xds(t,:);
        NetForces(1,:) = NetForces(t,:);
        % Discharge Valve
        NetForced(1,:) = NetForced(t,:);
        xdd(1,:)       = xdd(t,:);
        xd(1,:)        = xd(t,:);

        t = 1;

        fprintf('saved %d \n', SaveIdx)
    end
    %% time Step evaluation
    cMax = FindMax(c,t); % Calculate the
    Solver.dt = Solver.dx * Solver.CFL / cMax;
end

TIME = toc
```

A.2 Simulation Data

```
% Definition of the simulation setup.
% it is required to define:
```

```
% - Folder location: place where the data will be stored
% - NumCic:  number of the cyclethat must be simulated (included the rump
% pump cycle)
% - CFL: courant number for stability, in case of uncompressible and
% neglecting the convective term, if the number is euql to one it will be
% the analytical water hammer solution
% - toll: tollerance of the Newthon-Rhapson method for the valve
% - Cyclerump: rum pump cycle that can be also not integer number
% - Angle0: vector of the shift phase of the chambers
% - lambda: value of the ratio radius / Length
% - NumOfPumps: number of chamber. !! it required to set different layout
% in the suction abd discharge definition.
Folder     = ToBeDefined; % Folder location to save data
NumCic     = ToBeDefined;    % number of cycle to simulate, with the rump pump cycle include
CFL        = ToBeDefined;  % Courant Number for stability should be less than 1 for avoiding instability due to increasing in the pressure higher than the initial pressure.
NodeT      = 1e5;  % number of node to pre allocated the memory, this
```

```
% value will affect the number of saving and the time of seaving
toll        = 1e-8; % tollerance value of the Newton rapshon method Absolute error
Cyclerump   = ToBeDefined;     % number of cycle that will be use to reach the regime value it can be also not integer
Angle0      = [0 240 120]; % shif angle of the piston
lambda      = ToBeDefined;    % the ratio radius / length of the crank shaft

NumOfPumps = 3;  % Number of Pump

%% Process data simulation
SPM         = ToBeDefined;    % Stroke per minutes
pIn         = ToBeDefined;     % Suction pressure [Bar]
pCha        = ToBeDefined;  % Chamber pressure  [Bar]
pOut        = ToBeDefined;  % Discharge pressure [Bar]
TimeRump    = 60/SPM * Cyclerump; % Time for running the pump
TF          = 60/SPM * NumCic;  % End Time
g           = 9.81; % gravity acceleration
```

A.3 Suction Layout

The same idea of layout definition was applied for the discharge and the chamber pump part.

```
% The suction line will consider all the ducts and the pipe that are
% connected each other creating a network that end with the suction valve
% interface. The Dimension must be define following the criteria of:
%
% Dimension . ”Name” . Specific magniture
%
% Dimension is always required to consider a struct funcion where all the
% data will be stored.
% ”Name” the name that you can use for the part
% Specific magniture, namely the characteristic of the function:
% L --> Length [m]
% d --> Diameter [m]
% A --> Area [m] To calculate is possible to use the function CircleArea
% th --> Thickness of the pipe [m] in water hammer formulation it is
% usefull to calculate the speed of sound, but in the pump the differences
% in the speed of sound propagation will require also an initialization
% point that sometime create sporious oscillation, for this reason the
% speed of sound will be calculated as bulk moduls formulation in the
% liquid phase and the brennen formulation for the two/three phase
% nu —> Poisson contraction coefficient
% E  —> Young modulus
%% Dimension SBC
Dimension.SBC.L  = .5;
Dimension.SBC.d = .254;
Dimension.SBC.A  = CircleArea(Dimension.SBC.d);
Dimension.SBC.th = 0.01;
Dimension.SBC.nu = 0.3; % Poisson
Dimension.SBC.E  = 21e10; % Young
%% Dimension IN
Dimension.In.L  = .5;
Dimension.In.d  = 0.2545;
Dimension.In.A  = CircleArea(Dimension.In.d);
Dimension.In.th = 0.01;
Dimension.In.nu = 0.3; % Poisson
Dimension.In.E  = 20.5e10; % Poisson
%% Dimension L
Dimension.L.L  = 8.160;
Dimension.L.d = .254;
Dimension.L.A  = CircleArea(Dimension.L.d);
Dimension.L.th = 0.01;
Dimension.L.nu = 0.3; % Poisson
Dimension.L.E  = 21e10; % Poisson
%% Dimension AccS
Dimension.AccS.L  = .5;
Dimension.AccS.d = .254;
Dimension.AccS.A  = CircleArea(Dimension.AccS.d);
Dimension.AccS.th = 0.01;
Dimension.AccS.nu = 0.3; % Poisson
Dimension.AccS.E  = 21e10; % Poisson
%% Dimension AccSC
Dimension.AccSC.L  = 1;
Dimension.AccSC.d = .254;
Dimension.AccSC.A  = CircleArea(Dimension.AccSC.d);
Dimension.AccSC.th = 0.01;
Dimension.AccSC.nu = 0.3; % Poisson
Dimension.AccSC.E  = 21e10; % Poisson
%% Dimension R
Dimension.R.L  = 1;
Dimension.R.d = .254;
Dimension.R.A  = CircleArea(Dimension.R.d);
Dimension.R.th = 0.01;
Dimension.R.nu = 0.3; % Poisson
Dimension.R.E  = 21e10; % Poisson
%% Dimension S13
Dimension.S13.L  = 1.125;
Dimension.S13.d = .254;
Dimension.S13.A  = CircleArea(Dimension.S13.d);
Dimension.S13.th = 0.01;
Dimension.S13.nu = 0.3; % Poisson
Dimension.S13.E  = 21e10; % Poisson
%% Dimension S23
Dimension.S23.L  = 1.125;
Dimension.S23.d = .254;
Dimension.S23.A  = CircleArea(Dimension.S23.d);
Dimension.S23.th = 0.01;
Dimension.S23.nu = 0.3; % Poisson
Dimension.S23.E  = 21e10; % Poisson
```

```
%% Dimension SC1
Dimension.SC1.L  = .341;
Dimension.SC1.d = .254;
Dimension.SC1.A  = CircleArea(Dimension.SC1.d);
Dimension.SC1.th = 0.0016;
Dimension.SC1.nu = 0.3; % Poisson
Dimension.SC1.E  = 21e10; % Poisson
%% Dimension SC2
Dimension.SC2.L  = .341;
Dimension.SC2.d = .254;
Dimension.SC2.A  = CircleArea(Dimension.SC2.d);
Dimension.SC2.th = 0.01;
Dimension.SC2.nu = 0.3; % Poisson
Dimension.SC2.E  = 21e10; % Poisson
%% Dimension SC3
Dimension.SC3.L  = .341;
Dimension.SC3.d = .254;
Dimension.SC3.A  = CircleArea(Dimension.SC3.d);
Dimension.SC3.th = 0.01;
Dimension.SC3.nu = 0.3; % Poisson
Dimension.SC3.E  = 21e10; % Poisson
%% Num
SuctionName = fieldnames(Dimension);
SuctionNum  = length(SuctionName);
%% Create Backup of the script in the directory Back up
FileName     = mfilename;
% newbackup    = sprintf('%sbackup.m',FileName)
 currentfile = strcat(FileName, '.m');
```

```
SimulationBackupFolder = [directory,FolderName,'\
    BackupData'];
copyfile(currentfile,SimulationBackupFolder)
```

A.4 Suction Accumulator

The same idea of accumulator definition was applied for the discharge part.

