

Master thesis
TVVR 13/5010

Numerical simulation of three-dimensional flows in water storage tanks

Application to Norra Ugglarps Reservoirs in South Sweden

Julien Jaunâtre



**Division of Water Resources Engineering
Department of Building and Environmental Technology
Lund University**

Avd för Teknisk Vattenresurslära

Div. of Water Resources Engineering

TVVR-13/5010

ISSN 1101-9824

Numerical simulation of three-dimensional flows in water storage tanks

Application to Norra Ugglarps Reservoirs in South Sweden



LUNDS UNIVERSITET
Lunds Tekniska Högskola



Julien Jaunâtre

Cover: Lake Vombsjön, 2012-05-07. Photo by J. Jaunâtre

Copyright© Julien Jaunâtre, 2013

Report no: TVVR 13/5010

All photographs are taken by the author unless anything else is stated

Division of Water Resources Engineering
Department of Building and Environmental Technology
Faculty of Engineering, Lund University, Sweden

Together with

Sydvatten AB
Hyllie, Malmö

Exchange student from

Hydraulics, Civil & Environmental Engineering: Master program
Grenoble INP-Ense3, France

ABSTRACT

The main purpose of this thesis was to simulate the three-dimensional flow patterns in tanks for water storage. The practical intentions of the simulations were to identify water-quality problems related to insufficient water exchange. The water storage reservoirs at Norra Ugglarp in South Sweden were investigated as a special case study. In this thesis, the open source CFD software OpenFoam was chosen to simulate flow features in tanks.

The first part of this thesis focused on developing an understanding of the case study; that is, the distribution of water in West Skåne and especially the functioning of Norra Ugglarps reservoirs. An overview of possible mixing problems in the tank was detailed and presented.

As modelling was chosen to identify mixing problems, mechanisms and equations about flow modelling were first discussed before going into the OpenFOAM software itself. The modelling strategy in OpenFoam do not differs from other CFD software; nevertheless, setting up the case and post processing the results were quite unusual and discussed in some detail.

When it came to the modelling itself, in order to become familiar with the software, some simple test setups were first explored in two-dimensional tanks of simple geometry. Then, the simulations were extended to three-dimensional situations, still for rectangular tanks. Finally, the software was employed to simulate the flow in the reservoirs at Norra Ugglarp.

From the modelling results, it was possible to derive the following conclusions. The incoming velocity is a key parameter as injection of water should be powerful enough to reach the surface and renew the upper layer water. Water level range is another important parameter to renew more volume in the tanks. All conclusions tend to prove that the reservoirs at Norra Ugglarp suffer from insufficient mixing, mainly due to both the weak incoming velocity of new water and the limited water level range in use.

Thus, solutions to alleviate poor mixing are presented, such as increasing the water level range or monitoring existing valves to ameliorate the filling regarding water circulation.

Keywords: drinking water storage tank, rectangular tank, Computational Fluid Dynamics (CFD), numerical modelling, 2D, 3D flow, cluster, Lunarc.

SAMMANFATTNING

Det huvudsakliga syftet med detta arbete var att simulera tredimensionella flödesmönster i tankar för lagring av vatten. De praktiska avsikter med simuleringarna var att identifiera vattenkvalitetsproblem relaterade till bristande vattenutbyte. De vattenmagasin som reservoarer vid Norra Ugglarp i Skåne utgör undersöktes i en särskild fallstudie. I denna avhandling har den öppna CFD källkoden OpenFOAM valts för att simulera cirkulationen i tankarna.

Den första delen av rapporten fokuserar på en detaljerad förståelse av fallstudien, som omfattar fördelningen av vatten i västra Skåne och särskilt funktionen hos Norra Ugglarps reservoarer. En översikt över möjliga omblandningsproblem i reservoarerna studerades i detalj och sammanfattades.

Modellering valdes för att identifiera omblandningsproblem samt grundläggande mekanismer och ekvationer för flödesmodelleringen diskuteras först innan man går in på själva programvaran OpenFOAM. Modelleringsstrategin i OpenFOAM skiljer sig inte nämnvärt från annan CFD-programvara, även om upprättandet av det studerade fallet och efterbearbetningen av resultaten var ganska ovanligt, vilket speciellt diskuteras i rapporten.

Vad gäller själva modelleringen, för att bli bekant med programmet, har några enkla testuppställningar först utforskats omfattande två-dimensionella tankar av enkel geometri. Därefter utsträcktes simuleringarna till tre-dimensionella situationer för rektangulära tankar. Slutligen användes programvaran för att simulera flödet i Norra Ugglarps reservoarer.

Från modelleringsresultaten var det möjligt att erhålla följande slutsatser. Hastigheten hos den inkommande strålen är en viktig parameter och den bör vara tillräckligt hög så att strålen kan nå ytan och förnya övre vattenskikt. Variationen i vattennivå är en annan viktig parameter för att förnya volymen i tankarna. Resultaten från föreliggande beräkningar visar att Norra Ugglarps reservoarer lider av dålig omblandning, främst på grund av den låga hastigheten hos den inkommande strålen av nytt vatten och den begränsade variationen i vattennivå.

Lösningar för att förbättra omblandningen presenteras, till exempel att öka variationen i vattennivå och att övervaka befintliga ventiler för att förbättra fyllningsprocessen med avseende på vattencirkulationen.

Nyckelord: dricksvatten ackumulatortank, rektangulär tank, Computational Fluid Dynamics (CFD), numerisk modellering, 2D, 3D flöde, kluster, Lunarc

RÉSUMÉ DÉTAILLÉ

Un stage de dernière année de 5 mois clôture la formation ingénieure suivie à Grenoble INP – Ense3. Ce « projet de fin d'études » a été réalisé sous la forme d'une thèse de master, à Faculté d'Ingénierie à l'université de Lund (Lunds Tekniska Högskola, Lunds Universitet) au sein du département *Ingénierie des ressources en eaux* (Teknisk Vattenresurslära), en partenariat avec l'entreprise Sydvatten. Après une formation dans la filière Hydraulique, Ouvrages et Environnement (HOE) à Grenoble INP-Ense3, dont le dernier semestre d'études en échange à Lunds Tekniska Högskola, et déjà deux stages dans les domaines de la gestion de l'eau et de sa distribution ; c'est tout naturellement que ce projet a été choisi, dans la continuité d'un projet professionnel mûrissant.

L'objectif général de cette thèse est de simuler des modèles d'écoulement en trois dimensions dans des réservoirs de stockage d'eau potable. Le but final de ces simulations est d'identifier d'éventuels problèmes de qualité d'eau liés à des échanges insuffisants. Les réservoirs de stockage d'eau potable situés à Norra Ugglarp dans le sud de la Suède ont été l'objet d'une étude de cas (Norra Ugglarp, Genarp, Lunds kommun, Skåne).

Dans cette thèse, le logiciel open source CFD OpenFoam a été choisi pour simuler les caractéristiques d'écoulement dans les réservoirs (CFD : Computational Fluid Dynamics, ou encore en français Mécanique des Fluides Numériques MFN). OpenFoam est un code de calcul qui devient actuellement répandu. Ce code se manipule avec un macro langage de programmation de haut niveau. Aussi, comme tout logiciel open source, ce programme et toute mise à jour sont gratuits, ce qui était le dernier argument pour préférer OpenFoam à d'autres concurrents (OpenCFD Ltd (ESI Group), 2013. *OpenFoam: The open source CFD toolbox*).

Pour mener cette étude, un premier examen de la bibliographie existante a été mené, en particulier des études sur les problèmes de qualité de l'eau dans les réservoirs de stockage pour l'approvisionnement en eau potable et sur la simulation numérique d'écoulement dans ces réservoirs. Des études de cas intéressantes ont été examinées et résumées brièvement dans ce mémoire. Certaines études ont montré notamment que la température jouait un rôle important dans le phénomène de mélange de réservoir. Une stratification peut survenir dans les réservoirs, ce qui empêche les échanges d'eau entre les différentes couches de température. De plus, une différence de température entre

l'eau entrant dans le réservoir et celle déjà présente un mouvement de densité. Si l'arrivée d'eau est plus froide - soit plus dense – que l'eau déjà présente dans le réservoir, cette eau froide aurait tendance à rester dans le fond du réservoir. L'effet inverse se produirait pour une eau entrante de moins grande densité.

Le cas d'étude concerne des réservoirs d'eau potable de l'entreprise Sydvatten. Sydvatten est une entreprise appartenant aux collectivités locales qui produit de l'eau potable pour plus de 800 000 habitants dans l'ouest de la région Skåne (Cf. Figure II-1). L'eau potable produite par l'entreprise provient du lac Bolmen dans la région Småland et du lac Vombsjön de la région Skåne (Cf. Figure II-2). Si nécessaire, il est également possible de prendre de l'eau du lac Ringsjön dans la région Skåne. Un tunnel d'eau brute de 80 km de long apporte gravitairement l'eau du lac Bolmen à la station d'eau potable de Ringverket, près du lac Ringsjön. Une autre station d'eau potable – Vombverket – est présente près du lac Vombsjön. L'eau traitée dans ces deux stations est ensuite transportée vers des réservoirs de stockage d'eau potable, avant d'être distribuée aux municipalités. Le cas à l'étude se penche sur cette seconde ressource, et plus particulièrement les réservoirs d'eau potable de Norra Ugglarp situé en aval de la station de production d'eau potable de Vombverket.

Actuellement, ces deux réservoirs ne fonctionnent pas comme réservoirs de stockage à proprement parlé, ce qui signifie que l'eau serait stocké dans les réservoirs d'abord, puis distribuée aux consommateurs. Ces réservoirs sont auxiliaires à la station de traitement de Vombverket. Les pompes à la station de Vombverket sont des pompes à débit variable. Vombverket fournit de l'eau directement aux municipalités, parmi lesquelles par exemple Malmö (300.000 habitants). La consommation varie toute la journée. Mais l'usine n'adapte pas le débit à la consommation exacte. Pour un débit donné par la station, si la consommation y est supérieure, les réservoirs de Norra Ugglarp fournissent le complément. Si l'eau descend dans les réservoirs et atteint une consigne basse, les pompes à Vombverket démarrent/ augmentent vers la gamme de débit suivante. Ceci jusqu'à ce que le niveau d'eau dans le réservoir atteigne une valeur de consigne élevée déterminée. Les réservoirs sont alors considérés pleins. Alors, le débit provenant de l'usine est trop élevée, l'usine de production d'eau réduit le débit des pompes. Si au contraire, dans le cas de la diminution de la consommation, pour le même débit donné, les réservoirs se remplissent avec l'eau en excès. Cela se fait jusqu'à ce que le niveau d'eau dans le réservoir atteigne un niveau élevé.

Le renouvellement de l'eau est l'un des facteurs clés pour décrire le mélange dans des réservoirs. Pourquoi pense-t-on que l'eau n'est pas suffisamment renouvelée? Le problème est que la capacité réelle des réservoirs est juste suffisante pour assurer une réserve de stockage pour une période relativement courte. Par mesure de précaution, les réservoirs sont maintenus à un niveau élevé, suivant un faible marnage (cf. Figure II-7). Ainsi, seulement un petit volume, représenté au dessus de la ligne en pointillé dans la Figure II-7, est renouvelé pour un remplissage ou une vidange donné.

En outre, ces réservoirs ne possèdent qu'une seule conduite dans le fond pour le refoulement et la distribution d'eau. Le plus grand réservoir, d'un volume maximal de 18300m³ possède une seule conduite refoulement – distribution à proximité du centre du réservoir cylindrique, dirigée vers le haut et perpendiculaire au fond du réservoir. Le petit réservoir, d'une capacité maximale de 3000 m³ possède une seule conduite de refoulement – distribution, mais avec deux issues : une perpendiculaire au fond dirigé vers le haut près de la paroi, et une autre au refoulement tangentiel au diamètre intérieur du réservoir (cf. figure II-8). Mais les deux réservoirs ne possèdent qu'une seule conduite d'adduction en eau potable dans le fond.

Grâce à ces deux conditions susmentionnées, l'entreprise Sydvatten est préoccupée par le mélange dans ces réservoirs, ce qui impacte sur la qualité de l'eau. Pour confirmer ou réfuter l'idée d'une mauvaise circulation de l'eau dans ces réservoirs, Sydvatten est curieuse de comprendre le modèle d'écoulement et les profils de vitesse dans ses réservoirs. Cette étude a été au cœur de cette mission. Le logiciel CFD OpenFoam a été utilisé pour simuler des modèles d'écoulement dans les réservoirs. Pour ce cas d'étude, le solveur *interMixingFoam* a été choisi. Il s'agit d'un "solveur pour 3 fluides incompressibles, dont deux sont miscibles, en utilisant une méthode VOF (Volume of Fluid) pour capturer l'interface" (OpenCFD, 2013, *section Standard solvers*).

Pour les modélisations, 3 phases ont donc été évaluées : une phase air, une phase représentant l'eau existant dans le réservoir, considérée comme âgée, et l'eau entrante a été représentée par une dernière phase. Ces deux phases sont miscibles, l'interface avec l'air suivi grâce à la méthode VOF. Le lecteur peut se demander pourquoi les équations pour l'eau et l'air sont toutes deux résolues et non seulement la diffusion d'un traceur représentant l'eau fraîche dans l'eau existante des réservoirs. Le problème est que ces cas étudient le remplissage – ou vidange – des réservoirs, ce qui signifie que le niveau d'eau va augmenter au cours

du calcul. Dans le cas de la résolution des équations de l'eau seulement, un maillage en mouvement serait nécessaire pour suivre l'évolution du niveau d'eau dans le réservoir. Outre le fait que la mise en œuvre d'un maillage mobile serait plus compliqué, utiliser le solveur *interMixingFoam* a permis de prouver que l'eau reste toujours à la surface et que l'air n'est pas entraîné dans l'eau par des tourbillons et des remous.

Une partie théorique sur les équations régissant l'écoulement dans les réservoirs a été développée au début de cette étude. La stratégie de modélisation d'OpenFoam ne diffère pas des autres logiciels de CFD, néanmoins la mise en place de cas d'étude et la visualisation des résultats sont plutôt inhabituel et ont fait l'objet également d'une section approfondie.

Pour se familiariser avec le logiciel, certaines configurations de tests simples ont d'abord été abordées, tout d'abord des réservoirs en deux dimensions à géométrie simple. Ensuite, les simulations ont été étendues aux situations en trois dimensions, cependant, toujours pour des réservoirs de formes rectangulaires. Enfin, le logiciel a été utilisé pour simuler l'écoulement dans les réservoirs de Norra Ugglarp dans le but de vérifier si les problèmes actuels de qualité de l'eau sont liés à la configuration de l'écoulement dans le réservoir.

Bien que le cas du réservoir rectangulaire en 3D suivait celui en 2D, poursuivre vers le réservoir réel a été difficile. Obtenir un maillage approprié a été d'une certaine difficulté puisqu'il a fallu passer par l'utilisation d'un logiciel tiers pour la création du design (Salome est un logiciel open source pour la création de structure 3D – modèles CAO); et ensuite l'utilisation d'un outil inhérent à OpenFoam pour la création du maillage. Un maillage satisfaisant a été obtenu pour un peu moins de cellules 200 000. Il s'est vite avéré nécessaire de recourir à plus de puissance pour avoir de certains résultats. L'accès tardif à un cluster a permis d'augmenter la rapidité des calculs, même si plus de temps serait nécessaire pour obtenir des résultats de scénarios complets. Cependant, l'ensemble des scénarios étudiés liste les conclusions suivantes.

Le débit d'entrée dans le réservoir est un paramètre clé en ce qui concerne le mélange. En effet, l'amplitude de la vitesse du jet lors du remplissage doit être suffisamment forte pour être capable de propulser l'eau à travers la colonne d'eau de la cuve. Le marnage est un autre paramètre important à considérer. Pour des raisons de sécurité, les gestionnaires d'eau potable utilisent un faible marnage, sur une position élevée, ce qui induit de faible renouvellement d'eau au sein du

réservoir. L'utilisation d'un marnage limité, à 'haute altitude' dans un réservoir n'est pas bénéfique pour le renouvellement ou circulation d'eau puisque seulement un faible volume de la cuve est renouvelé. Un compromis doit être trouvé entre la sécurité et une bonne mixité de l'eau.

Toutes les observations des modélisations s'accordent à dire que les réservoirs de Norra Ugglarp souffrent de mauvais mélange, tant par la faible vitesse d'arrivée d'eau dans les réservoirs que par le minuscule marnage utilisé aujourd'hui.

Pour remédier à ces problèmes de mélange, quelques solutions ont été proposées. L'ajout d'une conduite montante pour le refoulement et l'alimentation du réservoir par la partie supérieure semble une solution pertinente pour améliorer grandement le mélange dans le réservoir. L'eau sera alors forcée d'entrer dans le haut du réservoir, et sera drainée vers l'actuelle conduite de distribution située à la base du réservoir, créant ainsi une circulation générale naturelle. Cependant, cette solution peut être difficile à mettre en place techniquement. L'augmentation du marnage serait une solution simple pour améliorer le mélange, mais préjudiciable pour le volume de stockage disponible en cas de défaillance. L'asservissement des vannes d'accès aux réservoirs permettrait de nombreuses possibilités. Par exemple, les réservoirs peuvent être isolés l'un de l'autre, le débit provenant de Vomverket dirigés vers un seul et même réservoir. Un débit d'arrivé élevé dans un réservoir entrainerait un plus grand mélange.

Au moment où ce rapport a été réalisé et cette thèse sur le point d'être achevée, il pourrait être demandé d'effectuer de plus amples études sur les réservoirs dans Norra Ugglarps. En espérant que ce rapport soit assez clair pour fournir des informations utiles à quiconque aurait besoin de poursuivre ce travail, l'auteur de cette thèse sera volontiers contacté pour plus de détails. Néanmoins, un dernier chapitre fut consacré aux prochaines études susceptibles d'être mises en œuvre.

NOTATIONS

The following symbols are used in this thesis:

e.g.	<i>exempli gratia</i> = for instance (Latin)
CFD	Computational Fluid Dynamics
cf.	<i>confer</i> (Latin)
DN	Diameter nominal (expressed in mm)
VOF	Volume of Fluid
U	Velocity
g	Acceleration due to gravity
p	Pressure
t	Time
∂	Partial derivative
PDE	Partial differential equation
CAD	Computer-aided design
i.e.	<i>id est</i> = that is (Latin)
1/2/3D	One/two/three dimensional (in space)

Greek letters

α_i	Fraction of fluid i
μ	Dynamic viscosity
ρ	Density of fluid
σ	Standard deviation

ACKNOWLEDGEMENTS

Lund, June 2013

Pleased to have undertaken this master thesis for five months, performed at the Division of Water Resources Engineering, Department of Building and Environmental Technology, Lund University in Sweden, together with the company Sydvatten; I would like to give special attention to the hereafter people:

To my supervisor, Magnus Larson, Professor from the Water Resources Engineering Division, for his guidance and numerous advices, who encouraged me with his undying optimism throughout this project;

To Sydvatten, for all of you who welcome me both in the main office in Malmö Hyllie and on-site in Vombverket or Norra Ugglarps reservoirs; And particularly Fedja Rustempasic, Development Engineer at Sydvatten, for his friendly and helpful supervision, his constant interest as well as willingness for this thesis;

To Kenneth M. Persson, who introduced me to Sydvatten and without whom I would not have carried out this interesting venture;

To Johan Nilsson, Division of Structural Mechanics - Department of Building Sciences and Robert-Zoltán Szász, Division Fluid Mechanics - Department of Energy Sciences for their help dealing with Computational Fluid Dynamics and the software OpenFOAM;

To Marie Brink, for her devotion to her work as international exchange coordinator and saving help;

To my family in France for their great support and encouragement from a distance;

To my friends, here and there, for your smiles and shared moments together, making my life joyful;

Thanked be all of you.

