University of South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2020

Process Technology

Evaluation of De-oiler Performance Using Numerical CFD Technique

Yovita Hariyanti

Faculty of Technology, Natural Sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis, 2020

Title: Evaluation of De-oiler Performance Using Numerical CFD Technique

Number of pages: 77

Keywords: De-oiler, De-oiling hydrocyclone, Computational Fluid Dynamics, Hydrocyclone, OpenFOAM, TwoPhaseEulerFoam, Oil-water separation, Liquid-liquid separation, Euler-Euler multiphase model.

Student: Yovita Hariyanti

Supervisor: Joachim Lundberg, Ph.D.

External partner: Subsea 7

Availability: Confidential

Summary:

When producing oil from the reservoir, it comes together with water, gas, sand, and other impurities. Excluding gas, all of them are unexpected waste. The water or simply called produced water is the biggest problem due to its quantity and oil content. A produced water treatment should be performed before it is discharged to the sea or re-injected into the wells. External partner of this thesis, Subsea 7 is now developing a subsea separation technology called SPU (Subsea Processing Unit). SPU consists of some separation units, one of them is a de-oiler/hydrocyclone. This thesis will only focus on the de-oiler unit to reduce oil concentration in the produced water to be under 20-30 ppmv.

A literature review is conducted to choose de-oiler geometry based on separation efficiency. SALOME is used to draw and make a de-oiler mesh. The CFD simulation is conducted using OpenFOAM with the Euler-Euler model. *twoPhaseEulerFoam* solver is modified from its original example called *bubbleColumn*. The literature review has been done to examine various geometry and types of de-oiler for oil and water. Based on efficiency, the geometry for simulation is selected. From the results, the oil volume fraction in the water is found to be high which leads to low efficiency. The pressure drop is obtained to be very low compared to most of the literature, which is 0.02272 bar for 54.696 L/min and 0.0424 bar for 82.045 L/min. With the assumption of efficiency 76%, the number of de-oilers needed to separate 60000-80000 barrels of produced water per day varies between 25-162.

The University of South-Eastern Norway takes no responsibility for the results and conclusions in this student report.

Preface

First I would like to express my gratitude to my supervisor Dr. Joachim Lundberg for his willingness to be my supervisor and believed me to do this project. Thank you for your guidance, patience, help, time, and support during the thesis work.

Second I would like to thank Sigbjørn Daasvatn and Christian Wathne from Subsea 7 who give me the opportunity to work on this very interesting project, support me, and provide me a space in the office and information with regards to the thesis.

Third, I would like to thank my wonderful parents, Lay Sin Hoa and Yohanes Hariyanto for their never-ending love and financial support, my brother Geoffrey Hariyanto, and my sister Stephani Hariyanti for their support from far away in Indonesia.

Finally, I would also use this opportunity to thank my fellow master students and seniors at USN campus Porsgrunn and thank all my friends in Porsgrunn, Stavanger, and Indonesia for their love and caring.

Porsgrunn, 1st June 2020

Yovita Hariyanti

Contents

Nomenclature

Latin letters

Symbol	Explanation	Units
$\bar{M}_{\varphi}$	averaged inter-phase momentum transfer term	kg m s^{-1}
$\overline{Re}_{\varphi}^{eff}$	Reynolds stress	N m^2
$\overline{Re}_{\varphi}^{effC}$	Correction component of Reynolds stress term	N m^2
$\overline{Re}_{\varphi}^{effD}$	Diffusive component of Reynolds stress term	N m^2
$\vec{U}_{\varphi}$	Average velocity of phase φ	m s^{-1}
$\bar{U}_{\varphi}^{T}$	total phase velocity	m s^{-1}
$\tilde{V}_i$	Feed velocity	m s^{-1}
$\dot{m}_{HPO,oil}$	Mass flow rate of oil in heavy phase outlet	kg s^{-1}
$\dot{m}_{HPO,water}$	Mass flow rate of water in heavy phase outlet	kg s^{-1}
$\dot{m}_{HPO}$	Mass flow rate in heavy phase outlet	kg s^{-1}
$\dot{m}_{LPO,oil}$	Oil mass flow rate in light phase outlet	kg s^{-1}
$\dot{m}_{LPO,water}$	Water mass flow rate in light phase outlet	kg s^{-1}
$\dot{m}_{LPO}$	Mass flow rate in light phase outlet	kg s^{-1}
$\dot{m}_{feed,oil}$	Mass flow rate of oil in feed	kg s^{-1}
$\dot{m}_{feed,water}$	Mass flow rate of water in feed	kg s^{-1}
$\dot{m}_{feed}$	Mass flow rate of feed	kg s^{-1}
h_1	height at position 1 (in Bernoulli's equation)	m

h_2	height at position 2 (in Bernoulli's equation)	m
A_1	cross sectional area position 1 see Figure 2.19	m^2
A_2	cross sectional area position 2 see Figure 2.19	m^2
A_c	Area calculated using D_c	m^2
A_i	Cross-sectional area of feed inlet	m^2
A_o	overflow or vortex finder inner wall cross sectional area	m^2
C_d	Drag coefficient	-
D_c	biggest de-oiler diameter i.e. diameter of the cylinder section	m
D_c	cyclone wall at its maximum I.D., that is, in the cylindrical section	m
D_f	equivalent diameter of feed inlet.	m
D_i	Diameter of the de-oiler at the inlet level	m
D_o	Diameter assume 0.25Dc	m
E_{ds}	Separation efficiency/efficiency	-
K_{ds}	concentration of dispersed phase (oil) in HPO	%-vol
K_i	oil concentration in the feed	%-vol
M_D	Drag force	Newton
P_1	pressure at position 1 (in Bernoulli's equation)	Pa
P_2	pressure at position 2 (in Bernoulli's equation)	Pa

P_{feed}	Pressure in the feed/inlet stream	bar or psig or Pa
$P_{overflow}$	Pressure in the overflow	bar or psig or Pa
$P_{underflow}$	Pressure in the underflow	bar or psig or Pa
$Q_{HPO,oil}$	Volumetric flow rate of oil in heavy phase outlet	$m^3 h^{-1}$ or $m^3 s^{-1}$
$Q_{HPO,water}$	Volumetric flow rate of water in heavy phase outlet	$m^3 h^{-1}$
Q_{HPO}	Volumetric flow rate in heavy phase outlet stream	$m^3 h^{-1}$ or $cm^3 s^{-1}$
$Q_{LPO,oil}$	Volumetric flow rate of oil in light phase outlet	$m^3 h^{-1}$ or $cm^3 s^{-1}$
$Q_{LPO,water}$	Volumetric flow rate of water in light phase outlet	$m^3 h^{-1}$ or $cm^3 s^{-1}$ or $m^3 s^{-1}$
Q_{LPO}	Volumetric flow rate in light phase outlet stream	$m^3 h^{-1}$ or $cm^3 s^{-1}$
$Q_{feed,oil}$	Volumetric flow rate of oil in feed stream	$m^3 h^{-1}$ or $cm^3 s^{-1}$
$Q_{feed,water}$	Volumetric flow rate of water in feed stream	$cm^3 s^{-1}$
Q_{feed}	Feed volumetric flow rate	$m^3 h^{-1}$ or gal min^{-1} or $m^3 s^{-1}$
R_w	Water flow ratio	-
$\overline{U}$	Average velocity, $\alpha_1\overline{U}_1 + \alpha_2\overline{U}_2$	$m s^{-1}$
U_1	Bubble velocity	$m s^{-1}$

U_1'	r.m.s of the fluctuations in the velocity of dispersed phase	m s^{-1}
U_2	Fluid velocity	m s^{-1}
U_2'	r.m.s of the fluctuations in the velocity of continuous phase	m s^{-1}
U_r	Relative velocity	m s^{-1}
V_k	volume of phase k	m^3 or liter
W_{oi}	Oil mass flowrate at the feed stream	kg s^{-1}
W_{oo}	Oil mass flowrate at the overflow	kg s^{-1}
$X_{HPO,oil}$	Oil concentration in heavy phase outlet (volume/volume)	-
$X_{feed,oil}$	Oil concentration in feed (volume/volume)	-
c_{water}	Water mass fraction in feed	-
dP_{oil}	Pressure drop between feed and overflow stream	bar or psig or Pa
dP_{water}	Pressure drop between feed and underflow stream	bar or psig or Pa
f_s	Flow split-ratio	-
$\bar{p}$	Average pressure	bar or psig or Pa
u_1	axial velocity at position 1 (in Bernoulli's equation)	m s^{-1}
u_2	axial velocity at position 2 (in Bernoulli's equation)	m s^{-1}
$u_{axial,feed}$	inlet axial velocity	m s^{-1}
u_{axial}	axial velocity	m s^{-1}

u_{mag}	magnitude velocity	m s^{-1}
u_{tang}	tangential velocity	m s^{-1}
C	Pressure drop constant	
D	Bubble diameter	μm
J	factor in de Gelder pressure drop correlation	-
Q	Volumetric flow rate	m^3 s^{-1}
Re	Reynold's number	-
V	total volume	m^3 or liter
d	vector between center of the cell P and N	
g	Gravitational force	m s^{-2}
r	radius of de-oiler	m

Greek letters

Symbol	**Description**	**Unit**
$\langle\alpha\rangle_{\nabla}$	notation for averaging procedure	-
$\nabla_f^{\perp}\phi$	gradient at the face f	-
ϕ_{φ}^{T}	total phase flux	-
C_t	turbulence response function	-
α_1	Volume fraction of dispersed phase	-
α_2	Volume fraction of continuous phase	-
α_k	Volume fraction for phase k	-
α_{φ}	Volume fraction for phase φ	-

$\eta_{dispersed}$	Dispersed efficiency	-
ν_2	kinematic viscosity of continuous phase	$m^2\ s^{-1}$
ν^t	turbulent kinematic viscosity	$m^2\ s^{-1}$
ν_φ	is the kinematic viscosity for phase φ	$m^2\ s^{-1}$
ν_φ^{eff}	effective viscosity	Pa s
ξ_∞	factor in de Gelder pressure drop correlation	-
ρ_1	Dispersed phase density	$kg\ m^{-3}$
ρ_2	Continuous phase density	$kg\ m^{-3}$
ρ_{mix}	Mixture of oil and water density	$kg\ m^{-3}$
ρ_{oil}	Oil density	$kg\ m^{-3}$
ρ_{water}	Water density	$kg\ m^{-3}$
ρ_φ	Density of phase φ	$kg\ m^{-3}$
ϕ_φ	Volumetric phase flux	-
ΔP	pressure drop	Pa or barg or psi
Δt	time	s or min
G	Grade efficiency	-
α	Ratio of the tangential velocity to the inlet velocity	-
β	Cone angle	^{o}
δ	small stabilising factor	-
η	Efficiency	-
ξ	dimensionless constant which is similar to the loss coefficient	-

π	Ratio of a circle circumference to its diameter	-
φ	Phase (dispersed or continuous)	-

1 Introduction

When producing oil from the reservoir, it comes together with water, gas, sand, drilling fluids, and drilling cuttings. Excluding gas, all of them are waste that is unexpected. The water or simply called produced water causes the biggest problem due to its quantity. On average, during the lifetime of reservoir one barrel of oil is produced together with four barrels of water. A separation process should be performed to remove the non-commercial water and preferably conducted near to the well to avoid transportation cost of water. Therefore, the idea of doing a subsea separation is considered. According to Skjefstad & Stanko (2019) and Aadal et al. (2016), subsea separation can increase production rates. (Campen, 2014, p. 2) (Young, G. A. B., Wakley, W. D., Taggart, D. L., Andrews, S. L., & Worrell, J. R., 1994, p. 37) (Aadal et al., 2016, p. 6-7) (Skjefstad & Stanko, 2019, p. 203)

After the separation process, the produced water can be dumped overboard with a restriction of small oil concentration in water, typically 20-30 ppm. Current oil and gas industries often perform gas, oil, and water (GOW) separation in a large vessel. This is because the operation works continuously, and long residence time is needed to perform the gravity-based separation. This type of separator might not be suitable in the subsea level because in the depth water thicker walls are needed and this will impact high installation cost. Some challenges then appear to have a subsea separation process. Space and weight limitation of the separation unit is one of them, therefore, it is critical to consider a compact and highly efficient subsea separation process. (Das & Jäschke, 2018, p. 138) (Campen, 2014, p. 2) (Skjefstad & Stanko, 2019, p. 204) (Liu, 2012, p. 116)

External partner of this thesis, Subsea 7 is now developing a subsea separation technology called SPU (Subsea Processing Unit). GOW separation will start from a harp pipe, an inclined pipe, a pipe in pipe separator, to a de-oiler (hydrocyclone). This thesis will focus on the de-oiler to purify the produced water. De-oiler uses a centrifugal force to separate two-phase feed (immiscible liquid-liquid) which makes it more compact, simple, and less weight. Besides, the advantages of the de-oiler are lack of motion sensitivity, no moving parts, cost-effectiveness, small installation footprint, and relatively easy to scale by adding units in a parallel manner. (Young et al., 1994, p. 37) (Bram, Hansen, Hansen, & Yang, 2018, p. 131) (Liu, 2012, p. 117)

To have a better understanding of how good the de-oiler separating performance, numerical simulation i.e. Computational Fluid Dynamics (CFD) is chosen to estimate the flow field. An analytical calculation cannot predict the flow field in centrifugal de-oiler due to its complexity. Besides, compared to experimental works, using CFD can save time, cost, and efforts when running several simulations with various operational conditions and geometries of hydrocyclone. (Slot, 2013, p. 7) (Kharoua, Khezzar, & Nemouchi, 2010, p. 752)

The method used in this work is firstly conducting a literature review to point out a good geometry and performance of de-oiler. After choosing geometry and type of cyclone based on separation efficiency, Computational Fluid Dynamics (CFD) simulations are done using OpenFOAM to find out the oil concentration of cleaned produced water. Two different inlet velocities are chosen to know how these can affect the pressure drop value.

The main goals of the project are to evaluate various types of hydrocyclones based on the efficiency and the geometry which is suitable for the subsea separation process, simulate CFD model for hydrocyclone, update the geometry of the hydrocyclone according to CFD

simulation results, make a recommendation regarding the optimal hydrocyclone geometry for varying process conditions. The complete task description can be found in Appendix A.

Chapter 2 consists of theories regarding physics behind the hydrocyclone, produced water as de-oiler feed, types and geometry of de-oiler, pressure drop and velocity, operating de-oiler in marginal fields and turn down conditions, the solver used in OpenFOAM, process control, and lastly, subsea specific design issue.

Chapter 3 covers the problem description, governing equation and numerical implementation, numerical schemes, and case variations.

Chapter 4 will tell about the CFD simulation: pre-processing, simulation, and post-processing.

Chapter 5 and 6 cover the results and discussion and finally chapter 7 will conclude all the work has been done.

2 Theory

Chapter 2 will generally explain a hydrocyclone in general, produced water, the mathematical formulation of the solver used, types of de-oiler and their geometry, operating conditions in the marginal oil fields, process control of de-oiler, and finally, the subsea specific design issues.

2.1 Hydrocyclone

"Cyclones are simple devices used to separate a dispersed phase from a continuous phase based on centrifugal force." If the continuous phase is a liquid, it is called hydrocyclone. Hydrocylone separates two phases of different densities with the aid of a strong centrifugal force generated by the swirling flow. The swirling flow arises because of the rotating momentum obtained by the tangential inlet duct or guide vanes. This increases the velocity magnitude. Soon after the flow reaches a tapered cylinder, rotating velocity increases moderately. Later, the flow will separate into two spiral flow namely vortex. The inner vortex (forced vortex) which has a smaller diameter will move towards an overflow part of the hydrocyclone while the outer vortex (free vortex) will discharge through the underflow, see Figure 2.1. Furthermore, the rotation intensity will decay along the hydrocyclone. (Kharoua, Khezzar, & Nemouchi, 2010, p. 738) (Liu, Chen, Cai, Han, & Xiong, 2018, p. 60, 62-63) (Saidi, Maddahian, & Farhanieh, 2012, p. 1827)

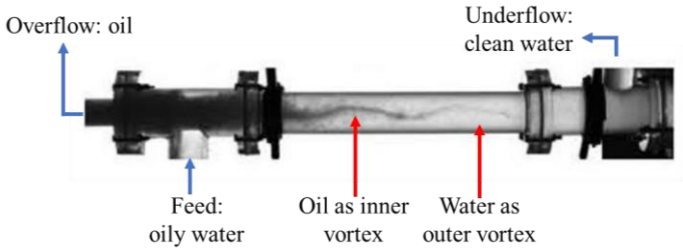


Figure 2.1: Hydrocyclones for oil/water separation. (Adapted from Liu et al., 2012, p. 118)

The hydrocyclone discussed in this work will focus on oil and water separation, especially for the de-oiling purpose. Later, the term *de-oiler* will be used in this work. Often in oil and gas operation, a conventional gravity separator is used to separate oil and water. Rely on gravity force, the magnitude of separation is 1 g while using de-oiler the separation magnitude can get as big as 2000-3000 g. However, a small difference of density in the de-oiler, droplet breakup and coalescence will make separation process more challenging. Further discussion of de-oiler types will be discussed later in sub-chapter 2.3. (Husveg, Johansen, & Bilstad, 2007, p. 294)

2.1.1 Efficiency

When comparing various types of hydrocyclone, one of the most important parameters is the efficiency. It must be seen carefully when comparing the efficiency because the definition can vary. The separation efficiency according to Liu et al. (2012, p. 121) mainly depends on the geometry of the hydrocyclone, operational parameters, and liquid physical properties. Using analysis dimensional, she agreed that efficiency is a function of Reynold's number and flow split-ratio. Here, the flow split-ratio f_s is the ratio between the overflow and inlet liquid

flow rate. Later in sub-chapter 2.6, the flow split ratio will be explained as a parameter to control a de-oiler.

Dispersed efficiency counts for the total amount of water and oil exit through the wrong outlet, it is defined by Campen (2014, p. 37) in equation (2.1) where Q is the volumetric flow rate [m^3/s]; LPO (low phase outlet) and HPO (heavy phase outlet) are referred to Light Phase Outlet and Heavy Phase Outlet, respectively.

$$\eta_{dispersed} = 1 - \frac{Q_{LPO,water} + Q_{HPO,oil}}{Q_{feed}} \tag{2.1}$$

Next, work of Simms, Zaidi, Hashmi, Thew, & Smyth (1992, p. 300) defined the separation efficiency E_{ds} as in equation (2.2). Equation (2.2) is mentioned as a common definition of de-oiler efficiency. Taking an example of water and sand hydrocyclone, the amount of sand separated from water can be described as a yield, however how dry the sand is another thing to describe. Therefore, for sand-water hydrocyclone yield and quality were defined. Yield is how much sand in mass can be separated and quality is how much water follows the sand in the outlet stream. Back to oil-water hydrocyclone or de-oiler, the yield is considered important. (Husveg, 2007, p. 20)

$$E_{ds} = 1 - \frac{K_{ds}}{K_i} \tag{2.2}$$

where K_{ds} and K_i are the concentration of dispersed phase (oil) in HPO and oil concentration in the feed, respectively.

Grade efficiency by (Braga, Huziwara, Martignoni, Scheid, & Medronho, 2015, p. 116) was defined in equation (2.3).

$$G = \frac{W_{oo}}{W_{oi}} \tag{2.3}$$

where W_{oo} and W_{oi} are oil mass flowrate at the overflow and inlet stream.

Next, the most common efficiency definition (2.2) is used to compare the de-oiler performance.

2.2 Produced water as de-oiler feed

Produced water consists of formation water and flood water. Also, it may be contained gas production and condensed water. Formation water exists naturally in the reservoir together with the oil and gas as shown in Figure 2.2. Meanwhile, the flood water is the reinjected produced water mixed with freshwater and/or seawater. Reinjecting produced water into the reservoir is performed to maintain the reservoir's pressure. (Miller, 1996, p. 1007) (Fluor Offshore Solutions, 2012, sec. 1, p. 2)

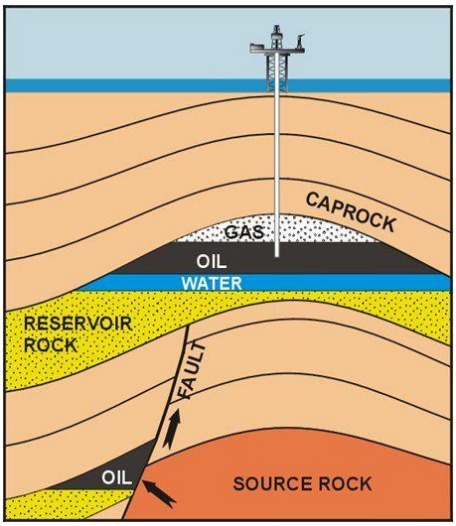


Figure 2.2: Formation water in a reservoir. (Bilstad, Nair, & Protasova, n.d., p. 18)

"Produced water is a complex mixture of organic and inorganic compounds and the largest volume of by-product generated during oil and gas recovery operations." In general, produced water contains dissolved and dispersed oil components, dissolved mineral, production chemicals, produced solids, and dissolved gases. Dissolved oil components are including BTEX (benzene, toluene, ethylbenzene, and xylene) and phenols whilst dispersed oil is PAHs (polyaromatic hydrocarbons) i.e. less soluble in water. (Igunnu & Chen, 2012, p. 157-159) (Fluor Offshore Solutions, 2012, sec. 3, p. 2)

Dissolved minerals are classified as cations and anions. For instance, Cl^- and Na^+ are an anion and cation which cause salinity. SO_4^{2-}, CO_3^{2-}, HCO_3^-, K^+, Ca^{2+}, Ba^{2+}, Mg^{2+}, Fe^{2+}, and Sr^{2+} are responsible for conductivity and scale formation. To avoid scale formation, production chemicals namely scale inhibitor is added. A lot of unexpected things happen when extracting gas/oil/water from the reservoir, such as wax, scale, and hydrate formation, bacterial growth, foam production, and corrosion. Therefore, other production chemicals are added such as hydrate inhibitor to prevent hydrate formation, emulsion breaker to improve separation, corrosion inhibitor to prevent corrosion, etc. (Igunnu & Chen, 2012, p. 157-159)

Moreover, produced water also contains produced solid such as sand and silt, clays, precipitated solid, corrosion and scale products, and dissolved gases such as CO_2, O_2, and H_2S. These gases are from chemical reactions, bacterial activities, or naturally existed. (Igunnu & Chen, 2012, p. 159)

As mentioned in chapter 1, the focus on this work is to treat produced water by occupying the de-oiler. The feed of the de-oiler is the downstream of Pipe-in-Pipe (PiP) separator which is used for treating bulk water separation. This means the oil contained in the downstream of PiP separator will not be high and there will be a very little amount of sands because the is a desander before the PiP separator. The complete diagram of the subsea separation train is illustrated in Figure 2.3. The oil concentration at the inlet of de-oiler will be around less than 0.1 to 10.5%. (Xodus Group, 2020, p. 6, 22)

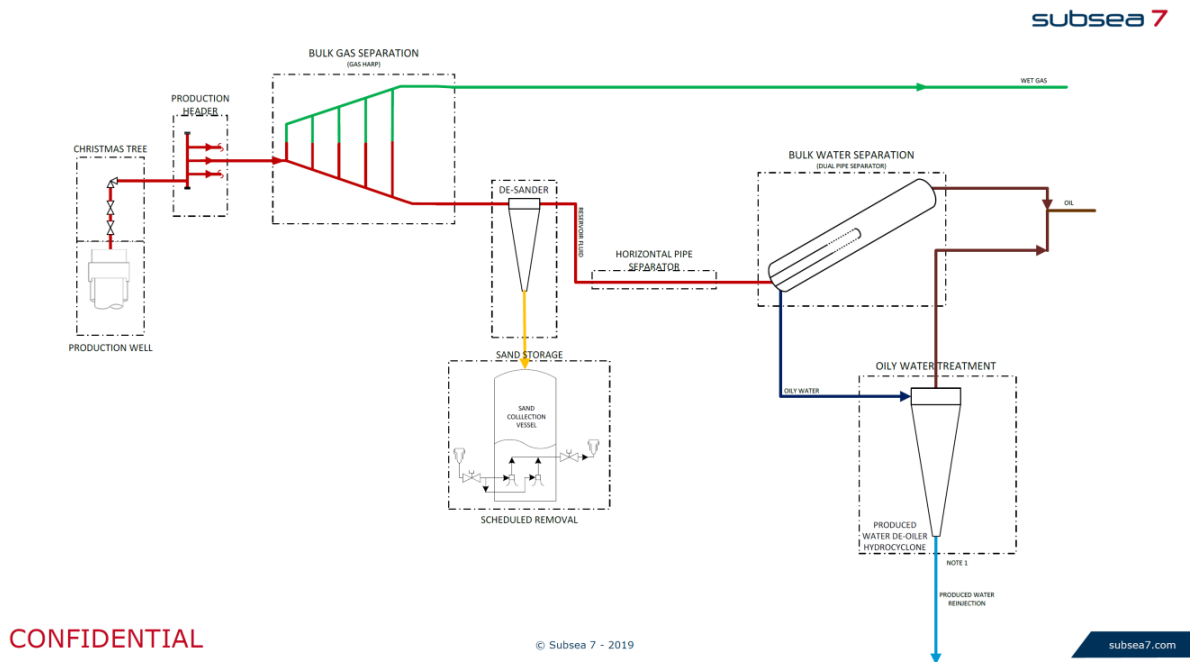


Figure 2.3: Process flow diagram of the proposed subsea separation train. (Xodus Group, 2020, p. 6)

Dealing with two immiscible liquids can be very difficult if one experiences a turbulence or agitation manner. In this condition, the shear force will break the dispersed phase into small droplets and form an emulsion. The emulsion is avoided as much as possible when separating oil and water because it prevents separation in a reasonable time. Its natural tendency to coalesce between these small droplets, however, this can take several mechanisms: sedimentation, aggregation, and coalescence. (Pettersen, 2008, p. 19)

The possibility of droplets to breakup and coalescence leans on many aspects: interfacial film, the existence of electrical/steric barriers, the viscosity of continuous phase, droplet size, temperature, pH, brine salinity, oil type, phase volumetric ratio. Due to the breakup and coalescence phenomenon, it is complicated to predict the size of oil droplets. There is a mathematical modelling technique called Population Balance Modelling (PBM) to predict the mean size of oil droplets. (Pettersen, 2008, p. 19-20)

However, in this work, a uniform droplet size will be used to reduce complexity and this was also done by de Araújo, Scheid, Loureiro, Klein, & Medronho (2020, p. 3) in their work.

2.3 De-oiler

De-oiler is a specific hydrocyclone that separates oil from water, in this case, the oil has a smaller portion compared to water. According to Bram et al. (2018), de-oiler is cost-effective because it has no moving parts. De-oiler has a small installation footprint and can be scaled up easily by adding more de-oilers in a parallel configuration. However, due to its simple design, the separation process will depend on the flow conditions. Besides, the de-oiler requires only small spaces and it is easy to manufacture, install, operate, and maintain. (Bram, 2018, p. 133) (Wolbert, Ma, & Aurelle, 1995, p. 1395) (Braga et al., 2015, p. 115)

There are some designs of de-oiler that had been developed, including axial and tangential de-oiler. Next, these two kinds of the de-oiler will be discussed.