```
% The accumulato of the suction line
% It is request to set up different parameters and the
     accumulator function
% in the simulink matlab library was taken as primary
    algorithms
% for further details check the accumulator helper.
% Three differen algorithm can be used:
% Acc - accumulator
% press - the pressure is considered constant at the
    pref value
% rhoQ  - no fluid i allow to pass, namelly rhoQ = 0
    at the interface
%
% Enter manually the volume of the accumulator in VF,
    Fluid at t = 0 inside
% the accumulator in Liter, and the VT, the total
    volume.

%% Suction Accumulator
pPreS   = pressure(ToBeDefined,'bar','Pa');    %[Bar] %
    Pressure at the precharge Accumulator where the
    volume is VT-VF
```

```
pMaxS    = pressure(ToBeDefined,'bar','Pa');    %[Bar] % Pressure at max, where the volume is VD
KSs      = [1e6, -1e5]; % constant value that influence the behaviour when the Volume of the accumulator is smaller than the minimum value
KDs      = [1e6, 1e6];  % constant value that influence the behaviour when the Volume of the accumulator is bigger than the minimum value
gamma    = 1;     % transform exponent
VFs      = L2m(60); % Volume of Fluid at t=0; Conversion Litres to meters
VTs      = L2m(150); % total volume
V0s      = VTs - VFs; % initial volume of the gass
%% Create Backup of the script in the directory Back up
FileName      = mfilename;
% newbackup    = sprintf('%sbackup.m',FileName)
currentfile = strcat(FileName, '.m');
SimulationBackupFolder = [directory,FolderName,'\BackupData'];
copyfile(currentfile,SimulationBackupFolder)
```

A.5 Valve Data

```
% Definition of the suction and discharge valve characteristi in terms of
% deign layout, spring forces and constant, and all the parameters request
% to calculate the forces that are acting across
```

```
themself.
%% Suction Valve
SuctionValve.do    = .25; %[m] Outer diameter
SuctionValve.di    = .20;  %[m] Valve_Nominal_diameter
SuctionValve.m     = 26.8;  %[kg]   mass
SuctionValve.h     = 0.024;  %[m] height
SuctionValve.theta = 25;  %[deg] degree of the Valve
SuctionValve.lg    = (SuctionValve.do+SuctionValve.di)/(2*cos(deg2rad(SuctionValve.theta)));  %[m] Gap_length
SuctionValve.CDJ   = 0.72; % Johnston Coefficien
SuctionValve.C     = 50; % Fluid Dumper coefficient [Ns/m]
% Valve Spring
SuctionValve.k     = 12500;  %[N/m] Spring costant
SuctionValve.FS0   = 435;  %[N]    Preload Spring
SuctionValve.xMax  = 0.04;  %[m]   Max position Valve
% Valve Fluid
SuctionValve.fdc   = [225 0 0.9 0]; % Flow_discharge_constants
SuctionValve.Dam   = [280 463]; % Damping_Costant
SuctionValve.ffc   = [0.72 0.86 0.07]; % Flow_force_constants Thield
SuctionValve.nu    = 8.9e-4; % Flow_force_constants
SuctionValve.fric  = 0.01;
SuctionValve.Zeta  = 1;
%% Discharge Valve
DischargeValve.do    = .25;
DischargeValve.di     = .2;   %[m] Valve_Nominal_diameter
DischargeValve.m      = 26.8;  %[kg]    mass
DischargeValve.h      = 0.024;  %[m] height
DischargeValve.theta = 25;  %[deg] degree of the Valve
DischargeValve.lg     = (DischargeValve.do+DischargeValve.di)/(2*cos(deg2rad(DischargeValve.theta)));  %[m] Gap_length
DischargeValve.CDJ    = 0.72;
DischargeValve.C      = 50; % Fluid Dumper coefficient [Ns/m]
% Valve Spring
DischargeValve.k      = 12500;  %[N/m] Spring costant
DischargeValve.FS0    = 435;  %[N]     Preload Spring
DischargeValve.xMax   = 0.04;  %[m]    Max position Valve
% Valve Fluid
DischargeValve.fdc    = [225 0 0.9 0]; % Flow_discharge_constants
DischargeValve.Dam    = [280 463]; % Damping_Costant
DischargeValve.ffc    = [0.72 0.86 0.07]; % Flow_force_constants
DischargeValve.nu     = 8.9e-4; % Flow_force_constants
DischargeValve.fric   = 0.01;
DischargeValve.Zeta   = 1;
%% Create Backup of the script in the directory Back up
FileName      = mfilename;
% newbackup     = sprintf('%sbackup.m',FileName)
```

```
currentfile = strcat(FileName, '.m');
SimulationBackupFolder = [directory,FolderName,'\BackupData'];
copyfile(currentfile,SimulationBackupFolder)
fprintf('NB check the viscosity of the valve model in the valve Data ')
```

A.6 Fluids Characteristic

For the Slurry section:

```
% Fluid characteristic, considering all the principle value of the liquid
% and the vapour parameters for the slurry section
Slurry.rho      = 998; %[kg/m3] density  of the liquid
Slurry.p        = pressure(1,'bar','Pa');   %[Bar] Reference pressure of the fluid characteristic
Slurry.B        = 2.15e9; % bulk modulus
Slurry.nu       = 8.94e-4; % viscosity
Slurry.pv       = 2338; % Vapour pressure in Pa
Slurry.Temp     = 281; % tempreature of the simulation in K
Slurry.RVapour  = 461.5; % Gas vapour costant value
Slurry.RGas     = 287; % Gas Air constan value
Slurry.aGas     = 1e-10; % Non condensable gas fraction
Slurry.gamma    = 1; % 1 isotehrmal, 1.4 adiabatic of the disolved gas
g               = 9.81; % Gravity aceleration value
%% Create Backup of the script in the directory Back up
FileName     = mfilename;
```

```
% newbackup    = sprintf('%sbackup.m',FileName)
currentfile = strcat(FileName, '.m');
SimulationBackupFolder = [directory,FolderName,'\BackupData'];
copyfile(currentfile,SimulationBackupFolder)
```

For the propelling section:

```
% Fluid characteristic, considering all the principle value of the liquid
% and the vapour parameters for the propelling section
Propelling.rho      = 875; %[kg/m3] density  of the liquid
Propelling.p        = pressure(1,'bar','Pa');   %[Bar] Reference pressure of the fluid characteristic
Propelling.B        = 1.37e9; % nulk modulus
Propelling.nu       = 8.94e-4; % viscosity
Propelling.pv       = 2338; % Vapour pressure n Pa
Propelling.Temp     = 281; % tempreature of the simulation in K
Propelling.RVapour = 461.5; % Gas vapour costant value
Propelling.RGas     = 287; % Gas Air constan value
Propelling.aGas     = 1e-10; % Dissoved gas fraction
Propelling.gamma    = 1; % 1 isotehrmal, 1.4 adiabatic
%% Create Backup of the script in the directory Back up
FileName      = mfilename;
% newbackup    = sprintf('%sbackup.m',FileName)
currentfile = strcat(FileName, '.m');
SimulationBackupFolder = [directory,FolderName,'\BackupData'];
```