Julien Jaunâtre

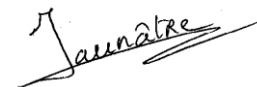


TABLE OF CONTENTS

ABSTRACT	i
SAMMANFATTNING	ii
RÉSUMÉ DÉTAILLÉ	iii
NOTATIONS	viii
ACKNOWLEDGEMENTS	ix
I. INTRODUCTION	1
I.1 Background.....	1
I.2 Objectives.....	2
I.3 Procedure.....	2
II. CASE STUDIES	4
II.1 Distribution of water in Skåne.....	4
II.2 Norra Ugglarps Reservoirs.....	6
II.3 Water renewal.....	8
II.4 Scenarios of studies.....	9
III. SIMULATION OF WATER FLOWS IN STORAGE TANKS	13
III.1 Basic mechanisms.....	13
III.2 Governing equations	13
III.3 Numerical solution methods	17
III.4 Example of previous studies.....	17
IV. OpenFOAM.....	19
IV.1 Modelling strategy.....	19
IV.2 Software structure.....	20
IV.3 Solution schemes.....	21
IV.4 Boundary and initial conditions.....	22
IV.5 Output handling and analysis.....	23
IV.6 Using the Cluster	24
V. MODELLING TWO-DIMENSIONAL FLOW IN A STORAGE TANK	26

V.1	Test setup	26
V.2	Simulation cases	27
V.3	Results	28
VI.	MODELLING THREE-DIMENSIONAL FLOW IN A STORAGE TANK.....	34
VI.1	Test setup	34
VI.2	Simulation cases	34
VI.3	Results	35
VII.	MODELLING THE FLOW IN NORRA UGGLARPS' RESERVOIRS.....	41
VII.1	First reservoir.....	41
VII.2	Second reservoir.....	41
VII.3	Simulation cases	43
VII.4	Results	45
VIII.	Limit of the study.....	48
IX.	Further studies	50
X.	SOLUTIONS TO PROVIDE MORE MIXING.....	52
X.1	Implementing an upper pipe.....	52
X.2	Increasing tidal range	53
X.3	Regulating the reservoirs	54
	CONCLUSION	56
	REFERENCES	58
	Appendix I – Calculation for defining scenario of study.....	I
	Appendix II- Code for the tank case (Norra Ugglarps)	III
	Appendix III- Submitting script for the cluster	XII

I. INTRODUCTION

I.1 Background

Water storage tanks have a major role when it comes to water consumption networks. Their main function is to provide storage of drinking water for further consumption. They are often situated at a high altitude, or/and designed as an elevated tower to provide a certain total head in the water supply network. They permit a balance between production of drinking water from the plant and continuous consumption by municipality consumers and industries. Given their current filling, they ensure reserve storage in case of network breakdown, problems in production plants or even for fire fighting.

Because of the above requirements, water systems try to keep water levels in storage tanks high. Filling in and draining out follow a rather small water level range located high in the reservoir. This might lead to a poor mixing in the tank. The presence of a single inlet/outlet, the design itself of the tank (large height compared to the diameter) or other parameters can also reduce the water mixing. Small variations in water level imply that the water stays longer in the network before reaching the customers and the problem of stagnant water appears. A weak mixing leads to inert water or stratified zones, which can induce declining disinfectant residuals (Lemke and DeBoer, 2012).

The awareness of possible unwanted impacts on drinking water has prompted a better understanding of the hydraulics in storage tanks. Computational fluid dynamics (CFD) has been used to solve and analyse many problems that involve fluid flows since the 1960s. The use of CFD to study hydraulics in water storage tanks has been limited in the past, although actual CFD softwares have the potential to describe main patterns of flow in such tanks (Nordblom, 2004).

Physical measurements of mixing parameters, for instance measuring three-dimensional flows, are arduous work, raising many questions. How difficult would it be to have an accurate estimate of the flow over an entire tank, over depth and length, with flow meters? Would the introduction of an instrument modify the flow itself while measuring it? A scale physical model, or even prototype are sometimes used to reproduce as closely as possible the behaviour of studied cases. Those physical reproductions require, however, experts skills and investments (Encyclopaedia Wikipedia, *Maquette en hydraulique*). In the following case studied,

the presence of one single inlet/outlet, as well as the lack of accurate measurements of flow at the entrance of the reservoirs, makes it difficult to use a simple water balance to understand the water exchange. The exploitation of a numerical CFD model disregards the aforementioned issues and may be used to understand the hydraulics of tanks.

I.2 Objectives

The overall objective of this thesis is to simulate the three-dimensional flow patterns in tanks for water storage. The practical purposes of the simulations are to identify water-quality problems related to insufficient water exchange. The water storage reservoirs at Norra Ugglarp in South Sweden will be investigated as a special case study (Norra Ugglarp, Genarp, Lunds kommun, Skåne).

In this thesis, the open source CFD software OpenFoam was chosen to simulate flow features in tanks. OpenFoam is a modelling code that is becoming widespread, handled with a high-level programming language. Also, as an open source software, program and update are free of charge; which was a main reason to prefer OpenFoam to other competitors (OpenCFD Ltd (ESI Group), 2013. *OpenFoam: The open source CFD toolbox*).

From the modelling results one will try to identify mixing problems in such tanks and thereafter provide some solutions to remediate them.

I.3 Procedure

In order to conduct this study, first of all a review of the appropriate literature was carried out, including studies on water-quality problems in storage tanks for water supply and on numerical simulation of the flow in such tanks. Interesting case studies were reviewed and briefly summarized.

Afterwards the numerical software OpenFOAM was applied to simulate the three-dimensional flows in water storage tanks. To become familiar with the software, some simple test setups were first explored, primarily two-dimensional tanks of simple geometry. Focus was on steady-state conditions and various measures were investigated to quantify the water exchange in gross terms. Then the simulations were extended to three-dimensional situations, however, still for simple geometries.

Finally, the software was employed to simulate the flow in Norra Ugglarps reservoirs with the purpose to investigate if the existing water-quality problems are related to the prevailing flow patterns in the reservoir. Solutions to alleviate poor mixing were presented. Data on water levels and flows measured at the reservoir were used in the modelling.

II. CASE STUDIES

II.1 Distribution of water in Skåne

Sydvatten is a municipally owned company that produces drinking water for 800 000 inhabitants of the province Skåne. Skåne is the southernmost county of Sweden.

The drinking water produced by the company comes from Lake Bolmen in Småland and Lake Vomb in Skåne (please look at the map in Figure II-2 for more understanding). If necessary it is also possible to draw water from the Lake Ringsjön in Skåne. The laws and management concerning each lake determine the maximum amount of water that can be withdrawn.

An 80km-tunnel transfers raw water from the lake Bolmen to the reservoirs in Äktaboden. The construction started in 1975 to counter the foreseen lack of water in Skåne.

The tunnel has a cross section of 9m² and water flow is governed by gravity. Then water is pumped to Ringsjöverket. Water from the lakes Ringsjön and Vombsjön are treated, respectively, in Ringsjöverket and Vombverket: two water purification plants in Skåne (Sydvatten, 2010. *Fresh Drinking Water*).

Treated water from the water plants is delivered to West-Skåne, to connection points in each municipality. They are responsible for the distribution of water from this connection point to the customers. Every day, Sydvatten provides drinking water for municipality inhabitants, as well as for industries and healthcare. Approximately 70 million cubic metres are produced each year (Sydvatten, 2012; *Årsredovisning 2012*).



Figure II-1: Skåne, Southern Sweden



Figure II-2: Map of Sydsvatten water distribution system. Sydsvatten 2013, from Årsredovisning 2012
 Purple dot : Norra Ugglarps Reservoirs

II.2 Norra Ugglarps Reservoirs

Our focus of interest is the reservoirs of Norra Ugglarp (located at the purple dot in Figure II-2). Raw water from the lake Vombsjön is purified in the drinking water treatment plant in Vombverket.



Photo II-3: Lake Vombsjön, 2012-05-07

The plant produces an average of 900 l/s of drinking water. The drinking water is pumped to an elevated reservoir in Norra Ugglarp, and then it is distributed to Malmö, Burlöv, Staffanstorp, Swedala, and Vellinge.

The first reservoir was built in 1969 and has a capacity of 3000m³.



Photo II-4: Reservoir n°1, 2012-04-02



Photo II-5 : Reservoir n°2, 2012-04-02

The second reservoir was built slightly after the first one, in 1974, to match an increasing consumption. It has a capacity of 18300 m³.

Those two reservoirs are right now not functioning as a storage capacity reservoir, meaning that water would be stored in the reservoirs first and then distributed to the consumers. Those reservoirs are auxiliary to the water treatment plant in Vombverket.

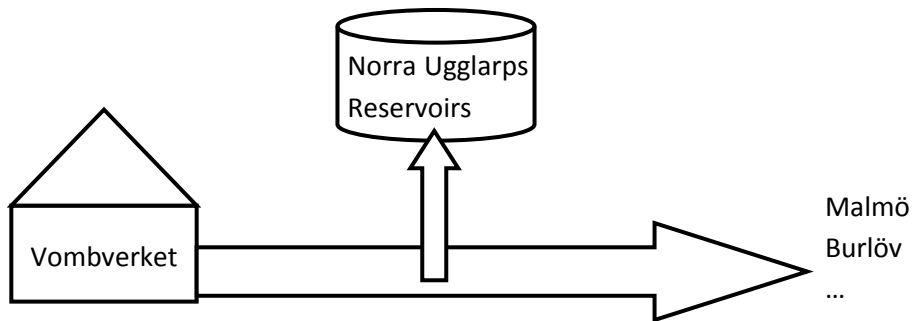


Figure II-6: Sketch of Vombverket water distribution system

The pumps in Vombverket are variable flow pumps. Vombverket delivers water directly to the municipalities, among them for instance Malmö (300,000 inhabitants). The consumption is varying constantly during the day; however, the plant does not adapt the flow to the exact consumption. If the consumption is higher than the flow provided by the water plant, the reservoirs provide the additional supply. When water drains down to a certain set point in the reservoirs, the pumps in Vombverket start/increase flow to the next step. This is done until the water levels in the reservoirs reach a pre-determined high set point. The reservoirs are then considered to be full. Thus, the flow coming from the plant is too high; the water production plant reduces the flow from the pumps to save some energy, but still water is ensured for consumptions.

On the contrary, in the case the consumption decreases, assuming that the pumps are still giving the same flow, the reservoirs are filled with the excess water. This is continued until the water levels in reservoirs reach the high level.

II.3 Water renewal

Water renewal is one of the key factors to quantify the mixing in tanks. Thus, the following question that can be posed: Why is the water not renewed at a sufficient rate? The problem is that the actual capacities of the tanks are just sufficient to ensure a storage reserve for a relatively short period of time. As a precaution, reservoirs are maintained to a quite high water level, using a small water level range (cf. Figure II-7). Thus only a small volume, represented above the dashed line in Figure II-7, is renewed for a given filling or draining.

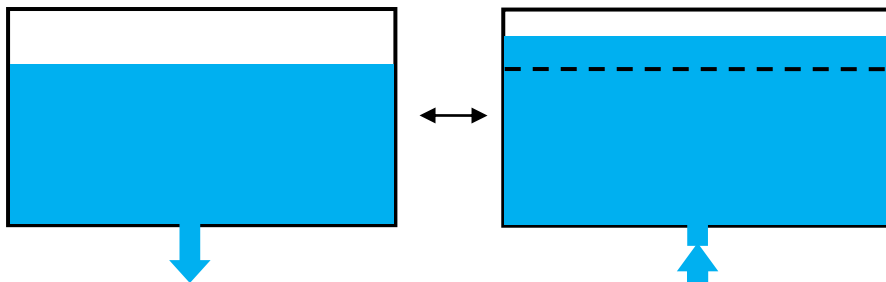


Figure II-7: Sketch of tanks used on their high capacity
Above the dash line: volume of water which is renewed

Moreover, those reservoirs possess only one pipe at their base for the inlet and outlet flows. The larger reservoir possesses one single inlet/outlet close to the centre of the cylindrical reservoir, directed upwards and perpendicular to the bottom of the reservoir. The smaller one has only one pipe but two inlets/outlets: one perpendicular to the bottom directed upwards close to the wall, and another one tangential to the diameter inside the reservoir (cf. Figure II-8). Both reservoirs have inlets and outlets at the bottom and no entrance exists from the top.

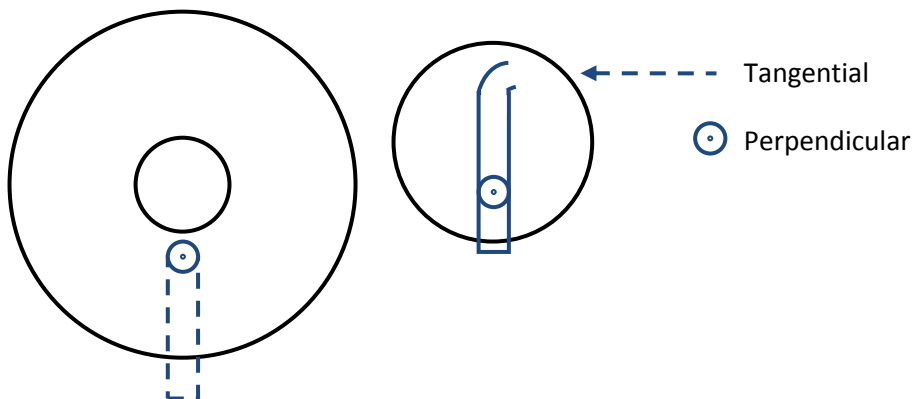


Figure II-8: Sectional view from the top of reservoirs. Right Reservoir 1 / Left Reservoir 2

Due to the aforementioned conditions, the company Sydvatten is concerned about the water mixing in the tanks, which may impact the water quality. It is thought that plug flow could occur in the reservoirs: only the lower part of the tank would be renewed, considered as a new plug, pushing the old water plug upwards. Thus the upper part would rarely be renewed, and water would become aged.

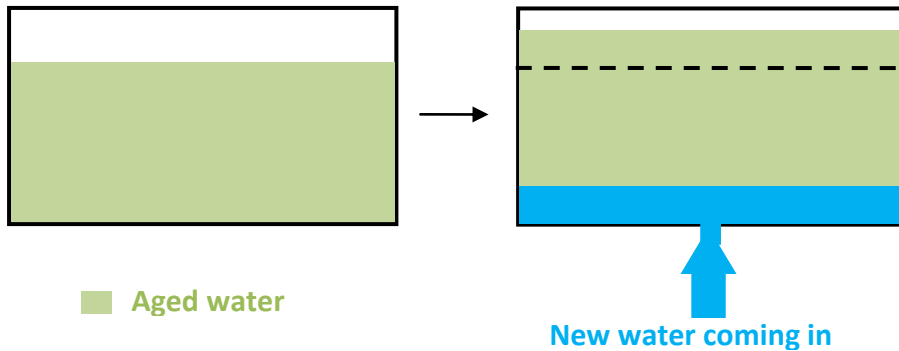


Figure II-9: Assessment of the lack of mixing in the tanks

II.4 Scenarios of studies

To confirm or disprove the above hypothesis about the mixing in the tanks, Sydvatten wants to understand the flow pattern and velocity profiles in the reservoirs. This aim was the core of the present thesis. The CFD software OpenFOAM was used to simulate the flow patterns in the tanks. (A detailed presentation of the software is given in the following section IV-OpenFOAM.)

To establish a first model set up, one needs to know which initial conditions to apply in the model, how much flow enters the reservoir, for how long, and what was the water level in the tank when the filling starts. The problem is that due to different consumption, various rates of flow occur during the day. Moreover, one cannot know exactly what the flow is that goes into the reservoir. No flow meters are present at the entrance of the reservoirs: the incoming/outgoing flows towards/from the reservoirs are thus calculated as the difference between production and consumption (assuming that friction and other losses are negligible). Scenarios and hypothesis about prevailing conditions in the study have been defined together with Sydvatten. As mixing is the focus of the study, a case where the flow towards the reservoir is at the minimum was chosen. To reach a defined set point (water level until the filling is performed) this case would take the longest time to fill the reservoirs. Through the beginning of the current year 2013,

the 7th May was considered interesting since a long filling time occurred during this day (cf. Figure II-9). Precise values of the produced and consumed water, as well as the water level in both reservoirs, were extracted from the monitoring by Sydsvatten.

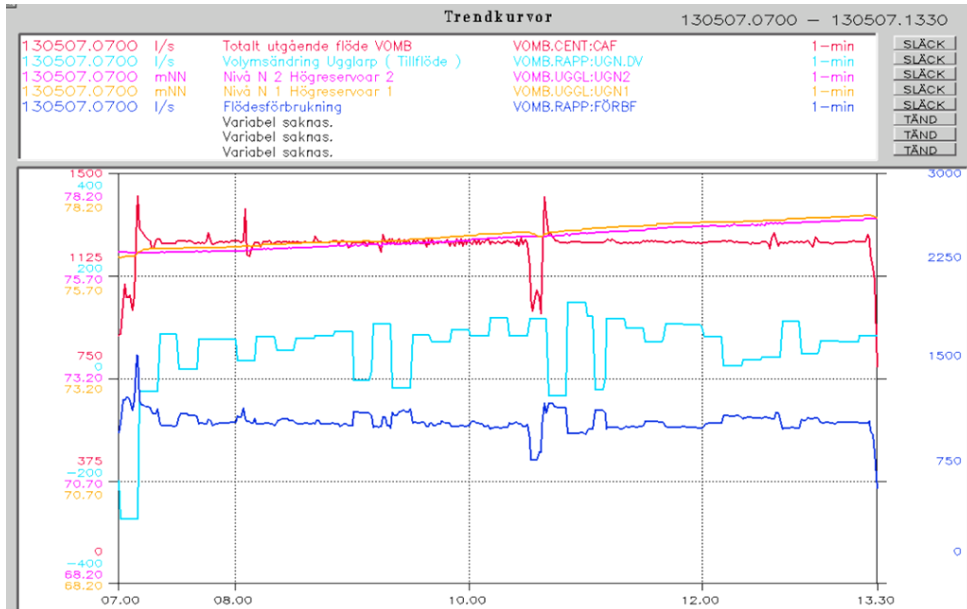


Figure II-10: Monitoring output for the 7th May between 7:00 and 13:30, Cactus software - Sydsvatten

The yellow and pink curves show the water level in Reservoir 1 and 2.

The dark blue line is the consumption.

The light blue line represents the difference in flow between production and consumption.

The red curve shows the total flow produced from Vombverket.

To simplify the problem the following assumptions were made:

- The flow produced by the station is assumed to be constant, here equal to 1225 l/s.
- An average flow towards the reservoirs was calculated as the mean value of the flow during the entire filling.

Those assumptions lead to this simplified state shown in Figure II-11.