2.3.1 Types of de-oiler

Based on the previous works of de-oiler design, types of de-oiler are listed in Table 2.1. The types are distinguished by their inlet geometry, swirl tube geometry, and flow direction. The following discussion in sub-chapter 2.3.1.1 and 2.3.1.2 will be mostly about the previous work and design of tangential and axial de-oiler. Then, in sub-chapter 2.3.2 the variables which have a strong influence on the de-oiler performance are explained.

Table 2.1: Hydrocyclone classification. (Campen, 2014, p. 3)

Type of de-oiler according to:	**Type**	
Inlet geometry	Tangential The inlet flow is in a radial direction	Axial The inlet flow is in an azimuthal direction
Swirl tube geometry	Traditional Has conical part(s)	Cylindrical Has no diameter reduction
Flow direction	Counter-current Heavy phase exits through the downstream side while light phase exits by the upstream part	Co-current Both phases exit through the downstream side

2.3.1.1 Tangential cyclone

Firstly, an important parameter for tangential de-oiler is the inlet configuration as shown in Figure 2.4. The main purpose is to provide strong tangential velocity while preventing the breakup of oil droplets. Twin inlets are considered as the best among others because it provides a stability of oil cores, however, a single inlet is preferably in production point of view. (Dirkzwager, 1996, p. 26) (Carlos et al., 2002, p. 354)

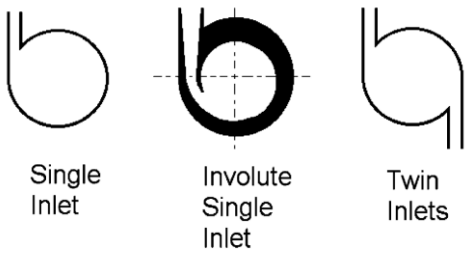


Figure 2.4: Inlet design (configuration) of tangential de-oiler. (Carlos et al., 2002, p. 354)

Secondly, a de-oiler design made by Colman and Thew inspired some works including work of Wolbert, Ma, & Aurelle (1995, p. 1396) and Young et al. (1994, p. 38). The intention of

the Colman and Thew's de-oiler was to clarify two liquids which have narrow density difference. As seen in Figure 2.5, the hydrocyclone consists of 1st conical part, 2nd conical part, and a cylindrical part. The 2nd conical part and the cylindrical part are added to increase the residence time. The principal is to have a swirling flow once the feed comes in a tangential direction. After swirling along the de-oiler, the concentrated emulsion exits through the overflow outlet, while the clarified output will flow towards the underflow outlet. (Wolbert, Ma, & Aurelle, 1995, p. 1396)

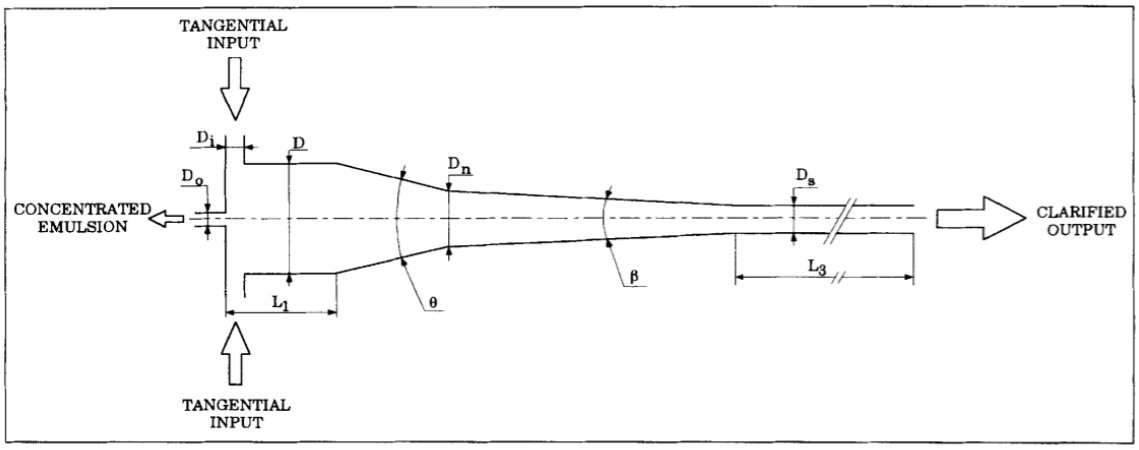


Figure 1. Colman and Thew's hydrocyclone design.
$D_n/D = 0.5$; $D_s/D = 0.25$; $D_i/D = 0.175$; $D_o/D < 0.05$; $L_1/D = 1$; $L_3/D = 15$; $\theta = 20°$; $\beta = 1.5°$.

Figure 2.5: Hydrocyclone design made by Colman and Thew. (Wolbert, Ma, & Aurelle, 1995, p. 1396)

Thirdly, a recent work in 2020 by de Araújo, Scheid, Loureiro, Klein, & Medronho (2020) analyzed a de-oiler using CFD simulation and experimental work. They focussed on varying 7 geometrical variables and used a software namely Design Expert to set up which computational experiments should be done. They carried out 17 computational experiments and decided to build a de-oiler based on the best grade efficiency. Then experimental works were conducted and compared with CFD analysis results. However, the oil volume fraction used in their work was much bigger compared to this thesis work. 40% of oil volume fraction made a big difference in the oil droplet diameter i.e. 250 μm. The de-oiler scheme is depicted in Figure 2.6. (p. 2, 6-7)

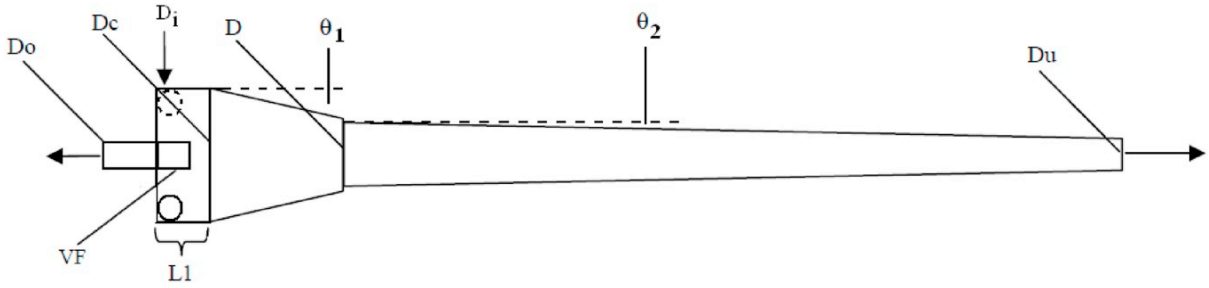


Figure 2.6: Hydrocyclone design made by de Araújo et al. with constant Dc = 3 cm. (de Araújo et al., 2020, p. 2)

Next, another work conducted by Braga et al. (2015, p. 115-118) used an ANSYS CFD simulation to develop a de-oiler. The principle is similar to Colman and Thew's

hydrocyclone. In his work, the overflow diameter Do and underflow diameter Du shown in Figure 2.7 were varied.

The de-oiler performance was measured using grade efficiency and water flow ratio. Grade efficiency was defined as a ratio between the oil mass flow rate at the overflow and the feed stream. A water flow ratio is the ratio of water mass flow rate at the overflow and the inlet flow. High grade efficiency and low water flow ratio are surely desired. In this study, the oil droplet size of 25 μm was used. After doing nine simulations in 1.13 seconds with time step 10^{-4}, the results were used to do statistical analysis, namely response surface and desirability function. The best overflow and underflow diameter were 10 and 15 mm, respectively. This geometry resulted in a grade efficiency of 72% and a water flow ratio of 62%. The complete geometry size of the de-oiler is listed in Appendix D. (Braga et al., 2015, p. 118, 122)

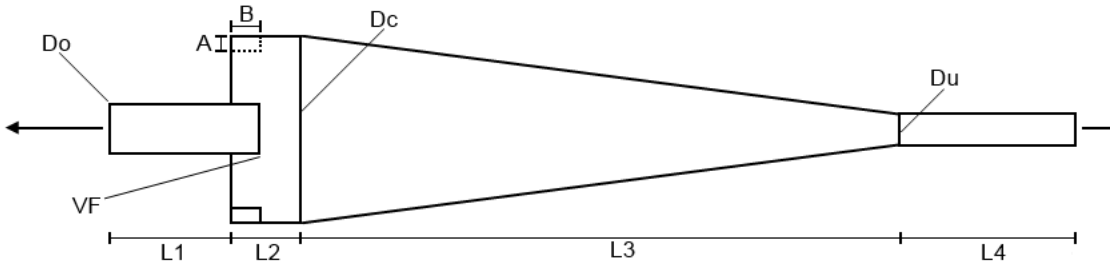


Figure 2.7: Hydrocyclone design developed by utilizing CFD simulation. (Braga et al., 2015, p. 117)

Lastly, work was done by Young et al. (1994) on investigating the effects of cylindrical length, cone angle, underflow length, underflow size, and feed size. A cylindrical length is required to reduce high shear stress when the flow comes through the inlet. By using a short cylindrical section near the inlet, the fluid experiences minimum drag force between the fluid and the cylinder wall. Increasing cylindrical length reduces separation efficiency. Concerning the cone angle, the bigger angles were proved to minimize the loss of angular momentum. In their conclusion, around 6^o of the cone angle was chosen as the best angle to perform separation in various flow conditions. The best geometry ratio is shown in Table 2.2. The variables are shown in Figure 2.8 (Young et al., 1994, p. 37, 44, 49)

Table 2.2: Geometry ratio of the de-oiler. (Young et al., 1994, p. 47)

Variable	**Value**
lu/Dc	18
du/Dc	0.33
di/Dc	0.25
α	6^o

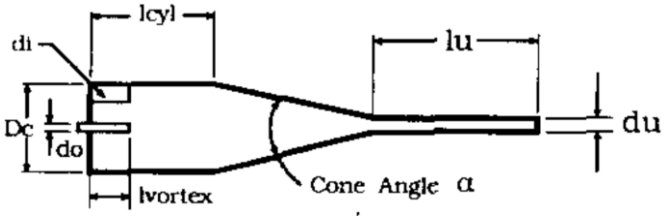


Figure 2.8: De-oiler geometry by Young et al. (1994, p. 40)

2.3.1.2 Axial inlet de-oiler

In the axial de-oiler, the inlet flow goes in the axial direction, hence it needs a swirl element to help it creates a swirling motion of the fluid. The swirl element itself consists of guide vanes attached to the tapering solid-body as depicted in Figure 2.9. A swirl element is mounted to the wall of the de-oiler body. The de-oiler body is called a swirl tube and the length of the swirl tube is proportional to the residence time. As illustrated in Figure 2.10, a smaller tube exists to pass the light phase outlet (LPO) namely pick-up tube. (Dirkzwager, p. 26, 28-29) (Campen, 2014, p. 71)

Figure 2.9: Strong swirl element (left) and weak swirl element (right). (Campen, 2014, p. 72)

A good example can be taken from Campen (2014, p. 65). As reported by him, the swirl element is the most critical part to design an axial de-oiler. The effect of three different types of swirl elements on the efficiency was studied. Strong and weak swirl elements were used in the experiments with the condition as follows: the volumetric flow rate is 56 m^3/h in a 170 cm long swirl tube with a diameter 10 cm and 50 mm pick-up tube. The different volumetric flow rate was chosen to conduct experiments using a large swirl element i.e. 30 m^3/h. The flow split used was equal to the oil/feed volumetric ratio. A strong swirl element generates a strong vortex, but it was found out that the weak swirl element showed the best dispersed efficiency. (Campen, 2014, p. 65, 71, 127-129, 159)

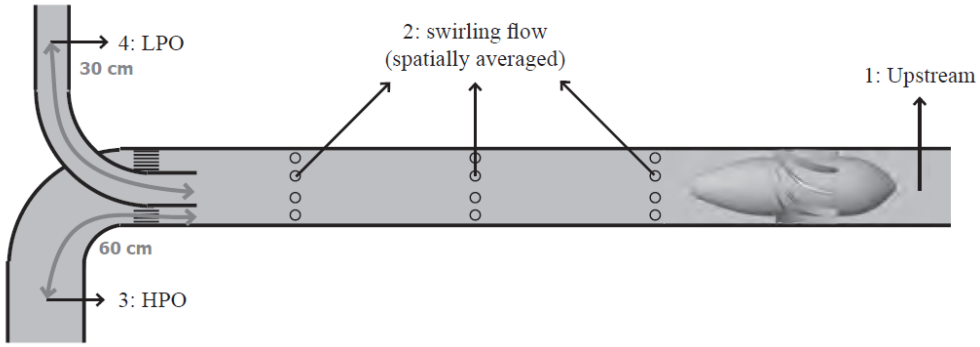


Figure 2.10: Axial cyclone with a swirl element and counter-current flow. (Campen, 2014, p. 124)

The constant swirl tube diameter was set to 10 cm even though the diameter size could affect separation. The oil droplets move on the radial distance to be collected in the middle. Increasing the diameter will raise the radial distance which droplets should travel, therefore make the separation process harder. However, a small diameter will raise the drag force which happened between the wall and oil droplets. This tears the droplets apart. (Campen, 2014, p. 156, 158)

The swirl tube length depends on the axial velocity of the oil droplets, therefore, it was varied from 110 cm to 190 cm. Analyses were performed on different swirl tube lengths using the weak swirl element. The longer the swirl tube length, the longer the residence time is. In contrast, swirl intensity decreases along the swirl tube. The optimum swirl tube length was 150 to 170 cm. Dispersed efficiency was gained approximately 88% for oil cut in feed 0.15 and volumetric flow rate 56 m^3/h. (Campen, 2014, p. 131)

Another work was conducted by Slot (2013, p. 35-36) who did a preliminary study of de-oiler geometry to see the advantages and drawbacks of it. Slot used long calculations and CASCADE software to design the vane shape for a certain tangential velocity. The length of the vane geometry can be various. The disadvantage of having longer vane is leading a droplet breakup while the advantage is structurally stronger mounted to the de-oiler wall.

Pressure drop and separation performance were inspected by conducting some analyses. Both variables were not strongly affected by changing the inlet directions from tangential to axial with guide vanes. Similarly, changing flow direction from co-current to counter-current affected neither pressure drop nor separation performance. Also, it was stated that the differences experienced by different geometry should not be emphasized because there was no clear outline among the comparison. As demonstrated in Figure 2.11, the final design includes a pick-up tube with 5 cm diameter, swirl element, flow straightener, a swirl tube with 10 cm diameter, and 1.7 m length. (Slot, 2013, p. 47-48)

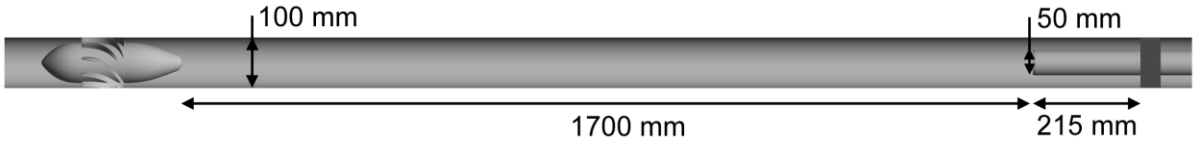


Figure 2.11: In-line separator with swirl element, pick-up tube, and flow straightener. (Slot, 2013, p. 48)

2.3.2 Influential variables for de-oiler performance

The performance of the de-oiler is strongly affected by three different factors: operational variables, feed physical characteristics, and de-oiler geometry. Some operational factors that affect separation performance are flow rate and flow split ratio. Besides, the physical characteristics of feed such as density difference between both phases and droplet size distribution have contributions to efficiency. Lastly, the de-oiler geometry impact on separation will be described. (Kharoua, Khezzar, & Nemouchi, 2010, p. 742) (Campen, 2014, p. 132)

Operational variables

Flow rate

De-oiler separation efficiency as a function of flow rate can be seen in Figure 2.12. Increasing a feed flow rate will raise the separation efficiency until a certain value, after that the efficiency will decrease drastically due to the increase of shear stress. The shear stress leads to droplets break-up and emulsifications. Besides, the too low flow rate will also decline the separation performance. (Kharoua, Khezzar, & Nemouchi, 2010, p. 743) (Liu et al., 2012, p. 123)

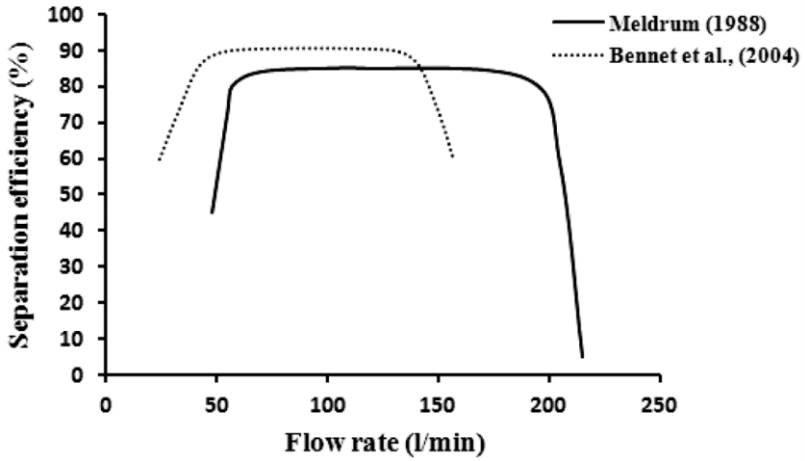


Figure 2.12: The effect of flow rate on the hydrocyclone separation efficiency. (Kharoua, Khezzar, & Nemouchi, 2010, p. 743)

Flow split-ratio

An experimental study was conducted by Liu et al. (2012, p. 122) using a cylindrical cyclone to investigate optimum flow split-ratio. The flow split-ratio is a ratio between the overflow and the mixture flow rate. The overflow and mixture flow rate can be seen in Figure 2.13. It was found that there was an optimum flow split-ratio to get maximum efficiency. According to Kharoua, Khezzar, & Nemouchi (2010, p. 744), too low flow split-ratio means less oil passes through overflow flow while too high flow split-ratio causes more water present in overflow.

Kharoua et al. (2010, p. 744) mentioned that the typical value of flow split-ratio is 2-3%. Similarly, Meldrum (1988, p. 671) agrees that the reject ratio around 1% gives very high oil-removal efficiencies. Reject ratio was defined as the ratio of overflow and underflow rate.

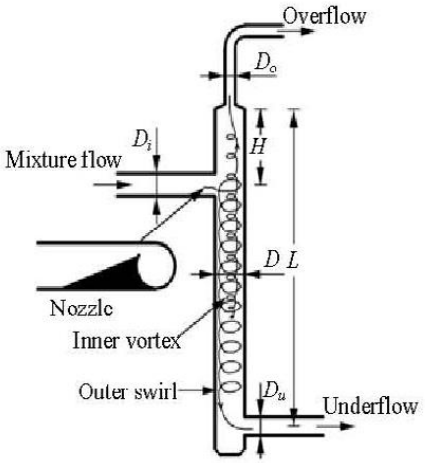


Figure 2.13: Cylindrical cyclone design. (Liu et al., 2012, p. 116)

Feed physical characteristics

Droplet size

Meldrum (1988, p. 670) mentioned that the bigger droplets will move faster to the inner vortex and a higher concentration of oil helps the droplets to coalescence and increase the mean size of oil droplets. Besides, Campen (2014, p. 141-142) studied the effect on droplet size using oil droplet median size 100, 300, and 500 μm with a flow rate of 10 m^3/h. As depicted in Figure 2.14, the oil droplet size has a significant effect on separation efficiency. The oil droplet of 500 μm has the best dispersed efficiency while 100 μm has the lowest dispersed efficiency. Therefore, droplet breakup should be avoided to maintain a better separation efficiency.

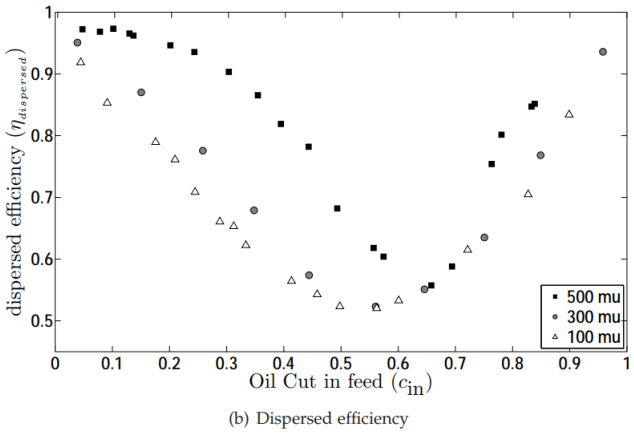


Figure 2.14: Droplet size affects the separation efficiency in a de-oiler. (Campen, 2014, p. 142)

Other physical characteristics such as a density difference and viscosity also affect the separation performance. These were briefly explained by Kharoua et al. (2010, p. 745). Oil viscosity decreases when the de-oiler feed temperature is increased. It was reported that the separation efficiency improved from 55.5% to 98% when heating the crude oil from 20 to 40^{o}C. However, no significant improvement happened when rising the crude oil temperature to the range of 40-60^{o}C. The density difference between the two phases is the driving force

for separation, the bigger the difference the faster the separation will take place. (Meldrum, 1988, p. 670)

Geometry

Cylindrical section

A cylindrical section is required to reduce high shear stress when the flow comes through the inlet. By using a short cylindrical section near the inlet, the fluid will not lose its angular momentum caused by the drag force between the fluid and the cylinder wall. (Young et al., 1994, p. 44)

Cone angle

Several experiments conducted by Young et al. (1994, p. 44-45, 47) presented that the cone angle of 3 and 1.5^{o} had a worse separation performance compared to 6, 9.5, and 20^{o}. These bigger angles were proved to minimize the loss of angular momentum. The loss of the angular momentum is due to the drag force between the fluid and the wall. In their conclusion, around 6^{o} of the cone angle was chosen as the best angle to perform separation in various flow conditions.

Blade deflecting angle

A study by Liu et al. (2018, p. 64) of axial de-oiler for pre-separation of oil and water shows that a bigger deflecting angle will result in a higher separation efficiency along with small or moderate tapered angle. However, the bigger deflecting angle will also produce a greater pressure drop. In their study, using 72^{o} blade deflecting angle, the highest efficiency can be reached is around 99% together with pressure drop above 350 kPa.

Blade height

Variations of five different blade heights were examined to know their relation to efficiency and pressure drop. The result was decreasing the blade height will rise separation efficiency as well as the pressure drop. Efficiency around 80% was attained by 10 mm blade height with more than 225 kPa of pressure drop. (Liu et al., 2018, p. 64)

2.3.3 Selecting de-oiler geometry

Three potential de-oilers from section 2.3.1 are selected to be compared to each other. Their performance is assessed based on the efficiency defined in equation (2.2).

Firstly, work from Braga et al. (2015, p. 115) used a CFD simulation to find the suitable overflow and underflow diameter to get the best separation efficiency and the flow ratio. The feed condition was oil and water with oil mass fraction 1%, 25 microns droplets size, and 1.8 L/s flow rate. The geometry used in this work is shown in Figure 2.15. Braga et al. (2015, p. 122) observed ten simulations using the Euler-Euler multiphase model with different overflow and underflow diameter.

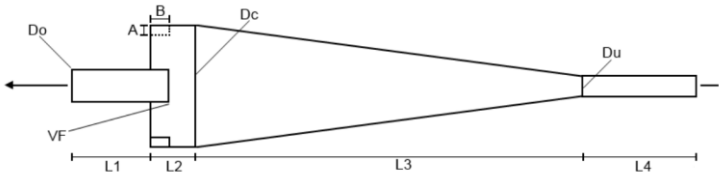


Figure 2.15: De-oiler design. (Braga et al., 2015, p. 117)

As mentioned in the sub-chapter 2.1 efficiency formula (2.2) is used to compare de-oiler performances in different articles. The grade efficiency is 72% while the efficiency of the de-oiler is 67.5%. The important findings from this article are: increasing the overflow diameter will rise grade efficiency while increasing underflow diameter will reduce the grade efficiency. (Braga et al., 2015, p. 119, 122)

Secondly, an axial de-oiler is seen to be a potential de-oiler to investigate more detail. The article by Campen (2014) provided detailed information about geometry, efficiency, and flow condition. The dispersed efficiency was used to describe how well the separation was conducted in axial hydrocyclone. The cyclone was used to separate 56.5 m^3/h with feed oil cut 0.125 and was reported to have dispersed efficiency 87.8%. Using an efficiency formula mentioned in equation (2.2) the efficiency is 51.7%. (Campen, 2014, p. 131-132)

Lastly, an article by Young et al. (1994) used the geometry ratio shown in Table 2.2. The median oil droplet size was 50 μm and inlet volumetric flow rate 50 gpm. The efficiency is found to be 73%. (Young et al., 1994, p. 47)

After comparing three de-oilers above, the most suitable de-oiler for subsea separation is the tangential cyclone by Young et al. (1994). The main reason is it has the best separation efficiency i.e. 73%. Besides, the tangential de-oiler is more simple to design compared to the axial de-oiler.

2.4 Pressure drop and velocity

A de-oiler exploits the fluid pressure energy to develop the separation power, therefore there must be a loss of pressure or often called a pressure drop. In the CFD simulation, the pressure field is gained at many different points along the de-oiler. However, investigating whether the value is correct or not is a must thing to do. In this work, no experiment is committed, so, the pressure drop result will be compared to either literature or a pressure drop correlations. There are many pressure drop correlations for hydrocyclone mentioned in Bradley's work (1965, p. 92, 96-97). The pressure drop correlation from de Gelder is chosen for this work and explained as follows:

pressure drop is determined by the capacity Q_{feed} as can be seen in equation (2.4).

$$Q_{feed} = \xi \cdot A_i \left(\frac{2\Delta p}{\rho}\right)^{0.5} \tag{2.4}$$

where A_i is a cross-sectional area of feed inlet, Δp is a pressure drop, and ξ is a dimensionless constant which is similar to the loss coefficient defined in equation (2.5). (Bradley, 1965, p. 92, 315, 317).

$$\xi = \frac{\xi_\infty}{1 - \frac{J\, D_c^2}{6\, D_f^2}\left(\frac{2}{Re\, sin\frac{\theta}{2}}\right)^{0.5}} \tag{2.5}$$

where ξ_∞ and J are factors that rely on the de-oiler design, these factors are presented in Figure 2.16. and Figure 2.17, respectively. D_c is the biggest de-oiler diameter i.e. diameter of the cylinder section while D_f is equivalent diameter of feed inlet. (Bradley, 1965, p. 92-94)

Previously, pressure drop in equation (2.4) is only part of the total pressure drop. The total pressure drop ΔP_{total} is then given in equation (2.6).

$$\Delta P_{total} = (\xi^{-2} - 1)\ 0.5\ \rho\ u_{tang}^2 + 0.5\ \rho\ u_{axial}^2 \tag{2.6}$$

where u_{tang} and u_{axial} are the tangential and axial velocity [m/s]. The calculation is listed in Appendix E.