```
copyfile(currentfile ,SimulationBackupFolder)
```

A.7 Initialization Data

```
%% Initialization of simulation parameters
% |------------|
% |    Inlet    |
% |------------|
pIn            = pressure(pIn,'bar','Pa');   %Pa
[cIn, rhoIn]   = SpeedOfSound(pIn, Solver,Slurry);
uIn            = 0;
rhoQIn         = 0;
alphaVIn       = 0;
alphaGIn       = Slurry.aGas*(Slurry.p ./pIn ).^(1/Slurry.gamma);
% |----------------------|
% |    Chamber Slurry    |
% |----------------------|
pChaSlurry                   = pressure(pCha,'bar','Pa');   %Pa
[cChaSlurry, rhoChaSlurry]   = SpeedOfSound(pChaSlurry, Solver,Slurry);
uChaSlurry                   = 0;
rhoQChaSlurry                = 0;
alphaVChaSlurry              = 0;
alphaGChaSlurry              = Slurry.aGas*(Slurry.p ./pChaSlurry ).^(1/Slurry.gamma);
% |--------------------------|
% |    Chamber Propelling    |
% |------------------------------------------|
pChaPropelling                    = pressure(pCha,'bar','Pa');  %Pa
[cChaPropelling, rhoChaPropelling] = SpeedOfSound(pChaPropelling, Solver,Propelling);
uChaPropelling                    = 0;
rhoQChaPropelling                 = 0;
alphaVChaPropelling               = 0;
alphaGChaPropelling               = Propelling.aGas*(Propelling.p ./pChaPropelling ).^(1/Propelling.gamma);
% |--------------|
% |    Outlet    |
% |--------------|
pOut            = pressure(pOut,'bar','Pa');  %Pa
[cOut, rhoOut]  =  SpeedOfSound(pOut, Solver,Slurry);
uOut            = 0;
rhoQOut         = 0;
alphaVOut       = 0;
alphaGOut       = Slurry.aGas*(Slurry.p ./pOut ).^(1/Slurry.gamma);
%% Create Backup of the script in the directory Back up
FileName      = mfilename;
% newbackup     = sprintf('%sbackup.m',FileName)
currentfile = strcat(FileName, '.m');
SimulationBackupFolder = [directory,FolderName,'\BackupData'];
copyfile(currentfile,SimulationBackupFolder)
```

A.8 Valve dynamics

```
function [x, xdot, NetForce, Agap, CF] = ValveModel(p1,p2,rho,x0,xdot0,Vgap,Valve,Solver,NetForce0,t)
%==================================
%     FG <--||
%             ||
%             ||        /\
% P1          ||\      /  \     P2
%             ||  \   / <-- FS
%             ||   \/  <-- FS0
%             ||
%             ||
%==================================
%            |----> x
dt = Solver.dt;
% For Jonhston model the maximum area possible is when CF is equal to one
% then the value to not overcome is: a = A * cos(theta) /CD;
% Valve Dimension calculation;
dm        = 0.5 *(Valve.do + Valve.di);
AreaValve = CircleArea(dm);
g         = 9.81;
%% Gap Velocity
dp    = p1-p2;
% Bernulli Speed Velocity
switch Solver.Speed
    case 'Bernulli'
```

```
        Vgap = sqrt(2 * abs(dp) /rho) *sign(dp);
    case 'Coefficient'
        Vgap = sqrt(2 * abs(dp) /(rho * Valve.CDJ)) * sign(dp);
    case 'Eulerian'
    otherwise
        error('No Method specify')
end
%% Force Coefficient
switch Solver.Force
    case 'Thiel'
        dh = x0 * 2 * sin(deg2rad(Valve.theta));
        Re = abs(rho * Vgap * dh / Valve.nu) + 1e-6;
        CF = Valve.ffc(1) - Valve.ffc(2) * x0 / dm - Valve.ffc(3) * log10(Re);
    case 'Johnston'
        Agap =  pi * dm * x0 * sin(deg2rad(Valve.theta)) *(1-x0/(2*dm)*sin(2*(deg2rad(Valve.theta))));
        CF   = 1- 2 * Valve.CDJ * Agap / AreaValve * cos(deg2rad(Valve.theta)) + 2 * (Valve.CDJ * Agap / AreaValve)^2;
    case 'Constant'
        CF = Valve.CDJ;
    otherwise
        error('No Method specify')
end
%% Damping
switch Solver.Damping
```

```
    case 'Constant'
        C   = Valve.C;
    otherwise
        error('No Method specify')
end
%% Forces
Force(1) = AreaValve * CF * dp;     % Pressure Force
Force(2) = Valve.m * g + Valve.FS0; % Cosyant Force
Force(3) = C * xdot0;   % Damping Force
%% Spring Force
if strcmp(Solver.Spring,'Power')
    Force(4) = 1370323771520  * x0^6 -...
               185636068328   * x0^5 +...
               9625259842.50  * x0^4 -...
               231629611.63   * x0^3 +...
               2512028.98     * x0^2 +...
               1977.13        * x0   +...
               4.3383;   % Sproing Force
elseif strcmp(Solver.Spring,'Linear')
    Force(4) = Valve.k * x0;   % Sproing Force
elseif strcmp(Solver.Spring,'Cost-Linear')
    Force(4) = 209 + Valve.k * x0;   % Sproing Force
end
NetForce = Force(1) - sum(Force(2:4));
switch Solver.Integrate
    case 'AB'
        if t > 3 % ADAM BASHForth
            xdot = AdamBashforth(xdot0,NetForce/Valve.
                m,NetForce0/Valve.m,dt);
                x     = AdamBashforth(x0,xdot,xdot0,dt);
            else % EULER
                xdot  = Euler(xdot0,NetForce/Valve.m,dt);
                x    = Euler(x0,xdot,dt);
            end
        case 'EU'
            xdot  = Euler(xdot0,NetForce/Valve.m,dt);
            x    = Euler(x0,xdot,dt);
end
%% Limitation Of Valve
 if x >= Valve.xMax
       x = Valve.xMax;
       xdot = 0;
elseif x <= 0
       x    = 0;
       xdot = 0;
 end
 %% GAP AREA
Agap =  pi * dm * x * sin(deg2rad(Valve.theta)) *(1-x/(2*dm)*sin(2*(deg2rad(Valve.theta))));
```

A.9 Valve Gap Mass flow rate

```
function [rhoQgap, Vgap] = GapFlow(Agap,rho,fvN,aN,fv1,a1,rhoQgap0,dt,toll,Valve)
% rhoQgap =
% GapFlow(Agap,rho,CF,pN,rhoqN,cN,AN,p1,rhoq1,c1,A1,toll,Valve,section)
% The value of the rhowQgap is moltiply by the vlave seat angle, if this is
```