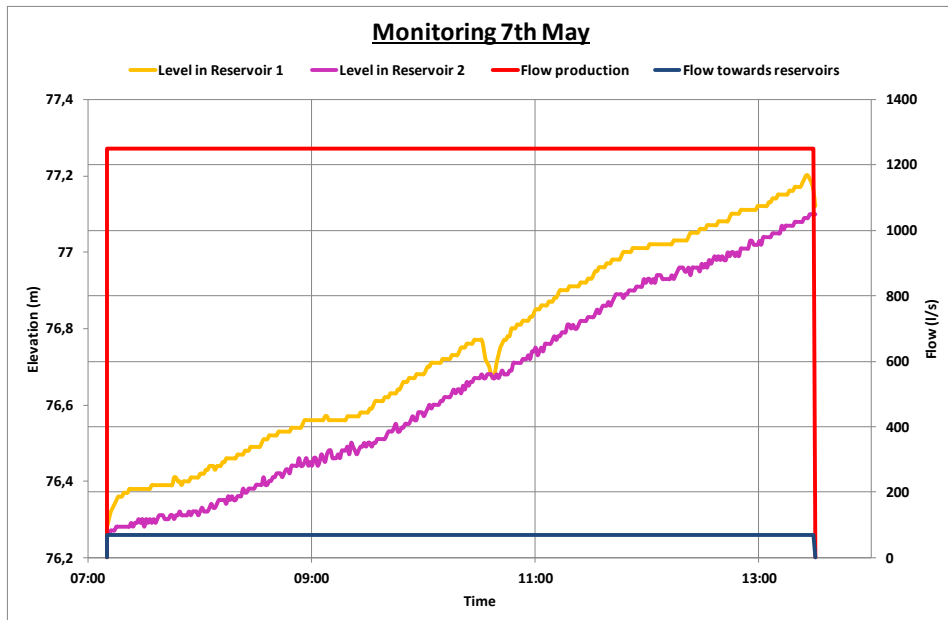


Figure II-11: Simplification of the first study case

The simplified case as shown in the figure above is then:

- filling the tank from 7:10 to 13:30
- flow production of 1250 l/s
- flow towards both reservoirs is 69 l/s

However, since simulations were done separately for each tank, one needs to know the flow that goes into each tank. Since no flow meters are present at the reservoirs, some calculations were made to estimate the flow into each tank. The detailed calculations are explained in Appendix I.

To sum up, the first modelling scenario to be studied aimed to reproduce this filling case:

- Filling reservoir for 6h10
- Start with Reservoir 1 or 2 with a water level up to the elevation 76,30m, that is 10,20m of water in the first reservoir and 9,30m of water in the second
- For reservoir 1, $v_{inlet R1} = 0,003m/s$
- For reservoir 2, $v_{inlet R2} = 0,16m/s$

Table II-12: First scenario of study

After this scenario was set up, and second one was studied involving a case where the tanks are draining down to 1m above the pipe before the water production plant starts to fill them up again, assuming the same velocity:

- Start with both reservoir with 1m of water inside
- For the reservoir 1, $v_{inlet R1} = 0,003m/s$
- For the reservoir 2, $v_{inlet R2} = 0,16m/s$
- Filling up until the same setpoint value as the first scenario

Table II-13: Second scenario of study

The purpose with the first scenario is to investigate the ‘actual’ mixing, meaning the most critical situation that occurs occasionally nowadays. The second situation is selected to provide another case to compare with, and answer the issue how the mixing may improve when a different starting set point is used before filling the tanks.

III. SIMULATION OF WATER FLOWS IN STORAGE TANKS

III.1 Basic mechanisms

Before going into the governing equations, some basic fluid mechanisms of relevance will be discussed. For the specific case study where the tank possesses only one inlet at the base, one main parameter should be the flow rate entering the tank, or the inlet velocity. Considering that the tank has already water in it, a strong velocity would permit the incoming jet to penetrate the existing water in the tank and reach the surface. On the contrary, a low inlet velocity would just affect the layer above the inlet, which in turn would only slightly influence the layer above and so on, creating something similar to a plug flow.

The strength of the inlet velocity on one side needs to counteract the pressure field on the other side. Indeed, all cases studied in the following part with at least a minimum of 9m water depth in the tank. The more water that is present in the tank, the more hydraulic pressure faces the incoming jet, due to the water column above it.

Due to the viscosity, or the turbulent stresses occurring in flow of the studied cases (Cf. Reynolds Number, section III-2), the incoming jet will lose energy as it is ascending into the reservoir. In fact, the upward-transported jet particles are affected by the shear stress from the surrounding water particles.

Temperature is another major parameter, even though it is not considered hereafter in this study. If the stagnant water in the tank would have a different temperature than the incoming water, the density would differ influencing the flow. A colder – meaning denser – inflow will have a tendency to sink in warmer ambient water. Gradients in temperature, or even stratification could occur, preventing water exchange between layers.

III.2 Governing equations

This section provides the governing equations employed in the numerical model and the numerical solution method. The modelling approach is based on the eddy-viscosity concept and volume of fluid (VOF) for tracking and locating the free surface.

The equations were solved using OpenFoam 2.2.0 CFD software package. (More details are given in the following section IV). For our case study, the latest solver *interMixingFoam* was chosen. This is a “solver for 3 incompressible fluids, two of which are miscible, using a VOF method to capture the interface” (OpenCFD, 2013, *section Standard solvers*).

In the case of the Norra Ugglarp reservoirs, the air will be simulated as well as the water; one phase will represent the existing water in the tank, considered as aged or “old” water, and incoming water will be represented by another phase. Those two phases are miscible, and the interface towards the air is tracked with the VOF method. Readers may wonder why the water and the air phase are simulated together and not only the diffusion of new water in the existing water in tanks by using a concentration tracer. A problem is that the studied case of the filling (or draining) of the tanks implies that the water level will rise during computations. In the case of solving the equations for the water only, a moving mesh will be required to follow the moving water surface. Assumption would then be made that the water stays still at the surface. Beyond the fact that implementing a moving mesh would be more complicated, using the solver *interMixingFoam* has permitted us to prove that the water stays still at the surface, and that the air is not entrained into the water by whirlpools and swirls.

The equations describing the motion of the flow are the Navier-Stokes (momentum) equations and the continuity equation. For the VOF method, those equations are solved for each phase, and then the physical properties of one fluid are calculated as weighted averages based on the volume fraction of the 3 fluids in one cell.

The momentum equations (Navier-Stokes equation) takes the form:

$$\frac{\partial \rho U}{\partial t} + \nabla \cdot (\rho U U) - \nabla \cdot (\mu + \mu_t) \nabla U - \rho g = -\nabla p - F_s \quad (\text{III.1})$$

Where each term represents:

- $\frac{\partial \rho U}{\partial t}$ the unsteady acceleration
- $\nabla \cdot (\rho U U)$ The convective acceleration
- $-\nabla \cdot (\mu + \mu_t) \nabla U$ Shear stresses
- $-\rho g$ Gravity

- $-\nabla p$ Pressure gradients
- $-F_s$ Surface tension forces (occurs only at free surfaces) cf. below

The continuity equation takes the form:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho U) = 0$$

But since the water flow is considered as incompressible ($\rho = \text{constant}$), this equation could be simplified as

$$\nabla \cdot U = 0 \quad (\text{III.2})$$

To estimate whether the flow is laminar or turbulent, the ratio of inertial forces to viscous forces, known as the Reynolds Number, is calculated

$$Re = \frac{\rho v L}{\mu}, \text{ where} \quad (\text{III.3})$$

- ρ Fluid density
- v Fluid velocity
- L Characteristic travel length
- μ Dynamic viscosity

A Reynolds Number over 4000 characterises a turbulent flow, or the ratio $\frac{\rho L}{\mu} = \frac{10^3 * 0,7}{10^{-3}} = 7 * 10^5$. Unless considering very small velocities (under $\frac{4000}{7 * 10^5} \approx 5 * 10^{-3}$, our study case is considered turbulent.

Turbulence is modelled by the eddy-viscosity concept. The turbulent viscosity μ_t in equation (III.1) is obtained from the turbulent kinetic energy, k , and the dissipation rate, ε , using the standard $k - \varepsilon$ model (Rodi, 1993)

The eddy viscosity is given by

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (\text{III.4})$$

Turbulent kinetic energy (after neglecting buoyancy forces)

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_i k)}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right) + G_k - \rho \varepsilon \quad (\text{III.5})$$

The dissipation rate is obtained from

$$\frac{\partial(\rho\varepsilon)}{\partial t} + \frac{\partial(\rho u_i \varepsilon)}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_i} \right) + C_{1\varepsilon} \frac{\varepsilon}{k} G_k - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} \quad (\text{III.6})$$

$G_k = \mu_t S^2$ is the shear production term

$S^2 = ((\partial u_i)/(\partial x_j) + \partial u_j/\partial x_i) \partial u_i/\partial x_j$ is the squared modulus of the mean rate of strain tensor.

The constants of the $k - \varepsilon$ model take the following standard values: $C_\mu = 0.09$, $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$, $C_{1\varepsilon} = 1.44$ and $C_{2\varepsilon} = 1.92$ (CFD online, 2007, *Standard k-epsilon entry*).

The volume of fluid in a cell is calculated as $F_{vol} = \alpha_i V_{cell}$, where V_{cell} is the volume of a unity cell of the mesh and α_i is the fraction of the fluid i in a cell. The fraction α should vary between 0 and 1. In our case, the interMixingFoam will associate α_1 with air, α_2 with the aged water and α_3 with the new water. For instance, if the cell is completely filled with the new water then $\alpha_3 = 1$ and if it is filled with another fluid then its value should be 0. The scalar function α can be computed from a separate transport equation that takes the form

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha U) = 0 \quad (\text{III.7})$$

The momentum equations and the continuity equation are in reality solved for the fraction α_1 and α_2 ; then the remaining fluid is deducted as follow:

$$\alpha_3 = 1 - \alpha_1 - \alpha_2$$

The surface tension F_s in the momentum equation (III.1) is computed as:

$$F_s = \sigma \kappa(x) n$$

where σ is the surface tension coefficient ($\sigma = 0.5$) and n is a unit vector normal to the interface that is calculated from

$$n = \frac{\nabla \alpha}{|\nabla \alpha|}$$

and κ is the curvature of the interface that is calculated from

$$\kappa(x) = \nabla n$$

III.3 Numerical solution methods

The solution of the coupled Navier-Stokes equation and Continuity equation is not straightforward since an explicit equation for the pressure is not available. One of the most common approaches is to derive the momentum equation and by substituting it in the continuity equation. More details about numerical solution methods will be discussed directly together with the solver `interMixingFoam` used in this study in the section IV.3 Solution schemes.

III.4 Example of previous studies

A review of appropriate literature was carried out to refer to previous studies focusing on modelling the flow in drinking water storage tanks. Even though such studies have been limited in the past, cases of interest that have been performed should be mentioned and briefly reviewed here:

- Martinez-Solano et al., 2010. *Modelling flow and concentration field in rectangular water tanks.*

This study was carried out with the CFD software FLUENT®, to model flow and concentration field of a tracer within a 3D rectangular tank. The classical two equation $k - \epsilon$ model was used to approach turbulence. The case study included a tank with a volume of 1500 m³ and dimensions 20x25 m² and a water depth of 3m. The inflow to and outflow from the tank were 50l/s. The inlet and outlet were located in opposite corners of the tank. (Their case study was smaller than the Norra Ugglarp reservoirs: considering both reservoirs as a single one, the tanks have a total capacity of 21300m³ for a mean inflow to the tanks of 69l/s). Mixing was evaluated through injection of a tracer through a third pipe. The mesh was splitted into hexahedral cells, finer around the inlet and outlet points, ending up with around 19000 cells. Several parameters for the mixing process were compared, and the main conclusion of this study was that 82% of the volume was under complete mixing conditions.

- Lemke A. and DeBoer D.E., 2012. *Effect of storage tank mixing on water quality*

This thesis presents the results from a study of seven storage tanks used in the South Dakota rural water systems (USA). Seven different high tanks were under examination. The design and placement of inlets were studied, but emphasis was placed on the impact of temperature and concentration of chlorine on mixing. Thus temperature and water sampling were performed to examine the reservoir

behaviour at varying heights. Hydraulic and mixing parameters were used to determine mixing capabilities, as well as a chlorine decay modelling using the CompTank software (Grayman et al. 2000) based on CFD. Conclusions on these practical experiments were the following. The higher the ratio height/diameter is, the poorer the mixing will be. Adding a riser pipe (inlet on the top of the reservoir) to a storage tank is an effective way of promoting mixing in the tank.

- Nordblom, O., 2004. *Mixing and Stagnation in Drinking water storage tanks*

Last but not least, this PhD study aimed to assess mixing conditions in drinking water storage tanks. After measurements of temperatures and flow rates in two drinking water storage tanks, numerical modelling was accomplished to determine parameters governing the initiation of stagnation in tanks. Fluent 6 was used to resolve the governing equations of the flow, based on the eddy-viscosity concept. Results showed that the thermal stratification has a significant impact on the mixing conditions as well as denser inflow (due to a temperature below the temperature in the tank). Denser inflows tend to stay in the lower part of the tank and the water exchange during a fill-and-draw cycle is reduced. Hence, for a certain temperature difference between the incoming and the stored water, a higher inflow velocity is required to avoid stagnation in deep tanks and at high temperatures. Numerical calculations confirmed those experimental results, leading to a criterion implying that complete mixing occurs when the height of rise of the inlet jet exceeds some fraction of the initial water depth in the tank. The validation study suggested that calculations based on the standard $k - \varepsilon$ model are sufficiently accurate to draw correct conclusions about the mixing characteristics. Also, horizontal or upwardly inclined inflow was preferable in tanks with height-to-width ratios below 0.25, whereas vertical inflows are better in tanks with initial height-to-width ratios greater than 0.5.

IV. OpenFOAM

Computational fluid dynamics (CFD) is a part of fluid mechanics used to solve and analyse many problems that involve fluid flows that has been employed since the 1960s. Among softwares that exist on the market nowadays, the OpenFOAM® (Open Field Operation and Manipulation) CFD Toolbox was used to carry out the simulations in this thesis. This is a free, open source CFD software package including meshing tools, many solvers and post processing applications distributed by OpenCFD Ltd. The OpenFoam code is an object-oriented toolkit written in C++. Since it is a free software package, OpenCFD distribute OpenFoam for Linux operating systems (OpenCFD Ltd (ESI Group), 2013 and ESI Group, 2013. *First OpenFOAM User Conference*).

If not familiar with Linux, one should start with a tutorial to be able to handle more rapidly the OpenFoam software. At the start of this thesis, a very well conducted tutorial was followed in order to understand Linux {in French} (SiteDuZéro, 2013: *Reprenez le contrôle à l'aide de Linux*). To return to our main concern, details about the OpenFoam structures and handling are provided in the following sections.

IV.1 Modelling strategy

The modelling strategy in OpenFoam do not differ from others and follows the same basic procedure:

- Pre-processing:
 - Define clearly a study case with a geometry to investigate
 - The volume considered is divided into discrete cells, which form a grid called the mesh.
 - Physical models are defined – in our case governing equations for the flow
 - Boundary and initials conditions are defined
- Simulations are launched and equations are solved iteratively
- Post-processing: Visualisation and analysis results from simulations

Following this modelling strategy, the structure of OpenFoam is constructed as different directories in a case directory to be able to run an application.

IV.2 Software structure

To run a *case* in OpenFoam a directory is created that must contain the following files, described in Figure IV-1.

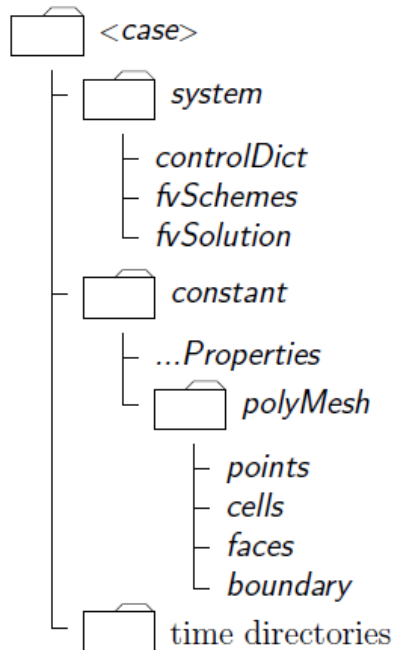


Figure IV-1: Structure of a case study in OpenFOAM, *OpenFoam 2013*

The *constant* directory contains all physical properties of the three fluids, e.g., the density of water in the *transportProperties* file and the gravitational acceleration g . This folder also includes a subdirectory *polymesh* required to build the mesh. The *blockMesh* utility provided by OpenFoam is used for generating simple meshes of blocks of hexahedral cells. When it comes to generate more complex meshes of hexahedral cells, it is possible to import a CAD file in OpenFoam and use *snappyHexMesh* tool to create the mesh. The latter technique has been used to generate the 3D cylinder of the Norra Ugglarp reservoirs (cf. Chapter VII)

The *system* directory contains several dictionaries. All information for starting, ending, time step and saving date are found in *controlDict*. The discretization schemes used in the calculations are set up in *fvSchemes*. Equation solvers, tolerances and other algorithms are located in *fvSolution*. To be able to decompose a case to run it in parallel mode, a dictionary *decomposeParDict* is present in this system directory. Finally, in the case of a free surface problems,

initial volume fractions of the three fluids (if using *interMixingFoam*) need to be defined. The dictionary *setFieldsDict* divides the domain into three parts and gives different alpha value for each region.

The ***time*** directory contains a sub-directory for each time step of writing results (defined in *controlDict*). For instance, pressure p and velocity U are found in each time directory. In the case of free surface problems, values of alpha will be also found in each step. OpenFoam must be initialised, for initial and boundary conditions. This is done by imposing fields in the 0 directory, for instance pressure field is initialised through the file 0/p.

An example of each file can be founded in appendix II for the tank directory.

IV.3 Solution schemes

A first-order implicit scheme (Euler) was used for the temporal discretization. Gauss linear (second order) scheme is used for the discretization of the divergence terms. The Laplacian terms are discretized using the central difference scheme.

As said previously, in section III.2, the solver *interMixingFoam* was used to perform the modelling of the flow. Since both the momentum equations and continuity equation have no analytic solutions, discretization needs to be done.

The solver *interMixingFoam* is an improvement of the *pisoFoam* solvers, which is a transient solver for incompressible flow. The PISO algorithm (Pressure Implicit with Splitting of Operators) is used to solve Navier-Stokes equation in unsteady problems. As already pointed out, once those two equations are solved for each phase, where *interMixingFoam* uses a VOF method to track the interface between the two fluids.

Without going into the details, here is a summary of the PISO algorithm: (OpenWiki, 2009. Entry: *The PISO algorithm in OpenFOAM*)

“The algorithm can be summed up as follows:

1. Set the boundary conditions.
2. Solve the discretized momentum equation to compute an intermediate velocity field.
3. Compute the mass fluxes at the cells faces.

4. Solve the pressure equation.
5. Correct the mass fluxes at the cell faces.
6. Correct the velocities on the basis of the new pressure field.
7. Update the boundary conditions.
8. Repeat from 3 for the prescribed number of times.
9. Increase the time step and repeat from 1.”

Convergence of the numerical solution in both stationary and transient cases was judged mainly by examining the CLF condition (Courant, R.; Friedrichs, K.; Lewy, H.; 1928). In fact, PISO is based on the assumption that the discretization of the Navier-Stokes equation may be safely frozen through a series of pressure correctors, which is only valid at small time-steps (Gram, 2009). To ensure convergence of PDEs used by OpenFOAM, the Courant number, Co , should always be below 1. The Courant Number is defined as

$$Co = \frac{\Delta t |u|}{\Delta x}$$

where Δt is the time step, $|u|$ the magnitude of the velocity through the element and Δx the size of the element in the direction of the velocity.