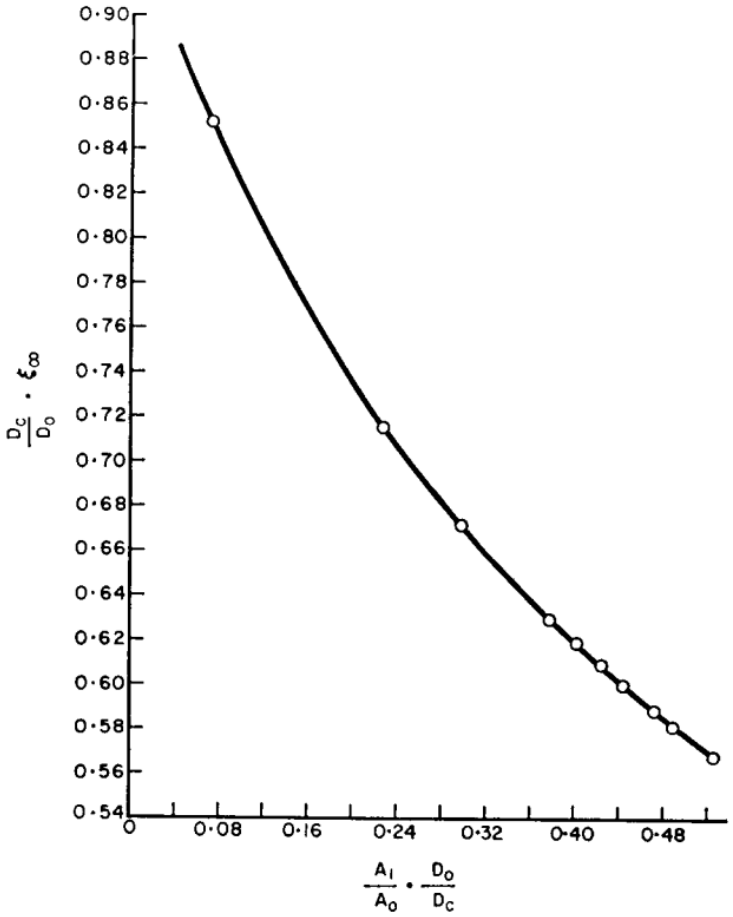


Figure 2.16: A relationship between $\frac{A_i}{A_o} \cdot \frac{D_o}{D_c}$ and $\frac{D_c}{D_o} \cdot \xi_\infty$. (Bradley, 1965, p. 93)

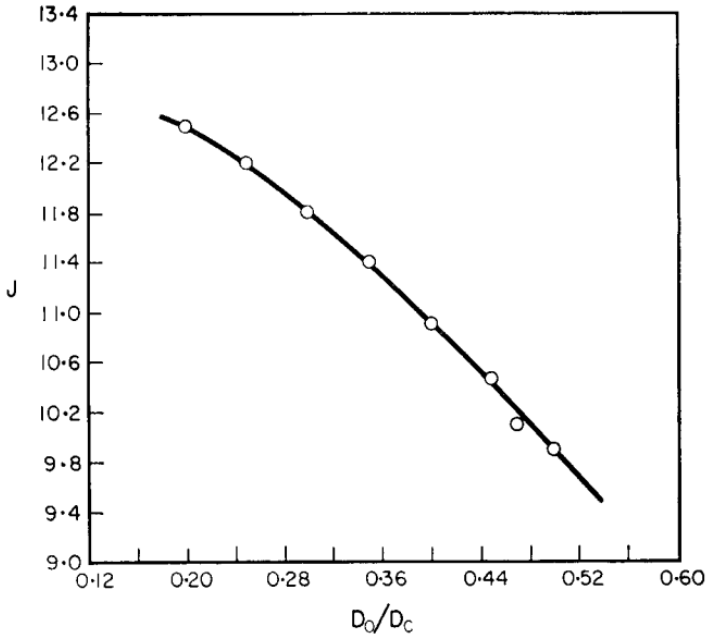


Figure 2.17: A relationship between $\frac{D_o}{D_c}$ and J. (Bradley, 1965, p. 94)

Bernoulli's equation explains the relationship of kinetic, potential, and pressure energy as can be seen in equation (2.7) and Figure 2.18.

$$P_1 + \frac{1}{2}\rho u_1^2 + \rho g h_1 = P_2 + \frac{1}{2}\rho u_2^2 + \rho g h_2 \tag{2.7}$$

where P_1 and P_2 are pressure energy in locations 1 and 2, see Figure 2.18. the second term in equation (2.7) is the kinetic energy per unit volume, and the third term is potential energy per unit volume.

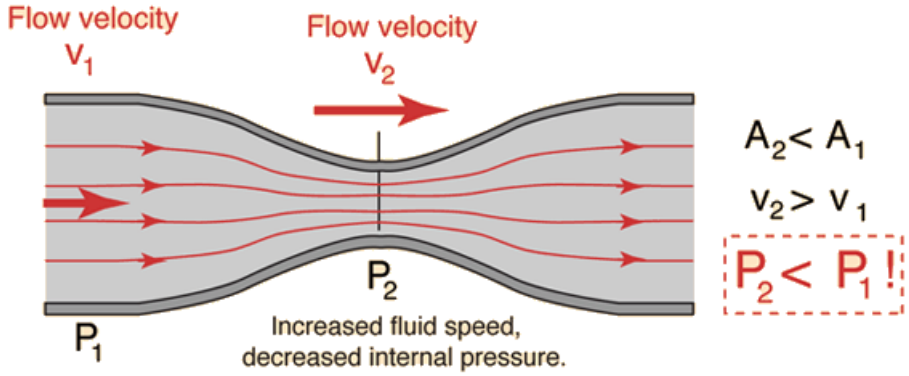


Figure 2.18: Bernoulli's equation adapted from Nave. (2017)

Magnitude velocity for 3D (x, y, and z.direction) is defined as is shown in equation (2.8). (Learning About Electronics, 2018)

$$magnitude = (x^2 + y^2 + z^2)^{0.5} \tag{2.8}$$

According to the continuity equation, the volume flow rate into the pipe must be equal to the volume flow rate at the outlet. As shown in Figure 2.19, the shaded area is the volume at a time Δt. The continuity equation is shown in equation (2.9). (King, 2018, p. 258)

$$A_1 u_1 \Delta t = A_2 u_2 \Delta t \tag{2.9}$$

where A_1 and A_2 are the cross-sectional area, u_1 and u_2 are the axial velocities, see Figure 2.19. At the same length of time, the equation (2.9) can be simplified into equation (2.10). (King, 2018, p. 258)

$$A_1 u_1 = A_2 u_2 \tag{2.10}$$

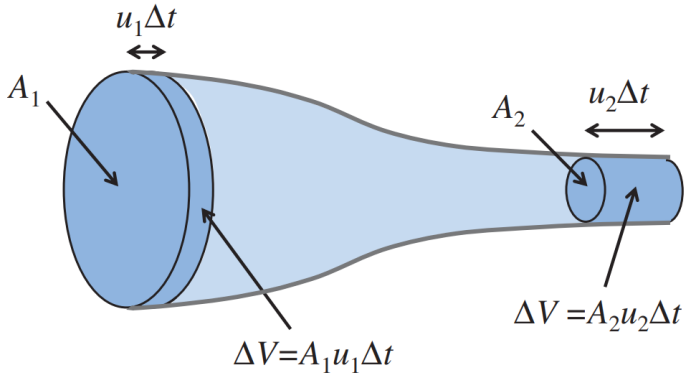


Figure 2.19: Fluid flows in a decreasing cross-sectional area from left to right. (King, 2018, p. 258)

The cartesian (x, y, and z) data of velocity is found in cartesian coordinate, calculation in Ms. Excel is performed to find the tangential velocity, some of the calculation example is presented in Appendix H.

2.5 Operating in marginal fields & turn down conditions

A marginal field is an unexploited discovery of a small reserve/pool. The reasons can be a lack of local infrastructure and profitable consumers, not economically viable (very small sizes of reserves/pool), and technological constraints. On the other hand, once economical or technical situations alter, the field can be exploited and become a commercial field. (Society of Petroleum Engineers, n.d.)

Generally, marginal fields have less flow rate of gas, oil, and water or in other words, less produced water to be treated in the de-oiler. As knowing from before, an optimum flow rate and a minimum pressure of 100 psi are required to achieve satisfying separation. Marginal fields are neighboured with a larger field. So, the stream from the marginal fields can be routed to the subsea separation train located in the seabed to be treated together with the other well stream from both marginal and bigger fields. (2B1st Consulting, 2012) (Wathne, C., personal communication, May 2020) (Stewart & Arnold, 2009, p.187)

Along the lifetime of an oil field, water cut is typically increased while the total volume flow rate is decreased. This means that the facility will be operated below the design specification, namely turn down. Turn down conditions will affect azimuthal velocity as well as separation

performance, therefore, it is recommended to design several cyclones running in parallel which allows each of the de-oiler to run in optimum flow rate. Alternatively, the recycle flow stream through a surge tank will also help to overcome a low flow rate problem, especially in the turndown period. (Campen, 2014, p. 158-159) (Fluor Offshore Solutions, 2012, sec. 6, p. 9)

2.6 Solver

An analytical calculation cannot predict a flow field in centrifugal de-oiler due to its complexity. Therefore, numerical simulations are chosen to estimate the flow. Firstly, this sub-chapter will explain briefly the multiphase, then describe the solver namely *twoPhaseEulerFoam*, and lastly, tell the reason why the Euler-Euler approach is used to solve multiphase flows. (Slot, 2013, p. 7)

2.6.1 Multiphase

Multiphase

"A multiphase flow is a fluid flow consisting of more than one phase component and have some level of phase separation above molecular level." Two-phase flow can be categorized as a gas-liquid mixture, gas-solid mixture, liquid-solid mixture, or immiscible liquid-liquid. According to phase morphology, it can be classified as a dispersed system and a separated system. The dispersed phase exists between a continuous phase as non-contiguous isolated regions. The system is often called dispersed-continuous flow. Meanwhile, the separated system has a clear interface between one phase to another and it is called continuous-continuous flow. Figure 2.20 depicted distinctly the difference between these two systems. (Wolf Dynamics, n.d.)

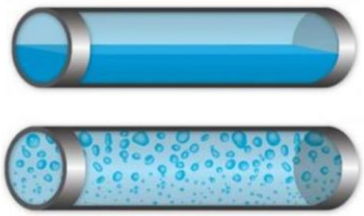


Figure 2.20: Separated (top) and dispersed (bottom) system. (Wolf Dynamics, n.d.)

The separated system can be treated using the Euler-Euler approach (Volume of Fluid). The dispersed system, on the other hand, can be treated using either the Euler-Euler approach (multi-fluid and mixture models) or the Euler-Lagrangian approach (particle tracking). Both mixture model and Volume of Fluid model are a simplified version of the full Eulerian model. In this work, the system is dispersed-continuous where oil is the dispersed phase and water is the continuous phase. Therefore, the option is either Euler-Euler or Euler-Lagrangian. (Wimshurst, 2019, 4:09) (Wolf Dynamics, n.d.)

2.6.2 *twoPhaseEulerFoam* solver

The Lagrangian approach considers the fluid phase as a continuum and the particulate as single particles while the Eulerian sees both particulates and fluid as a continuum and solves the appropriate continuum equations for each of the phases. The Euler-Euler approach is selected and the reasons will be explained later. (Durst, Miloievic, & Schönung, 1984, pp. 101)

Using the Euler-Euler approach, each phase has its volume fraction and velocity field but only has one pressure field. The volume fraction for phase k α_k is defined in equation (2.11) which V_k and V is the volume of phase k and total volume, respectively. (Manni, 2014, p. 1) (Guen, 2016, p. 6)

$$\alpha_k = \frac{V_k}{V} \tag{2.11}$$

In this case, the oil is dispersed in the water because the oil has a much lower volume fraction i.e. less than 10%. From now, the dispersed phase will have subscript 1 and the continuous phase will use subscript 2 in the equations. Because there are only two phases, the summation of the alphas will be one, see equation (2.12).

$$\alpha_1 + \alpha_2 = 1 \tag{2.12}$$

twoPhaseEulerFoam is a solver for compressible fluid for two phases which one of them is a dispersed phase and the other is a continuous phase. It uses the Euler-Euler approach. The solver is modified based on an available previous example in OpenFOAM namely *bubbleFoam*. (Manni, 2014, p. 1) (Wolf Dynamics, n.d.)

Equation of states (EoS)

In the case of compressible fluids which has density variations due to the pressure and temperature changes, the linkage between energy equation and the linkage between mass and momentum equations is provided by the EoS. However, in this case, where the density change is assumed zero due to incompressible fluid, the linkage mentioned before does not exist. Therefore, the flow field can be solved using mass and momentum equations only. Additionally, the heat transfer is also not relevant in this case, so the energy equation is not necessary. (Versteeg & Malalasekera, 2007, pp. 21)

Interfacial forces

The interfacial force is expressed by different sub-forces such as drag, virtual mass, lift, wall lubrication, and turbulent dispersion force. Next, some forces will be shortly described whether they have a significant impact in this case or not. (Guen, 2016, p. 8)

Drag force

To give an illustration, Figure 2.21 shows a liquid droplet falls due to a gravitational force. At the same time, the liquid droplet is also decelerated by a drag force. The drag force shown in the equation (2.13) is often dominating. In this case, the drag force is the most important compared to other interfacial forces. (Guen, 2016, p. 9) (Slot, 2013, p. 23)

$$M_D = -\frac{3}{4}\frac{C_d}{D}\rho_2\alpha_1|U_r|U_r \tag{2.13}$$

where M_D is a drag force, C_d is the drag coefficient, D is bubble diameter, ρ_2 is the continuous phase, and U_r is the relative velocity expressed in equation (2.14).

$$U_r = U_1 - U_2 \tag{2.14}$$

Here, U_1 is bubble velocity and U_2 is the fluid velocity. (Guen, 2016, p. 9)

Furthermore, the equation (2.13) can be written as equation (2.15) with the involvement of a Reynolds number. The drag coefficient with Reynolds number expression C_{dRe} is modelled using Schiller and Naumann model. C_{dRe} can be computed with equation (2.16) for Reynolds number smaller or bigger than 1000. The model is chosen because in this work the dispersed phase is assumed to be spherical. Besides, the model is widely used and quite simple. (Guen, 2016, p. 9)

$$M_D = -\frac{3}{4}\frac{C_d Re_1}{D^2}\nu_2\rho_2\alpha_1 U_r = -\frac{3}{4}\frac{C_{dRe}}{D^2}\nu_2\rho_2\alpha_1 U_r = -KU_r \tag{2.15}$$

$$C_{dRe} = \begin{cases} 24.0/Re\ (1.0 + 0.15\ Re_1^{0.687}) & if\ Re_1 \leq 1000 \\ 0.44\ Re_1 & if\ Re_1 \geq 1000 \end{cases} \tag{2.16}$$

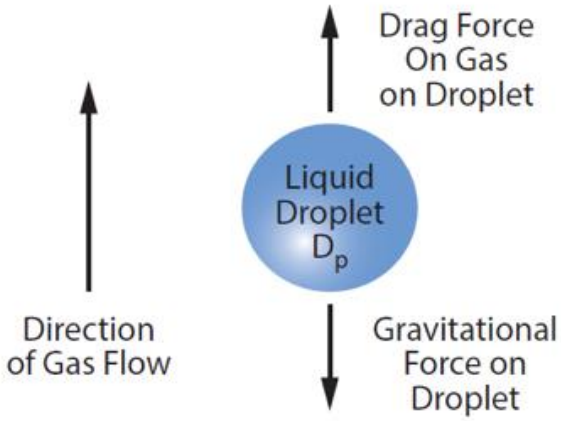


Figure 2.21: Forces experienced by a liquid droplet. (Moshfeghian, 2015)

Virtual mass force

When droplets are moving in a quiescent fluid, the acceleration of droplets can affect the surrounding fluid. The surrounding fluid will accelerate and some of the mass of the fluid which will be brought by the bubble is defined as a virtual mass, see Figure 2.22. The force to accelerate the fluid will act on the droplet in the opposite direction. The virtual mass force is usually neglected when it comes to a gaseous environment is brought by solid or liquid particulate because the mass of the gas is very small. However, in the case of two immiscible liquids the magnitude of this forces depends on the density different. In this case, the density

difference is small, and the virtual mass force will also be small, therefore, it can be neglected. (Paladino & Maliska, n.d., p. 1) (Slot, 2013, p. 26) (Liu et al., 2018, p. 61)

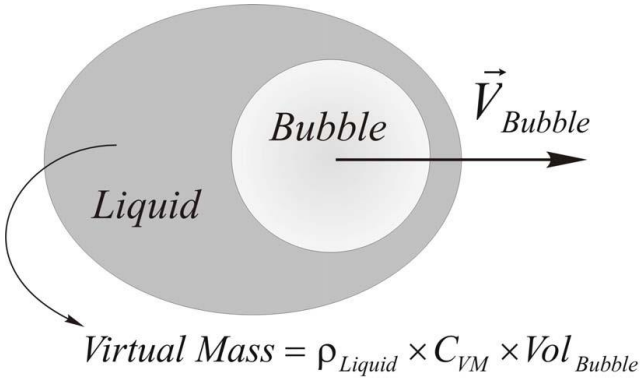


Figure 2.22: Virtual mass of liquid and bubble interaction. (Maliska & Paladino, 2006, p. 953)

Lift force

Lateral or lift force is a force acted in the dispersed phase perpendicular to its velocity direction. According to Guen (2016, p. 10) the lift coefficient C_L for a spherical bubble is constant ($C_L = 0.5$) with positive direction i.e. move towards the wall. Furthermore, Slot (2013, p. 27) used Saffman-Mei model to account lift force in their axial hydrocyclone. (Kolev, 2012, p. 70)

On the other hand, Noroozi & Hashemabadi (2009, p. 1886) stated that the lift force is often neglected in hydrocyclones application because its insignificant value compared to drag force. Similarly, Campen (2014, pp. 98) found out that the Saffman lift is much smaller than the drag force i.e. around two magnitude smaller, see Figure 2.23. Therefore, in this case, the lift force is assumed negligible.

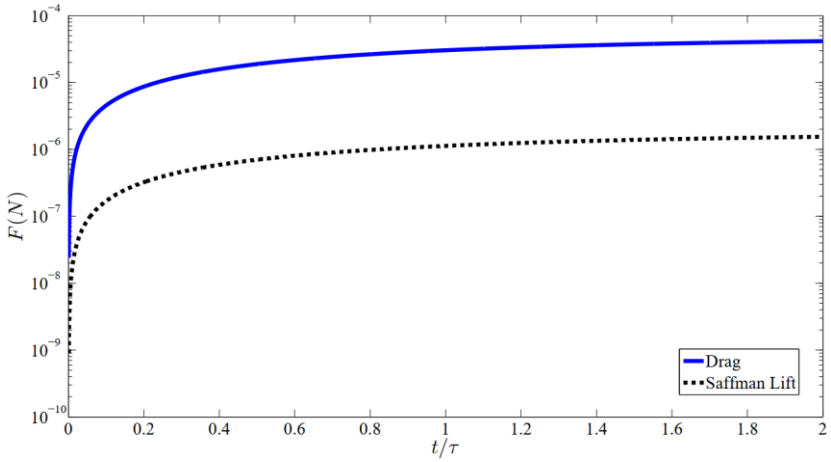


Figure 2.23: A comparison of drag and Saffman lift force experienced by 10 μm oil drop size accelerated in centrifugal flow field with azimuthal velocity 10 m/s. (Campen, 2014, p. 98)

2.6.3 Euler-Euler vs Euler-Lagrange approach

The Lagrangian approach tracks every particle. Newton's law of motion is used to find the velocity of every single particle and the integration of the velocity results in a particle path. This path/location and velocity are then used to get the momentum transfer terms e.g. transfer of drag force between two phases. (Durst, Miloievic, & Schönung, 1984, p. 103)

In the Euler approach, the particle continuity equation is used to solve the particle volume fraction instead of determining the individual location of the particle. Together with particle velocity, the momentum transfer terms can be derived. (Durst, Miloievic, & Schönung, 1984, p. 103)

Durst et al. (1984, p. 108) investigating how these two approaches gave different results on particulate two-phase flow in a vertical pipe with various particle concentrations. They found that the most influential variables on predicting the two-phase flow are:

- Large particle accelerations or decelerations
- High volumetric concentration (alpha)
- Non-uniform alpha distributions

When large particle accelerations occur, the Lagrangian approach has some advantages for predicting the flow behaviour. It gives more detail information about the particle phase. On the other hand, the Euler approach showed some advantages when it is used for high volume fraction. They concluded that both approaches are very similar. (Durst et al., 1984, p. 101)

The Euler-Lagrange approach is considered to be closer to reality because of the low volume fraction of oil in water. However, this approach needs to track thousands of particles and simulate the collision between every particle. This leads to more computational efforts and it is time-consuming. (Guen, 2016, p. 8)

In conclusion, since the oil volume fraction used in this work is equal to 10%, the Euler-Euler approach is used. The Euler-Lagrange approach needs a higher computational effort and as aforementioned, both approaches are very similar. Some experiments are shown in Table 2.3 used various approaches for both ranges of volume fractions.

Table 2.3: Euler-Euler and Euler-Lagrange approach used in various cases of de-oiler.

No.	Article by:	Inlet oil concentration	Approach & case
1	Liu et al. (2018, p. 65)	1, 3, 5, 10 %-vol	Euler-Euler
			For a volume fraction less than 10% Eulerian-Lagrangian model can be used while more than 10%-vol oil, Eulerian-Eulerian model can be used.
2	Huang (2005, p. 829, 832)	20-30%-vol	Euler-Euler
			It is stated that Euler-Euler model is suitable for above 10% oil concentration. A high volume fraction of oil is separated from water using a hydrocyclone.
3	Nascimento et al. (2012, p. 102)	1, 2, 3 %-mass	Euler-Lagrangian
			Performed a deoiling hydrocyclone for oil-water separation

4	Kharoua et al. (2010, p. 750)	-	10%-vol of dispersed phase volume fraction act as a threshold to go from Euler-Lagrange to Euler-Euler model. A review of hydrocyclones for de-oilling applications.

2.7 Process control

As reported by Husveg et al. (2007, p. 295) flow rate and flow split control are the keys to controlling hydrocyclone performance criteria. Flow rate must be maintained between the minimum and maximum volumetric flow rate, $Q_{min,}$ and Q_{max}, respectively. Flow split control defined in equation (2.17) must be maintained since the volumetric flow rate may vary. Besides, by controlling a certain flow split hydrocyclone efficiency can be retained. To control the flow split, they introduced a variable namely pressure drop ratio (PDR) or pressure drop constant C defined in equation (2.19).

According to Meldrum (1988, p. 670), there is a linear relation between reject ratio mentioned in equation (2.18) and C. Increasing pressure drop constant enlarges pressure gradient between the inlet and overflow pressure and escalates reject ratio which means increase the concentrated oily water flow in the overflow as shown in Figure 2.24.

$$Flow\ split = \frac{Q_{LPO}}{Q_{feed}} .100\% \tag{2.17}$$

$$Reject\ ratio = \frac{Q_{LPO}}{Q_{HPO}} \tag{2.18}$$

$$PDR = C = \frac{dP_{oil}}{dP_{water}} = \frac{P_{feed} - P_{overflow}}{P_{feed} - P_{underflow}} \tag{2.19}$$

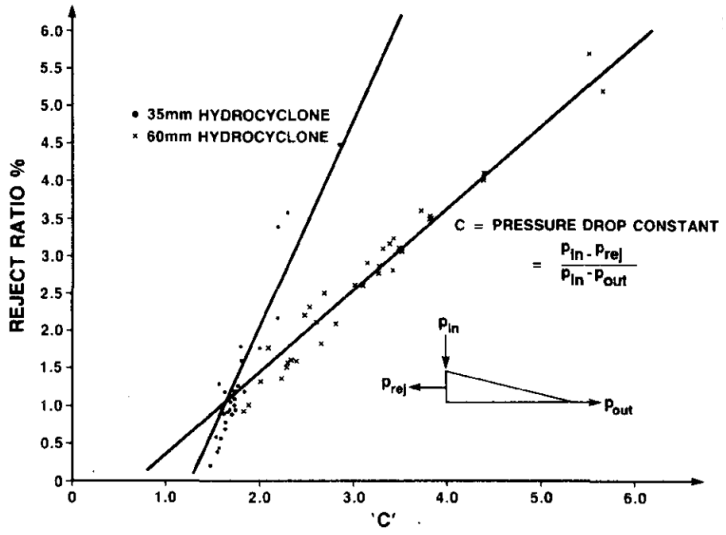


Figure 2.24: Linear relation between reject ratio and pressure drop constant. (Meldrum, 1988, p. 671)

Figure 2.25 depicts the example of the de-oiler control system. It maintains the pressure drop ratio by measuring pressure values in the feed, overflow, and underflow. The main objective is to get a stable flow rate and flow split by controlling the underflow and overflow control valve. (Husveg et al., 2007, p. 296) (Campen, 2014, p. 159)

Flow rate control

The underflow control valve is regulated by two controllers LC01 and LC02. LC01 has an operator-prescribed set point and input from the level transmitter. In our case, the hydrocyclone is located after the pipe-in-pipe separator, the level transmitter will not be relevant. The output from LC01 will be a set point for LC02 and output from DP01 will be an input for LC02. This will control the underflow control valve (LCV) in two ways. Firstly, when the flow rates increase the dP_{water} is also increasing. This will reduce the PDR according to equation (2.19). Furthermore, the PCV will regulate so that the dP_{oil} will increase and restore the PDR set point. The bigger the flow rate increase, the stronger the PDR deviation is. This impacts the efficiency to drop off. Secondly, when flow rates decrease to Q_{min}, the dP_{water} is also decreased, and PDR will increase. The flow split rises and for a while will drag water to the overflow. Unlike the first case, the efficiency is not affected negatively. (Husveg et al., 2007, p. 296, 298)

Flow split control

Overflow control valve (PCV) controls the flow split uses input from PCD01 (pressure differential controller) while PCD01 has two inputs i.e. setpoint (SP PDR) and PDY01. PDY01 value is from equation (2.19), which is a ratio of dP_{oil} and dP_{water} or DP02 and DP01, respectively. (Husveg et al., 2007, p. 296)

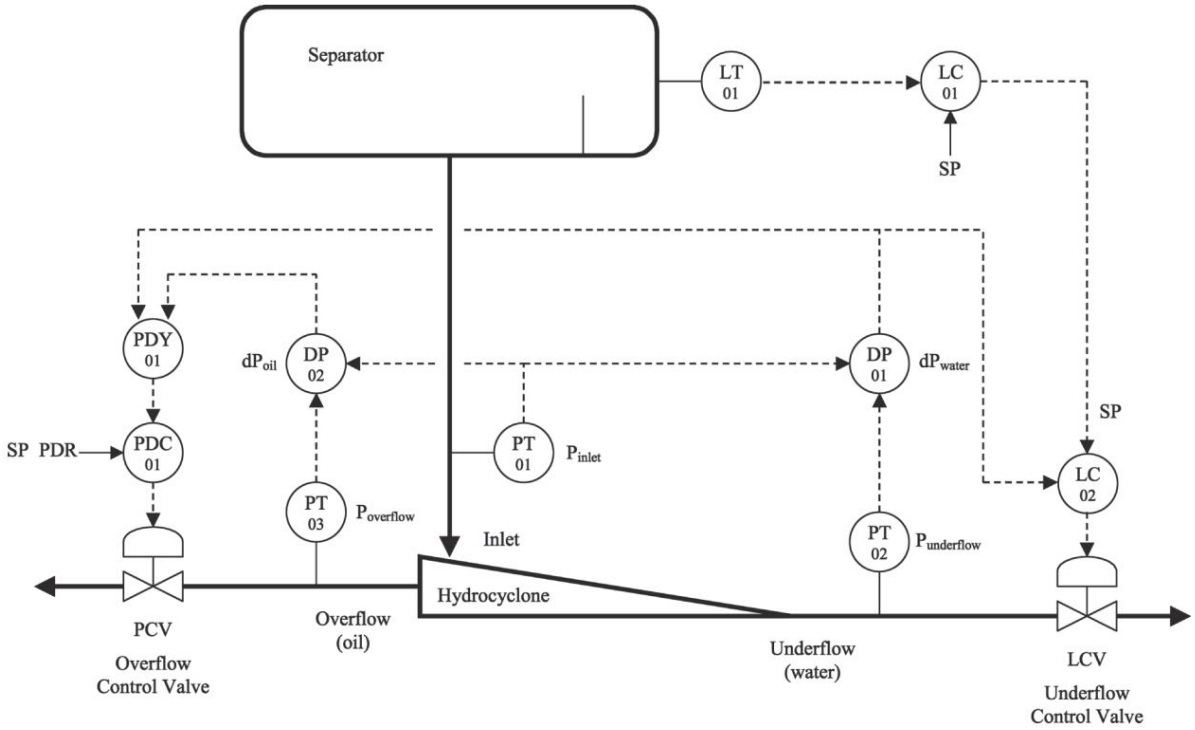


Figure 2.25: Hydrocyclone control scheme example. (Husveg et al., 2007, p. 296)

2.8 Subsea specific design issue & recommendations

Process control system

The process control system for subsea separation is supposed to be more simple compared to the top side control system. This is related to the data transfer rate, data volume, and the limitation of the bandwidth. Using a traditional umbilical cable is seen as a challenge to handle the data transfer rate. Fiber optic cable can lessen the challenge by providing lightning-fast data transfer rates. Most possible, the design of the subsea process control system will require a significant effort and cost like the Tordis project in Norway. (Fluor Offshore Solutions, 2012, p. 6-84)

Power system

Distribution of power is seen as a major challenge for subsea separation. The distribution for pumps, measurement systems, and control valves. The pumps, for instance, require a very high power supply. Fortunately, the de-oiler does not need a pump to operate if the inlet pressure is sufficient. However, if a minimum pressure of 100 psi is not exceeded, it is recommended to use a low-shear pump e.g. progressive-cavity. What's more, the pipe connecting from a pump to the de-oiler should allow oil coalescence. (Stewart & Arnold, 2009, p.187) (Fluor Offshore Solutions, 2012, sec. 6, p. 84)

Maintenance difficulties

The subsea separation unit is expected to sit in the seabed for at least 5 years without maintenance. The poor accessibility to do maintenance brings another challenge, especially when scaling products or sands in the produced water clog the reject port of de-oiler. The small amount of sand that may be left from the previous separation unit can causes erosion in the de-oiler cone/tapered section. Optimizing the desander unit to prevent sand comes to the de-oiler is seen as one of the possible solutions. While performing a chemical injection is necessary to prevent scaling and wax problems. (Bai & Bai, 2019, p. 777) (Stewart & Arnold, 2009, p. 188)

Other challenges

Harsh subsea environment, high external pressure, low ambient temperature in deepwater are very important factors to consider when determining size and materials for the separation units. Higher pressure in salinity environment and low temperature means a thicker de-oiler is needed and this points higher material cost. (Bai & Bai, 2019, p. 777)

3 Modelling

3.1 Problem description

The separation of oil-water and pressure loss in the de-oiler are the main concerns in this work. The produced water out of the de-oiler is expected to have a concentration of around 20-30 ppmv oil while the feed oil concentration will be less than 0.1 to 10.5%. Pressure drop is interesting to see the energy consumption for operating a de-oiler. (Das & Jäschke, 2018, p. 138) (Xodus Group, 2020, p. 6, 22)

The complexity of separating the liquid-liquid phase leads to doing a simplification, including neglecting break-up and coalescence phenomenon, using a constant diameter of oil droplets, and neglecting some of the insignificant inter-phase forces (explained in 2.6.2).