```matlab
% set to 0 than the flow is directly projected to the axial direction
%
% This function gives the flow value across the valve, evaluaing the
% magniture with newton raphson method limited to 100 simulation. If the
% limitation is exceed the value of the gap flow will be set to the last
% time step and declared in a print view.
% The parameters request to solve the function are:
% - Agap: Area of the gap calculated with the ValveModel
% - rho: the density of the flow that goes inside the valve, normally is
% given by the Suction density for the suction valve and chamber density
% for the discharge.
% - CF: coefficient of the loss, tipically calculated from the ValveModel
% - pN: pressure of the innet duct (suction for the suction valve, Chamber
% for the discharge)
% - rhoqN: flowrate of the innet duct (suction for the suction valve,
% Chamber for the discharge)
% - cN: speed of sound of the innet duct (suction for the suction valve,
% Chamber for the discharge)
% - AN, Area of the inner duct(suction for the suction valve,
% Chamber for the discharge)
% - p1: pressure of the Outlet duct (suction for the suction valve, Chamber
% for the discharge)
% - rhoq1: flowrate of the Outlet duct (suction for the suction valve,
% Chamber for the discharge)
% - c1: speed of sound of the Outlet duct (suction for the suction valve,
% Chamber for the discharge)
% - A1, Area of the Outlet duct(suction for the suction valve,
% Chamber for the discharge)
% - toll: tollerance of the solver method
% - Valve: Characteristic of the Valve, if use the notation of the Valve
% data will be SuctionValve or DischargeValve
% - section: 'discharge' or 'suction'%

p1  = fv1(1,1);
RQ1 = fv1(2,1);
c1  = fv1(6,1);
L12 = fv1(9,1);

pN  = fvN(1,end);
RQN = fvN(2,end);
cN  = fvN(6,end);
```

```
LN1 = fvN(8,end);

if Agap > 0
    dh = sqrt(4*Agap/pi);
    AK = (Valve.fric * Valve.lg / dh + Valve.Zeta + 1);
    BK = 2 * rho * Agap *  Valve.lg;
    CK = 2 * Agap^2 * rho;
    M = BK / dt + CK * (c1^2 / (a1*L12) - cN^2/(aN*LN1));
    W = CK * ((p1-pN) + cN^2/(aN*LN1)*RQN - c1^2/(a1*L12)*RQ1) - BK / dt * rhoQgap0;
    Err  = 1;
    rhoQguess = rand;
    while Err > toll
        f          = rhoQguess * (abs(rhoQguess) * AK + M) + W;
%           f          = rhoQguess * (abs(rhoQguess) + M/AK) + W/AK;
        df         = 2 * AK *abs(rhoQguess) + M;
%           df         = 2 *abs(rhoQguess) + M / AK ;
        rhoQgap    = rhoQguess - f/df;
        Err        = abs(rhoQguess - rhoQgap);
        rhoQguess = rhoQgap;
    end

    rhoQgap = rhoQguess;
    Vgap    = rhoQguess ./ (Agap .* rho);
else
```

```

    rhoQgap = 0;
    Vgap = 0;
end
```

A.10 Accumulator

```
function [pF, VF] = Accumulator(VT, VF, V0, FV, G, dt, A, pPreC, pMax, Fluid, KD, KS, fix)%
% The accumulator return the pressure pF and the volume of the fluid VF inside
% the accumulator itself. The algorithms consists into a pre charged gas,
% and can simulate a bladder, a piston or a diaphragm behavior.
%
%         |---------------------------|
%         |        ||          |      |
% FV ---|    VF    || VG          VD  |
%         |        ||          |      |
%         |---------------------------|
%
% The motion of the separator between the fluid chamber and the gas chamber
% is restricted by two hard stops that limit the expansion and contraction
% of the fluid volume. The fluid volume is limited when the fluid chamber
% is at capacity and when the fluid chamber is empty. The hard stops are
% modeled with finite stiffness and damping. This means that it is possible
% for the fluid volume to become negative or greater than the fluid chamber
% capacity, depending on the values of the hard-stop stiffness coefficient
% and the accumulator inlet pressure.
% However the volume can be limited, causing numerical issues.
%
% PARAMETERS
% VT - Total volume of the accumulator [m^3]
% VF - previous time step of the fluid volume [m^3]
% V0 - Initial condition of Gas VT - VF(1)  [m^3]
% FV - The Value of the previous time step of the pipe connected to the
% Accumulator
% G  - Gamma, politropic expansion
% dt - delta time  [sec]
% A  - Area of the pipe connected to the accumulator
% pRecC - PreCharge Gas pressure at V0 [Pa]
% pMax  - Max pressure of the accumulator [Pa]
% Fluid - Fluid struct description
% KD - [KDmax KDmin] Costant dampening pressure when the volume exceed the
% maximum value (VF > (VT-V0)) and the minimum Volume(VF < 0);
% KS - [KDmax KDmin] Costant stiffness pressure when the volume exceed the
% maximum value (VF > (VT-V0)) and the minimum Volume(VF < 0);
% fix - 'Yes' the limitation in the volume at minimum 0 L, 'limited0.1' at
% 10% of the V0 and 'limited0.2' at 20%
%
% further information:
% https://nl.mathworks.com/help/physmod/hydro/ref/gaschargedaccumulator.html
if nargin == 10
    KD = [1e6 1e6];
    KS = [1e6 1e6];
elseif nargin == 11
    if strcmp(KD,'Yes')
        KD = [1e6  1e6];
        KS = [1e6 -1e6];
        fix = 'Yes';
    end
elseif nargin == 12
    fix = 'No';
elseif nargin < 13
    error('Not enough input')
end
Kacc = pPreC*V0^G;
VD    = (Kacc / pMax)^(1/G); %Volume of dead
VC    = VT - VD;             % Max volume of the fluid
AK = VT - VF - dt / FV(5,end) * FV(2,end) + dt / FV(5,end) * A * FV(8,end) / FV(6,end)^2 * FV(1,end);
BK = dt / FV(5,end) * A * FV(8,end) / FV(6,end)^2;
```

```
Err = 1;
Tol = 1e-6;
p = rand;
while Err > Tol
    f  = p - Kacc / (AK-BK*p)^G;
    df = 1 - BK * G * Kacc * (AK-BK*p)^(-G-1);
    pG = p - f / (df+eps);
    Err = abs(p - pG);
    p   = pG;
end
rho  = Fluid.rho * exp(1/Fluid.B*(pG - Fluid.p));
QF   = (FV(2,end) + A * FV(8,end) / FV(6,end)^2 * (pG - FV(1,end))) / rho;%FV(5,end);
VF   = VF + dt * QF;
if strcmp(fix,'Yes')
    VF = VF * (VF>0);
elseif strcmp(fix,'limited0.1')
    if VF < (-0.1*V0)
        VF = 0;
    end
elseif strcmp(fix,'limited0.2')
    if VF < (-0.2*V0)
        VF = 0;
    end
end
QFp = QF * (QF > 0);
QFm = QF * (QF < 0);
ps = (KS(1)*(VF-VC)+KD(1)*QFp*(VF-VC)) * (VF>VC) + (KS(2)*VF-KD(2)*QFm*VF) * (VF < 0);
```