Moreover, according to OpenFOAM (2013, UserGuide, *Chapitre 2 Tutorial*): “Time step control is an important issue in free surface tracking since the surface-tracking algorithm is considerably more sensitive to the Courant number Co than in standard. Ideally, we should not exceed an upper limit $Co \approx 0.5$ in the region of interface.”

Since the propagation velocity is not easy to predict, *interFoam* offers an automatic adjustment of the time step. While choosing this option, users should define a maximum Co . A maximum value of 0.5 was judged enough to ensure converging results. A maximal time step of $\Delta t = 0.05s$ was chosen for the 2D case, but this had to be decreased to $\Delta t = 0.01s$ for convergence in the 3D simulations.

IV.4 Boundary and initial conditions

The boundaries of the geometry are generated by *blockMesh* (or *snappyHexMesh*) by employing the boundary file in the *polyMesh* directory. In

this thesis, the Norra Ugglarp case contains a list of 5 boundary patches: *extWall*, *intWall*, *lowerWall*, *atmosphere* and *pipeWall*. The user should attribute a standard type for each of the aforementioned patches. The atmosphere has a standard patch, i.e., just an entity on which boundary conditions can be specified, as well as the pipe. Other patches are defined as walls; a plain patch to apply special wall surface modelling can be used if needed. When patches are defined, boundary and initial conditions are set in the first time directory read by the solver, which is usually the *0/* directory. The top of the tank is free towards the atmosphere and permits both inflow and outflow (of air). The combination of boundary conditions for pressure and velocity is the following:

- For the pressure; at wall boundaries, the *buoyantPressure* boundary condition is applied to the pressure field, which calculates the normal gradient from the local density gradient; on the top, the pressure is *totalPressure* which is a fixed value calculated by $p_0 = p + \frac{1}{2}\rho|U|^2$.
- Velocity at every wall is fixed to 0, defined as a *pressureInletOutletVelocity* on the top layer, which applies a zero gradient on all components except where there is inflow, in which case the tangential component is fixed to 0.
- For each alpha phase, a *zeroGradient* condition applies on walls; the same conditions for flows outwards on the top and pipe layer, and a fixed value when the flow is inwards. (OpenCFD, 2013. *Chapter 2 Tutorials*)

In the same files are the initial conditions defined for the velocity (for instance *inlet velocity { value (0 0,5 0); }*), the pressure, and also the starting values of each phase alpha.

IV.5 Output handling and analysis

Once the calculations have been completed, users can launch the post-processing tool provided with OpenFOAM, called ParaView. ParaView is open-source software for visualisation application and can be launched directly in OpenFoam by taping the single command *paraFoam* within the case study. A pipeline browser lists the case opened as well as the modules selected. The properties panel contains the input fields for your case (such as pressure and velocity), and the display panel contains the setting for visualising the data for a given case. For more details about the handling of ParaView, please consult either *OpenFoam: Chapter 6 Post-processing* (OpenCFD Ltd (ESI Group), 2013.) or ParaView website (Kitware, 2013).

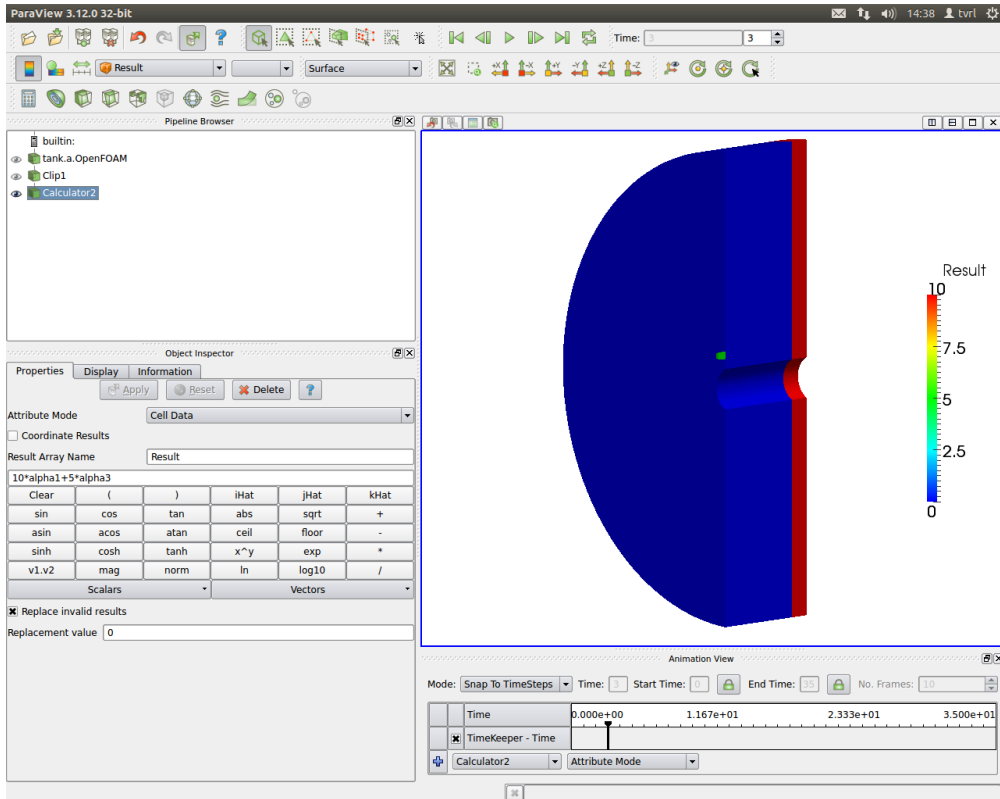


Figure IV-2: The paraFoam windows / tank case

In this thesis, different modules were used to be able to analyse the data from the calculations:

- Clip and slice tools were used to be able to look inside a 3D volume, as shown in figure IV-2 above
- Different phases (air, aged water and fresh water) or the pressure were directly shown by using the corresponding alphas in the properties panel
- For the velocity, some filters such as *Cell centers* then *Glyph* were used to display velocity as arrows, instead of a plain field colour
- The Calculator was also used to emphasize certain fields or neglect others (for instance to display only water velocity and neglect air velocity).

IV.6 Using the Cluster

The computer provided by the department for this thesis has only one processor. When starting dealing with 3D calculations, it appeared that this single processor was not powerful enough to handle such jobs. Just to illustrate this, the final tank case, with the coarse mesh, had 181744 cells to compute for. Some single

simulations required so much computing power that a single processor would require days or even years to complete the calculations. Thus, an access to the cluster at Lund University was required. Generally speaking, a cluster consists of a set of connected computers that work together so that they can be viewed as single supercomputer. A Cluster provides computational power through the use of *parallel programming*, a technique for coordinating the use of many processors for a single problem.



Figure IV-3: Picture of Platon cluster at Lunarc. Lunarc, 2013. (Tab Systems / Platon)

A small size project (max 5000 core hours/ month) was given to this thesis to be able to run cases on the Platon cluster. Due to the queuing systems, starting of submitted jobs could occur immediately or being put with lower priority in the queue depending of several parameters such as the number of processors needed, hours of computation. To find a compromise between quick launching and a larger amount of used processors, it was usually required for this thesis to solve cases in parallel for 24h of calculation, using *nodes:2 ppn:8*: i.e., using in total 16 processors (2 nodes, each one including 8 processors). More details about the cluster, applying for a project, submitting project can be found on the Lunarc webpage (Details in references). The submitting script used for launching is provided in Appendix III: this script may save some time for news users, reducing their effort when launching jobs on the cluster.

V. MODELLING TWO-DIMENSIONAL FLOW IN A STORAGE TANK

V.1 Test setup

To become familiar with the software, a test case was first explored, investigating a two-dimensional tank of simple geometry. Actually, “OpenFoam always operates in a 3 dimensional Cartesian coordinate system and all geometries are generated in 3 dimensions. OpenFOAM solves the case in 3 dimensions by default but can be instructed to solve in 2 dimensions by specifying a ‘special’ *empty* boundary condition on the boundaries normal to the (3rd) dimension for which no solution is required.” (OpenCFD, 2013. *Chapter 2 Tutorials*)

After reviewing several tutorials and becoming more familiar with the model, it appeared that the solver *interMixingFoam* would be the most appropriate for the cases studied. In fact, this solver is able to compute for “3 incompressible fluids, two of which are miscible, using a VOF method to capture the interface”. In other words, in the present case, fresh and aged water – obviously miscible – can be tracked, as well as air in the tank thanks to this solver.

The setup of the first case study is presented below. This case is a simple rectangular tank of length 21m and height 15m. (The width is here 1m to act as a ‘2D’ model.) The mesh generator supplied by OpenFoam, *blockMesh*, generates meshes from the dictionary *blockMeshDict* specified in the *constant/polyMesh* directory. (NB: For the following of this study, the reader should note that an example of the script for the 3D tank case can be found in Appendix II.)

A uniform mesh of 21 by 15 cells is used initially, as described in Figure V-1 below. Each cell has a volume of 1m^3 .

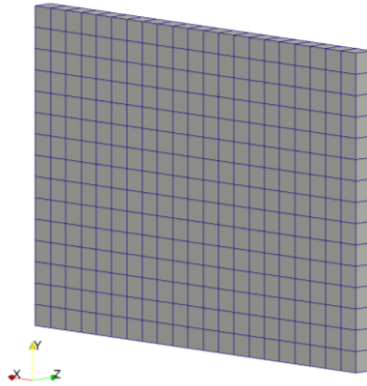


Figure V-1: Mesh for the 2D case – Domain: $x[0\ 21]$ $y[0\ 15]$ $z[0\ 1]$

Boundary conditions (in the blockMeshDict file) were set as: top face-atmosphere and inlet were simple patches, leftWall, rightWall, and base as walls, whereas front and back remained empty in this 2D case.

V.2 Simulation cases

A case where the rectangular tank was already filled with 10m of water was studied. Above the free surface of water, air is present in the tank. The inlet is placed in the middle of the base of the tank, and it injects fresh water with a velocity of 1m/s. Since the inlet represent one side of a cell, its section is 1m^2 , and thus the inflow inwards this reservoir is $Q = 1\text{m}^3/\text{s}$.

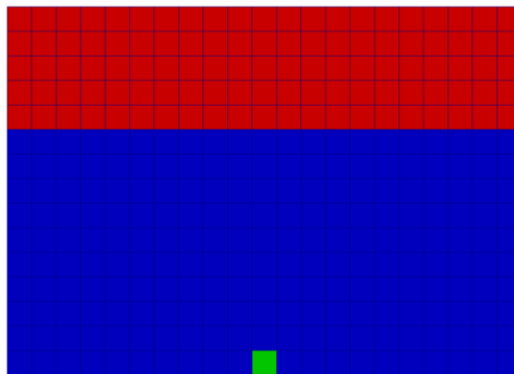


Figure V-2: Defined initial values. Green: Fresh water / Blue: Old water / Red: Air

A simulation case where the reservoir was filled up to 13m was simulated (which means that injection of water lasted 60s). Due to the small number of cells – 315 cells – and a maximal $\Delta t = 0.05\text{s}$ – simulations for this case converged rapidly (48s clock time for a one core processor).

V.3 Results

Results were post-processed with ParaView, employing the single command *paraFoam* within the case directory. Please find below in figure V-3 general results at the end of the simulation.

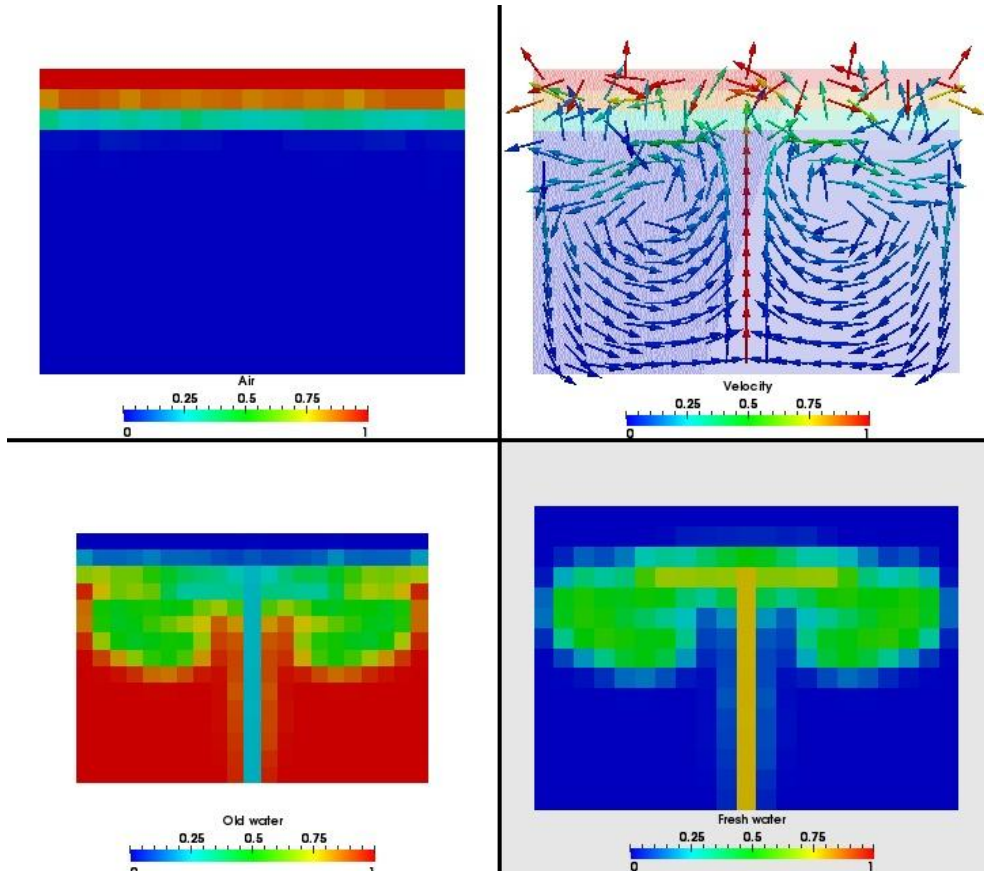


Figure V-3: flow in a rectangular case in 2D, after 60s of 1m3/s.

As seen, it is possible to present several views at the same time in *paraFoam* and to observe simultaneously different fields. Upper left picture is the air, computed by alpha1 in the solver. A cell completely filled with air has a fraction of 1, in red in the legend, and a completed submerged cell has an air fraction of 0, blue in the picture. Following the same concept, bottom left view represents the fraction of Old water in the tank: a complete fraction of old water is shown in red, absence of old water in the cell is shown in blue. On the bottom right, the very same concept is used for the Fresh water. Finally, on the upper right side, one can follow the velocity pattern in the tank. Attention must be drawn to the fact that the velocity is computed by the solver both for both air and water. Since the water velocity field is

of primary interest, one should ignore the velocity arrows above the water surface. Thus, the background colour illustrates the air fraction field in order to grasp where the water surface is located and only velocities below this level should be considered. The focus is now on intermediate and final results.

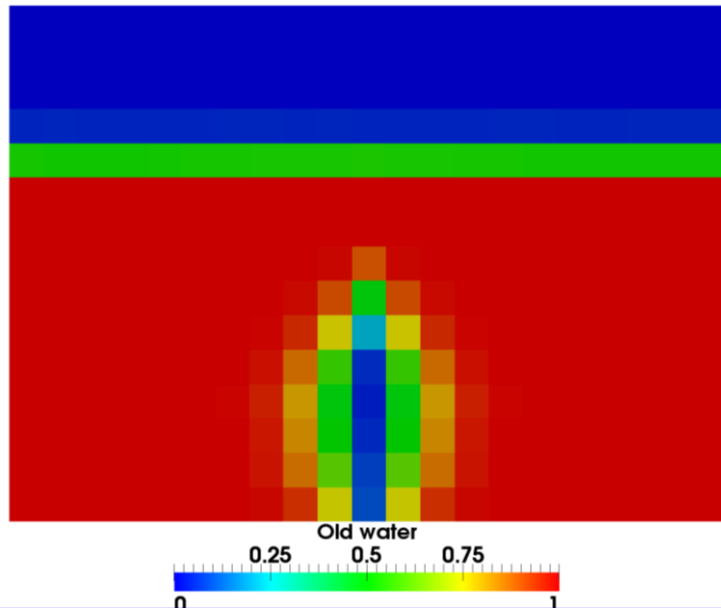


Figure V-4: Old water fraction in the tank after 12s of inflow

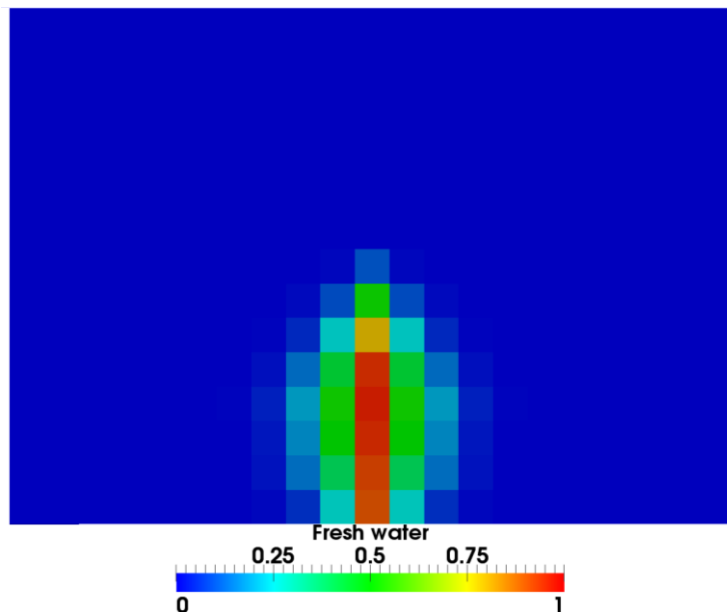


Figure V-5: Fresh water fraction in the tank after 12s of inflow

In the two pictures above, (Figure V-4 and V-5), since the air stays still above the water surface, results show exactly the same picture but in contrast. Indeed, under the water surface, cells are filled either with aged water, fresh water, or a fraction of both. After 12s of flow, the jet is only fresh water at its center, but the jet starts diffusing at the sides.