The interesting output of the simulation will be an oil volume fraction and the pressure drop so that the efficiency of a single de-oiler can be investigated. Next, the governing equations needed to be solved and the solution algorithm will be explained.

3.2 Governing equations and numerical implementation

In this case, the `twoPhaseEulerFoam` is used to solve the problem with the Eulerian-Eulerian approach. `twoPhaseEulerFoam` is a solver for compressible fluid for two phases which one of them is the dispersed phase and the other is a continuous phase. (Manni, 2014, p. 1)

The solver uses PIMPLE or PISO algorithm depending on the keyword write in the PIMPLE control directory in *fvSolution* file. This is explained more detail in sub-chapter 4.1.7. Next, the numerical implementation will be discussed together with governing equations included in `twoPhaseEulerFoam` solver. (Manni, 2014, p. 7)

3.2.1 Mass conservation equations

Firstly, mass conservation in equation (3.1) is solved by using MULES (Multidimensional Universal Limiter with Explicit Solution). The detailed process is explained by Damián (2013, p. 37).

$$\frac{\partial}{\partial_t}(\alpha_1\rho_1) + \nabla \cdot (\alpha_1\rho_1 U_1) = 0 \tag{3.1}$$

3.2.2 Momentum equations

As aforementioned, the two-fluid model or Euler-Euler model treats each phase as a continuum. Each phase is represented by averaged conservation equations. For incompressible fluids, dispersed phase, the averaged momentum equation is expressed in equation (3.2) and the continuity equation in equation (3.3). The momentum equation is useful to obtain predicted phase velocities. (Rusche, 2002, p. 92, 104)

$$\frac{\partial \alpha_\varphi \bar{U}_\varphi}{\partial t} + \nabla \cdot \left(\alpha_\varphi \bar{U}_\varphi \bar{U}_\varphi\right) + \nabla \cdot \left(\alpha_\varphi \bar{R}_\varphi^{eff}\right) = -\frac{\alpha_\varphi}{\rho_\varphi}\nabla \bar{p} + \alpha_\varphi g + \frac{\bar{M}_\varphi}{\rho_\varphi} \tag{3.2}$$

$$\frac{\partial \alpha_1}{\partial_t} + \nabla \cdot (\bar{U}_1 \alpha_1) = 0 \tag{3.3}$$

where φ is a phase; either dispersed ($\varphi = 1$) or continuous ($\varphi = 2$), $\bar{R}_\varphi^{eff}$ is the Reynolds (turbulent) and viscous stress, $\bar{M}_\varphi$ is the averaged inter-phase momentum transfer term. Summing up the continuity equations for both phases, equation (3.4) can be obtained.

$$\nabla \cdot (\bar{U}) = 0 \tag{3.4}$$

where $\bar{U} = \alpha_1 \bar{U}_1 + \alpha_2 \bar{U}_2$. (Rusche, 2002, p. 93)

Firstly, let us consider the discretization of the convective and diffusive terms of the momentum equation. Rusche (2002, p. 101-102) decomposed the Reynolds stress term into a diffusive component $\overline{Re}_\varphi^{effD}$ and a correction component $\overline{Re}_\varphi^{effC}$ as shown in equation (3.5).

$$\overline{Re}_\varphi^{eff} = \overline{Re}_\varphi^{effD} + \overline{Re}_\varphi^{effC} \tag{3.5}$$

where $\overline{Re}_\varphi^{effD}$ can be defined as $\overline{Re}_\varphi^{effD} = -\nu_\varphi^{eff} \nabla \bar{U}_\varphi$, it is explained more detail in Appendix B.1. To have a "phase-intensive" momentum equation, equation (3.2) is divided by the volume fraction, given in equation (3.7).

Equation (3.6) is then substituted to equation (3.7) becomes equation

(3.8).

$$\overline{Re}_\varphi^{eff} = \overline{Re}_\varphi^{effD} + \overline{Re}_\varphi^{effC} = -\nu_\varphi^{eff} \nabla \bar{U}_\varphi + \overline{Re}_\varphi^{effC} \tag{3.6}$$

$$\frac{\partial \bar{U}_\varphi}{\partial t} + \bar{U}_\varphi \cdot \nabla\, \bar{U}_\varphi + \nabla \cdot \left(\overline{Re}_\varphi^{eff}\right) + \frac{\nabla \alpha_\varphi}{\alpha_\varphi} \cdot \overline{Re}_\varphi^{eff} = -\frac{\nabla \bar{p}}{\rho_\varphi} + g + \frac{\bar{M}_\varphi}{\alpha_\varphi \rho_\varphi} \tag{3.7}$$

$$\frac{\partial \bar{U}_\varphi}{\partial t} + \bar{U}_\varphi \cdot \nabla \bar{U}_\varphi + \nabla \cdot (-\nu_\varphi^{eff} \nabla \bar{U}_\varphi + \overline{Re}_\varphi^{effC}) + \frac{\nabla \alpha_\varphi}{\alpha_\varphi} \cdot (-\nu_\varphi^{eff} \nabla \bar{U}_\varphi + \overline{Re}_\varphi^{effC}) = -\frac{\nabla \bar{p}}{\rho_\varphi} + g + \frac{\bar{M}_\varphi}{\alpha_\varphi \rho_\varphi}$$

$$\frac{\partial \bar{U}_\varphi}{\partial t} + \bar{U}_\varphi \cdot \nabla \bar{U}_\varphi - \nabla \cdot (\nu_\varphi^{eff} \nabla \bar{U}_\varphi) + \nabla \cdot \overline{Re}_\varphi^{effC} - \frac{\nabla \alpha_\varphi}{\alpha_\varphi} \cdot \nu_\varphi^{eff} \nabla \bar{U}_\varphi + \frac{\nabla \alpha_\varphi}{\alpha_\varphi} \overline{Re}_\varphi^{effC} = -\frac{\nabla \bar{p}}{\rho_\varphi} + g + \frac{\bar{M}_\varphi}{\alpha_\varphi \rho_\varphi} \quad (3.8)$$

$$\frac{\partial \bar{U}_\varphi}{\partial t} + (\bar{U}_\varphi - \nu_\varphi^{eff} \frac{\nabla \alpha_\varphi}{\alpha_\varphi}) \cdot \nabla \bar{U}_\varphi - \nabla \cdot (\nu_\varphi^{eff} \nabla \bar{U}_\varphi) + \nabla \cdot \overline{Re}_\varphi^{effC} + \frac{\nabla \alpha_\varphi}{\alpha_\varphi} \overline{Re}_\varphi^{effC} = -\frac{\nabla \bar{p}}{\rho_\varphi} + g + \frac{\bar{M}_\varphi}{\alpha_\varphi \rho_\varphi} \quad (3.9)$$

Defining a new variable called a total phase velocity $\bar{U}_\varphi^T$ mentioned in equation (3.10), equation (3.9) becomes equation (3.11). (Rusche, 2002, p. 101-102)

$$\bar{U}_\varphi^T = \bar{U}_\varphi - \nu_\varphi^{eff} \frac{\nabla \alpha_\varphi}{\alpha_\varphi} \quad (3.10)$$

$$\frac{\partial \bar{U}_\varphi}{\partial t} + \bar{U}_\varphi^T \cdot \nabla \bar{U}_\varphi - \nabla \cdot (\nu_\varphi^{eff} \nabla \bar{U}_\varphi) + \nabla \cdot \overline{Re}_\varphi^{effC} + \frac{\nabla \alpha_\varphi}{\alpha_\varphi} \overline{Re}_\varphi^{effC} = -\frac{\nabla \bar{p}}{\rho_\varphi} + g + \frac{\bar{M}_\varphi}{\alpha_\varphi \rho_\varphi} \quad (3.11)$$

Consider only the left-hand side, the convective, diffusive, and time derivative term can now be discretized, given in equation (3.12). (p. 103)

$$\left[\left[\frac{\partial \bar{U}_\varphi}{\partial t}\right]\right] + \left[\left[\nabla \cdot \left(\phi_\varphi^T [\bar{U}_\varphi]_{\phi_\varphi^T, S}\right)\right]\right] - \left[\left[\nabla \cdot (\phi_\varphi^T)[\bar{U}_\varphi]\right]\right] - \left[\left[\nabla \cdot (\nu_\varphi^{eff} \nabla [\bar{U}_\varphi])\right]\right] + \nabla \cdot \overline{Re}_1^{effC} + \frac{\nabla \alpha_\varphi}{\langle \alpha_\varphi \rangle_\nabla + \delta} \cdot \overline{Re}_1^{effC} = \cdots \quad (3.12)$$

where ϕ_φ^T is the total phase flux and it is calculated by interpolating equation (3.10). The result is then given in equation (3.13). The last term of equation (3.12) is explained in Appendix B.2.

$$\phi_\varphi^T = \phi_\varphi - \nu_{\varphi f}^{eff} \frac{\nabla_f^\perp \alpha_\varphi}{\alpha_{\varphi f} + \delta} \quad (3.13)$$

where ϕ_φ is the volumetric phase flux, and $\nabla_f^\perp$ can be found in Appendix B.2.

Now, the right-hand side of equation (3.11): $-\frac{\nabla\bar{p}}{\rho_\varphi}+g+\frac{\bar{M}_\varphi}{\alpha_\varphi\rho_\varphi}$ will be discretized. The inter-phase force, especially turbulent drag force and pressure gradient need a special treatment to avoid pressure-velocity decoupling and oscillations. The relation between dispersed and continuous inter-phase force term is: $\bar{M}_1=-\bar{M}_2$, where $\bar{M}_1=M_D+A_\alpha\nabla_\alpha=-KU_r+A_\alpha\nabla_\alpha$. The drag force is treated semi-implicitly in both phases. The r.h.s. of the momentum phase equation is written in a discretized term as shown in equation (3.14).

$$\ldots=-\frac{\nabla\bar{p}}{\rho_\varphi}+g+\mathcal{M}_\varphi-\frac{A_\alpha\nabla_{\alpha_\varphi}}{\alpha_\varphi\rho_\varphi} \tag{3.14}$$

The complete momentum equation is given in equation (3.15). (Rusche, 2002, p. 103-104, 107)

$$\left[\left[\frac{\partial\bar{U}_\varphi}{\partial t}\right]\right]+\left[\left[\nabla\cdot\left(\phi_\varphi^T\left[\bar{U}_\varphi\right]_{\phi_\varphi^T,S}\right)\right]\right]-\left[\left[\nabla\cdot\left(\phi_\varphi^T\right)\left[\bar{U}_\varphi\right]\right]\right]-\left[\left[\nabla\cdot\left(\nu_\varphi^{eff}\nabla\left[\bar{U}_\varphi\right]\right)\right]\right]+\nabla \cdot\overline{Re}_1^{effC}+\frac{\nabla\alpha_\varphi}{\langle\alpha_\varphi\rangle_\nabla+\delta}\cdot\overline{Re}_1^{effC}=-\frac{\nabla\bar{p}}{\rho_\varphi}+g+\mathcal{M}_\varphi-\frac{A_\alpha\nabla_{\alpha_\varphi}}{\alpha_\varphi\rho_\varphi} \tag{3.15}$$

3.2.3 Phase momentum correction equations

The semi-discretized form of the momentum equation is shown in equation (3.16).

$$\left(\mathcal{A}_\varphi\right)_D\bar{U}_\varphi=\left(\mathcal{A}_\varphi\right)_H-\frac{\nabla\bar{p}}{\rho_\varphi}-\frac{A_\alpha\nabla\alpha_\varphi}{\alpha_\varphi\rho_\varphi} \tag{3.16}$$

where $(\,)_H$ is the "H" operator and $(\,)_D$ is diagonal matrix coefficients ($\mathcal{A}_D\equiv[D]$), $\mathcal{A}$ defines the system of linear algebraic equations. The final phase momentum correction equation is obtained by rearranging equation (3.16), see equation (3.17). This equation is useful to correct the velocities after the pressure field is updated. The updated pressure field is obtained by solving the pressure equation which will be explained next. (Rusche, 2002, pp. 90-91, 104)

$$\bar{U}_\varphi=\frac{\left(\mathcal{A}_\varphi\right)_H}{\left(\mathcal{A}_\varphi\right)_D}-\frac{\nabla\bar{p}}{\rho_\varphi\left(\mathcal{A}_\varphi\right)_D}-\frac{A_\alpha\nabla\alpha_\varphi}{\alpha_\varphi\rho_\varphi\left(\mathcal{A}_\varphi\right)_D} \tag{3.17}$$

3.2.4 Pressure equations

Firstly, the mixture pressure equation (3.18) is rearranged using the volumetric continuity equation (3.4) and formulated at the cell faces. (Rusche, 2002, p. 105-106)

$$\nabla\cdot\left(\alpha_{1f}\phi_1+\alpha_{2f}\phi_2\right)=0 \tag{3.18}$$

Next, the volumetric phase flux ϕ_φ is derived by doing an interpolation (using central differencing) of the momentum correction equation (equation ((3.17))). The discretization of

the terms which are proportional to the pressure gradient (equation (3.19)) and phase fraction gradient (equation (3.20)) is done separately. The first term of r.h.s. equation (3.20) is estimated by re-substituting velocity gained in equation (3.15) while the second term is due to turbulent drag term. (Rusche, 2002, p. 105-106)

$$\phi_\varphi = \phi_\varphi^* - \left(\frac{1}{\rho_\varphi \left(\mathcal{A}_\varphi\right)_D}\right)_{\mathrm{f}} |S|\, \nabla_f^\perp \bar{p} \tag{3.19}$$

$$\phi_\varphi^* = \left(\frac{\left(\mathcal{A}_\varphi\right)_H}{\left(\mathcal{A}_\varphi\right)_D}\right)_{\mathrm{f}} \cdot S - \left(\frac{1}{\rho_\varphi \left(\mathcal{A}_\varphi\right)_D}\right)_{\mathrm{f}} \frac{A_{\alpha f} |S|\, \nabla_f^\perp \alpha_\varphi}{\alpha_{\varphi f}} \tag{3.20}$$

Lastly, the pressure equation is obtained by combining equation (3.18) and (3.19), see equation (3.21). This pressure equation results in corrections for correcting pressure, fluxes, and velocities. (Rusche, 2002, p. 105-106)

$$\left[\!\left[\nabla \cdot \left(\left(\alpha_{1f} \left(\frac{1}{\rho_1 (\mathcal{A}_1)_D}\right)_f + \alpha_{2f} \left(\frac{1}{\rho_2 (\mathcal{A}_2)_D}\right)_f\right) \nabla[\bar{p}]\right)\right]\!\right] = \nabla \cdot \left(\alpha_{1f} \phi_1^* + \alpha_{2f} \phi_2^*\right) \tag{3.21}$$

3.2.5 Solution algorithm

The solution algorithm taken from Rusche (2002, p. 117) is displayed in Figure 3.1.

for `N` from 1 to `NOuterCorrectors` **do**

> Solve the mass conservation in equation (3.1) for the dispersed phase to get α_1^{new}.
>
> Using equation (2.12) to get α_2^{new}.
>
> Update the inter-phase forces mentioned in sub-chapter 2.6.2.
>
> Construct and discretize the momentum equation (explained afterward).
>
> Solve the predicted phase velocities from phase momentum equation (equation (3.15)).
>
> **for** `N2` from 1 to `NCorrectors` **do**
>
> > Prediction of flux using equation (3.20).
> >
> > Construct and solve pressure equation (3.21).
> >
> > Correct fluxes using equation (3.19).
> >
> > Pressure relaxation
> >
> > Correct velocities (3.17)

End

Solve for turbulent equation

End

Figure 3.1: Solution procedure of `twoPhaseEulerFoam` solver. (Guen, 2016, p. 6-7, 22)

3.3 Numerical schemes

This sub-chapter will explain the numerical schemes occupied in OpenFOAM simulations to discretize terms in the equations mentioned earlier. The schemes used in OpenFOAM are specified in *fvSchemes* file and can be seen in section 4.1.8.

a. Time schemes

First or second order of time derivative $\frac{\partial}{\partial t}, \frac{\partial^2}{\partial t^2}$ or `ddtSchemes` in OpenFOAM. The time scheme used is `Euler`: transient, first-order implicit.

b. Gradient schemes

The `gradSchemes` sub-dictionary in OpenFOAM contains the gradient terms. In this work, `Gauss linear` is chosen. The `Gauss` entry means a standard finite volume discretization of Gaussian integration while the `linear` entry is the `interpolationSchemes` means a linear interpolating or central differencing. (OpenCFD Ltd (ESI Group), n.d.)

c. Divergence schemes

The `divSchemes` sub-dictionary in OpenFOAM contains divergence terms $\nabla \cdot ( )$ except Laplacian terms (will be explained later). The advective term (e.g. $\nabla \cdot \left(\alpha_\varphi \overline{U}_\varphi \overline{U}_\varphi\right)$ in the momentum equation) is included in these schemes. All divergence schemes are based on the `Gauss` integration and be noted that the `Gauss` entry should be followed by `interpolationScheme` i.e. `Gauss <interpolationScheme>`. The entry 'V' schemes such as `Gauss limitedLinearV 1` is a specialized version made for vector fields. Other schemes used are `Gauss vanLeer` (second-order, unbounded), `Gauss limitedLinearV 1`, and `Gauss limitedLinear 1`. (Greenshields, 2018) (OpenFOAM: User Guide v1912. (2016-2017) (OpenCFD Ltd (ESI Group), n.d.)

d. snGradSchemes

The surface normal gradient scheme is included in `snGradSchemes` sub-dictionary. The sub-dictionary is also used in `laplacianSchemes`. The surface normal gradient is a gradient component normal to the face, for example $\nabla_f^\perp \alpha_\varphi$ in equation (3.13). The format is `default uncorrected`. The calculation for gradient normal to the face is second-order accurate if the vector connecting the cell center is orthogonal to the face. For most of the mesh, it does not have high orthogonality. To maintain the second-order accuracy, a non-orthogonal correction can be added to an orthogonal component, namely `corrected`

scheme. In this work, the average mesh non-orthogonality is below 5^{o}, and it is recommended to use `uncorrected` schemes. (Greenshields, 2018) (OpenCFD Ltd (ESI Group), n.d.)

e. laplacianSchemes

Laplacian terms ∇^2(e.g. $\nabla \cdot (\nu_{\varphi}^{eff} \nabla \bar{U}_{\varphi})$ in the momentum equation) is included in the `laplacianSchemes` sub-dictionary. Gauss scheme is the only option to discretize and it needs `interpolationScheme` for the diffusion coefficient ν_{φ}^{eff} in this example, and needs a surface normal gradient scheme for $\nabla \bar{U}_{\varphi}$. So, the format is `Gauss <interpolationScheme> <snGradScheme>`. `Gauss linear uncorrected` (Bounded, first order, non-conservative). (OpenCFD Ltd (ESI Group), n.d.)

f. Interpolation schemes

The `interpolationSchemes` sub-dictionary interpolating term typically interpolated from the cell center to the face center. There are many kinds of interpolation schemes available in OpenFOAM, but the `linear` which specified as a default one is used in most cases. The format is `default linear`. (OpenCFD Ltd (ESI Group), n.d.) (Greenshields, 2018)

3.4 Case variations

There are two simulations presented in this work. Both simulations have a similar condition except for the velocity boundary condition. Run 1 has 0.2 and 0.5 m/s for axial and tangential velocity, respectively. Run 2 has bigger velocities that are 0.5 and 0.7 m/s for axial and tangential velocity in the boundary condition.

The produced water volumetric flow rate is assumed 60000-80000 barrels per day.

4 CFD Simulation

CFD simulation with OpenFOAM

The case structure in OpenFOAM consists of *0* folder, *constant* folder, and *system* folder. *0* folder contains initial and boundary conditions. *Constant* folder encloses the mesh data (*polyMesh*), and other files such as *phaseProperties*, *thermophysicalProperties*, and *turbulenceProperties*. Lastly, the *system* folder has *controlDict* file where the time step, *endTime* can be changed, computational schemes file, and *fvSolution* file e.g. to change the algorithm, set under relaxation factor, and tolerance. The case structure for simulation called Run1 and Run2 is given in Figure 4.1.

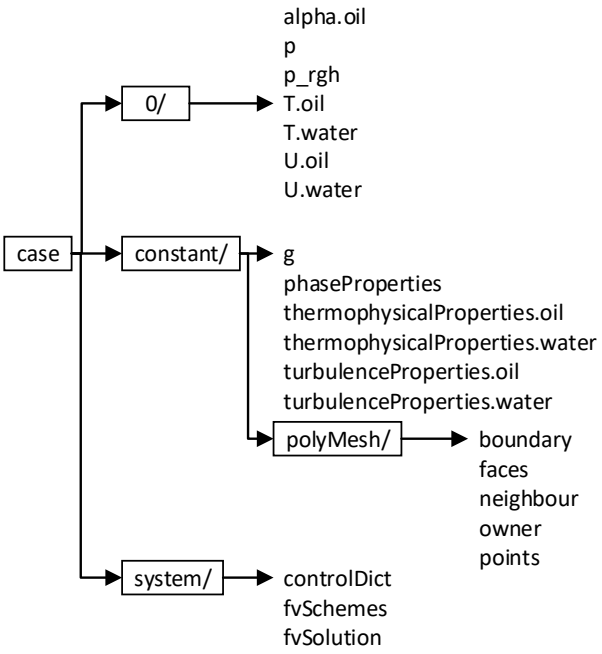


Figure 4.1: Case structure of Run1 and Run 2.

4.1 Pre-processor

There are three main elements are contained in the CFD codes: a pre-processor, a solver, and a post-processor. A pre-processor consists of a definition of the geometry i.e. computational domain, grid or mesh generation, selection of the model, set-up of fluid properties, and suitable boundary conditions. (Versteeg & Malalasekera, 2007, pp. 2)

Then, a solver, a finite volume method is used in this work. The outline is integrating the governing equations over a control volume, doing a discretization (conversion of integral equations into a system of algebraic equations), and solving by iterations. The post-processor is used to visualize the simulation results. ParaView is used in this work and it is included in the OpenFOAM package. (Versteeg & Malalasekera, 2007, pp. 3)

In this section, geometry and mesh will be discussed first. Then, the set-up in OpenFOAM is done by modifying the files in the 0, constant, and system folder. Lastly, the simulation and post-process will be pointed out.

4.1.1 Geometry and mesh generation

Geometry generation

The geometry or computational domain used in this work is adapted from Young et al. (1994, p. 40). Both geometry and mesh are created by open-source software namely SALOME. The hydrocyclone geometry is depicted in Figure 4.2 and the dimensions are listed in Table 4.1.