```
pF = pG + ps;
```

A.11 T Junction

```
function pTJ = TJunction(VarA, VarB, VarC, AA, AB, AC, type)
% [pT, rhoQT1, rhoQT2, rhoQT3] =
% TJunction(rhoQ1,p1,c1,A1,rhoQ2,p2,c2,A2,rhoQ3,p3,c3,A3,type)
%
% Return the junction pressure
% The type of juction must be specify: '1NN','NNN', , N11' and '111
% If there are 1 pipe that goes to the juction and two duct that start from
% that, the type will be 'N11', always the different value should be define
% first. Also in theat function is required a tollerance value, as default
% set to 1e-1 Pa, do to the numerical problem.
%
%
% Type N11        |      1NN
%         /       |        \
%  A  N  / 1 B    |   B   N \   1  A
% ------+         |          +------
%        \ 1 C    |   C   N /
%         \       |        /
Toll = 1e-2;
if strcmp(type,'1NN')
```

```
    num = double((VarA(2,1) - VarB(2,end) - VarC(2,end)) - AA * VarA(9,1)   / VarA(6,1).^2   * VarA(1,1) + AB * VarB(8,end) / VarB(6,end).^2 * VarB(1,end) + AC * VarC(8,end) / VarC(6,end).^2 * VarC(1,end));
    den = double(AB * VarB(8,end) / VarB(6,end).^2  + AC * VarC(8,end) / VarC(6,end).^2  - AA * VarA(9,1)   / VarA(6,1).^2    );
    pTJ = double(num / den);
elseif strcmp(type,'N11')
    num = double((VarA(2,end) -  VarB(2,1) - VarC(2,1)) - AA * VarA(8,end) / VarA(6,end).^2 * VarA(1,end) + AB * VarB(9,1) / VarB(6,1).^2   * VarB(1,1) + AC * VarC(9,1)   / VarC(6,1).^2   * VarC(1,1));
    den = double(AB * VarB(9,1) / VarB(6,1).^2 + AC * VarC(9,1) / VarC(6,1).^2 - AA * VarA(8,end) / VarA(6,end).^2  );
    pTJ = double(num / den);
elseif strcmp(type,'111')
    num = double((VarA(2,1) - VarC(2,1) - VarB(2,1)) - AA * VarA(9,1) / VarA(6,1).^2 * VarA(9,1) + AB * VarB(9,1) / VarB(6,1).^2 * VarB(9,1) + AC * VarC(9,1) / VarC(6,1).^2 * VarC(9,1));
    den = double(AB * VarB(8,1) / VarB(6,1).^2 + AC * VarC(8,1) / VarC(6,1).^2 - AA * VarA(8,1) / VarA(6,1).^2);
    pTJ = double(num / den);
elseif strcmp(type,'NNN')
```

```
    num = double((VarA(2,end) - VarC(2,end) - VarB(2,
        end)) - AA * VarA(8,end) / VarA(6,end).^2 *
        VarA(1,end) + AB * VarB(8,end) / VarB(6,end).^2
         * VarB(1,end) + AC * VarC(8,end) / VarC(6,end)
        .^2 * VarC(1,end));
    den = double(AC * VarC(8,end) / VarC(6,end).^2 +
        AB * VarB(8,end) / VarB(6,end).^2 - AA * VarA
        (8,end) / VarA(6,end).^2);
    pTJ = double(num / den);
elseif strcmp(type,'NN1')
    num = double((VarB(2,end) + VarC(2,end) - VarA
        (2,1)) + AA * VarA(9,1)    / VarA(6,1).^2    *
        VarA(1,1) - AB * VarB(8,end) / VarB(6,end).^2 *
         VarB(1,end) - AC * VarC(8,end) / VarC(6,end)
        .^2 * VarC(1,end));
    den = double(AA * VarA(9,1)    / VarA(6,1).^2 - AB
        * VarB(8,end) / VarB(6,end).^2 - AC * VarC(8,
        end) / VarC(6,end).^2);
    pTJ = double(num / den);
else
    error('Juction error definition')
end
%%
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

FloorValue = floor(pTJ);
CeilValue  = ceil(pTJ);
if abs((pTJ - FloorValue)) < Toll
    pTJ = FloorValue;
```

```
elseif abs((pTJ - CeilValue)) < Toll
    pTJ = CeilValue;
end
```

A.12 Junction

```
function pTJ = Junction(VarA, VarB, AA, AB)
% pTJ = Junction(VarA, VarB, AA, AB)
%
%             A                                  B
% -----------------------------|-------------------------------------
%                              |
%  --- N-1 ---|---  N  ---|---  1  ---|---  2  ---
%                              |
% -----------------------------|-------------------------------------
% This function returns the value of the pressure at the juction where VarA
% is the end extreme and VarB is the starting pipe.
% VarA and VarB are the Function value preallocated in memory, namely FV
% meanwhile the AA and AB are the area value
% there is a tollerance value impose at 1e-5 Pa, for which lower value will
% be set to ceil or floor value to prevent numerical error.
% It happened that for some pipe value, the variation
```

```
in pressure, even
% smaller, like noise value creare a water hammer effect that variate step
% by step the value of the mass flow rate. In same cases the code diverge

Toll = 1e-5; % <---
num = (VarA(2,end) - VarB(2,1)) + AB * VarB(9,1)   / VarB(6,1)^2   * VarB(1,1) - AA * VarA(8,end) / VarA(6,end)^2 * VarA(1,end);
den = AB * VarB(9,1)   / VarB(6,1)^2  - AA * VarA(8,end) / VarA(6,end)^2;
pTJ = num / den;
FloorValue = floor(pTJ);
CeilValue  = ceil(pTJ);
if abs((pTJ - FloorValue)) < Toll
    pTJ = FloorValue;
elseif abs((pTJ - CeilValue)) < Toll
    pTJ = CeilValue;
end
```

A.13 Speed of Sound

```
function [cL, rhoL] = SpeedOfSound(pL, Solver,Fluid,Dimension)
% [cL, rhoL] = SpeedOfSound(pL, Solver,Fluid,Dimension)
% gives the value of the liquid speed of sound and the density related with
% the algorithm of the Solver definition.
```

```
if nargin == 3
    if or(strcmp(Solver.LiquidSpeed,'Constant'),strcmp(Solver.LiquidSpeed,'Bulk'))
        Dimension = 0;
    else
        error('Non defined Dimension in Speed of Sound')
    end
end
switch Solver.LiquidSpeed
    case 'Constant'
        rhoL = Fluid.rho;
        cL   = Fluid.c;
    case 'Korteweg'
        beta = 2 / ((Dimension.d  + 2 * Dimension.th )^2 - Dimension.d^2) *((1-Dimension.nu)*Dimension.d ^2+(1+Dimension.nu)*(Dimension.d + 2 * Dimension.th)^2);
        rhoL = Fluid.rho * exp(1/Fluid.B*(pL - Fluid.p));
        cL   = sqrt(Fluid.B/Fluid.rho);
        cL  = sqrt(cL^2 /((1+ rhoL * beta * cL^2 / Dimension.E)));
    case 'Simpson'
        rhoL = Fluid.rho * exp(1/Fluid.B*(pL - Fluid.p));
        cL   = sqrt(Fluid.B/Fluid.rho);
        cL  = cL /(1+(Fluid.B / Dimension.E * Dimension.d / Dimension.th)*(1-Dimension.nu
            ^2));
    case 'Bulk'
        rhoL = Fluid.rho .* exp(1 ./ Fluid.B .*(pL - Fluid.p));
        cL   = sqrt(Fluid.B ./ Fluid.rho);
end
```

A.14 Mesh Grid