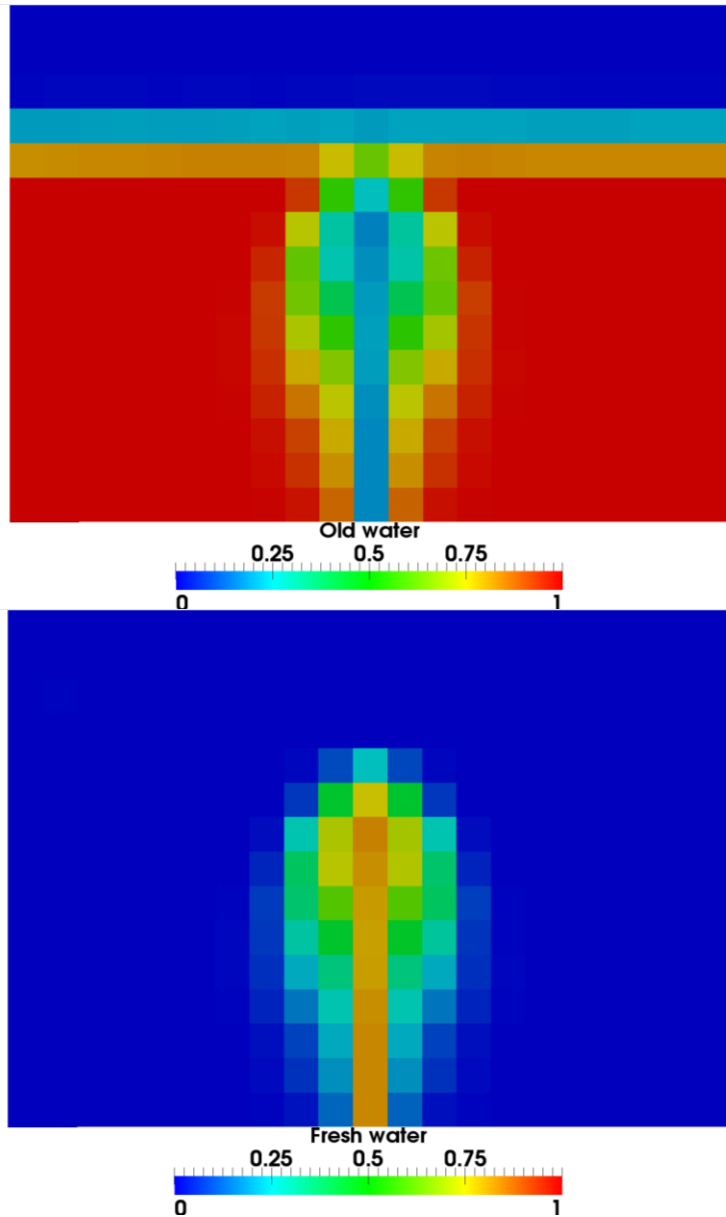


Figure V-6: Old (upper part) and Fresh (bottom) water fraction in the tank after 22s of inflow

In figure V-6, the inside of the jet is not completely fresh water, but a mixture of roughly 75% of fresh water and 25% of aged water. It also starts to diffuse on each side. More important, the velocity of the jet is strong enough to reach the surface.

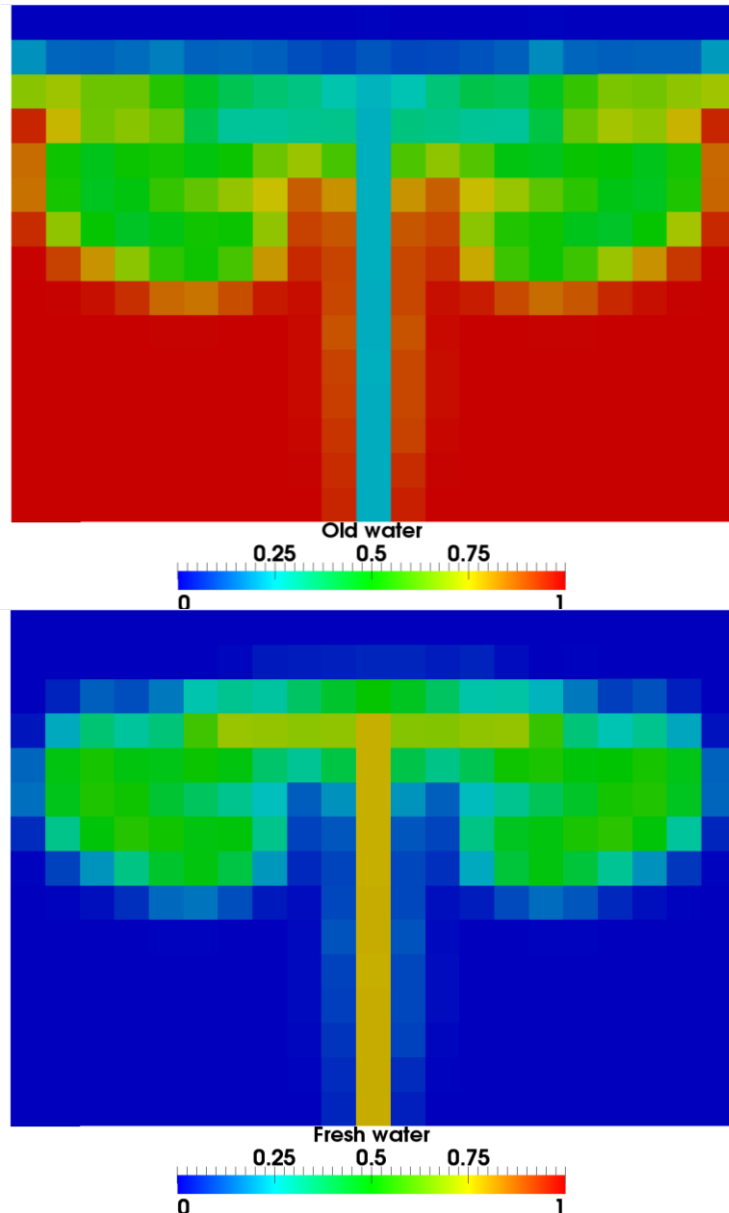


Figure V-7: Old (upper part) and Fresh (bottom) water fraction in the tank after 60s of inflow

Figure V-7 above presents results at the end of the simulation; after 60s of $1\text{m}^3/\text{s}$ inflow. Considering the fraction of fresh water injected in the tank compared to the existing one ($\frac{1\text{m}^3/\text{s} \cdot 60\text{s}}{21\text{m} \cdot 13\text{m}} = \frac{60}{273} = 22\%$), the proportion of fresh water would be 22%

and then the proportion of old water 78%. A perfect mixing would then correspond to a plain cyan colour for the fresh water and plain yellowish colour for the aged water. Instead what can be observed is that the incoming jet reaches the surface and then spreads out laterally, both to the right and left. Two circulation cells appear in a large top layer of the reservoir. Those cells are inside green, which means an equal fraction of old and new water. The jet path also seems to be established, going straight to the surface and not diverging as much as in the beginning.

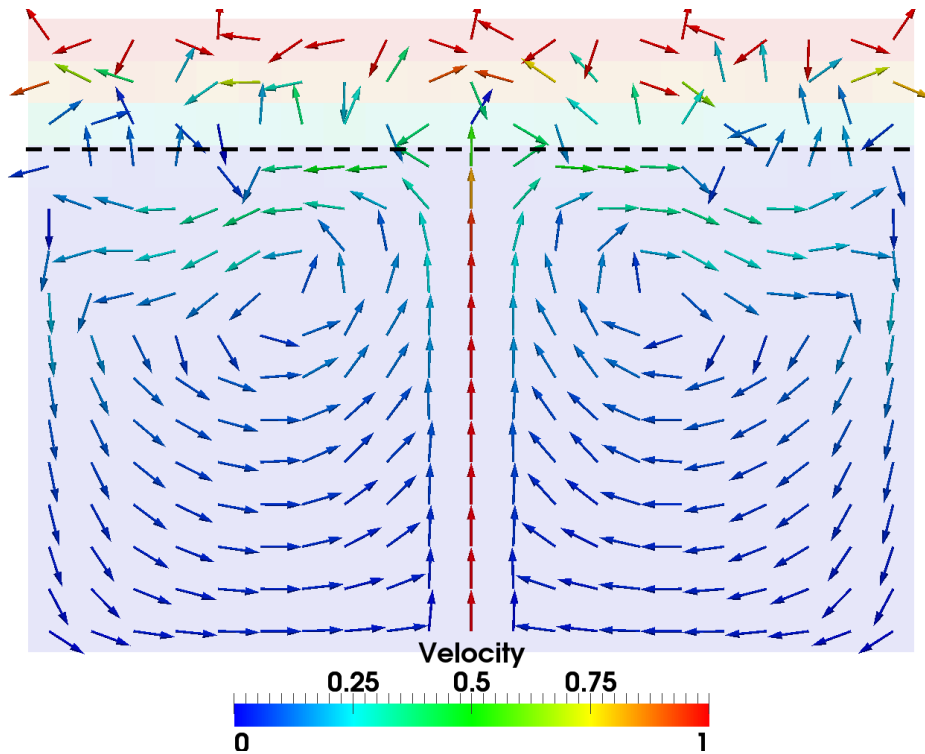


Figure V-8: Velocity profile inside the tank after 60s of inflow
Dashed line represents the still water surface

Figure V-8 presents the velocity field at the end of the simulation. Once again, the velocity is computed for both air and water phase. The background colour shows the air fraction to clearly distinguish water surface. The dashed line represents the still water surface and the velocity above it is the air velocity and should not be considered. Attention is also drawn to the fact that the arrows are all of the same size, independent of the velocity magnitude. Only the colour illustrates the magnitude of the velocity.

At the end of the simulation, the velocity field confirms the previous conclusions drawn by the fraction fields. The jet is powerful enough to reach the top layer and it ends up at the surface with a similar velocity to the initial one (1m/s). Diffusion occurs on the sides of the jet with a small velocity, the jet being concentrated straight in the middle. The jet separates then into two circulation cells, with half the velocity (0.5m/s) in the top of the cells, and generates a circulation on both sides, apparently down to the base of the tank. However, the further away from the water surface, the smaller the velocity, being close to zero at the bottom of the tank. Considering the previous fraction fields, one could assess that only the upper part of the reservoir, in the case with a strong jet, is renewed by the induced circulation of water in the tank.

Working with this case first made the author more familiar with the software and its general handling, including managing and interpreting the output. Even if this model deals with a 2D case, it provides a picture of the prevailing patterns in the tank regarding velocity and mixing.

VI. MODELLING THREE-DIMENSIONAL FLOW IN A STORAGE TANK

VI.1 Test setup

Another rectangular case, now in 3D, will be presented in this section. The setup of the 3D case is the following: a rectangular tank with a square base of 21m side, and a height of 15m. The hexahedral mesh was also once more designed with the blockMesh tool, slightly more complicated since all points, faces, and blocks needed to be defined one by one in Cartesian coordinates. The inlet was placed in the middle of the tank, and uniform meshing was applied, $21 \times 15 \times 21$ cells, for a total of 6615 cells (figure VI-1). Each cell had a volume of 1m^3

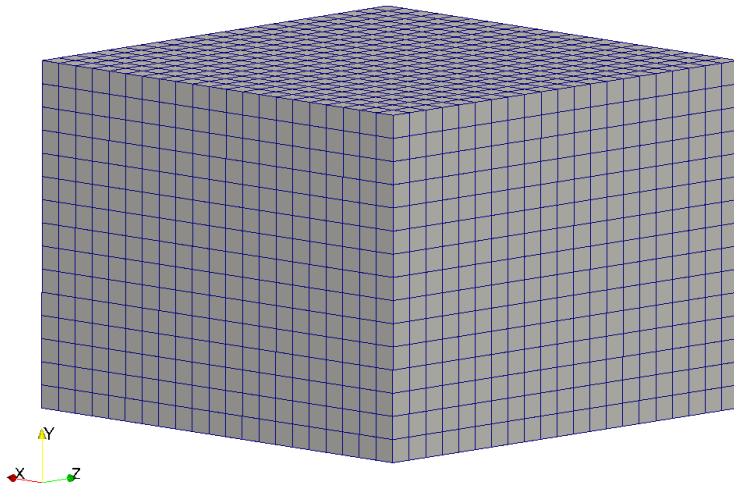


Figure VI-1: Mesh for the rectangular tank – Domain: $x[0\ 21]\ y[0\ 15]\ z[0\ 21]$

VI.2 Simulation cases

This simulation case represents a tank, already filled with 10m of water, which represents a volume of $21^2 \times 10 = 4410\text{m}^3$. As before, air occupies the upper part of the tank when starting the simulation. The inlet in the middle of the base of the tank injects fresh water with a velocity of 0.5m/s, which represents an inflow of $Q = 0.5\text{m}^3/\text{s}$. For a better understanding and to be able to display the results inside the tank, it was split up along two sides in the post processing, which permits one to see both inside and outside the volume (Figure VI-2).

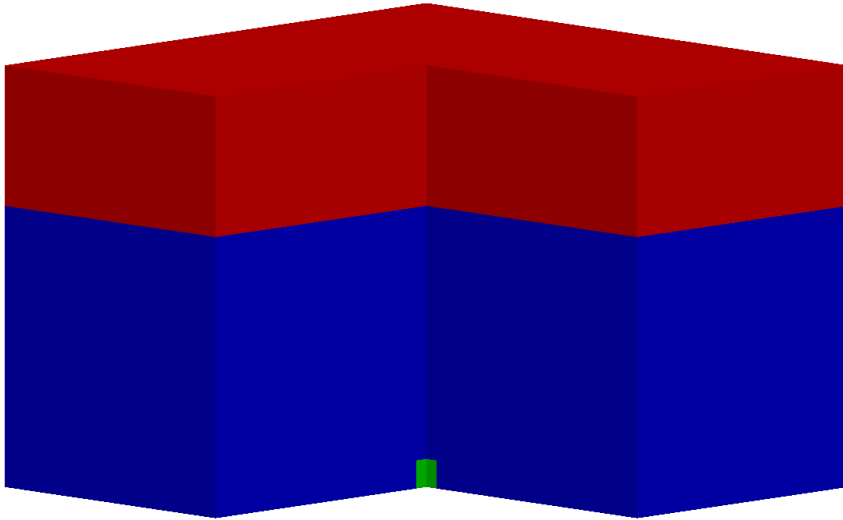
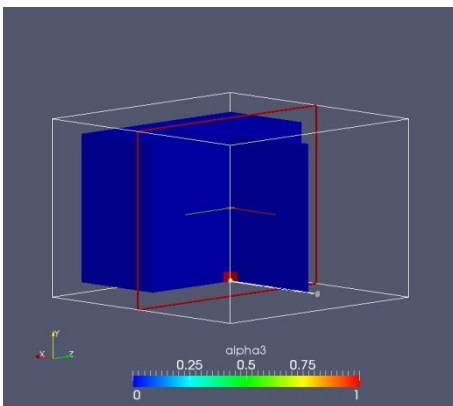


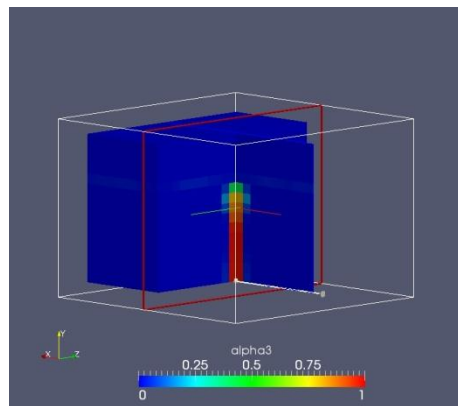
Figure VI-2: Definition of initial values. Green: Fresh water / Blue: Old water / Red: Air

A simulation where the tank was filled up to 12m was carried out (which means that the injection of water lasted 24min – real time). For 6615 cells and a maximum $\Delta t = 0.01s$, successful simulations converged in 31060s (that is about 8h40s).

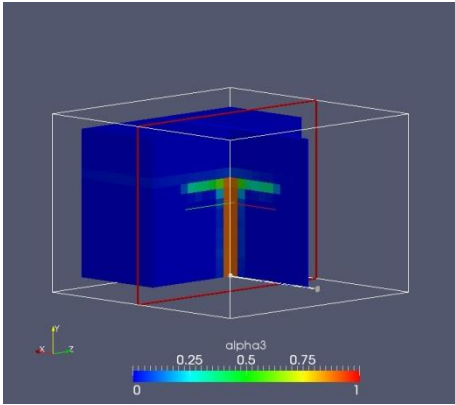
VI.3 Results



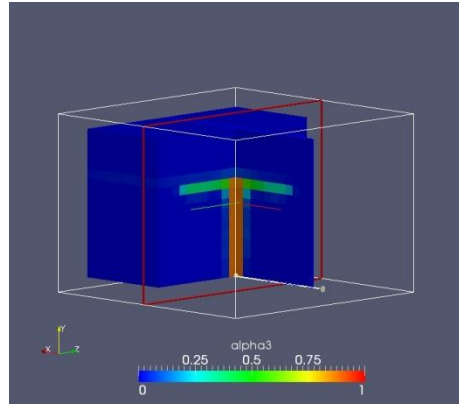
Picture 1: State of the tank at t=0s



Picture 2: State of the tank at t=40s



Picture 3: State of the tank at $t=0s$ $t=90s$



Picture 4: state of the tank at $t=120s$

Figure VI-3: Overview of the fresh water field through the first minutes of the simulation

As seen in Figure VI-3, even though the jet has a velocity of 0.5m/s, it seems powerful enough to reach the top layer and cause mixing at the surface. Fresh water spreads out in the top layer at the surface. Focus is now on the developments afterwards and the final results.

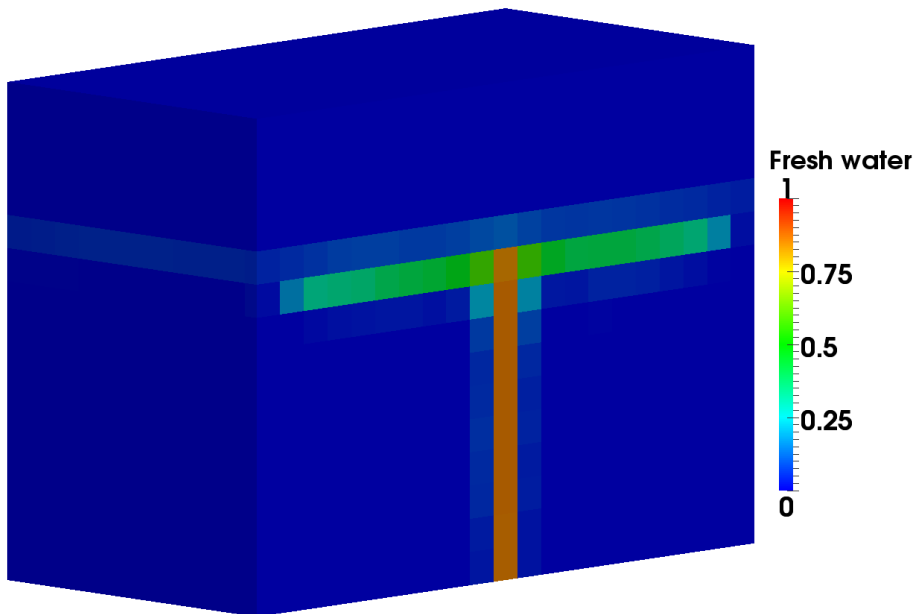


Figure VI-4: Fresh water fraction in the rectangular reservoir at $t=180s$

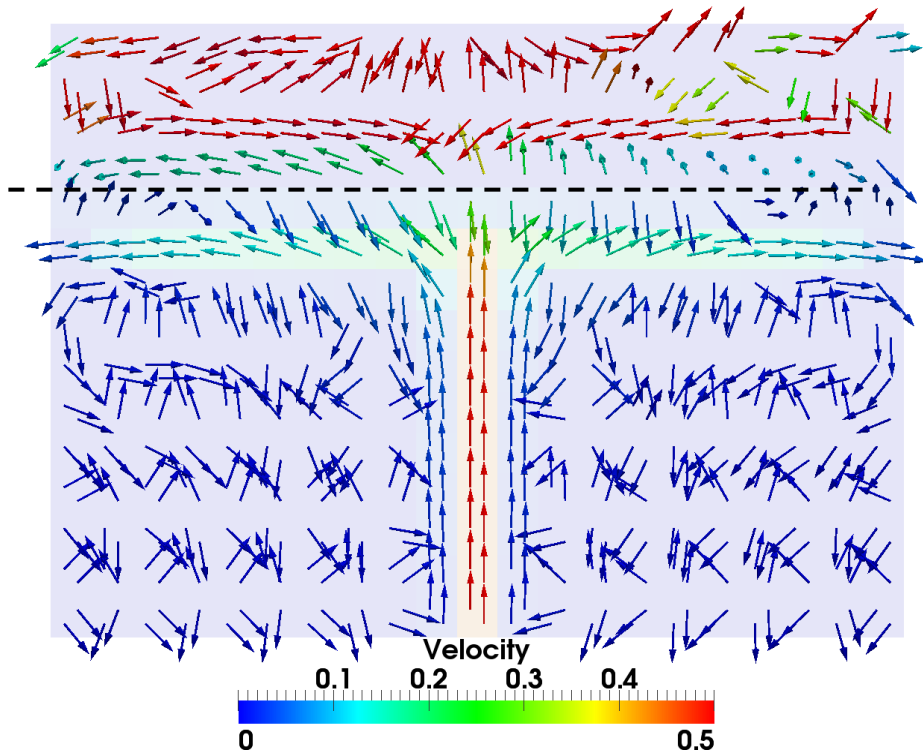


Figure VI-5: Velocity profile at $t=180s$ in the middle section of the rectangular reservoir
Dashed line represents the still water surface

After 180s, the jet has reached the surface and water movement is induced on both sides of the jets. The velocity is initially constant along the jet, equal to the inlet velocity of 0.5m/s, but it decreases to 0.2m/s on the surface.