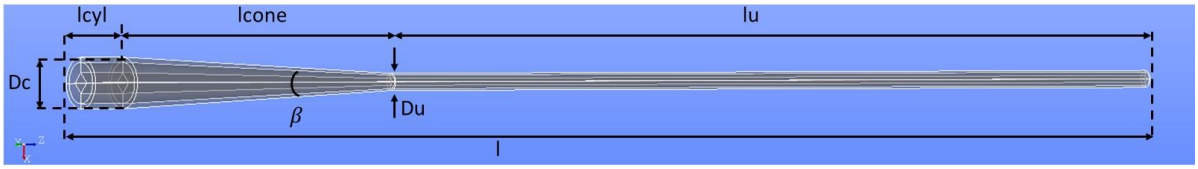


Figure 4.2: Hydrocyclone geometry adapted from Young et al. (1994, p. 40)

Table 4.1: Hydrocyclone classification. (Young et al., 1994, p. 40)

Variable	**Dimensions [m]**
Dc	0.0762
Du	0.0251
lcyl	0.0762
lcone	0.4871
lu	1.3716
l	1.9349
Variable	**Dimensions [º]**
alpha	6

Mesh generation

Over 50% of the time in the CFD project is spent on constructing geometry and mesh. Principally, using a larger number of cells will result in better accuracy of the solution. However, the larger number of cells require a longer time to solve as well as necessitate of higher specification of computer hardware. Therefore, the mesh should not be too fine or coarse. Besides, maintaining cell size changes to be smooth will avoid getting truncation errors. The finer mesh is needed to capture the details of the phenomenon happening in certain parts of the geometry. (Versteeg & Malalasekera, 2007, pp. 3) (Tresvik, 2016, p. 27)

For example, refinement in certain parts such as the shear layer near the wall is needed to capture the production of turbulent kinetic energy. For the de-oiler, refining the mesh in the center of the de-oiler is very important to reach the mesh convergence. It is needed to solve the azimuthal and axial velocity distribution accurately. (Slot, 2013, p. 52-53) (Saidi, Maddahian, & Farhanieh, 2012, p. 1824)

There are several choices for drawing and meshing a geometry for OpenFOAM simulation such as modifying a blockMeshDict file in OpenFOAM, using Gmsh, SALOME, and utilizing snappyHexMesh. For a simple geometry, modifying a blockMeshDict file in OpenFOAM is suitable while for more complex geometries, others are preferable. In this work, SALOME is used because the hydrocyclone geometry is not very simple. (Gullberg, 2017, p. 25)

The de-oiler geometry used in this work consists of a cylinder and truncated cone which has a circular area. Meshing a circular area in the way shown by Figure 4.3 will cause a singularity problem and the solution will not converge. To avoid this, the center part of the mesh is replaced by a square geometry. (Gullberg, 2017, p. 22)

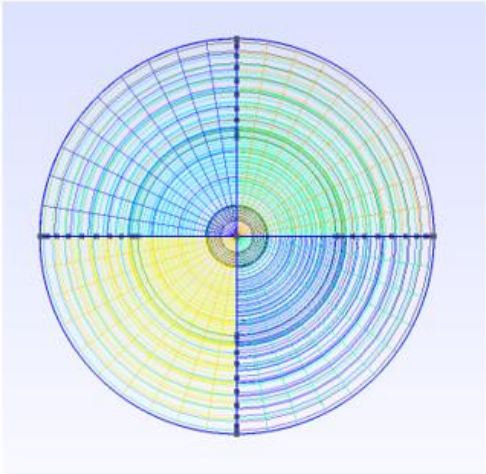


Figure 4.3: Singularity in the center part of a circular mesh. (Gullberg, 2017, p. 22)

The 3D mesh is built using SALOME by selecting hypotheses and algorithms listed in Table 4.3.

Then, the mesh is exported in the .unv file. It is converted in OpenFOAM using command `ideasUnvToFoam (meshname).unv`. An inlet part and a body part of the hydrocyclone mesh are shown in Figure 4.4. Furthermore, the mesh information is listed in Table 4.2.

Moreover, sub-meshes are created to refine certain areas of the geometry, sub-mesh is set by using Wire Discretisation algorithm, Number of Segments hypothesis, and Propagation of Node Distribution on Opposite Edges as an additional hypothesis. Sub-mesh is created near the wall. This is based on the work of de Araujo et al. (2020, p. 8) and Liu et al. (2018, p. 63). From their similar cases, a big velocity gradient begins around 77-100% of the radius. Therefore, the refinement of the mesh near the wall is created around 70% of the radius.

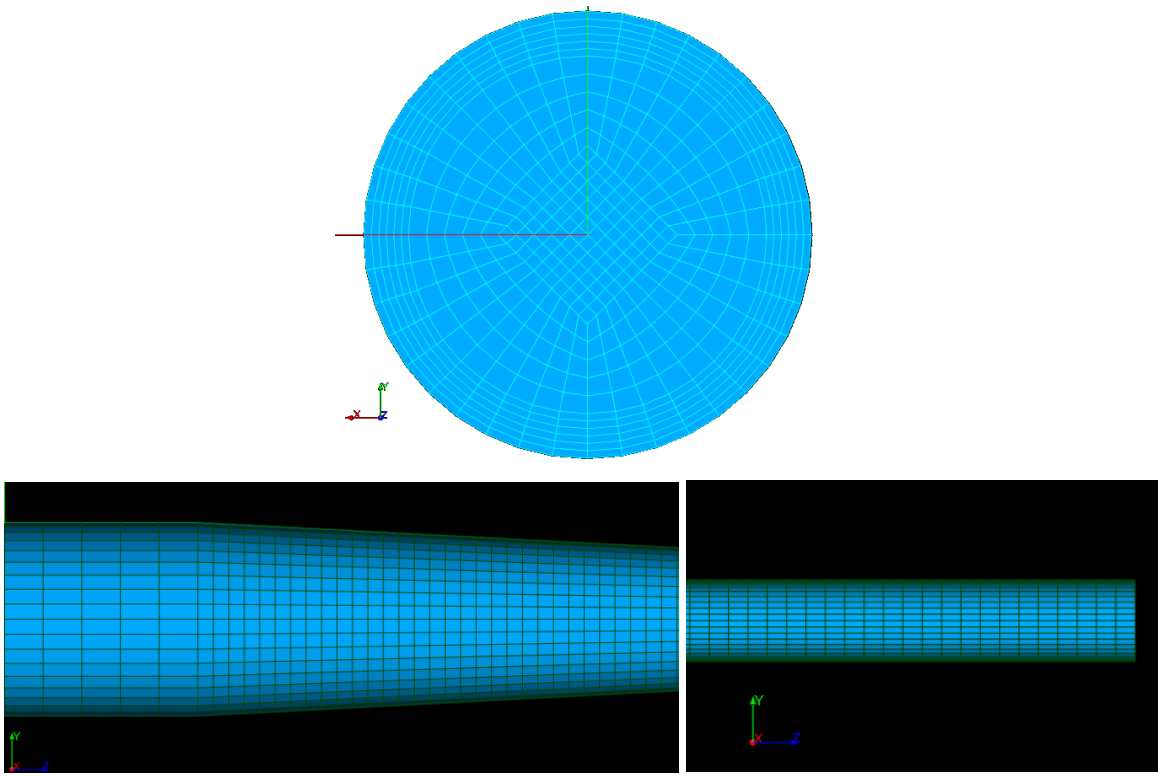


Figure 4.4: Inlet part of hydrocyclone mesh (top) and body parts; near inlet (bottom left) near outlet (bottom right).

Table 4.2: Mesh information from OpenFOAM.

Mesh stats	
Points	177276
Faces	517140
Internal faces	503460
Cells	170100
Faces per cell	6
Boundary patches	4
Overall number of cells of each type	170100
Max aspect ratio	14.4649
Mesh non-orthogonality	Max: 26.9412; Average: 4.80862
Max skewness	0.426479

Table 4.3: Mesh hypothesis and algorithms applied in SALOME

	Applied hypothesis	**Applied algorithms**
1D	Number of Segments	Wire Discretisation
2D	Quadrangle Parameters (with standard transition)	Quadrangle: Mapping
3D	-	Hexahedron (i, j, k)
Sub-mesh	Number of Segments Additional: Propagation of Node Distribution on Opposite Edges	Wire Discretisation

Lastly, the mesh quality can be measured by looking at the non-orthogonality, skewness, aspect ratio, and smoothness. The `checkMesh` or `checkMesh -allTopology -allGeometry` command will do the mesh quality check automatically to help analyze whether the mesh is appropriate to run a simulation or not. From the `checkMesh` utility, the mesh is ok. (Gullberg, 2017, pp. 25)

4.1.2 Boundary Conditions

In this work, `twoPhaseEulerFoam` solver is used by modifying the *bubbleColumn* example in openfoam7/tutorials/multiphase/twoPhaseEulerFoam/laminar/bubbleColumn. The boundary conditions in OpenFOAM The boundary condition used in this case can be seen in Table 4.4. The explanation for the terms inside Table 4.4 is listed in Table 4.5.

Table 4.4: The boundary conditions in Run 1.

		Patch					
		inletAnular		**inletCenter**		**outlet**	**walls**
Boundary conditions	**U.water**	swirlInletVelocity		swirlInletVelocity		pressureInletOutletVelocity uniform (0 0 0)	fixedValue uniform (0 0 0)
		axialVelocity	0.2	axialVelocity	0.2		
		tangentialVelocity	0.5	tangentialVelocity	0.5		
	U.oil	swirlInletVelocity		swirlInletVelocity			
		axialVelocity	0.2	axialVelocity	0.2	pressureInletOutletVelocity uniform (0 0 0)	fixedValue uniform (0 0 0)
		tangentialVelocity	0.5	tangentialVelocity	0.5		
	T.water	fixedValue uniform 300		fixedValue uniform 300		inletOutlet	zeroGradient
	T.oil	fixedValue uniform 300		fixedValue uniform 300		inletOutlet	zeroGradient

	p_rgh	fixedFluxPressure uniform 5e5	fixedFluxPressure uniform 5e5	prghPressure	fixedFluxPressure uniform 5e5
	p	calculated	calculated	calculated	calculated
	alpha.oil	fixedValue uniform 0.1	fixedValue uniform 0.1	inletOutlet	zeroGradient

Table 4.5: Boundary condition terms and meaning. (OpenFOAM: API Guide v1912, n.d.) (OpenFOAM, n.d.) (CFD Direct, (n.d.) (Cappelli, 2018, p. 9, 13, 16) (Technische Universität Wien, 2019)

Boundary condition	**Meaning**
`swirlInletVelocity`	inlet vector boundary condition with cylindrical coordinates: `axis` = axis of rotation, `origin` = origin of rotation, `axialVelocity` = axial velocity profile, `radialVelocity` = radial velocity profile, `tangentialVelocity` = tangential velocity profile.
`pressureInletOutletVelocity`	this velocity boundary condition is occupied to calculate the velocity at the outlet so that it satisfies the mass and momentum equation. Also, to satisfy the inlet and internal velocity and pressure.
`fixedValue`	fixed value constraint
`fixedFluxPressure`	This is used for pressure boundary condition. `zeroGradient` is generally used, but when body forces present, the pressure gradient is set to the provided value.
`calculated`	the boundary condition of pressure is calculated to keep the inside pressure as the one mentioned in the initialization
`inletOutlet`	when the flux is going towards the outside of the domain, `zeroGradient` is applied. When flux is moving into the domain, `fixedValue` is used.
`zeroGradient`	zero-gradient condition from patch internal field to patch faces
p_rgh	p_rgh is defined as hydrostaticless pressure. The 'p' is pressure boundary condition used for thermodynamics purposes. OpenFOAM solves the hydrostaticless pressure instead of solving 'p' in the pressure equation. Then the 'p' is calculated, $p = p_{rgh} + \rho g h$

alpha.*	volume fraction of *, e.g. alpha.air means air volume fraction.
T.*	temperature of *, e.g. T.air means air temperature.
U.*	velocity of *, e.g. U.air means velocity of air.

4.1.3 Phase properties

The phase properties are located in the *system* folder, to establish the relationship/interactions between dispersed and continuous phase. The explanation is listed in Table 4.6.

Table 4.6: *phaseProperties* terms and meaning. (cfdyna, n.d.) (OpenFOAM, n.d.) (Xodus Group, 2020, p. 8)

Terms	**Meaning**
`diameterModel`	bubble/droplet diameter = 60 μm
`blending`	to specify the dispersed and continuous phase locally for each cell.
`sigma`	interface surface tension = 0.043 N/m^2
`aspectRatio`	bubble aspect ratio (shape of the bubble)
`drag`	drag force
`virtualMass`	virtual mass force

4.1.4 Thermophysical properties

Thermophysical properties are given in Table 4.8, the detail physical properties see Appendix C. The selection of the thermophysical model is explained in Table 4.7.

Table 4.7: Thermophysical properties of water and oil. (Thomas, 2016, p. 36) (OpenFOAM, n.d.)

Terms (selected model)	**Meaning**
type heRhoThermo	general thermophysical model calculated based on enthalpy and density
transport const	assume constant dynamic viscosity and Prandtl number
thermo hConst	assumes a constant cp
equationOfState rhoConst	assume a constant density

Table 4.8: `thermophysicalProperties` terms, meaning, value, and units.

Terms	**Meaning**	**Values**		**Units**
		oil	**water**	
molWeight	molecular weight	170	18	[g/mol]
rho	density	856	996	[kg/m^3]
Cp	heat capacity	2130	4179	[J/kg.K]
Hf	heat of fusion	0	0	
mu	dynamic viscosity	0.0072	0.000855	[kg/m.s]
Pr	Prandtl number	127.8	5.828	-

4.1.5 controlDict

controlDict file is *system* folder where the time step, maximum Courant number, end time of simulation, etc. can be changed, more detail see Table 4.9.

Table 4.9: *controlDict* keyword, the example of selection, and meaning. (Huusari, 2015, pp. 108) (Greenshields, 2018) (Cappelli, 2018, p. 10)

keyword	**selection**	**meaning**
`application`	twoPhaseEulerFoam	solver used in this case is available for solving a multi-component phase. One phase but immiscible species are used in this work, namely oil and water. It is possible to add more species to this solver.
`startFrom`	latestTime or startTime	simulation start from
`stopAt`	endTime	end of simulation at endTime
`endTime`	100 (example)	to specify the end time
`deltaT`	0.0005 (example)	simulation time step
`writeControl`	adjustableRunTime	to write the data every `writeInterval` seconds by adjusting the time step
`writeInterval`	0.05 (example)	(described above)
`runTimeModifiable`	yes	to enable user to modify the parameters during simulation
`adjustTimeStep`	yes	to enable adjustable time step

maxCo	2 (example)	maximum courant number; e.g. fluid flow inside the de-oiler will not travel further than the length of two computational cells during a single time step. As the Courant number decrease, the accuracy and stability of the solution are increased, and time step needed is smaller.

4.1.6 decomposePar

To run the simulation using several cores, `decomposePar` is used to decompose the domain equally. `numberOfSubdomains` is the number of the processor/cores that want to be used for running the simulation. `method` used is `simple` to decompose in the direction which is specified in the `simpleCoeffs` subdictionary. E.g. `n (1 4 1)` to split the domain into 6 parts, 1 in the x-direction, 4 in the y-direction, and 1 in the z-direction. (openfoamwiki, 2009)

4.1.7 fvSolutions

PIMPLE algorithm is the merged between PISO and SIMPLE algorithm. The SIMPLE algorithm can have a relaxation factor which PISO does not. The Courant number (Co) in PISO should be less than 1 (leads to high computational cost), therefore to speed up the simulation, Co should be increased. By using the SIMPLE algorithm, the Co can be bigger than one and means to have a bigger time step and less computational cost.

However, in the usage of a normal SIMPLE algorithm, the pressure term is neglected, so the relaxation factor is needed. The `fvSolution` is explained in Table 4.10.

Table 4.10: `fvSolutions` keywords and meaning. (Darren, 2018) (openfoamwiki, 2018) (cfd-online, 2016)

keywords	**meaning**

`nOuterCorrectors`
 	
`(e.g. = 2)`
	Number of outer correctors of PIMPLE algorithm, pressure-momentum equation will be iterated until the convergence criteria are fulfilled before going to the next step. Set to a very high number between 50-1000 is recommended.
	activates PIMPLE loop, re-calculate the pressure-momentum coupling for two times.

`nCorrectors`	Number of inner corrector, set to small number 1 to 3 is recommended, if high number is set, it will leave the PIMPLE loop earlier and prevent extra calculation.
`nNonOrthogonalCorrectors`	To correct the pressure field more often, needed for high non-orthogonal mesh only.

`e.*`	Internal energy, the minimum iteration is set to 0, to exclude the effect of energy transfer because the temperature in this case is not the concern.

4.1.8 fvSchemes

fvSchemes dictionary contains 6 schemes subdictionaries and each contains keyword entries such as a `default` entry, `div(phi)`. For example, if `gradSchemes` is set to `default`, all gradient terms will be treated with the same schemes. Therefore, no need to specify different schemes. However, `default` entry is set and some schemes are speficied, the specified schemes will override the `default` scheme. Keyword identifier `div(phi, …)` is or the advective terms, where phi is defined as the (volumetric) flux, for example: `div(phi, U)` is for advection of velocity, `div(phi, e)` is for advection of internal energy, and `div(phi, k)` is for turbulent kinetic energy. All the OpenFOAM files are presented in Appendix G. (Greenshields, 2018)

4.2 Simulation and Post-processing

There are two simulations done in this work. The actual time required for running a simulation (Run 1) for 27.6 s is from 7th May to 11th June using 1 core i7 computer. Furthermore, the simulation of Run 2 took time from 15th May to 9th June for having a result of 7.25 s using 4 cores i5 computer. The results are taken by using ParaView. The *filters* used are *Plot Over Line*, *glyph*, and *stream tracer*.

The velocity data in cartesian coordinates are opened in *SpreadSheetView* and downloaded to be converted to the tangential velocities, it is calculated using Microsoft Excel.

5 Results

Chapter 5 covers the results from the CFD simulation including oil concentration, velocity profiles, and pressure along the de-oiler. There are two simulations done in this work. The Run 1 will be discussed mainly in the Results and Discussion chapter. Meanwhile, the Run 2 will be used as pressure and velocity references, as a comparison to Run 1. This is because the oil concentration needs more time to develop until it stable while Run 2 was only simulated to 7.25 seconds.

5.1.1 Oil concentration

The mean oil concentration (alpha oil) along the axial direction is increased over time, given in Figure 5.1. Also, the alpha oil is fluctuating along the time and not showing a clear increment. The boundary condition used for inlet alpha oil is 0.1. Figure 5.2 shows that near the inlet area, the oil has not been concentrated in the center. With increasing axial distance, the alpha oil is then starting to build up in the center of the de-oiler. Figure 5.3 shows the oil core position near the outlet and the alpha at x = 0, y = 0, and z = 1.9348 is 0.14.

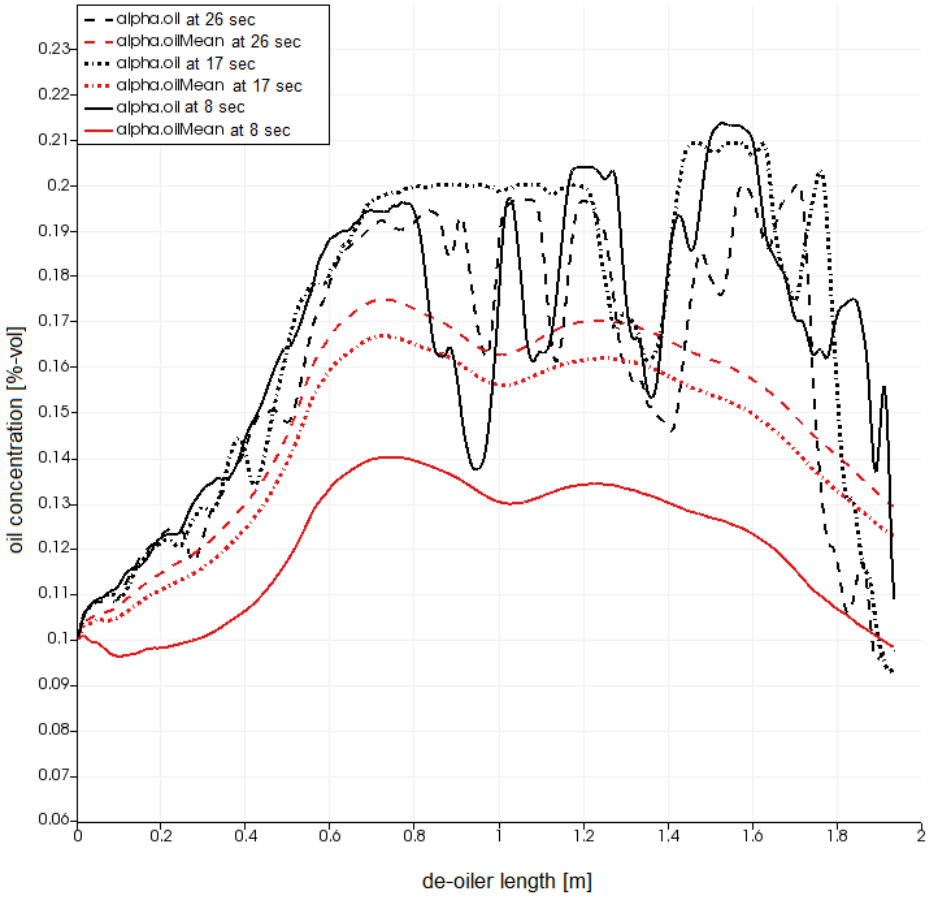


Figure 5.1: The oil concentration (alpha oil and alpha oil mean) along the axial for time = 8, 17, and 26 seconds.

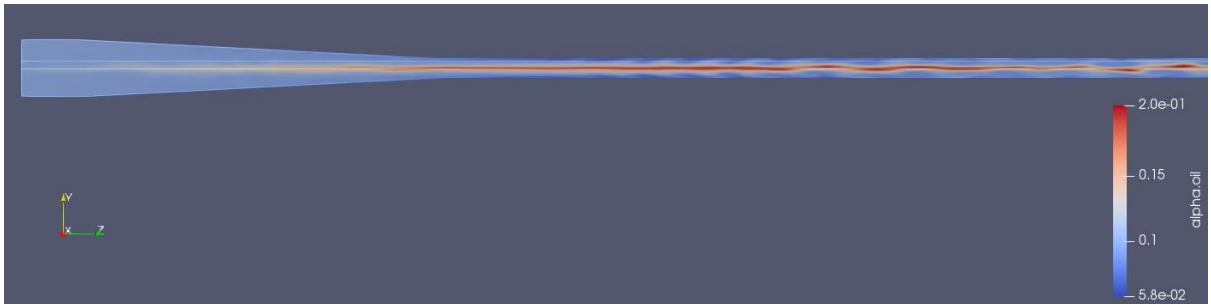


Figure 5.2: The appearance of the oil core at z = 0 to 1.3 m; x = 0, and y = 0.

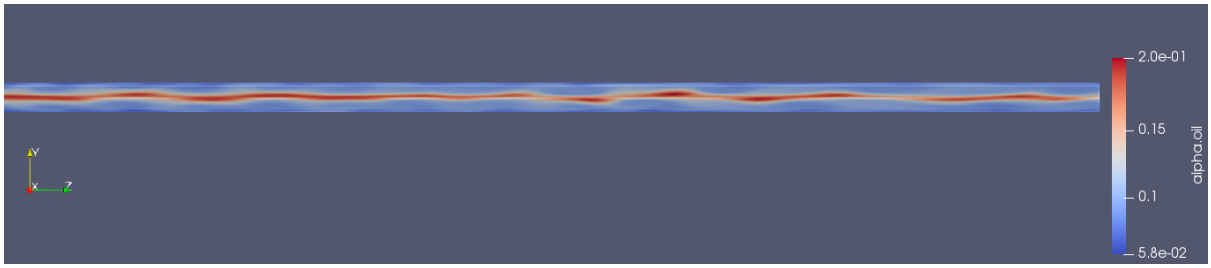


Figure 5.3: The appearance of oil core at z = 1.3 to 1.93 m; x = 0, and y = 0.

Next, in Figure 5.4, it is shown the position where the alpha oil in the water/HPO is inspected. The result is then given in Figure 5.5, the mean alpha oil in the HPO at z = 1.932 m is 0.088. The corresponding efficiency is 12%.

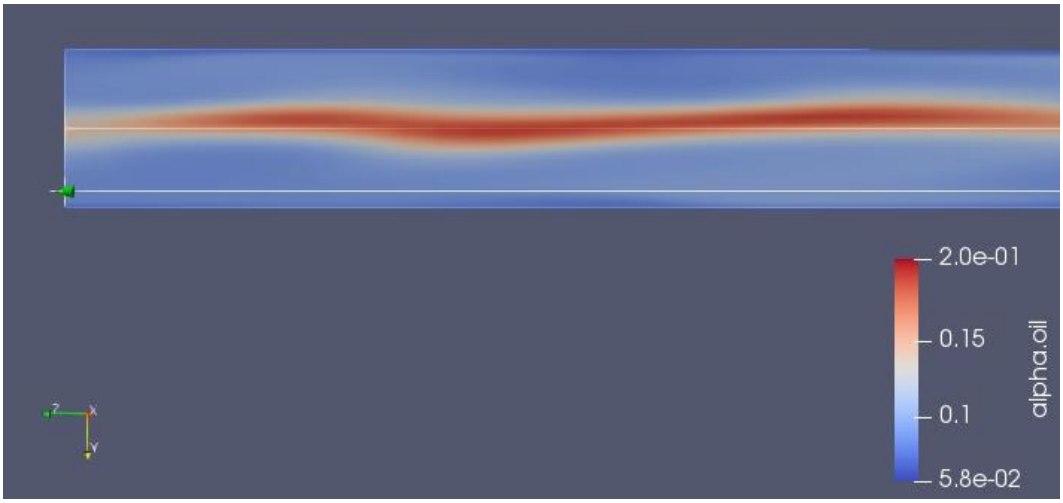


Figure 5.4: The sampling position at x = 0 and y = 0.01 along the z direction at time 27.6 s.

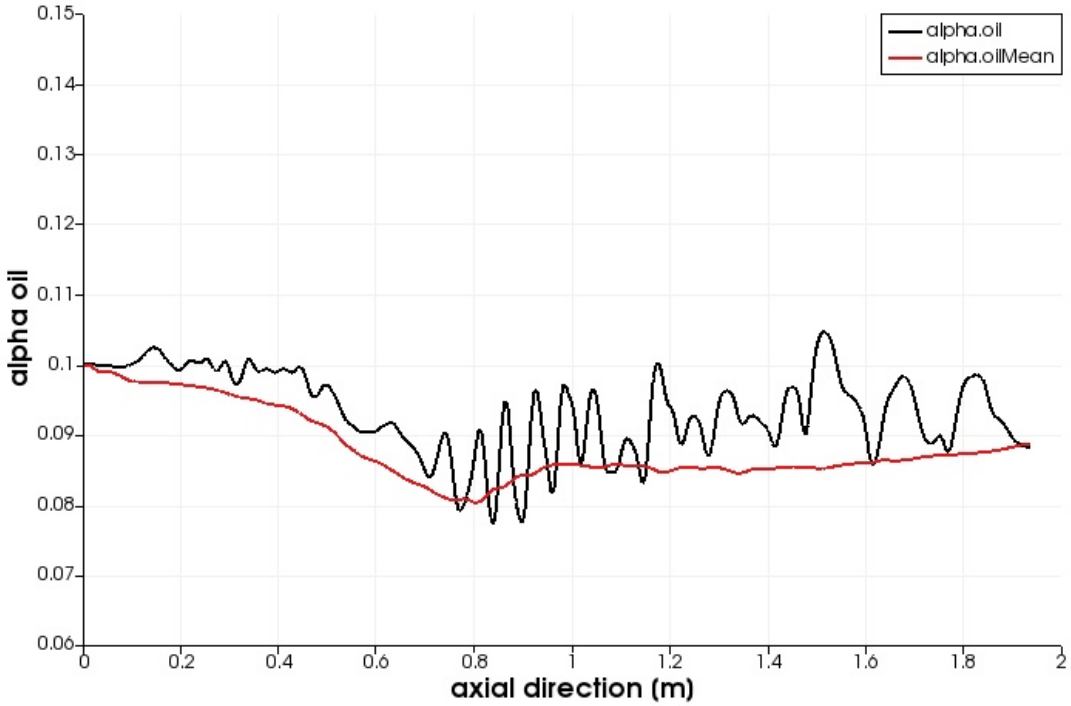


Figure 5.5: The oil concentration along axial direction at x = 0.01 m; y = 0 at time 27.6 s.