```
function [NodeX, NodeINT, dx, x, Xint, LF] = Staggrid(L0,LF,dx)
%
%[NodeX, NodeINT, dx, X, Xint] = Staggrid(L0,LF,dx)
%
% L0                          |<-dx-->|<->|dx/2   LF
% |---x---|---x---|---x---|---x---|---x---|---x---|---x---|---x---|
%     i=1     i=2     i=3                      i=N-1   i=N-1    i=N
%
% This algorithm create only even division of the section, and if the dx is
% not fullfil entierly the pipe, the length dx will be  added or removed to
% overcome the issues. It will be shown the amount of dimension changeg.
% Smaller is dx, smaller will be the different with the real dimension. The
% reason why is built like that is the possibility to
```

```
do not use
% interpolation scheme, because everywhere will be set the right number of
% nodes and the possibility to use the cavitation algorithm (that require
% even division of the pipe)

if LF < L0
    error('L must be bigger than L0');
end
Num_dx = ceil((LF - L0) / dx);
if mod(Num_dx, 2) ~= 0
    temp = LF;
    Num_dx = Num_dx + 1;
    LF = Num_dx * dx;
    fprintf(['Length change from ','%2.3f',' to ','%2.3f', ' [m] \n'],temp,LF)
end
NodeX    = Num_dx;
l0       = L0 + dx/2;
lF       = LF - dx/2;
x        = linspace(l0,lF,NodeX);
NodeINT  = NodeX+1;
Xint     = linspace(L0,LF,NodeINT);
end
```

A.15 MUSCL

```
function [VAR, FV] = MUSCL(dimension, X, XI, BC, BCinlet, BCoutlet, Solver, Fluid, FV)
```

```
% [VAR, FV] =
% MUSCL(dimension, X, XI, BC, BCinlet, BCoutlet, Solver, Fluid, FV)
% The function required:
% - Dimension: the characteristic of the duct
% - X: the vector of the node position
% - XI: the vector of the interface position
% - BC: the vector of the Boundary interface, the BC(1) is the value at the
% initial boundary meanwhile the BC(2) is the boundary at the N+1/2
% - Solver, the struct of the solver algorithms
% - Fluid: fluid characteristic
% - FV, namely the value of the parameters for the previous time step.
% for details check the paper of Zhou_2017B and Zhou_2017
% to understand the MUSCL algorithm.
 Toll = 1e-10;
SIZE = size(FV);
VAR  = zeros(SIZE(1),SIZE(2));
%% Variable calculated
Area      = CircleArea(dimension.d); % calculate the Area
%% Eingenvalue
% Eigenvalue calculate for the system of the pressure and flow rate
if strcmp(Solver.Convective,'No')
    Lambda(1,:) = -FV(6,:); % first no ceonvective
```

```
method eigenvalue
    Lambda(2,:) = FV(6,:); % second no ceonvective method eigenvalue
else
    Lambda(1,:) = (FV(7,:)- sqrt(FV(7,:).^2 + 4 * FV(6,:).^2 ))/2; % First
    Lambda(2,:) = (FV(7,:)+ sqrt(FV(7,:).^2 + 4 * FV(6,:).^2))/2; % Second
end
%% Boundary
% Depending on the Value set as boundary, automatically will calculate the value of the other parameter
% INLET
if strcmp(BCinlet,'p') % Calculate the inlet boundary condition  in case of pressure definition
    BDI(1) = BC(1);
    BDI(2) = FV(2,1) +  Area * Lambda(2,1) ./ FV(6,1).^2 * (BDI(1) - FV(1,1));
elseif strcmp(BCinlet,'fr') % Calculate the inlet boundary condition  in case of Flowrate definition
    BDI(2) = BC(1);
    BDI(1) = FV(1,1) + FV(6,1).^2 ./ (Area * Lambda(2,1)) * (BDI(2) - FV(2,1));
else
    error('Error to inlet Boundary condition')
end
% OUTLET
if strcmp(BCoutlet,'p') % Calculate the Outlet
```

```
    boundary condition  in case of pressure definition
    BDO(1) = BC(2);
    BDO(2) = FV(2,end) +  Area * Lambda(1,end) ./ FV(6,end).^2 * (BDO(1) - FV(1,end));
elseif strcmp(BCoutlet,'fr') % Calculate the outlet boundary condition  in case of Flowrate definition
    BDO(2) = BC(2);
    BDO(1) = FV(1,end) + FV(6,end).^2./   ( Area * Lambda(1,end))* (BDO(2) - FV(2,end));
else
     error('Error to outlet Boundary condition')
end
% etrapolation the others variable that are calculated by the first two
% Inlet
BDI(3) = extrapolation(FV(3,1:2), XI(1), X(1:2),'inlet','yes'); % alpha
BDI(4) = extrapolation(FV(4,1:2), XI(1), X(1:2),'inlet','yes'); % alphaG
BDI(5) = extrapolation(FV(5,1:2), XI(1), X(1:2),'inlet','yes'); % rho
BDI(6) = extrapolation(FV(6,1:2), XI(1), X(1:2),'inlet','yes'); % c, speed of sound
BDI(7) = extrapolation(FV(7,1:2), XI(1), X(1:2),'inlet','n');   % speed of fluid
% Outlet
BDO(3) = extrapolation(FV(3,end-1:end), XI(end), X(end-1:end),'outlet','yes'); % alpha
BDO(4) = extrapolation(FV(4,end-1:end), XI(end), X(end
-1:end),'outlet','yes'); % alphaG
BDO(5) = extrapolation(FV(5,end-1:end), XI(end), X(end-1:end),'outlet','yes'); % rho
BDO(6) = extrapolation(FV(6,end-1:end), XI(end), X(end-1:end),'outlet','yes'); % c, speed of sound
BDO(7) = extrapolation(FV(7,end-1:end), XI(end), X(end-1:end),'outlet','n');   % Speed of fluid
%% Interface Exrtrapolation
u     = zeros(SIZE(1),SIZE(2)+4);
Ind   = 3:SIZE(2)+2;
% adding two ghost cells at each edges
for i = 1:SIZE(1)-2
    u(i,1:2)                  = BDI(i); %define the inlet boundary condition
    u(i,SIZE(2)+3:SIZE(2)+4) = BDO(i); %define the outlet boundary condition
    u(i,Ind)                  = FV(i,:); %define the interior boundary condition
end
Ind      = 2:SIZE(2)+3;
SizeInd = length(Ind);
% Slope Limiter Function
Di = SlopeLimiter(u,Solver);
uL = u(1:2,Ind) - Solver.dx/2 * Di;
uR = u(1:2,Ind) + Solver.dx/2 * Di;
%% Evolution
% calculate the Eigenvalue at all the grid points
if strcmp(Solver.Convective,'No')
    Eig(1,1:SizeInd) = - u(6,Ind);
    Eig(2,1:SizeInd) =   u(6,Ind);
else
    Eig(1,1:SizeInd) = (u(7,Ind) - sqrt(u(7,Ind).^2 + 4 * u(6,Ind).^2))/2;
    Eig(2,1:SizeInd) = (u(7,Ind) + sqrt(u(7,Ind).^2 + 4 * u(6,Ind).^2))/2;
end
% Evaluating the Jacobian Matrix
A(1,1,1:SizeInd) = 0;
A(1,2,1:SizeInd) = u(6,Ind).^2 ./ Area;
A(2,1,1:SizeInd) = Area;
if strcmp(Solver.Convective,'No')
    A(1,1,1:SizeInd) = 0;
    A(1,2,1:SizeInd) = u(6,Ind).^2 ./ Area;
    A(2,1,1:SizeInd) = Area;
    A(2,2,1:SizeInd) = 0;
else
    A(1,1,1:SizeInd) = 0;
    A(1,2,1:SizeInd) = u(6,Ind).^2 ./ Area;
    A(2,1,1:SizeInd) = Area;
    A(2,2,1:SizeInd) = u(7,Ind);
end
%% Caracteristics formulation
ML(1,1,1:SizeInd) = Eig(1,1:SizeInd) ./ (Eig(1,1:SizeInd) - Eig(2,1:SizeInd));
ML(1,2,1:SizeInd) = u(6,Ind).^2 ./ (Area .* (Eig(2,1:SizeInd) - Eig(1,1:SizeInd))) ;
ML(2,1,1:SizeInd) = Area .* Eig(1,1:SizeInd) .* Eig(2,1:SizeInd) ./ (u(6,Ind).^2  .* (Eig(1,1:SizeInd)
 - Eig(2,1:SizeInd)));
ML(2,2,1:SizeInd) = Eig(2,1:SizeInd)./(Eig(2,1:SizeInd) - Eig(1,1:SizeInd));