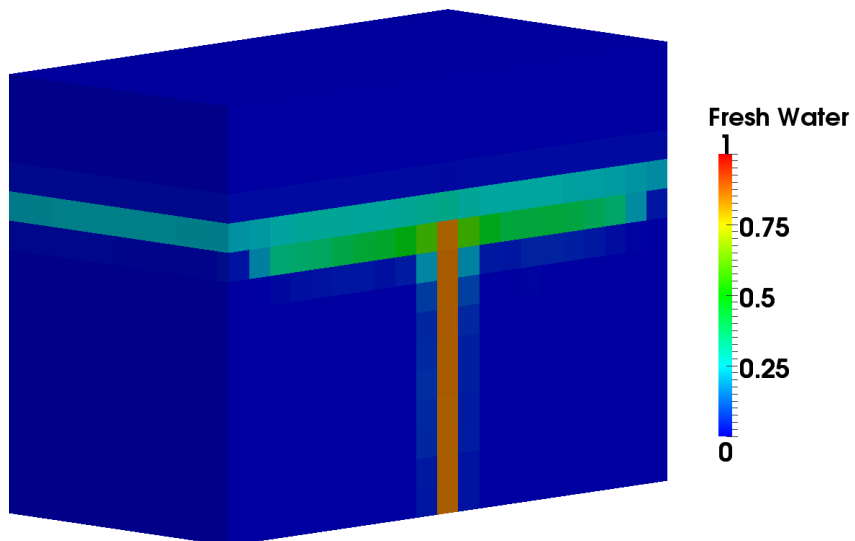


Figure VI-6: Fresh water fraction in the rectangular reservoir at $t=360s$

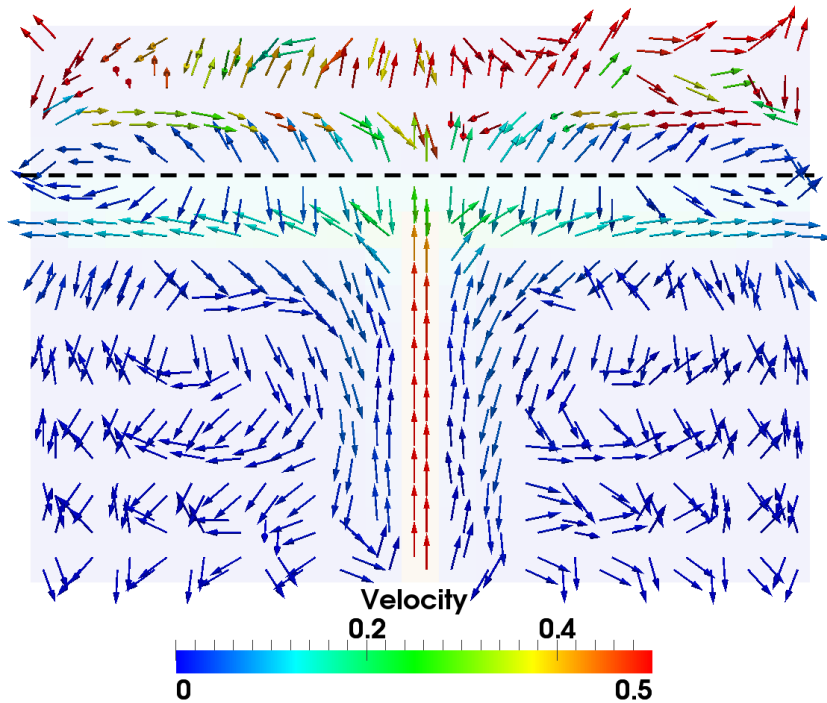


Figure VI-7: Velocity profile at $t=360s$ in the middle section of the rectangular reservoir

After $360s=6$ min, the fraction of fresh water are the same as described previously. The velocity profile also tends to follow the same pattern, even if some details are difficult to explain, for example, in some cases circulation against the main current seems to occur. However, since those velocities are represented in dark blue – meaning close to zero - readers should neglect influence of those velocities.

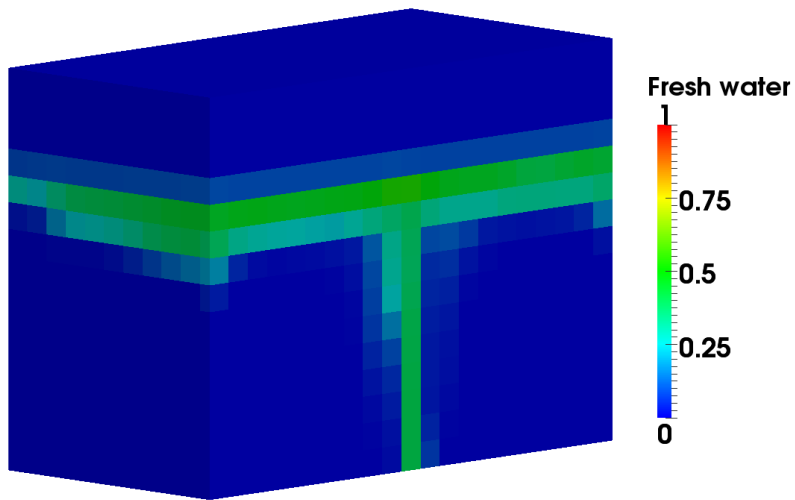


Figure VI-8: Fresh water fraction in the rectangular reservoir at t=720s

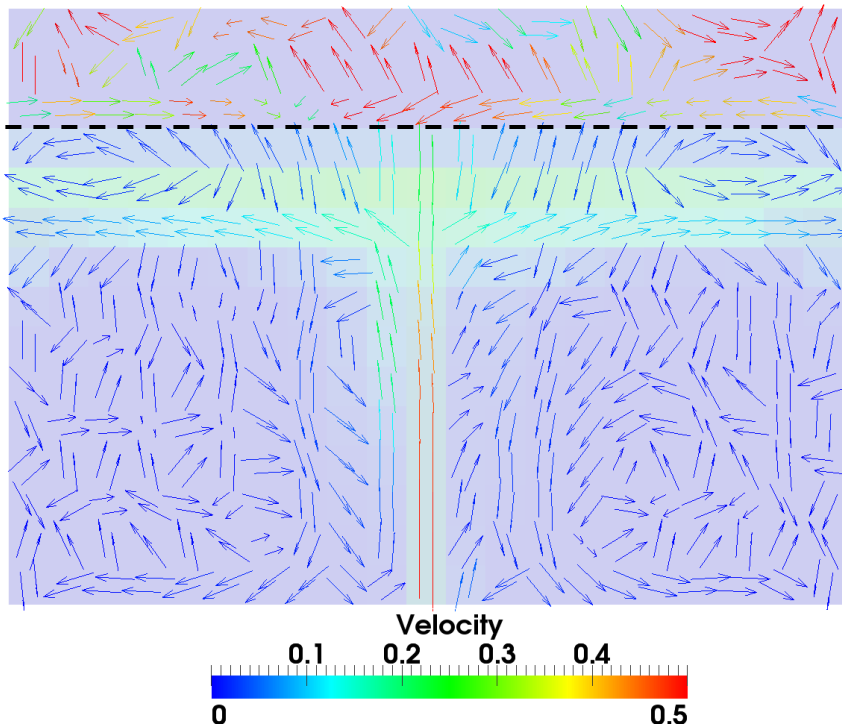


Figure VI-9: Velocity profile at t=720s in the middle section of the rectangular reservoir

At the end of the simulation, fresh water seems to go straight to the top, and then spread to the fourth side of the rectangular tank. No larger circulation cells are present compared to the 2D case. Fresh water seems just to stay afloat. One can notice from the velocity profile that old water appears to be entrained by the

upgoing jet. An important observation, especially visible on the fraction of fresh water graph, is that the solution looks not to be symmetric, although the initial conditions are the same on every side. This may be explained by the size of the mesh, which includes cubic cells of 1m^3 .

VII. MODELLING THE FLOW IN NORRA UGGLARPS' RESERVOIRS

VII.1 First reservoir

As previously mentioned in the presentation of the case study, the first reservoir was constructed 1969 and had a capacity of 3000m^3 . This only represents $\frac{3000\text{m}^3}{18300\text{m}^3} = 16\%$ of the capacity of the larger reservoir. Moreover, this reservoir possesses one single pipe but two inlet/outlet, which is thought to be beneficial for improving mixing. Since both reservoirs are actually connected with the same supply pipes, it is assumed that if the larger reservoir is sufficiently mixed, then the smaller reservoir will therefore also be well mixed. Thus, due to time constraints, the focus of the simulations was only on the larger reservoir.

VII.2 Second reservoir

This section will discuss the modelling of the larger reservoir and the studied scenarios. As a reminder, this reservoir was constructed in 1974 to ensure enough capacity to supply an anticipated growing population. Its full volume of 18300 m^3 resides in a cylinder tank of 46m in diameter, which includes an inside tower to provide access to the top with a 4.4m diameter. The inlet pipe is of DN700 located close to the middle in the bottom of the tank (cf. Figure VII-1).

Unlike the previous cases, *blockMesh* was not used to create the mesh. Actually, as acknowledged by the guide of the software itself, *blockMesh* utility is just good enough “for generating simple meshes of blocks of hexahedral cells, [...] the *snappyHexMesh* utility for generating complex meshes of hexahedral and split-hexahedral cells automatically from surface geometries” was used instead. (OpenCFD Ltd, 2013. *Chapter 5 Mesh generation and conversion*)

The use of the *snappyHexMesh* tool requires importing a geometry file before creating a mesh. The design of the reservoir was thus created with Salome, open source CAD (Computer-Aided Design) software that runs also under Linux distributions (Open CASCADE, 2013). The design of the tank is shown in Figure VII-1.

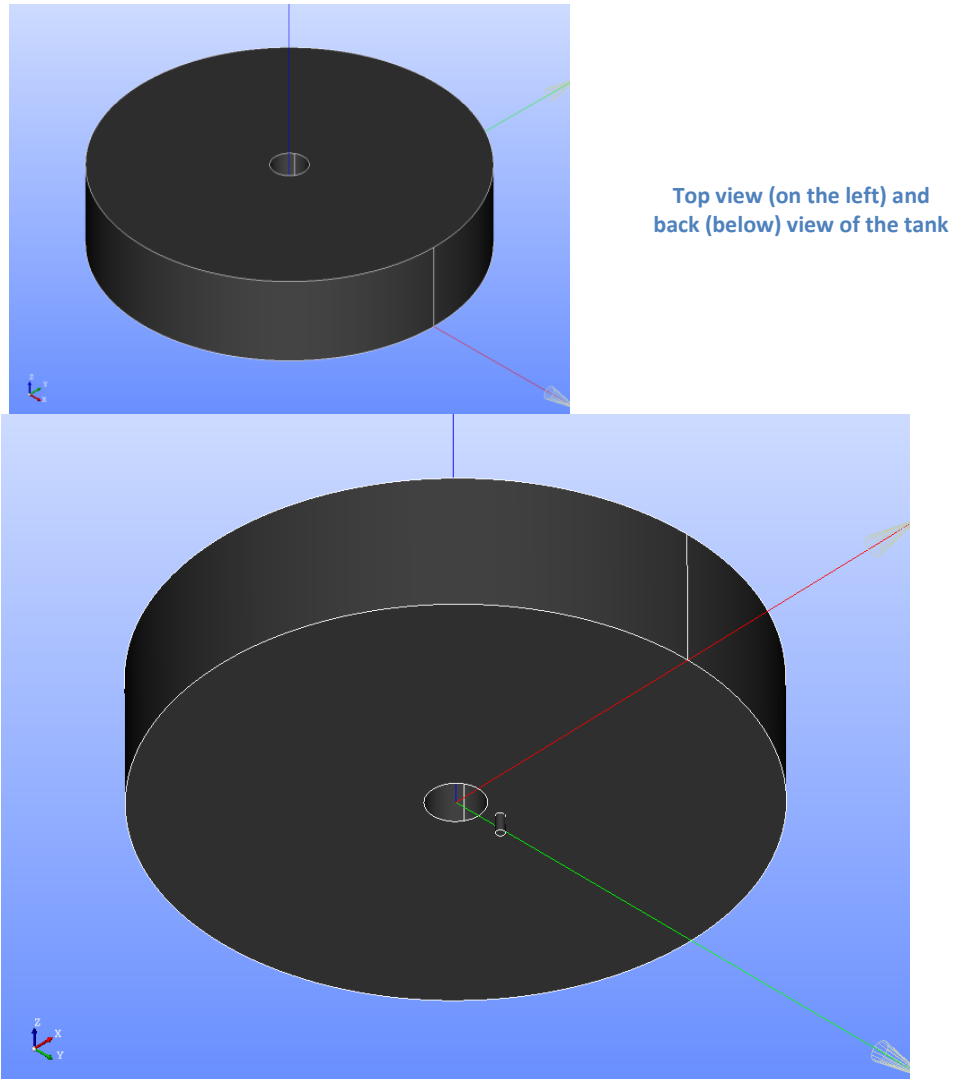


Figure VII-1: Design of the tank using Salome - Open CASCADE, 2013

After creating the geometry, a special format file was exported from Salome to OpenFOAM to be able to use the tool snappyHexMesh. This tool was not easy to utilize. Sharp edges were not well defined. It was actually a problem of the tool itself, corrected in the last version of OpenFoam 2.2.0 (OpenCFD Ltd. (ESI Group), March 2013. *Release releases OpenFOAM® 2.2.0*). The relatively small inlet pipe, compared to the large size of the tank, was extremely sensitive to the background size of the mesh, sometimes not even detected when the mesh was too coarse. Once it was detected, it was at the very first time steps not possible to apply to different part of the tank different boundary conditions, the entire entity tank+pipe considered as one single volume. Both problems were finally solved with solutions

provided by the forum CFD online (CFD Online, 2013. *How to create patch after snappyHexMesh*).

Resolving the problems of the patches was done by following this procedure:

- Once created the geometry in Salome, explode all blocks into surfaces
- Export each surface into a STL.ASCII files, and rename each solid in this files using your Linux distribution editor
- Gather all the files to one single file, for instance surface.stl
- Run *snappyHexMesh* with this surface.stl

Finally the design was obtained. The post-processed view is presented in Figure VII-2. Interested users will find the script to the *snappyHexMesh* dictionary in Appendix III.

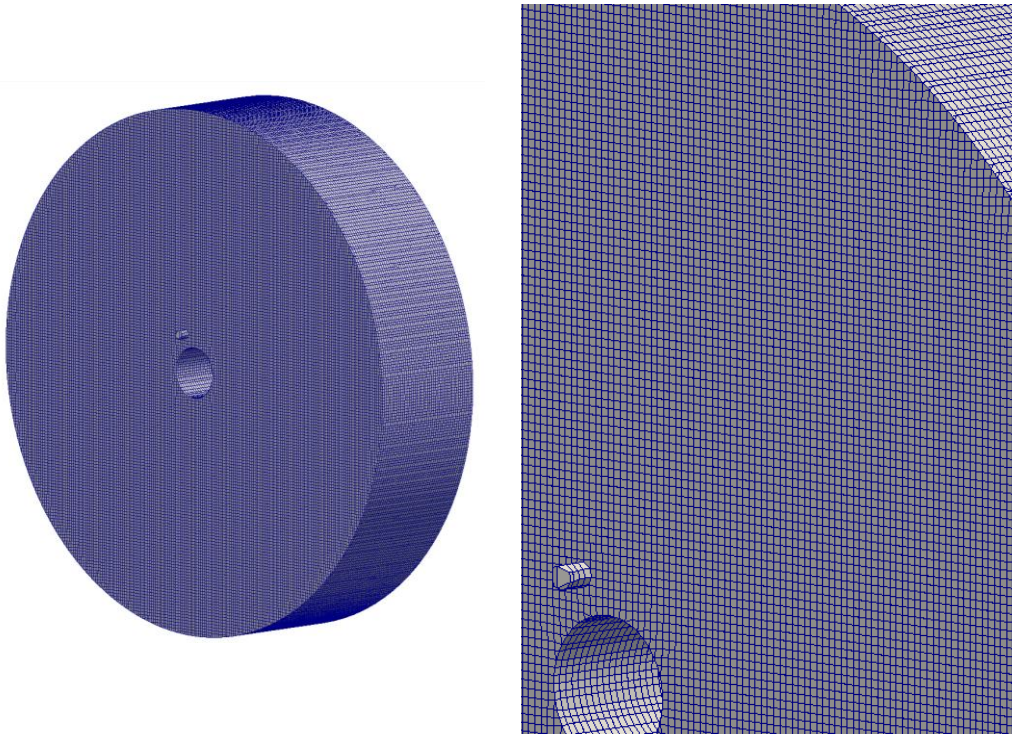


Figure VII-2: Coarse mesh for the tank – Note that the view is rotated right to 90°

VII.3 Simulation cases

As already discussed in section II.4 Scenarios of studies, the first case to be simulated is summarized in the following table:

- Filling reservoir for 6h10
- Start with Reservoir 2 with a water level up to the elevation 76,3m, that is 9,3m of water in the second
- For the reservoir 2, $v_{inlet R2} = 0,16m/s$

Table II-11: First scenario of study

As done in previous cases, each phase was setup in the setFields dict (script for the tank in Appendix II) and end up with the initiation point seen below in figure VII-3.