5.1.2 Velocity profiles

The axial and tangential velocity profiles are shown in this section. The velocity measurement is done at three different positions along the axial direction of the de-oiler. The positions are shown in Figure 5.6. At first, the axial velocity along the radial direction will be presented. Then, the tangential velocity along the radial direction is shown. Lastly, the axial velocity along the axial direction is presented and compared with a simple calculation of the continuity equation.

Figure 5.6: Different locations for measuring axial and tangential velocity.

Axial velocity

The axial velocity shown in Figure 5.7 is taken near the inlet of the de-oiler. The maximum axial velocity is seen around the center of the de-oiler while the minimum axial velocity i.e. zero is in the wall.

In Figure 5.8, the measurement is taken at the conical part of the de-oiler. The maximum axial velocity is near the center i.e. around 1 m/s while in the center the axial velocity is

decreased a little to around 0.7 m/s. Then, the axial velocity is declining continuously until it reaches the wall to 0 m/s.

Lastly, in Figure 5.9, the maximum velocity located in the center of the de-oiler is 2.2 m/s and decreases gradually to between 1.6 and 1.8 m/s. Then, the velocity falls significantly to near 0 m/s. The velocity is rising gradually from the axial position 0.04 m to 1.5 m.

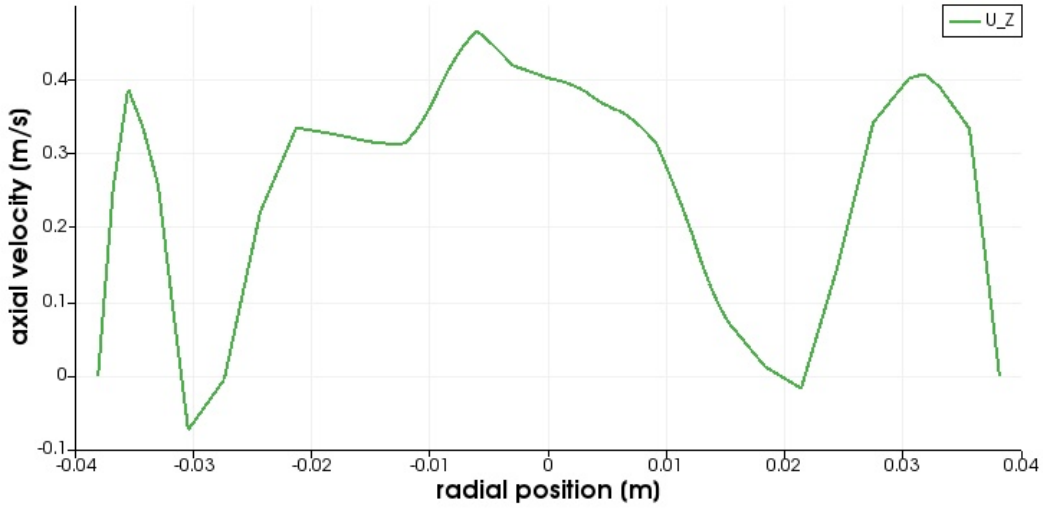


Figure 5.7: The axial velocity along the radial position at time = 27.6 s; x = 0 m; z = 0.04 m.

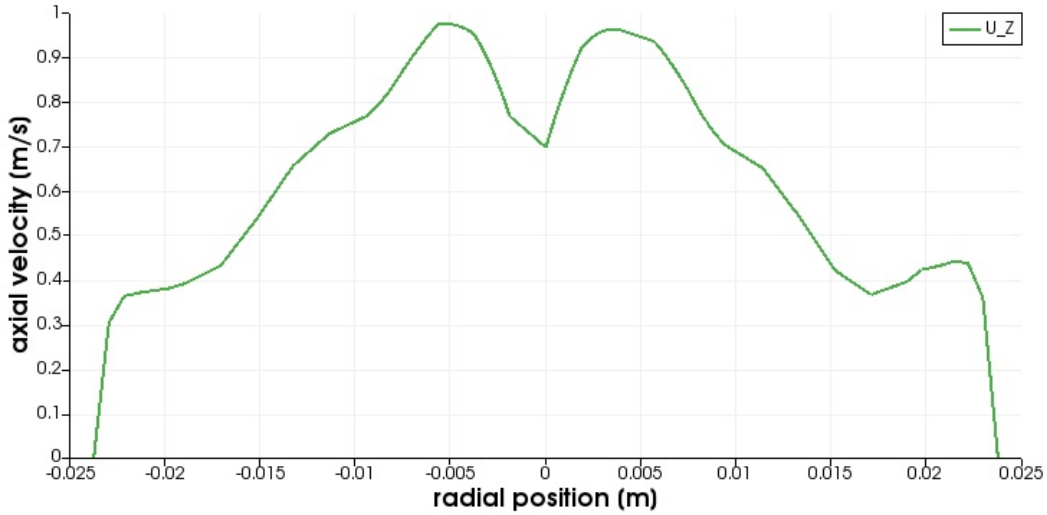


Figure 5.8: The axial velocity along the radial position at time = 27.6 s; x = 0 m; z = 0.35 m.

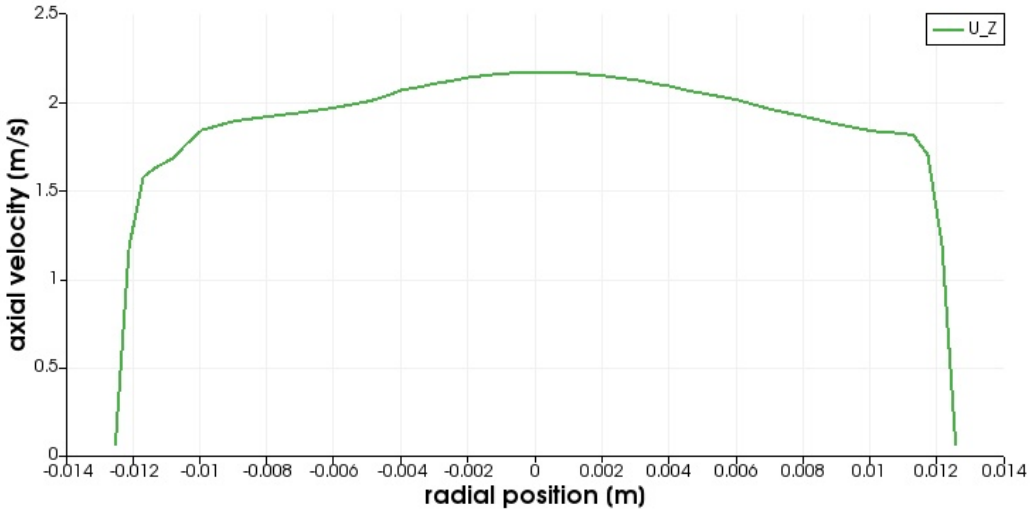


Figure 5.9: The axial velocity along the radial position at time = 27.6 s; x = 0 m; z = 1.5 m.

Tangential velocity

Similar to axial velocity, tangential velocities are also inspected in different axial at the de-oiler. Three figures which are Figure 5.10, Figure 5.11, and Figure 5.12 have a similar profile. The tangential velocity at a radial position of 0 m is almost 0 m/s in Figure 5.10, and 0 m/s in Figure 5.11, and Figure 5.12.

Next, Figure 5.10 and Figure 5.11 have the tangential velocities which are increased dramatically near the center. At a radial position around +0.01 or -0.01 m, the tangential velocities reach their maximum value i.e. 0.6796 m/s and 0.893 m/s for Figure 5.10 and Figure 5.11, respectively. It then decreased slowly until it reaches 0 m/s at the wall.

Figure 5.12 depicts the tangential velocity measured near the outlet of the de-oiler which is 1.5 m in axial position. The tangential velocity climbs gradually to its maximum i.e. 0.888 m/s at 0.01125 m and -0.01125 m/s at -0.6926 m.

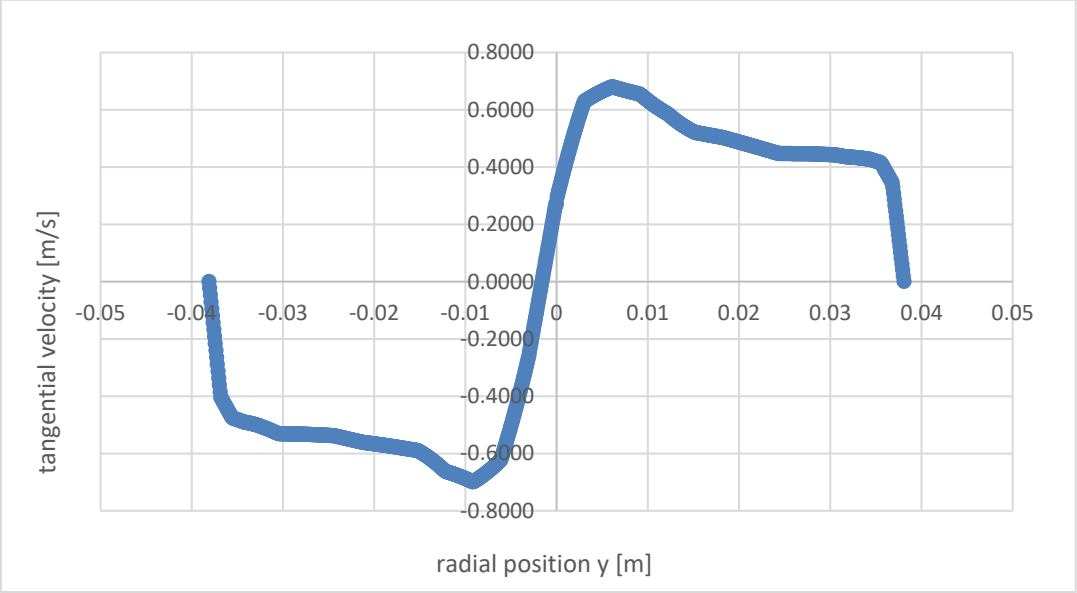


Figure 5.10: Tangential velocity at x = 0 m; y = -0.0381 to 0.0381 m; z = 0.04 m, t = 27.6 s.

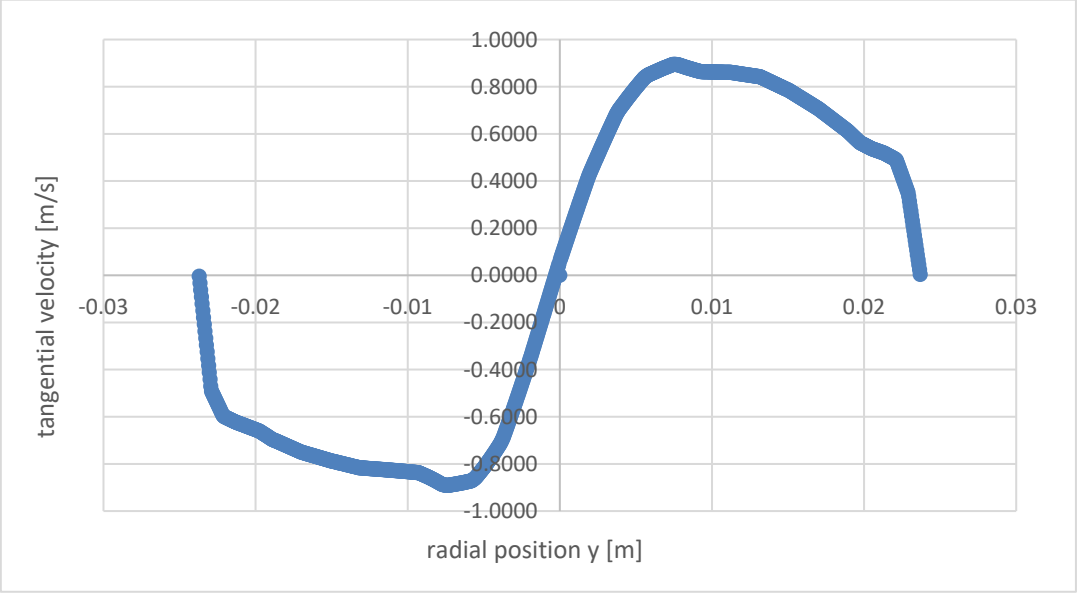


Figure 5.11: Tangential velocity at time = 27.6 s; x = 0 m; y = -0.0237 to 0.0237 m; z = 0.35 m.

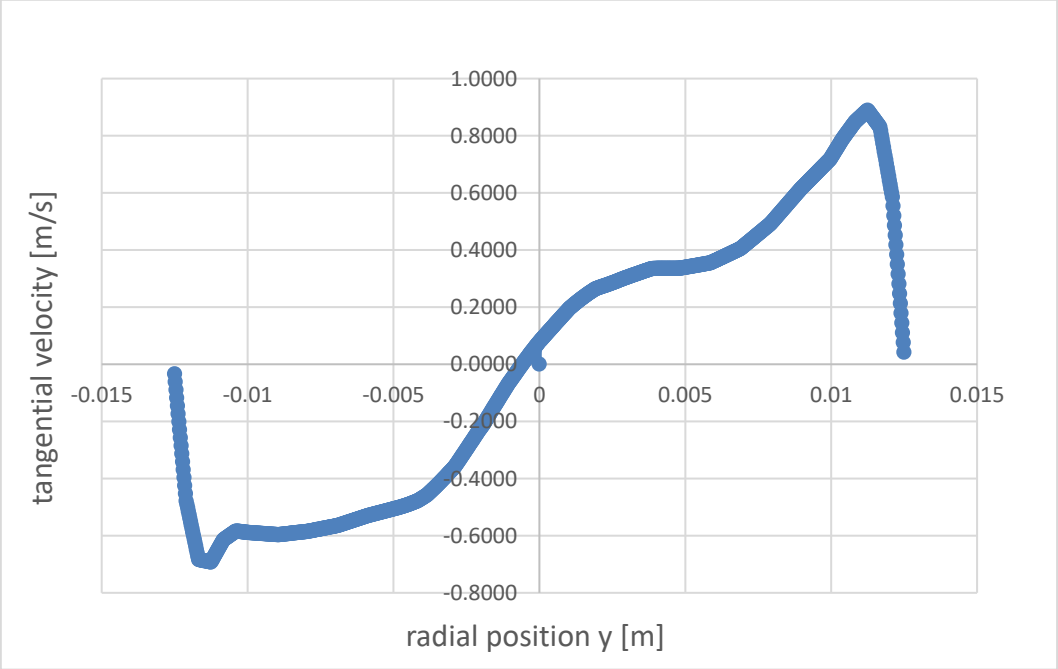


Figure 5.12: tangential velocity at x = 0; y = -0.0125 to 0.0125; z = 1.5 m, t = 27.6 s.

Figure 5.13 shows the growth of axial velocity along the axial position of the de-oiler. The inlet velocity is 0.2 m/s and it rises to 2.377 m/s at the axial position of 0.57 m. After that, the velocity seems to fluctuate in a certain range that is between 1.97 to 2.16 m/s until reaching the outlet velocity of 1.9146 m/s in the axial position of 1.9348 m. Run 1 gives a volumetric flow rate of 54.7 L/min.

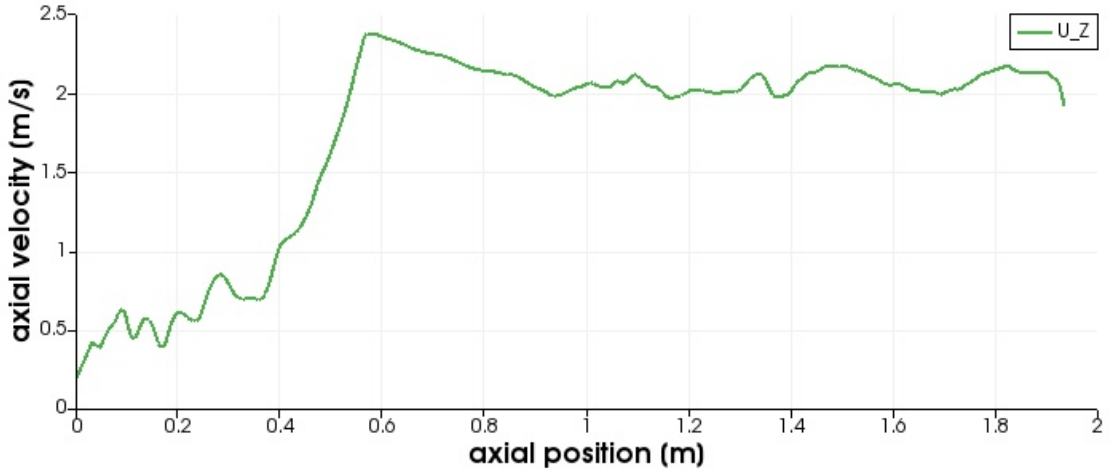


Figure 5.13: Run 1, magnitude velocity at time = 27.6 s; x = 0 m; y=0.01 m; z = 0 to 1.9348 m.

Run 2 has a bigger velocity boundary condition than Run 1 has. The axial velocity profile is given in Figure 5.14. The velocity in the inlet and outlet is 0.3 m/s and 2.4726 m/s, respectively, gives the volumetric flow rate 82.05 L/min.

Table 5.1 lists a comparison of axial velocity from OpenFOAM along the z-direction (axial position); x = 0; y = 0 m at time = 27.6 s for Run 1 and time = 7.25 s for Run 2.

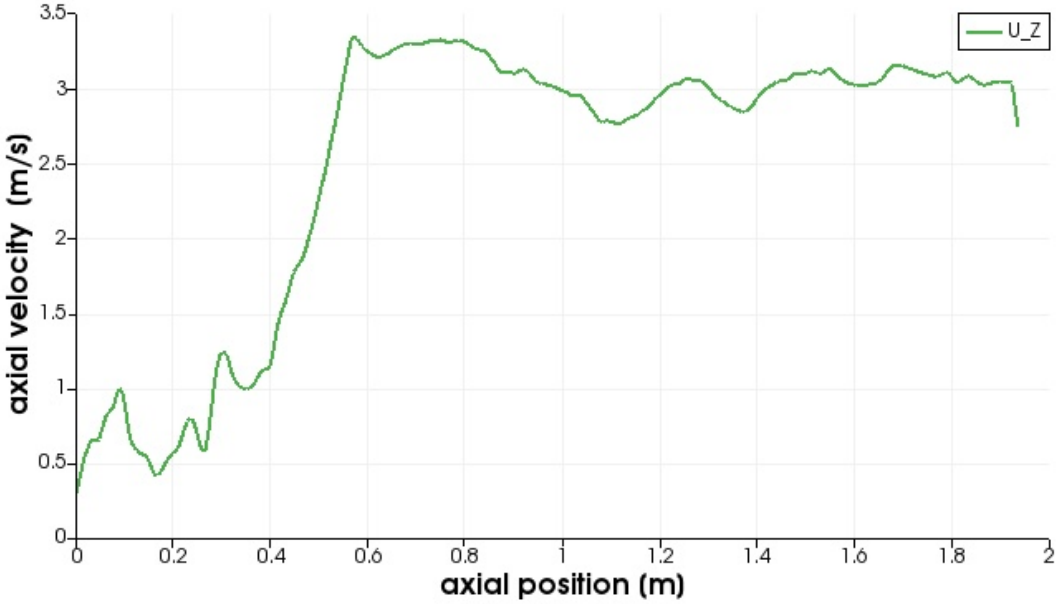


Figure 5.14: Run 2, magnitude velocity at time = 7.25 s; x = 0 m; y = 0 m; z = 0 to 1.9348 m.

Table 5.1: A comparison of axial velocity from OpenFOAM simulation and manually calculated continuity equation.

	A_1 [m^2]	A_2 [m^2]	u_1 [m/s]	u_2 [m/s]
Run1	0.004558	0.000496	0.2	1.9146
continuity equation	0.004558	0.000496	0.2	1.8365
Run2	0.004558	0.000496	0.3	2.7426
continuity equation	0.004558	0.000496	0.3	2.7548

5.1.3 Pressure

In Run 1, the pressure drops from 502272 Pa to 500000 Pa. It gives the total pressure drop of 2272 Pa or 0.0227 bar. The inlet and outlet magnitude velocities are 0.516 m/s and 2.00 m/s, respectively, and the inlet volumetric flow rate is 141.117 L/min.

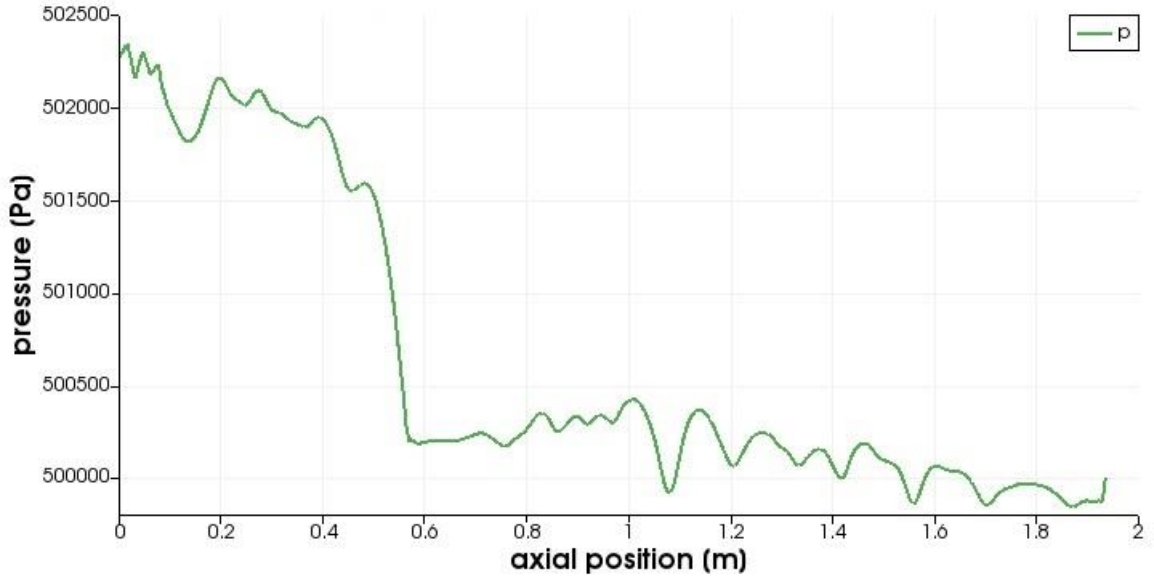


Figure 5.15: Run 1, the pressure along the de-oiler at time = 27.6 s; x = 0 m; y = 0.005 m; z = 0 to 1.9348 m.

Run 2 has the inlet pressure of 1039670 Pa and the outlet pressure of 1035430. This gives a pressure drop of 4240 Pa or 0.0424 bar. The inlet and outlet magnitude velocities are 0.73 m/s and 2.92 m/s, respectively, and the inlet volumetric flow rate is 199.643 L/min.

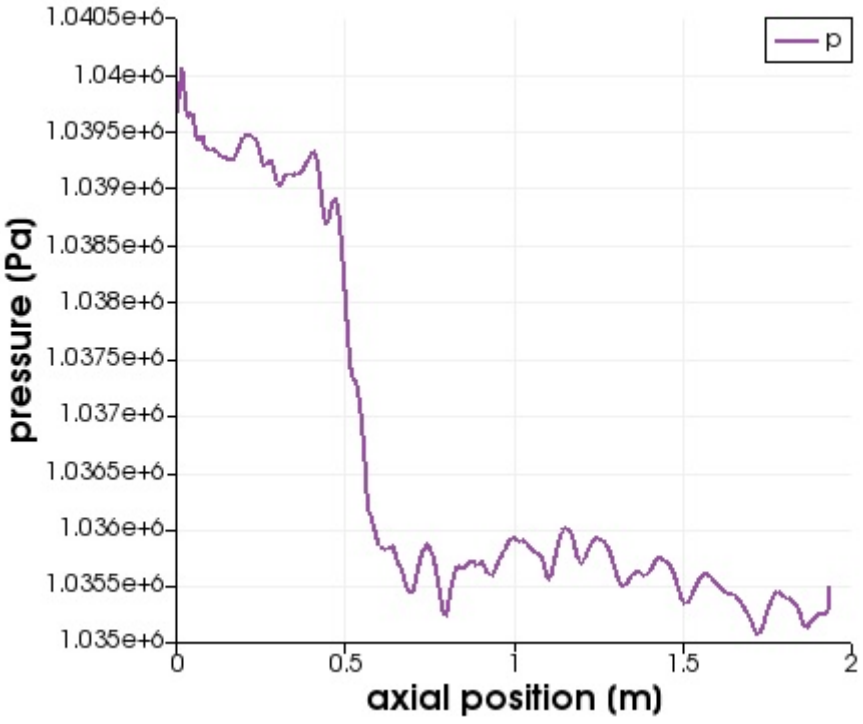


Figure 5.16: Run 2, the pressure along the de-oiler at time = 7.25 s; x = 0 m; y = 0.005 m; z = 0 to 1.9348 m.

Table 5.2 shows the pressure drop and velocity in both Run 1 and Run 2. The comparison of the pressure drop from some articles is presented in Figure 5.17. The pressure drop for Run 1

and Run 2 is relatively very low compared to others in Figure 5.17, even though it looks similar to the experimental results obtained by Meldrum (1988). From experimental work and CFD simulation conducted by Husveg et al. (2007) and Angelim et al. (2017), the pressure drop raises along with the flow rate increment. This also happens for Meldrum's (1988) work although the pressure loss difference between various flow rates is very small. The pressure drop obtained by Husveg's experiment (2007), Subsea 7 CFD simulation, and Kharoua's CFD simulation (2010) is corresponding to each other.

Lastly, the pressure drop is compared using de Gelder theoretical correlation. The pressure drop for the flow rate of 54.69 L/min is bigger than for 82.045 L/min. Nevertheless, de Gelder correlation predicts a very low pressure drop which is similar to Run 1 and Run 2. Detail data of Figure 5.17 can be found in Appendix F.

Table 5.2: The pressure drop and flow rate data for Run 1 and Run 2.