MR(1,1,1:SizeInd) = - Eig(2,1:SizeInd) ./ (Eig(1,1:SizeInd) - Eig(2,1:SizeInd));
MR(1,2,1:SizeInd) = - u(6,Ind).^2 ./ (Area .* (Eig(2,1:SizeInd) - Eig(1,1:SizeInd)));
MR(2,1,1:SizeInd) = - Area .* Eig(1,1:SizeInd) .* Eig(2,1:SizeInd) ./ (u(6,Ind).^2  .* (Eig(1,1:SizeInd) - Eig(2,1:SizeInd)));
MR(2,2,1:SizeInd) = - Eig(1,1:SizeInd) ./ (Eig(2,1:SizeInd) - Eig(1,1:SizeInd));
%% MUSCL
% interface evolution of the Boundary cell interface
for i = 1:length(uL)
    UL(:,i) = uL(1:2,i) + 0.5 * Solver.dt/Solver.dx * A(:,:,i) * (uL(1:2,i) - uR(1:2,i));
    UR(:,i) = uR(1:2,i) + 0.5 * Solver.dt/Solver.dx * A(:,:,i) * (uL(1:2,i) - uR(1:2,i));
end
%Flux at the interface
for i = 1:length(UL)-1
    fU(:,i)   = A(:,:,i) * ML(:,:,i) * UR(:,i) + A(:,:,i+1) * MR(:,:,i+1) * UL(:,i+1);
end
% Evaluation of the
for i = 1:2
    FV(i,:) = FV(i,:) - Solver.dt/Solver.dx * diff(fU(
i,:));
end
% Pressure at the Variable FV cannot be lower that the Vapour pressure
switch Solver.Cavitation
    case 'DGCM' % DISCRETE GAS CAVITY MODEL
        [FV, VAR] = DGCM(FV,VAR,dimension,Solver,Fluid);
    case 'DVCM'% DISCRETE Vapour CAVITY MODEL interpolated
        [FV, VAR] = DVCM(FV,VAR,dimension,Solver,Fluid);
    otherwise % No Cavity evaluation
        [FV, VAR] = DGNC(FV,VAR,dimension,Solver,Fluid);
end
%% Friction
Re = abs(mean(FV(7,:) .* dimension.d .* FV(5,:) ./ Fluid.nu)); % Reynold averaged value from the previous time step
f  = friction(Re,dimension,Solver);
%% RK4
f_termRK1 = - 1/8 * f .* Dimension.d .* FV(5,:) .*  FV(7,:) .* abs(FV(7,:)) ./ k0(I2);
g_termRK1 = - 9.81 .* FV(5,:) .* Dimension.A .* sin(deg2rad(Dimension.theta));
RK1      = Solver.dt * (f_termRK1 + g_termRK1);

rhoQ_K2    = FV(2,:) + RK1/2;
```

```
vRK2       = rhoQ_K2 / (FV(5,:) * Dimension.A);
f_termRK2 = - 1/8 * f .* Dimension.d .* FV(5,:) .* vRK2 .* abs(vRK2) ./ k0(I2);
g_termRK2 = - 9.81 .* FV(5,:) .* Dimension.A .* sin(deg2rad(Dimension.theta));
RK2        = Solver.dt * (f_termRK2 + g_termRK2);

rhoQ_K3    = FV(2,:) + RK2/2;
vRK3       = rhoQ_K3 / (FV(5,:) * Dimension.A);
f_termRK3 = - 1/8 * f .* Dimension.d .* FV(5,:) .* vRK3 .* abs(vRK3) ./ k0(I2);
g_termRK3 = - 9.81 .* FV(5,:) .* Dimension.A .* sin(deg2rad(Dimension.theta));
RK3        = Solver.dt * (f_termRK3 + g_termRK3);

rhoQ_K4    = FV(2,:) + RK3;
vRK4       = rhoQ_K4 / (FV(5,:) * Dimension.A);
f_termRK4 = - 1/8 * f .* Dimension.d .* FV(5,:) .* vRK4 .* abs(vRK4) ./ k0(I2);
g_termRK4 = - 9.81 .* FV(5,:) .* Dimension.A .* sin(deg2rad(Dimension.theta));
RK4        = Solver.dt * (f_termRK4 + g_termRK4);


FV(2,:) = FV(2,:) + (RK1 + 2*RK2 + 2*RK3 + RK4)/6;
FV(7,:) = FV(2,:) ./ (Dimension.A .* FV(5,:)); % speed
%% Update data
% Update the mass flowrate with friction
FV(2,:)   = FV(2,:) + Solver.dt / 6 *(RK1 + 2* RK2 + 2*
    RK3 + RK4);
FV(2,:)  = FV(2,:) .* (abs(FV(2,:)) > Toll); % Limitation of the sporious oscillation due to numerical data
VAR(2,:) = FV(2,:); % Flowrate

% Speed of the Fluid
FV(7,:)  = FV(2,:) ./ (Area .* FV(5,:)); % speed
VAR(7,:) = FV(7,:); % speed

% Eighenvalue
if strcmp(Solver.Convective,'No')
    FV(8,:) = -FV(6,:); % First
    FV(9,:) =  FV(6,:); % Second
else
    FV(8,:) = (FV(7,:) - sqrt(FV(7,:).^2 + 4 * FV(6,:).^2))/2; % First
    FV(9,:) = (FV(7,:) + sqrt(FV(7,:).^2 + 4 * FV(6,:).^2))/2; % Second
end

end
```

A.16 Friction

```
function f = friction(Re,Dimension,Solver,rough)
%  f = friction(Re,Dimension,Solver)
% function that gives back the friction factor, considering as the limit of
% turbolance and laminar 2300 Re. Refered with the
```