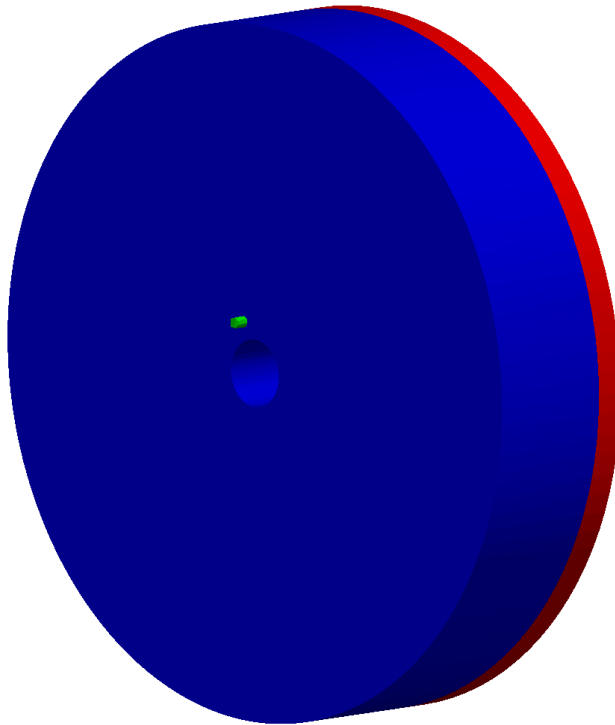


Figure VII-3: Definition of initial values. Green: Fresh water / Blue: Old water / Red: Air

One can already notice, before any simulation, that this situation describes the tank when the pumps start the filling-up procedure. Attention must be drawn on the very low water level range actually used by Sydvaatten on regular filling up: the set point for the filling up starts when water level drain down to 9.3m in the reservoir (shown in Figure VII-3), and stops when the filling up reaches 10.1m. The water level range is only 80 cm (i.e., the equivalent of $V = A_{pipe} * \Delta N = 1383m^2 * 0.8m = 1106m^3$). This case ended up having 181,744 cells. Each cell has a maximum volume of $0.3^3 = 0.027m^3$.

VII.4 Results

The first simulation that succeeded to run on the computer provided by the department - with a maximal $\Delta t = 0.05s$ was giving approximately 1s of the flow into the tank under 1h of calculation. A simple cross-multiplication shows that for solving our case of study, the calculations with this computer would take $21600h \approx 30 \text{ months}$. What could be done? Either simplify the case or ask for more powerful resources. Both solutions were study simultaneously.

- The first option was trying to make the mesh coarser. This solution was unfortunately not working since the simulations were very sensitive to the mesh size. Even trying to excessively decrease the time step was not enough for the calculations to converge.
- Another possibility was to perform the calculations on the cluster: Even while using 16 cores on the cluster, the calculations were very slow and getting the following results for 35s of flow in the tanks took 24h of calculation.

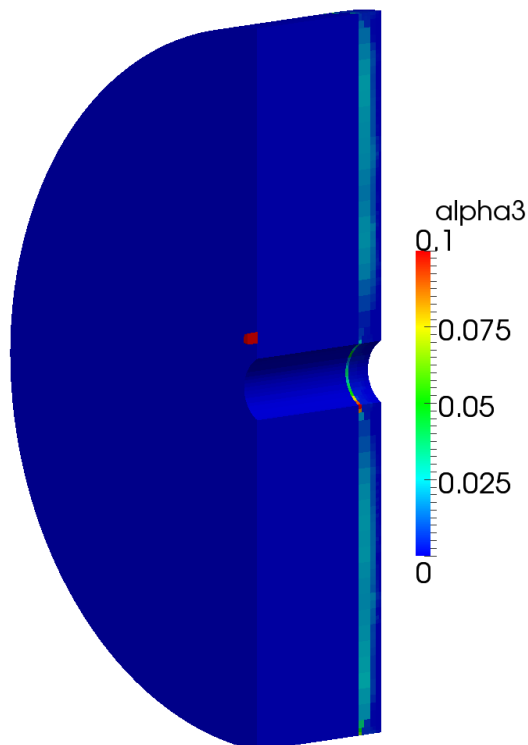


Figure VII-4: Fresh water fraction in the tank after 35s of filling up

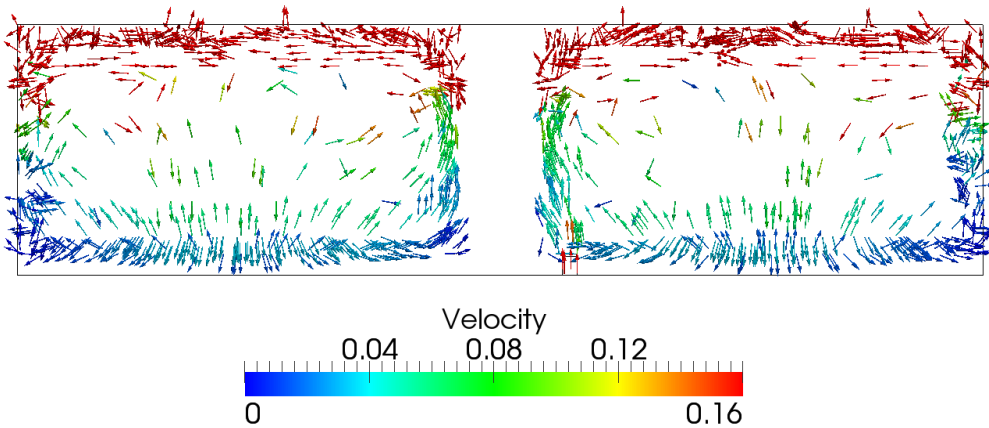


Figure VII-5: Velocity field in the tank after 35s of filling up

From the simulations one could obtain an overview of the first seconds of the flow in the tank. It appears in Figure VII-5 that on the right side of the tank (side of the inlet pipe) two cells of circulation seem to be present. The left side of the tank does not seem to induce any circulation cells after 35s.

- A second option was trying to go back to a rectangular tank with a square base of 41m, trying to catch the corresponding flow pattern as for the cylinder with a 46-4,4m diameter (Figure VII-6).

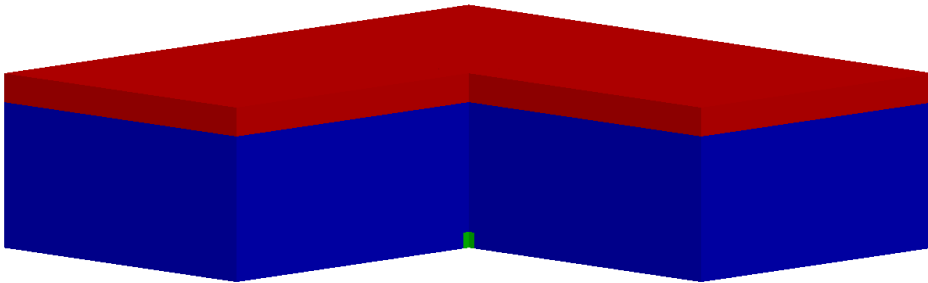


Figure VII-6: Definition of initial values. Green: Fresh water / Blue: Old water / Red: Air

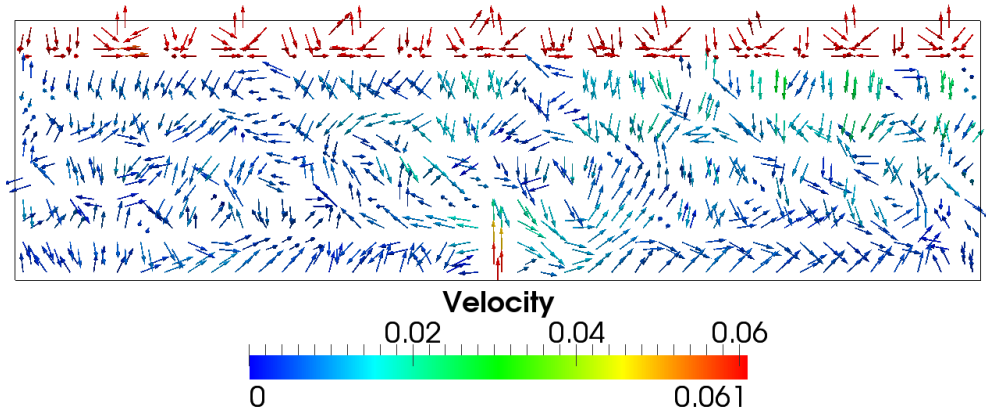


Figure VII-7: velocity profile after 53min of injection

In Figure VII-7, the jet seems to enter weakly into the tank and be blocked after few layers, around 4m from the tank bottom.

The fraction of water, due to the low volume injected compared to the actual one, where not able to track the new water.

VIII. Limit of the study

Even if the modelling tries to reproduce reality through accurate and detailed descriptions, some simplifications were done to perform the cases. The following parameters are not included, or limited throughout the study.

- A low slope is present on the floor of each tank to convey water from any place in the tank towards the discharge outlet. This difference in elevation, which reaches 27cm between the discharge pipe and the opposite wall, is modelled in all simulation as a flat base. Since the water depth in the tank is nowadays at minimum 8.5m, it is assumed that the impact of this small difference on the water column can be neglected. $\left(\frac{27\text{cm}}{8.5\text{m}} = 3,2\%\right)$
- The shape of the pipe itself was modified, both to simplify the model, but also to be able to be detected it using the meshing tool *snappyHexMesh*
As shown in figure VIII-1, the real pipe emerges from the bottom, with a slightly wider outlet than the diameter of the pipe. The modelled pipe is represented by a straight shape, discharging at the flat modelled floor. Moreover, the velocity was set to be the same over the entire section of the pipe, dividing roughly the incoming flow to the pipe by the cross section. However, a velocity profile occurs in pipe due to friction. Furthermore, the velocity profile might not be fully established in reality when the jet enters into the reservoir due to the closely located 90°-bend.



Figure VIII-1: Comparison between the real pipe and its modelling

- The surface tension between wall and fluid was ignored (by setting a *zeroGradient* in boundaries for each alpha field)
- The impact of temperature was not considered and this thesis uses a uniform temperature in the tank of 20°. It might not be true that the average temperature in Sweden will reach 20°. Even more important, Nordblom (2004, pp71-72) states that “thermal stratification can have a significant impact on the mixing conditions in drinking water storage tanks. [...] The [study of] tanks with denser inflow at the bottom showed that the risk of stagnation at the water

surface increased with increasing depths and temperatures of the stored water.”

- Pure water was considered in this study. However, since the concentration of chlorine will differ between existing water and incoming water, gradients in concentration of chlorine from the incoming water to the existing water may occur. One can assess that the mixing induced by the chemical process is dependant of the hydraulic one.

Another major limitation to the study was time. The entire time dedicated for this study was only five full months. Time was required first to grasp the world of Linux, with its huge capabilities, but also nebulous interface console. Time was required to understand the software OpenFoam. The basic meshing tool *blockMesh* was not straightforward enough to handle cylinder revolution as needed for the real reservoir. A solution provided by *snappyHexMesh* was to import the design from a CAD software. The software Salome was for the first time experienced to provide such a design file. Uploading version to resolve some features regarding the mesh with the tool *snappyHexMesh* has counterbalanced with incompatibility of the mesh constructed in the last version. When simple results were obtained, biased opinion would say that post-processing with an unclear ParaView was time consuming. When study case of the real reservoir ended up with almost 200'000 cells to deal with; it appeared that the one-processor computer provided by the department would not be sufficient. A solution was to access Platon, the cluster at Lund University. After applying for a project there, and finally getting access to it; submitted jobs on the cluster following different methods were more or less quite rapidly executed. Indeed, a combination of few processors required to process in parallel and a small wall time made submitted jobs end up fairly high in the queuing list.

Even if time was not enough to carry out all originally intended tasks in this thesis, everything was hopefully written and archived in order to be continued by someone else afterwards, if more modelling are required for the studied and similar cases.

IX. Further studies

At the time when this study was conducted, the issue to perform further investigations about the reservoirs in Norra Ugglarps was raised. As previously said, hopefully this thesis will be clear enough to provide useful information to anyone that intends to pursue this work; and the author of this thesis would be gladly contacted for further details. This section will detail further tasks that can be carried out as follow-ups to the present study.

Focus in this thesis was on the filling up on the larger tank. Modelling may be improved by increasing the mesh grid, and thus the accuracy of the results. Refinements around the pipe might be employed for further investigation of the turbulence at the base. Considering the bend just upstream the inlet of the tank and the associated velocity profile would alter the jet shape and somewhat influence the mixing.

Different scenarios regarding the filling of the reservoirs (i.e., by selecting different set points and water level ranges) seem to be a good idea to compare mixing in different situations in order to provide Sydsvatten with the best alternatives to ameliorate mixing. As well as filling, draining are interesting cases to study. May be the water will take different paths flowing into and out from the reservoir, creating different water circulation during filling up and emptying of the tank. Field measurements, although difficult to implement and expensive, are required to validate numerical models accuracy.

At present, both reservoirs are functioning together as a single system, and modelling on the second reservoir might not be necessary. After first modelling the larger reservoir, efforts should be made to ensure sufficient renewal of the water in it. Both reservoirs may be considered to act as a single system, since they are connected and the volume of the larger one (18330 m^3) is about six times larger than the smaller one (3000 m^3). Thus, such efforts might be sufficient to also ensure mixing in the second reservoir. Moreover, it should be remembered that the smaller reservoir has only one pipe but two inlets/outlets: one perpendicular to the bottom directed upwards and another one tangential to the diameter. This design must be more efficient considering mixing.

However, even though further modelling may be required to develop a basis for improving the mixing in the Norra Ugglarp reservoirs, the following section will attempt to provide some solutions.

X. SOLUTIONS TO PROVIDE MORE MIXING

As shown previously, reservoirs tend to suffer from poor mixing. In the following section some solutions to tackle this issue will be discussed.

X.1 Implementing an upper pipe

The presence of a single inlet/outlet at the base of each tank is to the major part responsible for the poor mixing. A solution could be the implementation of an encompassing inlet pipe at the top of the reservoir. Those pipes generally end up with either a bend or a swan neck duct. This new pipe connected to the actual inlet/outlet pipe would then provide water in the reservoir exclusively from the top. The original pipe will be used further on only to draw water out of the reservoir. To be efficient this method requires also the installation of a check valve (also known as non-return valve or one-way valve) to force water to be only transported up through the new inlet pipe, and to be evacuated from the reservoir through the original outlet pipe.

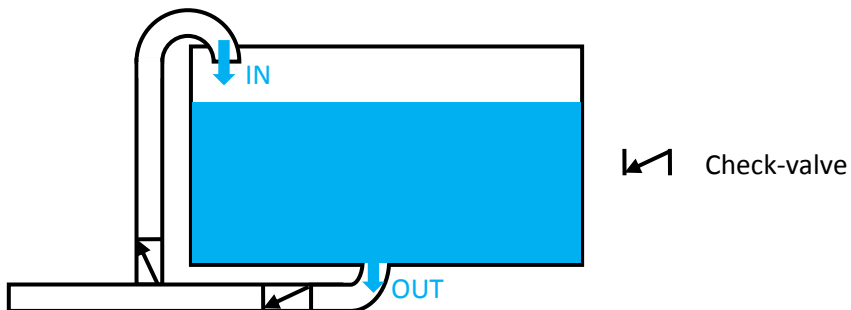


Figure X-1: Implementing a new pipe in reservoirs

This solution will improve the mixing for many reasons. First, the incoming water will be supplied to the upper layer in the reservoir, pushing the water already there towards the outlet, creating a natural turnover. Also, the waterfall induced by the upper inlet will develop some whirlpool that will contribute to enhance the mixing.

This solution still has some inconveniences. Technical issues will arise when it comes to connect the new inlet pipe to the existing system. The actual pipe is 700mm diameter large, and substantial work will be required. This solution also includes a check valve, and it might be complicated to introduce it in the existing

system. At last but not least, mastering those technical issues will require large investment from the company.

X.2 Increasing water level range

Another solution, which requires no technical modifications and no expenses, would be changing the actual water level range. As seen previously, precaution imposes that reservoirs are maintained to a quite high level of water, using a small range in water level. A solution would be to increase this range, or more precisely decrease the low set point. Reservoirs would drain more before Vombverket would start filling them up. (Cf. figure X-2 below)

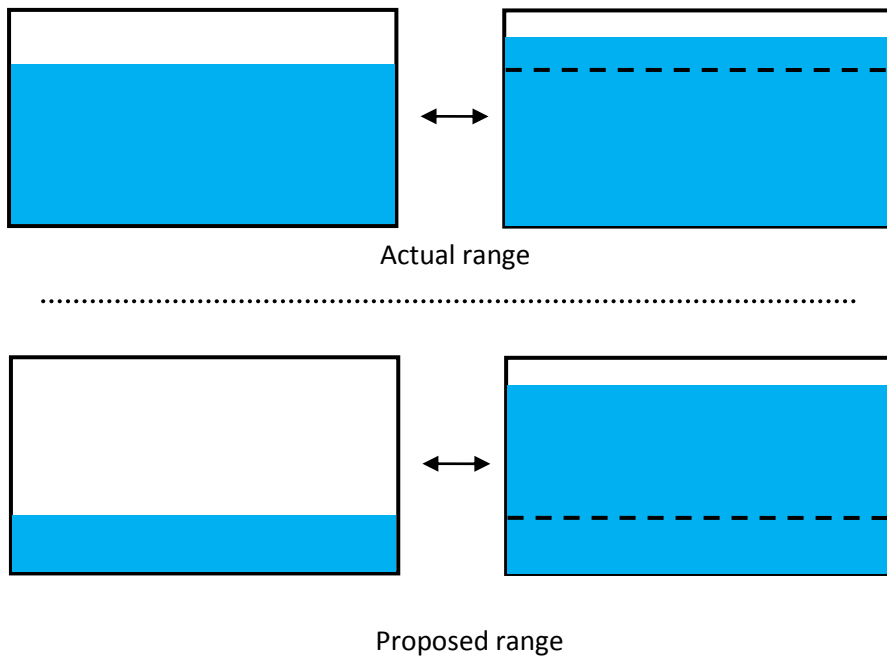


Figure X-2: Increase the water level range

The volume above the dashed line is the volume of new water coming into the reservoir. It is clear from the sketch above that the mixing will be better in the second case. More than half of the volume is renewed and hopefully this large volume incoming is enough to mix it completely with the aged water.

Further simulations could be done to test several low set points, and define from which starting water level in the tank mixing conditions are satisfactory.

This obvious and simple method just requires changing the monitoring of the starting and the filling up. However, this solution is still subject to debate since, for security reasons, decreasing the start set point will decrease the reserve storage in case of problems in Vombverket or network breakdown. The optimist opinion would conclude that the security concern is not a problem since:

1) Vombverket is working through several lines so there are few chances that a general breakdown affects all of them. Electric deficiency is counteracted by a powerful generators.

2) According to Sydvatten (*Fresh Drinking Water*, 2010) “The water mains network is comprised to a large extent of dual pipes”, which reduces the threat of a network breakdown possibility.

X.3 Regulating the reservoirs

Another solution would be to connect the existing valves to the actual filling and drawing. A simplified sketch of the actual presence of closing valves at the site of the reservoirs is presented in figure X-3. (NB: only the valves of interest have been presented here, but there are other valves as well located on the DN 900-pipe).

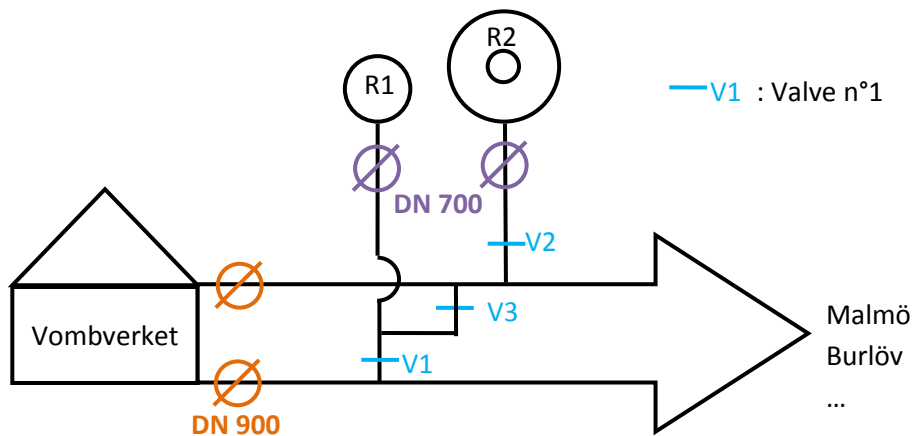


Figure X-3: Connecting the reservoirs

Currently, the two reservoirs are connected (through the pipe where the valve V3 is located) and functioning as if they were a single tank. Indeed, when excess water from Vombverket is not consumed and goes towards the reservoirs, even if some difference in roughness or friction would make that one reservoir fill faster than the other at the beginning, they would equilibrate because of this connection pipe.