Run	A1 [m^2]	A2 [m^2]	u_1 [m/s]	u_2 [m/s]	flow rate [L/min]	dP [bar]	de Gelder dP tot [bar]
Run 1	0.004558	0.000496	0.2	1.9146	54.69666	0.02272	0.02284
Run 2	0.004558	0.000496	0.3	2.7426	82.045	0.0424	0.01349

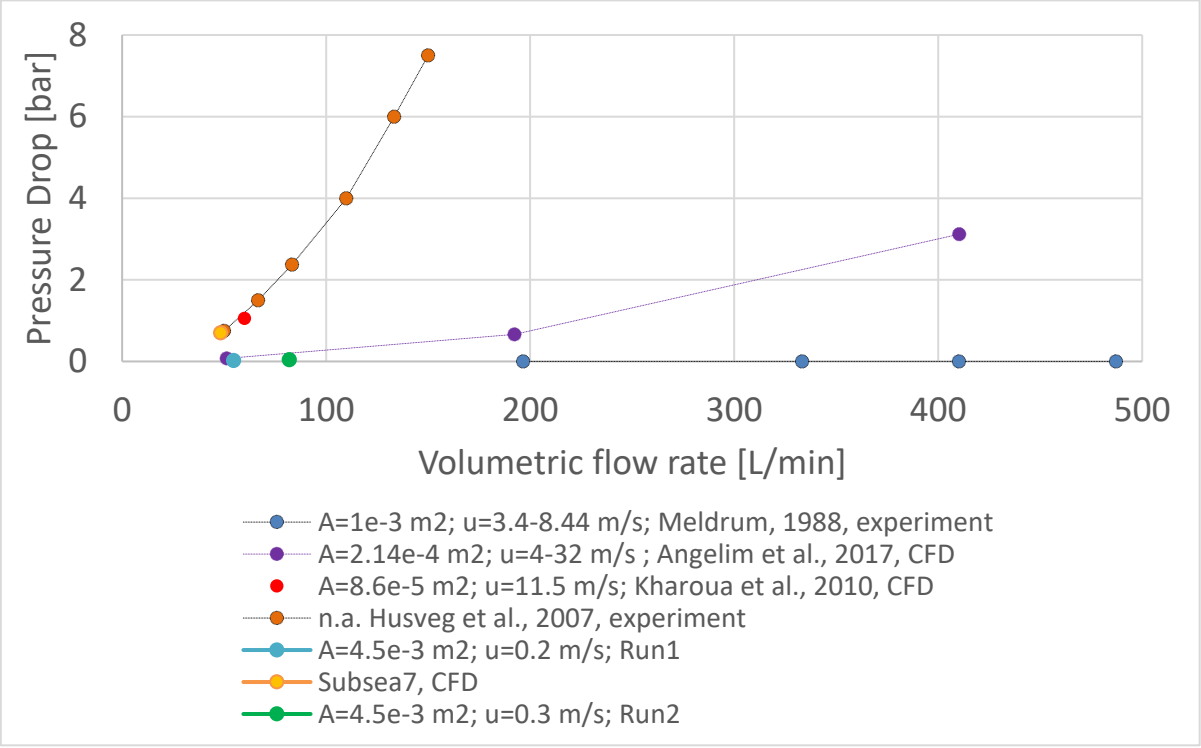


Figure 5.17: Pressure drop vs volumetric flow rate for different de-oiler cases.

5.1.4 Geometry

The geometry including the number of de-oilers needed to clean produced water range from 60000-80000 barrels per day is presented in Table 5.3. The simple geometry of the tangential inlet can be seen in Figure 5.18.

Table 5.3: The oil concentration of cleaned produced water with various flow rate and numbers of de-oiler needed.

Results

Variable / constant	Value		
Q [barrels per day]	60000	70000	80000
Q_{feed} [m3/h]	397.468	463.713	529.958
$Q_{feed,water}$ [m3/h]	357.721	417.342	476.962
$Q_{feed,oil}$ [m3/h]	39.747	46.371	52.996
$\frac{K_{ds}}{K_i}$	0.24	0.24	0.24
efficiency	0.76	0.76	0.76
K_{ds}	0.024	0.024	0.024
K_i	0.1	0.1	0.1
$Q_{HPO,oil}$ [m3/h]	9.444	11.018	12.592
Q_{HPO} [m3/h]	393.494	459.076	524.658
FS	0.010	0.010	0.010
Q_{LPO} [m3/h]	3.975	4.637	5.300
ppmv oil in water	23.44	23.44	23.44
$u_{axial,feed}$ [m/s]	0.2		
de-oiler capacity [m3/h]	3.28	3.28	3.28
n de-oiler	121.1	141.3	161.5
$u_{axial,feed}$ [m/s]	0.5		
de-oiler capacity [m3/h]	8.20	8.20	8.20
n de-oiler	48.45	56.52	64.59
$u_{axial,feed}$ [m/s]	1		
de-oiler capacity [m3/h]	16.41	16.41	16.41
n de-oiler	24.22	28.26	32.30

notes:

1. FS = flow split, defined in equation (2.17)
2. 1 barrel per day = 0.006624 m^3/h
3. assumed efficiency according to Young et al. (1994) = 76%
4. assumed overflow = 1% mentioned in 2.3.2

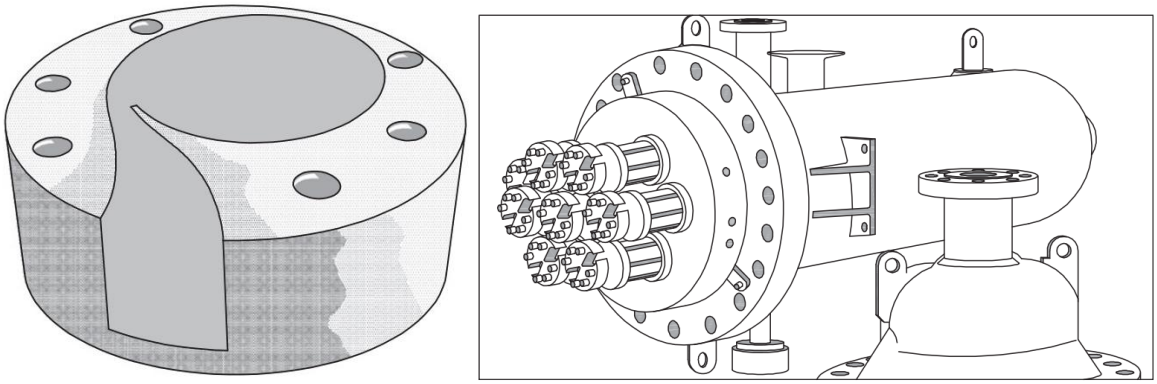


Figure 5.18: Tangential inlet design for de-oiler (left) and compact design of de-oilers in a vessel. (Stewart & Arnold, 2009, p. 185-186)

6 Discussion

Firstly, as mentioned in section 2.3.3, the geometry used in this work is from the article by Young et al. (1994). In Figure 5.4 and Figure 5.5, the alpha oil in the HPO is 0.088 which leads to efficiency 12%. Alpha oil in HPO is expected to be much lower i.e 0.024 to have an efficiency as big as 76% similar to in the article by Young et al. (1994, p. 47). This may be because of too short simulation time. As can be seen in Figure 5.1, the mean alpha oil is constantly increasing, which means the oil will concentrate in the center while the water (HPO) will have less and less oil over time. As a comparison, Liu et al. (2018, p. 63) did a CFD simulation for the axial hydrocyclone with axial inlet velocity 2-3 m/s. The time required to make the oil concentrated in the center is 21 s for alpha oil = 0.1, 33 s for alpha oil = 0.25, 50 s for alpha oil = 0.7, and 73 s to finally reach 0.9 of alpha oil. Then, it can be suspected that as time progresses, the center vortex can have a bigger alpha oil. Another reason is the inlet velocity is probably too low which can cause low swirl intensity. This was proven by Liu et al. (2018, p. 63), in their experiment, the efficiency was increased by having a bigger rotating velocity. Moreover, the data of alpha oil which is measured in x = 0, y = 0, and z = 1.9348 m is 0.14, however, the maximum of alpha oil can reach above 0.2.

Secondly, the axial and tangential velocity profiles for three different positions will be discussed, and the axial velocity profile along the axial direction will be explained. The axial velocity near the inlet and outlet depicted in Figure 5.7 and Figure 5.9, respectively, has its maximum value in the center of the de-oiler and minimum velocity near the wall. In the wall, the boundary condition which is `fixedValue` and `uniform (0 0 0)` is satisfied. The profile is in line with what is generally happened. (Araujo et al., 2020, p. 7)

On the other hand, Figure 5.8 presents an unusual profile of axial velocity. It appears as inverted “W”, a similar result was obtained by Araujo et al. (2020, p. 7-8). They stated that the physical reason for the inverted “W” axial velocity profile along the radial position is still unknown. Furthermore, the axial velocity located in the center of the de-oiler is rising gradually. From the inlet to the outlet, the cross-sectional area is reduced. The energy from inlet pressure is occupied to increase the kinetic energy. (Nave, 2017)

Tangential velocity is critical in separation using de-oiler. It relates to the centrifugal field inside the de-oiler which separates the oil and water. All tangential profile shapes have a similarity. The magnitude reaches its peak near the centerline. This is where the forced and free vortex region is separated. This is called a Rankine vortex. The rest is the tangential velocity is down to zero at the wall which satisfies the wall boundary condition. Similarly, the shape of the tangential velocity profile was obtained by Araujo et al. (2020, p. 7) and Saidi (2012, p. 1827) in their works.

The tangential velocity increases from the near inlet (cylindrical section) to the conical section of the de-oiler to the second cylindrical section. The conical section is intentionally used for preventing the loss of angular momentum caused by the drag force between fluid and the wall. Therefore, the increase in tangential velocity from the inlet to the outlet is expected. (Young et al., 1994, p. 44)

Next, the axial velocity along the de-oiler is shown in Figure 5.13 and Figure 5.14. The result in Table 5.1 shows that only a little difference between the axial velocity provided by OpenFOAM simulation and by the manually calculated simple continuity equation.

Thirdly, it is necessary to validate the model which is used in this work to predict the pressure fields. De Gelder pressure drop theoretical correlation is chosen to predict the pressure loss inside the de-oiler. The correlation predicts a very low pressure drop. This value is closer to the pressure drop resulted from Run 1 and Run 2, compared to other literature. However, a trend of the pressure drop obtained by de Gelder correlation is unexpected. It predicts a bigger pressure drop for a lower flow rate.

Besides, the pressure loss from Run 1 and Run 2 is compared to CFD and experimental works taken from various articles. Both Run 1 and Run 2 have a low pressure drop which is desired. By having a low pressure drop, the cleaned produced water can be re-injected to the well with lower energy requirement. However, the pressure drop in this simulation is assumed measured exactly in the inlet and outlet face of the de-oiler. In real condition, the pressure measuring devices are installed at some distance of the inlet, overflow, and underflow of de-oiler which may cause an additional pressure drop.

Lastly, with assumed efficiency 76% according to Young et al. (1994, p. 47), the produced water will contain only 23.44 ppmv which is less than 30 ppm. The number of de-oiler is varied according to the feed volumetric flow rate of produced water. The suggested inlet design of the de-oiler is shown in Figure 5.18. The tangential de-oiler is chosen because it has more simple geometry than the axial de-oiler, also a better efficiency, as already explained in 2.3.3. The design of the tangential cyclone inlet in Figure 5.18 makes a possibility to arrange the de-oiler to be more compact and suitable for the subsea separation system. Moreover, the challenge of subsea specific design issues and recommendations has already been explained in 2.8.

7 Conclusion

Various geometry and types of de-oiler for oil and water separation have been reviewed. Based on efficiency, the geometry for simulation is selected.

3D geometry and mesh have been generated using SALOME and an Euler-Euler model from *bubbleColumn* example has been modified to run a CFD simulation in OpenFOAM.

After 27.6 s of simulation, the oil concentration in the heavy phase outlet is still very high i.e 0.088 and the corresponding efficiency is 12%. Longer simulation time and a powerful computer are needed to obtain more data to see if the oil concentration in the heavy phase outlet can decline.

With the assumption efficiency 76%, the number of de-oilers needed to separate 60000-80000 barrels of produced water per day varies between 25-162. It depends on the velocity used.

The axial velocity profile shapes along the radial direction are in line with the literature, except for one with an inverted "W" profile and the physical reason for that is still unknown.

The tangential velocity is important for the separation and it goes up along axial direction.

The pressure drop is obtained to be very low compared to most of the literature, which is 0.02272 bar for 54.696 L/min and 0.0424 bar for 82.045 L/min.

7.1 Future work

For future work, a powerful computer with more cores is needed to simulate a complex simulation so that it can be run until sufficient end time.

The pressure drop seems to be very low, a further inspection of the model is needed.

The turbulent phenomena should be included after the model reach convergence.

Try different variations of inlet velocity boundary conditions and alpha oil.

References

[1] 2B1st Consulting. (2012). Tie-Back. Retrieved from https://www.2b1stconsulting.com/tie-back/

[2] Aadal, A., Djuraev, A., Costa, C., Milak, H., Dlima, M., & Tianqi, S. (2016). *Semester project, EiT Liquid-Liquid Separation Subsea*. Retrieved from https://www.semanticscholar.org/paper/Semester-project-%2C-EiT-Liquid-Liquid-Separation-Aadal-Djuraev/8341983ee92b5916ba259e8d1b0043ab7a0d5d8d

[3] Abulencia, J. P., & Theodore, L. (2009). Fluid Flow for the Practicing Chemical Engineer. Retrieved from https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470423851.app1

[4] Angelim, K., Lima, A., Souza, J., Neto, S., Oliveira, V., & Moreira, G. (2017). Applying CFD in the Analysis of Heavy Oil/Water Separation Process via Hydrocyclone. *International Journal of Multiphysics, 11*(2), 151-168. Retrieved from https://pdfs.semanticscholar.org/7135/c731b903dc9169f4fae68b4daf06ac0f3d9d.pdf

[5] Bai, Y., & Bai, Q. (2019). 26 - Subsea Processing. In Y. Bai & Q. Bai (Eds.), *Subsea Engineering Handbook (Second Edition)* (pp. 769-804). Boston: Gulf Professional Publishing. https://doi.org/10.1016/B978-0-12-812622-6.00026-9

[6] Bilstad, T., Nair, R. R., & Protasova, E. (n.d.). *Nanofiltration for Produced Water Reinjection for EOR*. [PowerPoint slides]. Retrieved from https://www.uis.no/getfile.php/13303879/IOR-senter/Crete_2016_Bilstad.pdf

[7] Bradley, D. (1965). The Hydrocyclone (1st ed.). Great Britain: Pergamon.

[8] Braga, E. R., Huziwara, W. K., Martignoni, W. P., Scheid, C. M., & Medronho, R. A. (2015). Improving Hydrocyclone Geometry for Oil/Water Separation. *Brazilian Journal of Petroleum and Gas, 9*(3), 115-123. doi:10.5419/bjpg2015-0012

[9] Bram, M. V., Hansen, L., Hansen, D. S., & Yang, Z. (2018). *Hydrocyclone Separation Efficiency Modeled by Flow Resistances and Droplet Trajectories. In Proceedings of the 3rd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production* (pp. 133-134). Esbjerg, Denmark: Aalborg University.

[10] Busch, S. (2015). *Project work: A twophaseEulerFoam tutorial*. Retrieved from http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2014/Sandra%20Busch/Tutorial_multiphaseEulerFoam_peered.pdf

[11] Campen, L. v. (2014). *Bulk Dynamics of Droplets in Liquid-Liquid Axial Cyclones*. (Doctoral thesis). Technische Universiteit Delft, Delft.

[12] Cappelli, D. (2018). *A detailed description of reactingTwoPhaseEulerFoam focussing on the links between mass and heat transfer at the interface*. Retrieved from http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2018/DarrenCappelli/Report_CAPPELLI.pdf

[13] Carlos, G., Juan, C., Shoubo, W., Luis, G., Mohan, R., & Ovadia, S. (2002). Oil/Water Separation in Liquid/Liquid Hydrocyclones (LLHC): Part 1 - Experimental Investigation. *SPE Journal, 7*(4), 353-372. doi:10.2118/81592-PA

[14] CFD Direct. (n.d.). Retrieved from https://cfd.direct/openfoam/user-guide/v6-boundaries/

[15] cfd-online. (2016). twoPhaseEulerFoam diverging. Retrieved from https://www.cfd-online.com/Forums/openfoam-solving/165213-twophaseeulerfoam-diverging.html

[16] cfdyna. (n.d.). *phaseProperties*. Retrieved from http://www.cfdyna.com/Home/OpenFOAM/phaseProperties.txt

[17] CIOL - EXXONMOBIL CHEMICAL CANADA. (2016). *EXXSOL™ D60 SOLVENT* [Safety Data Sheet]. Retrieved from https://www.chemical.net/content/images/uploaded/sds/Exxsol%20D60%20(12560-g).pdf

[18] Damián, S. M. (2013). *An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces* (Doctoral dissertation, Universidad Nacional del Litoral). Retrieved from https://drive.google.com/file/d/0B2lpdhG-Zh05Y0ZzOHVLb3lGekk/edit

[19] Darren. (2018). CFD: PIMPLE Algorithm. Retrieved from https://www.simscale.com/forum/t/cfd-pimple-algorithm/81418

[20] Das, T., & Jäschke, J. (2018). Modeling and control of an inline deoiling hydrocyclone ∗∗This work was supported by SUBPRO, the Norwegian Research Council and DNV GL. *IFAC-PapersOnLine*, *51*(8), 138-143. https://doi.org/10.1016/j.ifacol.2018.06.368

[21] de Araújo, C. A. O., Scheid, C. M., Loureiro, J. B. R., Klein, T. S., & Medronho, R. A. (2020). Hydrocylone for oil-water separations with high oil content: Comparison between CFD simulations and experimental data. *Journal of Petroleum Science and Engineering*, *187*, 106788. https://doi.org/10.1016/j.petrol.2019.106788

[22] Dirkzwager, M. (1996). *A New Axial Cyclone Design for Fluid-Fluid Separation* (Doctoral thesis). Technische Universiteit Delft, Delft.

[23] Durst, F., Miloievic, D., & Schönung, B. (1984). Eulerian and Lagrangian predictions of particulate two-phase flows: a numerical study. *Applied Mathematical Modelling*, *8*(2), 101-115. https://doi.org/10.1016/0307-904X(84)90062-3

[24] Elam, S. K., Tokura, I., Saito, K., & Altenkirch, R. A. (1989). Thermal conductivity of crude oils. *Experimental Thermal and Fluid Science*, *2*(1), 1-6. https://doi.org/10.1016/0894-1777(89)90043-5

[25] Engineering ToolBox. (2003). Specific Heat of some Liquids and Fluids. Retrieved from https://www.engineeringtoolbox.com/specific-heat-fluids-d_151.html

[26] Fluor Offshore Solutions. (2012). *Ultra Deep Water Discharge of Produced Water and/or Solids at the Seabed.* (09121-3100-01). Retrieved from https://rpsea.org/sites/default/files/2018-06/09121-3100-01-FR-UDW_Discharge_Produced_Water_Solids_Seabed-04-24-12_P.pdf

[27] Greenshields, C. (2014). OpenFOAM 2.3.0: Discrete Particle Modelling. Retrieved from https://openfoam.org/release/2-3-0/dpm/

[28] Greenshields, C. (2018). OpenFOAM v6 User Guide: 4.3 Time and data input/output control. Retrieved from https://cfd.direct/openfoam/user-guide/v6-controldict/

[29] Greenshields, C. (2018). OpenFOAM v6 User Guide: 4.4 Numerical schemes. Retrieved from https://cfd.direct/openfoam/user-guide/v6-fvschemes/

[30] Greenshields, C. (2018). OpenFOAM v6 User Guide: 4.4 Numerical schemes. Retrieved from https://cfd.direct/openfoam/user-guide/v6-fvschemes/

[31] Guen, R. L. (2016). *Numerical investigation of bubbles in channel flow Implementation and analysis of the influence of bubbles injection on the drag of a ship*. (Master's thesis). Chalmers University of Technology, Gothenburg.

[32] Gullberg, R. (2017). *Computational Fluid Dynamics in OpenFOAM Mesh Generation and Quality*. Retrieved from http://folk.ntnu.no/preisig/HAP_Specials/AdvancedSimulation_files/2017/project%20reports/CFD/Rebecca%20Gullberg%20-%20CFD_Mesh_Report.pdf

[33] Haddadi, B., Gößnitzer, C., Nagy, J., Natarajan, V., Zibuschka, S., & Chen, Y. (2019). *OpenFOAM® Basic Training*. Retrieved from https://www.cfd.at/downloads/FoamTutorial_8-ExampleEight.pdf

[34] Huang, S. (2005). Numerical Simulation of Oil-water Hydrocyclone Using Reynolds-Stress Model for Eulerian Multiphase Flows. *The Canadian Journal of Chemical Engineering, 83,* 829-834. https://doi.org/10.1002/cjce.5450830504

[35] Husveg, T. (2007). *Operational Control of Deoiling Hydrocyclones and Cyclones for Petroleum Flow Control*. (Doctoral thesis). University of Stavanger, Stavanger.

[36] Husveg, T., Johansen, O., & Bilstad, T. (2007). Operational Control of Hydrocyclones During Variable Produced Water Flow Rates—Frøy Case Study. *SPE Production & Operations, 22*(03), 294-300. doi:10.2118/100666-PA

[37] Huusari, L. (2015). *Analysis of Phase Separator Design Criteria Using Computational Fluid Dynamics* (Master's thesis, Lappeenranta University Of Technology). Retrieved from http://urn.fi/URN:NBN:fi-fe2015100814712

[38] Igunnu, E. T., & Chen, G. Z. (2012). Produced water treatment technologies. *International Journal of Low-Carbon Technologies, 9*(3), 157-177. doi:10.1093/ijlct/cts049

[39] Incropera, F. P., Dewitt, D. P., Bergman, T. L., & Lavine, A. S. (2013). *Principles of Heat and Mass Transfer* (7th ed.). Singapore: John Wiley & Sons Singapore Pte. Ltd.

[40] Kharoua, N., Khezzar, L., & Nemouchi, Z. (2010). Computational fluid dynamics study of the parameters affecting oil-water hydrocyclone performance. *Journal of Process Mechanical Engineering, 224*(2), 119-128. https://doi.org/10.1243/09544089JPME304

[41] Kharoua, N., Khezzar, L., & Nemouchi, Z. (2010). Hydrocyclones for De-oiling Applications—A Review. *Petroleum Science and Technology, 28*(7), 738-755. doi:10.1080/10916460902804721

[42] King, G. C. (2018). *Physics of Energy Sources* (1st ed.). New Jersey: John Wiley & Sons, Ltd.

[43] Kolev, N. I. (2012). Drag, lift, and virtual mass forces. In N. I. Kolev (Ed.), *Multiphase Flow Dynamics 2: Mechanical Interactions* (pp. 31-85). Berlin, Heidelberg: Springer Berlin Heidelberg.

[44] Learning About Electronics. (2018). Magnitude calculator. Retrieved from http://www.learningaboutelectronics.com/Articles/Magnitude-calculator.php

[45] Liu, H.-f., Xu, J.-y., Zhang, J., Sun, H.-q., Zhang, J., & Wu, Y.-x. (2012). Oil/water separation in a liquid-liquid cylindrical cyclone. *Journal of Hydrodynamics, Ser. B, 24*(1), 116-123. https://doi.org/10.1016/S1001-6058(11)60225-4

[46] Liu, M., Chen, J., Cai, X., Han, Y., & Xiong, S. (2018). Oil–water pre-separation with a novel axial hydrocyclone. *Chinese Journal of Chemical Engineering, 26*(1), 60-66. https://doi.org/10.1016/j.cjche.2017.06.021

[47] Liu, M., Chen, J., Cai, X., Han, Y., & Xiong, S. (2018). Oil–water pre-separation with a novel axial hydrocyclone. *Chinese Journal of Chemical Engineering, 26*(1), 60-66. https://doi.org/10.1016/j.cjche.2017.06.021

[48] Maliska, C. R., & Paladino, E. E. (2006). *The Role of Virtual Mass, Lift and Wall Lubrication Forces in Accelerated Bubbly Flows*. Retrieved from http://www.sinmec.ufsc.br/site/arquivos/g-gvwkdvqpux_the_role_of_virtual.pdf

[49] Manni, A. (2014). *Project work: An introduction to twoPhaseEulerFoam with addition of an heat exchange model*. Retrieved from http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2014/Alessandro%20Manni/ReportAM.pdf

[50] Meldrum, N. (1988). Hydrocyclones: A Solution to Produced-Water Treatment. *SPE Production Engineering, 3*(04), 669-676. doi:10.2118/16642-PA

[51] Miller, R. J. (1996). 7 - Petroleum Economics. In W. C. Lyons (Ed.), *Standard Handbook of Petroleum and Natural Gas Engineering* (pp. 985-1033). Houston: Gulf Professional Publishing. https://doi.org/10.1016/B978-088415643-7/50011-X

[52] Moshfeghian, M. (2015). Gas-Liquid Separators Sizing Parameter. Retrieved from http://www.jmcampbell.com/tip-of-the-month/2015/09/gas-liquid-separators-sizing-parameter/

[53] Nascimento, M. R. M., Bicalho, I. C., Mognon, J. L., Ataíde, C. H., & Duarte, C. R. (2012). Performance of a New Geometry of Deoiling Hydrocyclones: Experiments and Numerical Simulations. *Chemical Engineering Technology, 36*(1), 98-108. DOI: 10.1002/ceat.201200258

[54] Nave, R. (2017). Bernoulli Equation. Retrieved from http://hyperphysics.phy-astr.gsu.edu/hbase/pber.html

[55] Noroozi, S., & Hashemabadi, S. H. (2009). CFD Simulation of Inlet Design Effect on Deoiling Hydrocyclone Separation Efficiency. *Chemical Engineering & Technology, 32*(12), 1885-1893. doi:10.1002/ceat.200900129

[56] OpenCFD Ltd (ESI Group). (n.d.). 6.2 Numerical schemes. Retrieved from https://www.openfoam.com/documentation/user-guide/fvSchemes.php

[57] OpenFOAM. (n.d.). 5.2 Thermophysical models. Retrieved from https://www.openfoam.com/documentation/user-guide/thermophysical.php

[58] OpenFOAM. (n.d.). A.4 Standard boundary conditions. Retrieved from https://www.openfoam.com/documentation/user-guide/standard-boundaryconditions.php

[59] OpenFOAM: API Guide v1912. (n.d.). swirlInletVelocityFvPatchVectorField. Retrieved from https://www.openfoam.com/documentation/guides/latest/api/classFoam_1_1swirlInletVelocityFvPatchVectorField.html

[60] OpenFOAM: User Guide v1912. (2016-2017). Van Leer divergence scheme. Retrieved from https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-divergence-vanleer.html

[61] openfoamwiki. (2009). DecomposePar. Retrieved from https://openfoamwiki.net/index.php/DecomposePar

[62] openfoamwiki. (2018). OpenFOAM guide/The PIMPLE algorithm in OpenFOAM. Retrieved from https://openfoamwiki.net/index.php/OpenFOAM_guide/The_PIMPLE_algorithm_in_OpenFOAM

[63] Paladino, E. E., & Maliska, C. R. (n.d.). *Virtual mass in accelerated bubbly flows*. Retrieved from http://arquivos.info.ufrn.br/arquivos/2008242042348603252780cf13b053ed/paladino_eurotherm.pdf

[64] Parks, G. S., & Todd, S. S. (1929). Heats of Fusion of Some Paraffin Hydrocarbon. *Industrial & Engineering Chemistry, 21*(12), 1235-1237. doi:10.1021/ie50240a022

[65] Pettersen, T. S. (2008). *Dynamic model of offshore processing plant with emphasis on estimating environmental impact*. (Master's thesis). Norwegian University of Science and Technology, Trondheim. Retrieved from http://folk.ntnu.no/skoge/diplom/diplom08/pettersen_tone/Diplom_Tone.pdf

[66] Rusche, H. (2002*). Computational Fluid Dynamics of Dispersed Two-Phase Flow at High Phase Fractions* (Doctoral dissertation). Imperial College of Science, Technology & Medicine, London. Retrieved from https://spiral.imperial.ac.uk/handle/10044/1/8110

[67] Saidi, M., Maddahian, R., & Farhanieh, B. (2012). A parametric Study on Deoiling Hydrocyclones Flow Field. *International Journal of Mechanical and Mechatronics Engineering, 6*(9), 1824-1828.