```
solver method, different
% approaches can be applyed. The roughness is set as default 1e-5
% in case of use Darcy algorithm, a subroutine is performed when the
% friction became really small, with iteration step set to 0.0005.
% The following apprach can be used:
% - Constant, with friction value equal to 0.1
% - Moody
% - Wood
% - Eck
% - Swamee
% - Churchill
% - Jain
% - Chen
% - Round
% - Haaland
% - Darcy
if nargin == 3
    rough = 1e-5;
end
E = rough ./ Dimension.d;
if Re == 0
    f = 0;
elseif Re < 2300
    switch Solver.Friction
        case 'Null'
            f = 0;
        otherwise
            f = 64 / Re;
    end
else
    switch Solver.Friction
        case 'Constant'
            f = .01;
        case 'Moody'
            f = 0.0055 * (1 + (2e4 * E + 1e6 / Re)
               ^(1/3));
        case 'Wood'
            Psi = 1.62 * E^.134;
            f = 0.094 * E^0.225 + 0.53 * E + 88 * E
               ^.44 * Re^(-Psi);
        case 'Eck'
            f = (1/(- 2 * log10( E / 3.715 + (6.943 /
               Re)^.9))).^2;
        case 'Swamee'
            f = 0.25 / (log10( E/3.7 + 5.74 / Re .^
               0.9)).^2;
        case 'Churchill'
            f = ((1) / ( - 2 * log10(E / 3.71 + (7 /
               Re) ^.9))).^2;
        case 'Jain'
            f = ((1) / ( - 2 * log10(E / 3.715 +
               (6.943 / Re) ^.9))).^2;
        case 'Chen'
            f = ((1) / ( - 2 * log10(E / 3.7065 -
               5.0452 /  Solver.Re * log10(1/2.8257 *
                E^1.1098 + 5.8506 / Re^0.8981)))).^2;
        case 'Round'
            f = ((1) / ( 1.8 * log10(Re / (0.135 * Re
                * E + 6.5)))).^2;
        case 'Haaland'
            f = ((1) / ( - 1.8 * log((E / 3.7).^1.11 +
                6.9 / Re))).^2;
        case 'Darcy'
            B = 2.51 / Re;
            Err  = 1;
            toll = 1e-8;
            A = rough / (3.7* Dimension.d);
            fg   =  0.005;
            temp = fg;
            it = 0;
            while Err > toll
                fun  = 1 / sqrt(fg) + 2 * log(A + B /
                    sqrt(fg+eps));
                dfun = - 1 / (2 * fg^(3/2)) - B / (fg
                    ^(3/2) * (B/(sqrt(fg+eps) + A))) ;
                f = fg - fun/(dfun+eps);
                if f <= 0 % Algorithm to prevent
                    negative data
                    it = it +1;
                    f = temp - it*0.0005;
                end
                Err  = abs(fg - f);
                fg   = f;
            end
            otherwise
                f = 0;
      end
end
```

A.17 Integration

The Adam-Bashforth algorithm is:

```
function [x1] = AdamBashforth(x0,dxdt0,dxdt_p,dt)
% Adam Bashforth integration method second order in
    time
%
% [x1] = AdamBashforth(x0,dxdt0,dxdt_p,dt)
%
% required the the value of the previous time step x0,
     and the derivatieve
% information dxdt0, the derivative of the previous
    time step dxdt_p
% and the time step dt to obtain the function x1

x1  = x0 + 3/2* dt * dxdt0 - 0.5 * dt* dxdt_p;
end
```

The Euler algorithm is:

```
function [x1] = Euler(x0,dxdt,dt)
% Euler Explicit solver first order accuracy
%
% [x1] = Euler(x0,dxdt,dt)
%
% required the the value of the previous time step x0,
    and the derivatieve
% information dxdt and the time step dt to obtain the
   function x1

x1  = x0 + dt * dxdt;
end
```

Appendix B

Non Dimensional formulation

The momentum equation of the water hammer formulation for incompressible fluid, neglecting the source and the advective terms is given in equation B.1

$$\frac{\partial Q}{\partial t} + \frac{A}{\rho}\frac{\partial p}{\partial x} = 0 \tag{B.1}$$

Considering the spatial derivative as an incremental ratio, the equation B.1 takes the form shown in equation B.2.

$$\frac{\partial Q}{\partial t} + \frac{A}{\rho \Delta x} p = 0 \tag{B.2}$$

The Δx is the suction pipe length and can be defined as l. Rearranging the equation function B.2 result in:

$$\frac{pA}{\dot{Q}\rho l} = 1 \tag{B.3}$$

This equation can be approximate with the pump parameters. In fact the volume flow rate is directly proportional to the piston velocity profile, namely $\dot{Q} \propto \dot{v_p} A$. The equation of the derivative of the piston velocity is given by the formulation of the acceleration, reported in B.4.

$$\ddot{x}_p \approx r\omega^2[cos(\omega t + \theta_0) - \lambda cos(2\omega t + 2\theta_0)] \tag{B.4}$$

The acceleration is a sinusoidal function proportional to the power of the angular velocity that normally can be expressed as stroke rate per minute (SPM). Therefore, it is possible to consider the variation in the volume flow rate as $\dot{Q} \propto SPM^2 A$.

Finally the formulation of non dimensional treatment can be achieved in the form of normalized equation B.5.

$$\frac{p}{SPM^2 \rho l} = 1 \tag{B.5}$$

Appendix C

Friction non dimensional formulation

The momentum of equation is given in the equation C.1:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\frac{\partial \boldsymbol{u}}{\partial x} + gsin\alpha + \frac{f\pi\phi}{\rho A} = 0 \tag{C.1}$$

Using the following scale:

- Time: $\tilde{t} = \frac{\zeta L}{c}$, where ζ is a positive real parameter that takes into account the reflection time of a wave in a duct.
- Length: $\tilde{x} = c\tilde{t}$.
- Pressure: $\tilde{p} = \rho_0 c\boldsymbol{u}_0$, where ρ is the density and the subscription 0 is related to the steady condition.
- Wall stress: $\tilde{f} = \frac{\rho f \boldsymbol{u}^2}{8}$.
- Density: ρ_0;
- Vorticity time diffusion: T_D.

The momentum equations is rewritten in the form of equation C.2

$$\frac{\partial \tilde{\boldsymbol{u}}}{\partial \tilde{t}} + M\tilde{\boldsymbol{u}}\frac{\partial \tilde{\boldsymbol{u}}}{\partial \tilde{x}} + \frac{\rho_0}{\rho}\frac{\partial \tilde{p}}{\partial \tilde{x}} + g\frac{\zeta L}{\boldsymbol{u}c}sin\alpha + \frac{\zeta L}{\phi}M\frac{f}{2} + \zeta\Big(\frac{T_d}{L/c}\Big)\tilde{f} = 0 \tag{C.2}$$

The friction non dimensional term is therefore $\Gamma = \frac{\zeta LMf}{2\phi} + \frac{\zeta T_d}{L/c}$ [78].