[68] Simms, K. M., Zaidi, S. A., Hashmi, K. A., Thew, M. T., & Smyth, I. C. (1992). Testing of the Vortoil Deoiling Hydrocyclone Using Canadian Offshore Crude Oil. *Fluid Mechanics and Its Applications, 12*(-), 295-308. https://doi.org/10.1007/978-94-015-7981-0_19

[69] Skjefstad, H. S., & Stanko, M. (2019). Experimental performance evaluation and design optimization of a horizontal multi-pipe separator for subsea oil-water bulk separation. *Journal of Petroleum Science and Engineering, 176*, 203-219. https://doi.org/10.1016/j.petrol.2019.01.027

[70] Slot, J. J. (2013). *Development of a centrifugal in-line separator for oil-water flows*. (Doctoral thesis). University of Twente, Enschede. Retrieved from https://doi.org/10.3990/1.9789036535427

[71] Society of Petroleum Engineers. (n.d.). Field Developments and Technical Solutions – Marginal Fields. Retrieved from https://www.spe.org/en/training/courses/acai/

[72] Stewart, M., & Arnold, K. (2009). Chapter 3 - Produced Water Treating Systems. In M. Stewart & K. Arnold (Eds.), *Emulsions and Oil Treating Equipment* (pp. 107-211). Burlington: Gulf Professional Publishing. https://doi.org/10.1016/B978-0-7506-8970-0.00003-7

[73] Technische Universität Wien. (2019). *Tutorial Eight Multiphase*. Retrieved from https://www.cfd.at/downloads/FoamTutorial_8-ExampleEight.pdf

[74] Thew, M. T. (2000). Cyclones for oil/water separation. *Encyclopedia of Separation Science*, 1480-1490. doi:10.1016/B0-12-226770-2/04601-9

[75] Thomas, E. (2016). *CFD analysis of gas-particle flow in a scaled circulating fluidized bed* (Master's thesis, University of Stavanger). Retrieved from https://uis.brage.unit.no/uis-xmlui/bitstream/handle/11250/2411372/Engen%2C%20Thomas.pdf?sequence=1&isAllowed=y

[76] Tresvik, K. L. (2016). *Experimental and numerical investigation of oil-water dispersions* (Master's thesis). Norwegian University of Science and Technology, Trondheim.

[77] Tyvold, P. F. (2015). *Modeling and Optimization of a Subsea Oil-Water Separation System* (Master's thesis, Norwegian University of Science and Technology). Retrieved from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2351693

[78] Versteeg, H. K., & Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics*. England: Pearson Education Limited.

[79] Widnall, S., & Peraire, J. (2008). *Lecture L5 - Other Coordinate Systems*. Retrieved from https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16_07F09_Lec05.pdf

[80] Wimshurst, A. (Producer). (2019, 2 May 2020). *[CFD] Eulerian Multi-Phase Modelling* [Video clip]. Retrieved from https://www.youtube.com/watch?v=6BJauDTpCmo&t=334s

[81] Wolbert, D., Ma, B.-F., & Aurelle, Y. (1995). Efficiency Estimation of Liquid-Liquid Hydrocyclones Using Trajectory Analysis. *AIChE*, 41(6), 1395-1396. https://doi.org/10.1002/aic.690410606

[82] Wolf Dynamics. (n.d.). *A crash introduction to multiphase flows modeling OpenFOAM®*. [PowerPoint slides]. Retrieved from http://www.wolfdynamics.com/wiki/advanced_multiphase_flows1.pdf

[83] Xodus Group. (2020). *CFD Analysis of the Pipe Separator* (L-302678-S00-TECH-002). Retrieved from Subsea 7 S.A.

[84] Yang, C., & Mao, Z.-S. (2014). Chapter 2 - Fluid flow and mass transfer on particle scale. In C. Yang & Z.-S. Mao (Eds.), *Numerical Simulation of Multiphase Reactors with Continuous Liquid Phase* (pp. 5-73). Oxford: Academic Press.

[85] Yeoh, G. H., & Tu, J. (2010). *Computational Techniques for Multiphase Flows*. Oxford: Butterworth-Heinemann.

[86] Young, G. A. B., Wakley, W. D., Taggart, D. L., Andrews, S. L., & Worrell, J. R. (1994). Oil-water separation using hydrocyclones: An experimental search for optimum dimensions. *Journal of Petroleum Science and Engineering, 11*(1), 37-50. https://doi.org/10.1016/0920-4105(94)90061-2

Appendices

Appendix A Thesis description

Appendix B Additional explanation for mathematical term in section 3.2

Appendix C Physical properties of water and oil

Appendix D Geometry of de-oiler

Appendix E de Gelder pressure drop calculation

Appendix F Pressure drop in the de-oiler from various literature

Appendix G OpenFOAM files

Appendix H Tangential velocity calculation

Appendix A Thesis description

University of South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title:
Evaluation of de-oiler performance using numerical CFD technique.

USN supervisor:
Joachim Lundberg

External partner:
Subsea 7

Task background:
The benefits of subsea processing have been recognised for several decades. For much of this time, conventional technologies (i.e. topside processing facilities) have been adequate to achieve acceptable reservoir performance and project economics. With oil and gas exploration & production moving into deeper water and longer distance tiebacks the demands have never been greater to develop hydrocarbon fields using the most cost-effective production technology. Subsea processing technology is offering an economic benefit and improving production system operability, as well as lowered environmental impact.

The economic, technical, and environmental benefits of separating and disposing of produced water are apparent and widely accepted in the oil production industry world-wide.

Produced water handling is a challenge which affects the whole production process and an appropriate handling system is a key part of most production facilities. If discharge requirements are not met, there may be significant negative environmental and economic impact. Furthermore, produced water can take up valuable capacity in oil transport lines, reducing the amount of oil that can be produced.

This project will develop models for de-oiling of an incoming produced water stream using cyclone technology and provides a comparison of de-oiler performance for a range of flow rates, fluid compositions, and operating conditions.

Task description:

- Set the scope of the project.
- Literature review of previous work on modelling and design of de-oiling hydrocyclones (HC).
- Evaluation on standard conventional HC geometry vs inline HC vs Multi-cone HC
- CFD modelling of a de-oiling HC
- Establish and update design geometry model as progress is made
- Recommendations regarding optimal HC geometries for varying process conditions through the operating lifetime.

- Establish a calculation spreadsheet (Mathcad/Prime) for the de-oiling HC for recommended sizing and distribution of tubes within the pressure vessel.
- Evaluation of cleaning efficiency as function of ppm of oil in the produced water inlet to the HC.
- Describe the principles for the control system for operating the HC
- Evaluate subsea specific design issues and make recommendations
- Prepare report(s).

Student category:
PT students

Practical arrangements:
The thesis is intended one person in contact with the company.

Supervision:
As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:
Supervisor (date and signature):

10/ 1-2020
Joachim Lundberg

Student (date and signature):

09.01.2020
Yovita Hariyanti

Appendix Additional explanation for mathematical term in section 3.2

Appendix B.1 Effective viscosity

$$\nu_{\varphi}^{eff} = \nu_{\varphi} + C_t^2 \nu^t \tag{B.1}$$

where ν_{φ}^{eff} is the effective viscosity, ν^t is the turbulent kinematic viscosity defined in equation (B.2), and ν_{φ} is the kinematic viscosity for phase φ. While C_t is a turbulence response function i.e. the ratio of the root mean square of velocity fluctuations for dispersed and continuous, as shown in equation (B.3).

$$\nu^t = C_{\mu} \frac{k_{\varphi}^2}{\epsilon_{\varphi}} \tag{B.2}$$

$$C_t = \frac{U_1'}{U_2'} \tag{B.3}$$

where U_1' and U_2' are the r.m.s of the fluctuations in the velocity of dispersed and continuous phase, respectively. (Rusche, 2002, pp. 98-99)

Appendix B.2 Averaging procedure

$$\frac{\nabla \alpha_{\varphi}}{\langle \alpha_{\varphi} \rangle_{\nabla} + \delta} \tag{B.4}$$

δ denotes a small stabilising factor, $\langle \alpha \rangle_{\nabla}$ is the notation for averaging procedure i.e. average of α over the computational molecule of ∇ operator. (Rusche, 2002, p. 76, 79)

$$\nabla_f^{\perp} \phi = \boldsymbol{n} \cdot \nabla_f \phi = \frac{\phi_N - \phi_P}{|d|} \tag{B.5}$$

$\nabla_f^{\perp} \phi$ is the product of gradient $\nabla_f \phi$ normal to the face $\boldsymbol{n}$, where $\nabla_f \phi$ is a gradient at the face f, d is the vector between center of the cell P and N, see Figure 7.1. (Rusche, 2002, p. 76)

$$\nabla_f^{\perp} \alpha_{\varphi}$$

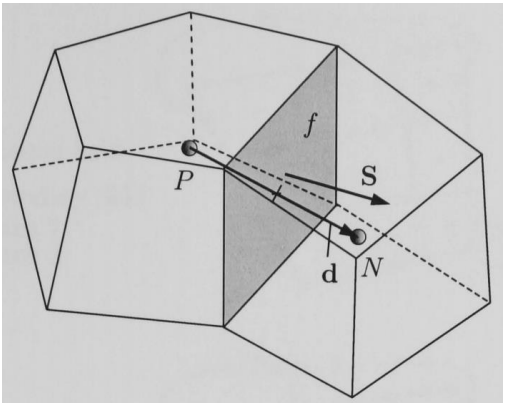


Figure 7.1: Parameters in finite volume discretization. (Rusche, 2002, p. 71)

Appendix B.3 Finite volume notation by Rusche (2002, p. 90)

Term	Term in finite volume notation	Discretized term
Time derivative	$\left[\!\left[\frac{\partial \rho[\phi]}{\partial t}\right]\!\right]$	$\frac{\rho_P^n \phi_P^n - \rho_P^o \phi_P^o}{\Delta t} V_P$
Convection term	$[\![\nabla \cdot (F[\phi]_{f(F,S)})]\!]$	$\sum_f F \phi_{f(F,S)}^n$
Diffusion term	$[\![\nabla \cdot (\Gamma \nabla[\phi])]\!]$	$\sum_f \Gamma_f S \cdot \nabla_f \phi^n$
Divergence term	$\nabla \cdot \phi$	$\sum_f S \cdot \phi_f^o$
Cell gradient term	$\nabla \phi$	$\sum_f S \, \phi_f^o$

Appendix C Physical properties of water and oil

Table 7.1: Physical properties of water and oil. [a](Xodus Group, 2020, p. 8) [b](Incropera, Dewitt, Bergman, & Lavine, 2013, p. 949) [c](Abulencia & Theodore, 2009, p. 555) [d](Engineering ToolBox, 2003) [e](Elam, Tokura, Saito, & Altenkirch, 1989, p. 1)

Component	Density [kg m^{-3}]	Dynamic viscosity [kg m^{-1} s^{-1}]	Molecular weight [kg $kmol^{-1}$]	Pr []	Interfacial tension[a] [N m^{-2}]	Heat capacity [J kg^{-1} K^{-1}]	Thermal conductivity [W m^{-1} K^{-1}]
Water[b] (1 atm, 300 K)	996	0.000855	18	5.828	43 x 10^{-3}	4179	0.613
Oil[c] (1 atm, 20°C)	856	0.0072	170	127.8		2130[d]	0.12[e] (273-323 K)

Appendix D Geometry of de-oiler

Table 7.2: Geometry of hydrocyclone. (Braga, 2015, p. 117)

Geometric variable	**Dimensions [mm]**
Length of inlet (A)	5
Height of inlet (B)	23
Length of vortex finder (VF)	48
Length of overflow tube (L1)	50
Length of cylindrical session (L2)	72
Length of conical session (L3)	810
Length of underflow tube (L4)	64
Length of the inlet tubes	40
Total length (L1+L2+L3+L4)	996

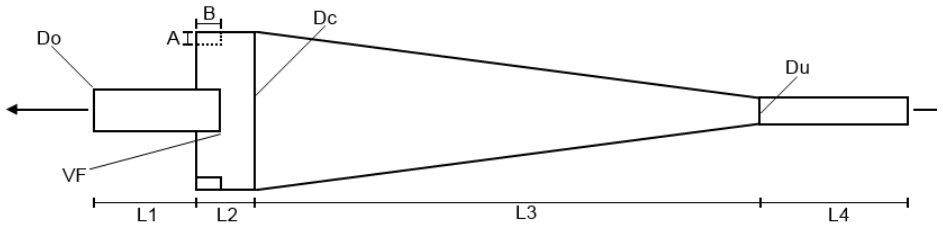


Table 7.3: Geometry of hydrocyclone. (Liu et al., 2012, p. 118)

Geometric variable	**Dimensions [m]**
Cyclone diameter D	0.05
Inlet diameter D_i	0.05
Overflow diameter D_o	0.05
Underflow diameter D_u	0.04
Total length of cyclone cylinder L	0.9
Height of the cylinder from top to the feed entrance H	0.1

Appendix E de Gelder pressure drop calculation

(Bradley, 1965, p. 92, 317) (Young et al., 1994, p. 46-47)

r	0.0381	0.0381
d	0.0762	0.0762
u_{mag} [m/s]	0.3	0.2
Q_{feed} [L/min]	82.0	54.7
Q_{feed} [m^3/s]	0.0014	0.0009
A_i	0.0003	0.0003
A_c	0.0046	0.0046
A_o	0.0003	0.0003
D_o	0.0191	0.0191
D_c	0.0762	0.0762
$\frac{A_i}{A_o}.\frac{D_o}{D_c}$	0.25	0.25
$\frac{D_o}{D_c}$	0.25	0.25
ξ_∞	0.1755	0.1755
J	12.2	12.2
ρ	982	982
θ [deg]	6	6
Re	19210	12807
ξ	-0.3891	-0.22582
u_{tang}	0.7	0.5
ΔP [Pa]	1348.505	2284.265
ΔP [bar]	0.0135	0.0228

Appendix F Pressure drop in the de-oiler from various literature

(Meldrum, 1988)

A 0.000961625 m^2

dP [kPa]	flow rate [BWPD]	dP [psi]	flow rate [L/min]	dP [bar]	u [m/s]
0.0031	1769.23	0.00045	196.58	0.000031	3.41
0.0246	3000.00	0.00357	333.33	0.000246	5.78
0.0415	3692.31	0.00602	410.26	0.000415	7.11
0.0585	4384.62	0.00848	487.18	0.000585	8.44
0.0900	5307.69	0.01305	589.74	0.000900	10.22
0.1246	6269.23	0.01807	696.58	0.001246	12.07
0.1862	7846.15	0.02700	871.79	0.001862	15.11

(Angelim et al., 2017, p. 154, 161)

L [m] 0.5508

D out [m] 0.0157

flow rate [m/s]	dP [Pa]	flow rate [L/min]	Pin [Pa]	P overflow [pa]	dP [bar]
4	7466	51.29712	108168	100702	0.07466
15	66519	192.3642	164005	97486	0.66519
32	311983	410.377	379702	67719	3.11983

(Kharoua, Khezzar, & Nemouchi, 2010, p. 121, 122)

Din [m] 0.0105

Ain [m^2] 8.65463E-05

u [m/s]	flow rate [m3/s]	flow rate [L/min]	dP [bar]
11.555	0.001	60	1.0546

(Husveg et al., 2007, p. 295)

geometry is not available

Flow rate [m3/h]	dP [bar]	flow rate [L/min]
3	0.75	50.00
4	1.5	66.67
5	2.375	83.33
6.6	4	109.91

8	6	133.33
9	7.5	150.00

Run 1 and Run 2

A [m2] 0.004558

	u [m/s]	dP [bar]	flow rate [L/min]
Run 1	0.2	0.02272	54.69
Run 2	0.3	0.0424	82.045

Subsea 7

A [m2] 0.000282

u [m/s]	dP [bar]	flow rate [L/min]
2.850	0.7	48.33

Appendix G OpenFOAM files

Run 1

0 folder

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions          [1 -1 -2 0 0 0 0];

internalField       uniform 1e5;

boundaryField
{
    inletCenter
    {
        type               calculated;
        value              $internalField;
    }
    inletAnular
    {
        type               calculated;
        value              $internalField;
    }
    outlet
    {
        type               calculated;
        value              $internalField;
    }
    walls
    {
        type               calculated;
        value              $internalField;
    }
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      binary;
    class       volVectorField;
    object      U.oil;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inletAnular
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.2;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.5;

        value              uniform (0 0 0);
    }

    inletCenter
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.2;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.5;

        value              uniform (0 0 0);
    }

    outlet
    {
```

```
        type            pressureInletOutletVelocity;
        phi             phi.oil;
        value           $internalField;
    }
    walls
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
}

// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      binary;
    class       volVectorField;
    object      U.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inletAnular
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.2;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.5;

        value              uniform (0 0 0);
    }

    inletCenter
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.2;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.5;

        value              uniform (0 0 0);
    }
    outlet
    {
        type               pressureInletOutletVelocity;
```

```
        phi                 phi.water;
        value               $internalField;
    }
    walls
    {
        type                fixedValue;
        value               uniform (0 0 0);
    }
}

// ************************************************************************* //
```

The remaining *0* folder, *constant* folder and *system* folder are similar with Run 2.

Run 2

0 folder

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      alpha.oil;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inletCenter
    {
        type            fixedValue;
        value           uniform 0.1;
    }
    inletAnular
    {
        type            fixedValue;
        value           uniform 0.1;
    }
    outlet
    {
        type            inletOutlet;
        phi             phi.oil;
        inletValue      uniform 0;
        value           uniform 0;
    }
    walls
    {
        type            zeroGradient;
    }
}

// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T.oil;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions          [0 0 0 1 0 0 0];

internalField       uniform 300;

boundaryField
{
    walls
    {
        type               zeroGradient;
    }
    outlet
    {
        type               inletOutlet;
        phi                phi.oil;
        inletValue         $internalField;
        value              $internalField;
    }
    inletCenter
    {
        type               fixedValue;
        value              $internalField;
    }
    inletAnular
    {
        type               fixedValue;
        value              $internalField;
    }
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions          [0 0 0 1 0 0 0];

internalField       uniform 300;

boundaryField
{
    walls
    {
        type               zeroGradient;
    }
    outlet
    {
        type               inletOutlet;
        phi                phi.water;
        inletValue         $internalField;
        value              $internalField;
    }
    inletCenter
    {
        type               fixedValue;
        value              $internalField;
    }
    inletAnular
    {
        type               fixedValue;
        value              $internalField;
    }
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      binary;
    class       volVectorField;
    object      U.oil;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inletAnular
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.3;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.7;

        value              uniform (0 0 0);
    }

    inletCenter
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.3;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.7;

        value              uniform (0 0 0);
    }

    outlet
    {
```

```
        type            pressureInletOutletVelocity;
        phi             phi.oil;
        value           $internalField;
    }
    walls
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
}

// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      binary;
    class       volVectorField;
    object      U.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inletAnular
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.3;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.7;

        value              uniform (0 0 0);
    }

    inletCenter
    {
        type               swirlInletVelocity;
        axis               (0 0 1);
        origin             (0 0 0);
        axialVelocity      constant 0.3;
        radialVelocity     constant 0;
        tangentialVelocity constant 0.7;

        value              uniform (0 0 0);
    }
    outlet
    {
        type               pressureInletOutletVelocity;
```

```
        phi                 phi.water;
        value               $internalField;
    }
    walls
    {
        type                fixedValue;
        value               uniform (0 0 0);
    }
}

// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions          [1 -1 -2 0 0 0 0];

internalField       uniform 1e5;

boundaryField
{
    inletCenter
    {
        type               calculated;
        value              $internalField;
    }
    inletAnular
    {
        type               calculated;
        value              $internalField;
    }
    outlet
    {
        type               calculated;
        value              $internalField;
    }
    walls
    {
        type               calculated;
        value              $internalField;
    }
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p_rgh;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions          [1 -1 -2 0 0 0 0];

internalField       uniform 1035496;//150psi

boundaryField
{
    inletCenter
    {
        type            fixedFluxPressure;
        value           $internalField;
    }
    inletAnular
    {
        type            fixedFluxPressure;
        value           $internalField;
    }
    outlet
    {
        type            prghPressure;
        p               $internalField;
        value           $internalField;
    }
    walls
    {
        type            fixedFluxPressure;
        value           $internalField;
    }
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

constant folder

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       uniformDimensionedVectorField;
    location    "constant";
    object      g;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -2 0 0 0 0];
value           (0 -9.81 0);


// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      phaseProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

phases (oil water);

oil
{
    diameterModel   constant;
    constantCoeffs
    {
        d           6e-5;
    }
    residualAlpha   1e-6;
}

water
{
    diameterModel   constant;
    constantCoeffs
    {
        d               1e-4;
    }

    residualAlpha       1e-6;
}

blending//to mix the effect of three models
{
    default
    {
        type            none;//options: linear, hyperbolic, none
        continuousPhase water;
    }
}
```

```
sigma
(
    (oil and water)     0.0253//Xodus pipe separator report
);

aspectRatio
(
    (oil in water)
    {
        type            constant;
        E0              1.0;//eotvos number
    }

);

drag
(
    (oil in water)
    {
        type            SchillerNaumann;
        residualRe      1e-3;
        swarmCorrection
        {
            type        none;
        }
    }
);

virtualMass//neglected
(
);

heatTransfer
(
    (oil in water)
    {
        type            RanzMarshall;
        residualAlpha   1e-4;
    }
);

phaseTransfer
(
);

lift//neglected
(
```

```
);

wallLubrication
(
);

turbulentDispersion
(
);

// Minimum allowable pressure
pMin             10000;


// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermophysicalProperties.oil;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
thermoType
{
    type            heRhoThermo;
    mixture         pureMixture;
    transport       const;
    thermo          hConst;
    equationOfState rhoConst;
    specie          specie;
    energy          sensibleInternalEnergy;
}
mixture
{
    specie
    {
        molWeight   170;
    }
    equationOfState
    {
        rho         856;
    }
    thermodynamics
    {
        Cp          2130;
        Hf          0;
    }
    transport
    {
        mu          0.0072;
        Pr          127.8;
    }
}
// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermophysicalProperties.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

thermoType
{
    type            heRhoThermo;
    mixture         pureMixture;
    transport       const;
    thermo          hConst;
    equationOfState rhoConst;
    specie          specie;
    energy          sensibleInternalEnergy;
}

mixture
{
    specie
    {
        molWeight   18;
    }

    equationOfState
    {
        rho        996;
    }

    thermodynamics
    {
        Cp          4179;
        Hf          0;
    }
    transport
    {
        mu          0.000855;
```

```
        Pr          5.828;
    }
}
// ************************************************************************* //



/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties.oil;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType  laminar;

// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties.water;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType  laminar;

// ************************************************************************* //
```

system folder

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     twoPhaseEulerFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         100;

deltaT          0.0005;

writeControl    adjustableRunTime;

writeInterval   0.05;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

adjustTimeStep  yes;
```

```
maxCo           2;//

maxAlphaCo      1;

maxDeltaT       1;

functions
{
    fieldAverage1
    {
        type            fieldAverage;
        libs            ("libfieldFunctionObjects.so");
        writeControl    writeTime;
        fields
        (
            U.oil
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }

            U.water
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }

            alpha.oil
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }

            p
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
        );
    }
}


// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      decomposeParDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

numberOfSubdomains 4;

method          simple;

simpleCoeffs
{
    n               (1 2 2);
    delta           0.001;
}

// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;//
}

divSchemes
{
    default                         none;

    div(phi,alpha.oil)              Gauss vanLeer;
    div(phir,alpha.oil)             Gauss vanLeer;

    "div\(alphaRhoPhi.*,U.*\)"      Gauss limitedLinearV 1;//
    "div\(phi.*,U.*\)"              Gauss limitedLinearV 1;//

    "div\(alphaRhoPhi.*,(h|e).*\)"  Gauss limitedLinear 1;
    "div\(alphaRhoPhi.*,K.*\)"      Gauss limitedLinear 1;
    "div\(alphaPhi.*,p\)"           Gauss limitedLinear 1;

    "div\(\(\(\(alpha.*\*thermo:rho.*\)*nuEff.*\)\*dev2\(T\(grad\(U.*\)\)\)\)\)" G
auss linear;
}

laplacianSchemes
{
```

```
    default         Gauss linear uncorrected;//try replace 'uncorrected' with 'lim
ited 0.777'
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         uncorrected;//try replace 'uncorrected' with 'limited 0.777'//
uncorrected
}


// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  7
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    alpha.oil
    {
        nAlphaCorr      1;
        nAlphaSubCycles 2;
    }

    p_rgh
    {
        solver          GAMG;
        smoother        DIC;
        tolerance       1e-4;
        relTol          0;
    }

    p_rghFinal
    {
        $p_rgh;
        relTol          0;
    }

    "U.*"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-5;
        relTol          0;
        minIter         1;
    }
```

```
    "e.*"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1;//1e-8
        relTol          0;
        minIter         0;//1
        maxIter         0;
    }
}

PIMPLE
{
    nNonOrthogonalCorrectors 1;//0 //correct press field more often
    nCorrectors      1;// if > 1 use piso
    nOuterCorrectors 50;//50-100
    residualControl
    {
      U     1e-5;//tolerance
      p     5e-4;//tolerance
    }
}

relaxationFactors
{
      fields
      {
            p         0.2;//0.3
            pFinal    1;
      }
      equations
      {
            "U.*"      0.3;
            "U.*Final" 1;
       }
}


// ************************************************************************* //
```

Appendix H Tangential velocity calculation

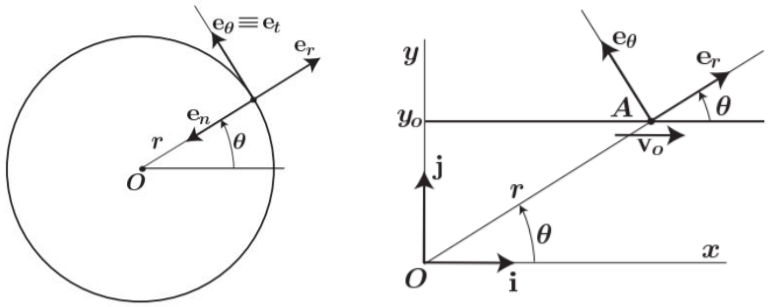
The Formula is given below:

$$r = (x^2 + y^2)^{0.5}$$

$$\theta = \tan^{-1}(y/x)$$

$$u_{tang} = -u\,sin\theta + v\cos\theta$$

Where r is radius, u is velocity in x-direction, v is velocity in y-direction, the other symbol can be seen in the figure below:



(Widnall & Peraire, 2008, p. 3-6)

u	v	w	x	y	y/x	tan-1(y/x)	u_tang
0.03409	-0.00154	0.085365	-0.0005	-0.0125	25	1.530818	-0.03412
0.061995	-0.00281	0.155241	-0.0005	-0.01248	25	1.530818	-0.06206
0.089899	-0.00407	0.225115	-0.0005	-0.01245	25	1.530818	-0.08999
0.117804	-0.00533	0.294991	-0.0005	-0.01243	25	1.530818	-0.11792
0.145708	-0.00659	0.364865	-0.0005	-0.0124	25	1.530818	-0.14586
0.173613	-0.00786	0.434741	-0.0005	-0.01238	25	1.530818	-0.17379
0.201517	-0.00912	0.504615	-0.00049	-0.01235	25	1.530818	-0.20172
0.229422	-0.01038	0.574492	-0.00049	-0.01233	25	1.530818	-0.22965
0.257327	-0.01164	0.644368	-0.00049	-0.0123	25	1.530818	-0.25759