The idea of regulating and monitoring those three valves bring many scenarios of utilization that could improve mixing. We could first of all disconnect the two reservoirs by closing V3. Then, the only way to fill up the first reservoir is by the pipe which is situated at V1, and the second one by the pipe at V2.

Thus, to ameliorate mixing many scenarios could be used:

- Considering excess water going towards the reservoirs, a solution could be to close one reservoir while filling up the first one, and then close the first one and start filling up the second. All the flow, nowadays divided into both reservoirs, will go into a single reservoir. It will increase the inlet flow into the reservoir; the higher the flow is into the reservoir, higher the mixing of the water will be.
- Following the same idea, one could start filling one reservoir, and then opening the connection valve V2 to play with the “communicating vessels” laws. Water will flow with high velocity from the full reservoir to the empty one; ensuring a high mixing in the latter reservoir.
- Same ideas can apply when draining the reservoirs: starting with the reservoirs disconnected, one reservoir can be drained first, then the second one drained; or even drain one and then equilibrate them again.

Multiple scenarios can be considered and all of them should be defined through the monitoring system to automate the actions needed, which imply regulating the flow through to the connected valves.

NB: Closing and opening valves should be done quite slowly to avoid water hammer and protect the pumps in Vombverket.

CONCLUSION

In summary, initially simulations were done in 2D and 3D; first, for rectangular cases, to obtain some typical flow patterns in tanks and gain general experience with the flow modelling, before carrying out modelling of the real tanks. Although the first cases followed each other quite well, moving to the real tank was arduous work, especially getting an appropriate mesh. A satisfactory mesh required over 200'000 cells, and it was realized that computations would take considerable time to obtain full scenario results. However, from all the cases carried out of the flow in water storage tanks, it is possible to list the following conclusions.

The incoming velocity is a key parameter regarding mixing. Indeed, the magnitude of the jet velocity should be strong enough to be able to propel fresh water across the entire water depth of the tank. The water level range is another important parameter to consider. For security reason, water systems keep reservoir levels high, using a small water level range.

Conclusions tend to demonstrate that using a small water level range is not beneficial for the mixing since a small volume of the tank is renewed. A compromise must be found between security and mixing of the water in the tank.

According to the literature review, another major parameter – not studied in this thesis – is temperature. Stratification could occur in tank, preventing exchange of water between different temperature layers. Moreover, a difference in temperature between the incoming water and existing one would induce some density movement. As the inflow is colder – i.e., denser than the water already present in the tank, this incoming colder water would tend to stay at the bottom of the tank. Reverse effects would occur for a less dense injection.

All conclusions tend to show that the reservoirs at Norra Ugglarp suffer from insufficient mixing, mainly due to both the weak incoming velocity of the fresh water jet and the limited water level range in use.

To tackle those mixing problems, some solutions have been suggested. Adding an external inlet pipe discharging at the top of the reservoir seems a relevant solution to greatly ameliorate mixing in the tank. Water will then be forced to enter in the top layer of the reservoir, and drain down to the existing outlet pipe making a 'natural' turnover. However, this solution may be technically demanding. On the contrary, increasing the water level range would be a solution to improve the mixing implying small costs, but it could potentially cause a more hazardous situation regarding the security storage. Monitoring access valves to reservoirs would allow mixing by many possibilities. For instance, reservoirs could be isolated and the flow coming from Vombverket directed to one single reservoir. Thus, higher inflow into a reservoir would be obtained, causing a higher mixing rate.

REFERENCES

Copyright© Julien Jaunâtre, 2013: Can be contact at jj jaunatre@live.fr

Report no: TVVR 13/5010

All photographs are taken by the author unless anything else is stated.

Harvard System of Referencing was used to provide detailed references. (Anglia Ruskin University, 2012)

Agnani ,E. 2012. *Welcome to snappyWiki!* [online] Available at : <https://sites.google.com/site/snappywiki/home> [Accessed May 2013]

Anglia Ruskin University, 2012. *Harvard system* [online] Available at: <http://libweb.anglia.ac.uk/referencing/harvard.htm> [Accessed May 2013]

CFD Online, 2007. *Standard k-epsilon model* [Online] Available at : http://www.cfd-online.com/Wiki/Standard_k-epsilon_model [Accessed February 2013]

CFD Online, 2009. *How to create patch after snappyHexMesh.* [online] Available at: <http://www.cfd-online.com/Forums/openfoam-meshing-utilities/61510-how-create-patch-after-snappyhexmesh.html> [Accessed Mars 2013]

Courant, R.; Friedrichs, K.; Lewy, H. (1928), *Über die partiellen Differenzengleichungen der mathematischen Physik* [online] English version available at: <http://www.archive.org/stream/onpartialdiffere00cour#page/n0/mode/2up> [Accessed June 2013]

Encyclopaedia Wikipedia, *Maquette en hydraulique* {in French} [online] Available at: http://fr.wikipedia.org/wiki/Maquette#En_hydraulique [Accessed June 2013]

Encyclopedia Wikipedia,2013. *Entry: Check-valve* [online] Wikipedia: the free encyclopedia [access May 2013] Available at: <http://en.wikipedia.org/>

Equations http://www.tfd.chalmers.se/~hani/kurser/OS_CFD/highLevelProgramming.pdf

ESI Group, 2013. *First OpenFOAM User Conference* [online] Available at: <http://www.esi-group.com/corporate/news-media/press-releases/2013-english-pr/first-openfoam-user-conference> [Accessed Mars 2013]

Hemida, H., 2013. *OpenFOAM tutorial: Free surface tutorial using interFoam and rasInterFoam – Chalmers university of Technology, Göteborg* [pdf] Available at: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2007/HassanHemida/Hassan_Hemida_VOF.pdf [Accessed Mars 2013]

Kitware, 2013. *ParaView application* [online] Available at : <http://www.paraview.org/> [Accessed June 2013]

Lemke A. and DeBoer D.E., 2012. *Effect of storage tank mixing on water quality* [pdf] Available at: <http://www.sdarws.com/PDF/SDRWRC/ImpactsofMixingonStorageTankWaterQuality.pdf> [Accessed April 2013]

Lunarc, 2013. *Center for scientific and technical computing at Lund University* [online] Available at: <http://www.lunarc.lu.se/> [Accessed June 2013]

Lunarc, 2013. *Center for scientific and technical computing at Lund University* [online] Available at: <http://www.lunarc.lu.se/lunarc/Systems/PlatonDetails> [Accessed June 2013]

Martinez-Solano et al., 2010. Modelling flow and concentration field in rectangular water tanks. *International Environmental Modelling and Software Society*, February 2010 [pdf] Available at (shorten): <http://www.iemss.org/iemss2010/papers/S02/S.02.11>. /<add title> [Accessed May 2013]

Mays,L.W, 2000. *Water distribution systems handbook*. New York: McGraw-Hill

Nordblom, O., 2004. *Mixing and Stagnation in Drinking water storage tanks* [manuscript] Göteborg: Chalmers University of Technology

Nordblom, O., Bergdahl, L., 2004. Initiation of Stagnation in Drinking Water Storage Tanks. *Journal of Hydraulic Engineering*, January 2004, p.49

Open CASCADE, 2013. *Salome: the Open Source Intergration Platform for Numerical Solution* [online] <http://www.salome-platform.org/> [Accessed Mars 2013]

OpenCFD Ltd (ESI Group), 2013. *OpenFoam: C++ documentation*, Mars 2013 [online] <http://www.openfoam.org/docs/cpp/> tabs Directory/interMixingFoam [Accessed June 2013]

OpenCFD Ltd (ESI Group), 2013. *OpenFoam: Chapter 2 Tutorials* [online] Available at: <http://www.openfoam.org/docs/user/tutorials.php> [Accessed January 2013]

OpenCFD Ltd (ESI Group), 2013. *OpenFoam: Chapter 6 Post-processing* [online] Available at: <http://www.openfoam.org/docs/user/paraview.php#x30-1780006.1> [Accessed Febuary 2013]

OpenCFD Ltd (ESI Group), 2013. *OpenFoam: The open source CFD toolbox*. [online] Available at: <http://www.openfoam.com/> [Access January 2013]

OpenCFD Ltd (ESI Group), 6th Mars 2013. *ESI-OpenCFD releases OpenFOAM® 2.2.0* [online] Available at: <http://www.openfoam.org/version2.2.0/> [Accessed Mars 2013]

OpenWiki, 2009. Entry: *The PISO algorithm in OpenFOAM* [online] Available at : http://openfoamwiki.net/index.php/The_PISO_algorithm_in_OpenFOAM [Accessed Feb 2013]

OpenWiki, 2009. Entry: *The SIMPLE algorithm in OpenFOAM* [online] Available at : http://openfoamwiki.net/index.php/The_SIMPLE_algorithm_in_OpenFOAM [Accessed Feb 2013]

Rodi, W., 1993. *Turbulence Models and Their Application in Hydraulics: A state-of-art-review*. IAHR Division Section on Fundamentals, Balkema, Rotterdam. [pdf] Available at: <http://cdlab2.fluid.tuwien.ac.at/LEHRE/TURB/Rodi1-46.pdf> [Access February 2013]

SiteDuZéro, 2013: *Reprenez le contrôle à l'aide de Linux* {in French} [online] Available at : <http://www.siteduzero.com/informatique/tutoriels/reprenez-le-controle-a-l-aide-de-linux> [Accessed January 2013]

Sydvatten, 2010. *Fresh Drinking Water* [pdf] Available at: [http://www.drickkranvatten.se/\\$2/file/sydvatten-engelska.pdf](http://www.drickkranvatten.se/$2/file/sydvatten-engelska.pdf) [Accessed May 2013]

Sydvatten, 2012. *Årsredovisning 2012* [manuscript] [http://www.drickkranvatten.se/\\$2/file/sydvatten-ar-2012-web-72dpi.pdf](http://www.drickkranvatten.se/$2/file/sydvatten-ar-2012-web-72dpi.pdf) [Accessed May 2013]

Appendix I – Calculation for defining scenario of study

Since each tank has its own design and volume, one needs to know the flow that goes into each tank. Since no flow meters are present on the reservoir site, some calculations were made to estimate the flow into each tank. The mean flow that goes towards both reservoirs is calculated as an average through the time of filling of the difference between the production and the consumption. $Q_{\text{towards both reservoirs}} = \text{average} (Q_{\text{prod}} - Q_{\text{cons}}) = 69 \text{ l/s}$.

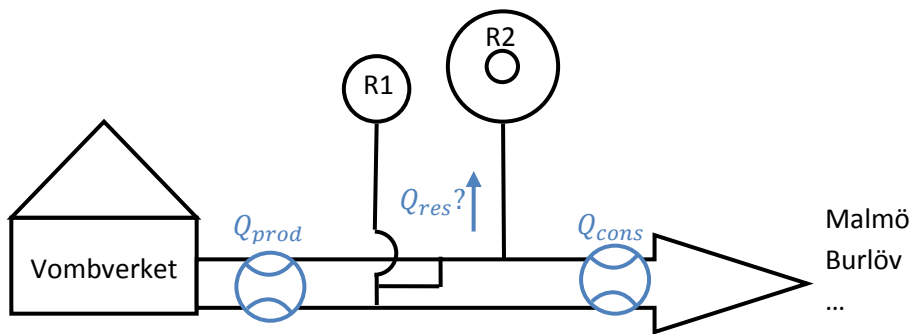


Figure AI-1: Simplified diagram of the network configuration

Knowing the cross section of each tank, and regarding the monitoring of their water level through time, one could estimate roughly the flow into each tank.

- The cross section of reservoir 1 is a disc of 18m diameter. So the area is: $A_{R1} = \frac{\pi \cdot \varnothing^2}{4} = \frac{\pi \cdot 18^2}{4} = 254 \text{ m}^2$. For the case of study, the monitor level rise from 76,28m to 77,12m, water level increase of $\Delta N_1 = 77,12 - 76,28 = 0,84\text{m}$ that is a corresponding volume of $V_{R1} = \Delta N_1 * A_{R1} = 0,84 * 254 = 213 \text{ m}^3$.

Since the filling of the tank is performed between 7:10 to 13:10, that is $\Delta T = 13:30 - 7:10 = 6,33\text{h}$, the flow into reservoir 1 is $Q_{in R1} = \frac{V_{R1}}{\Delta T} = \frac{213}{6,33}$

The flow into the reservoir 1 is $Q_{in R1} = 36\text{m}^3/\text{h} = 10\text{l/s}$.

- The cross section of the second reservoir is a disk with a whole in the middle (tower access to the top of the reservoir) so the area is $A_{R2} = \frac{\pi \cdot \varnothing^2}{4} - \frac{\pi \cdot \varnothing^2}{4} = \frac{\pi \cdot 46^2}{4} - \frac{\pi \cdot 4,4^2}{4} = 1646 \text{ m}^2$.

Following the same reasoning:

- the water level rise is $\Delta N_2 = 77,10 - 76,26 = 0,84m$
- the corresponding volume $V_{R2} = \Delta N_2 * A_{R2} = 0,84 * 1646 = 1383 m^3$
- the flow into the reservoir 2 is $Q_{in R2} = \frac{V_{R2}}{\Delta T} = \frac{1383}{6,33}$

Finally the flow into the reservoir 2 is $Q_{in R2} = 218m^3/h = 61l/s$.

One can notice that $Q_{in R1} + Q_{in R2} = 71l/s \approx Q_{average} = 69l/s$.

The model OpenFOAM requires the velocity at the inlet of each reservoir. Since the diameter of the inlet pipe for both reservoirs has a Diameter Nominal of 700mm, the inlet velocity is

- $v_{inlet R1} = \frac{Q_{in R1}}{A_{pipe}} = \frac{0,010}{0,385} = 0,03 m/s$
- $v_{inlet R2} = \frac{Q_{in R2}}{A_{pipe}} = \frac{0,061}{0,385} = 0,16 m/s$

Appendix II- Code for the tank case (Norra Ugglarps)

NB: to gain in visibility, standard headers provided by OpenFoam are removed from the code

```
// ** *0/u ** * //  
  
dimensions [0 1 -1 0 0 0 0];  
  
internalField uniform (0 0 0);  
  
boundaryField  
{  
  ext  
  {  
    type    fixedValue;  
    value   uniform (0 0 0);  
  }  
  int  
  {  
    type    fixedValue;  
    value   uniform (0 0 0);  
  }  
  base  
  {  
    type    fixedValue;  
    value   uniform (0 0 0);  
  }  
  
  atm  
  {  
    type    pressureInletOutletVelocity;  
    value   uniform (0 0 0);  
  }  
  
  pipe  
  {  
    type    fixedValue;  
    value   uniform (0 0 0.16);  
  }  
}  
  
// ***** //
```

```
// ** *0/p_rgh ** * //  
  
dimensions [1 -1 -2 0 0 0 0];  
  
internalField uniform 0;  
  
boundaryField  
{  
  ext  
  {  
    type    buoyantPressure;  
    value   uniform 0;  
  }  
  
  int  
  {  
    type    buoyantPressure;  
    value   uniform 0;  
  }  
  
  base  
  {  
    type    buoyantPressure;  
    value   uniform 0;  
  }  
  
  atm  
  {  
    type    totalPressure;  
    p0      uniform 0;  
    U       U;  
    phi     phi;  
    rho     rho;  
    psi     none;  
    gamma   1;  
    value   uniform 0;  
  }  
  
  pipe  
  {  
    type    buoyantPressure;  
    value   uniform 0;  
  }  
  
  defaultFaces  
  {  
    type    empty;  
  }  
}  
  
// ***** //
```

```

// *** 0/alpha1 *** //      //air phase
dimensions [0 0 0 0 0 0];
internalField uniform 0;

boundaryField
{
  ext
  {
    type zeroGradient;
  }

  int
  {
    type zeroGradient;
  }

  base
  {
    type zeroGradient;
  }

  pipe
  {
    type zeroGradient;
  }

  atm
  {
    type inletOutlet;
    inletValue uniform 1;
    value uniform 1;
  }

  defaultFaces
  {
    type empty;
  }
}

// ***** //

```

```

// *** 0/alpha2 *** //      //aged-water phase
dimensions [0 0 0 0 0 0];
internalField uniform 0;

boundaryField
{
  ext
  {
    type zeroGradient;
  }

  int
  {
    type zeroGradient;
  }

  base
  {
    type zeroGradient;
  }

  atm
  {
    type inletOutlet;
    inletValue uniform 0;
    value uniform 0;
  }

  pipe
  {
    type zeroGradient;
  }

  defaultFaces
  {
    type empty;
  }
}

// ***** //

```

```

// *** 0/alpha3*** //      //Fresh water phase
dimensions [0 0 0 0 0 0];

internalField uniform 0;

boundaryField
{
  ext
  {
    type    zeroGradient;
  }

  int
  {
    type    zeroGradient;
  }

  base
  {
    type    zeroGradient;
  }

  atm
  {
    type    inletOutlet;
    inletValue    uniform 0;
    value    uniform 0;
  }

  pipe
  {
    type    inletOutlet;
    inletValue    uniform 1;
    value    uniform 1;
  }

  defaultFaces
  {
    type    empty;
  }
}

// ***** //

// *** 3/polymesh/boundary*** //
//Created after snappyHexMesh and autoPatch 75 to
detect boundaries//

5
(
  atm
  {
    type    patch;
    nFaces    26280;
    startFace    539148;
  }
  pipe
  {
    type    patch;
    nFaces    36;
    startFace    565428;
  }
  ext
  {
    type    wall;
    nFaces    22880;
    startFace    565464;
  }
  int
  {
    type    wall;
    nFaces    2288;
    startFace    588344;
  }
  base
  {
    type    wall;
    nFaces    26288;
    startFace    590632;
  }
}
)

// ***** //

```

```

// * * * constant/polyMesh/blockMeshDict * * * // // * * * constant/dynamicMeshDict * * * //

//generate a hexahedral block requires for using the dynamicFvMesh staticFvMesh;
tool snappyHexMesh //

convertToMeters 1; // ***** //

a 50;
b 50;
c 13;
yi -25;
yf 25;
zi -2;
zf 11;
xi -25;
xf 25;

// * * * constant/g * * * //

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

// ***** //

(
($xi $yi $zi) //0
($xf $yi $zi) //1
($xf $yf $zi) //2
($xi $yf $zi) //3
($xi $yi $zf) //4
($xf $yi $zf) //5
($xf $yf $zf) //6
($xi $yf $zf) //7
);

// others files in constant/polyMesh are created
automatically mean creating the mesh

blocks
(
hex (0 1 2 3 4 5 6 7) ($a $b $c) simpleGrading (1 1 1)
);

//constant/triSurface folder should contain the .STL file
from your CAD file to generate the mesh via the tool
snappyHexMesh

edges
(
);

patches
(
);

mergePatchPairs
(
);

// ***** //

```

```

// *** constant/transportProperties *** //

// Air
phase1
{
    transportModel Newtonian;
    nu      nu [0 2 -1 0 0 0] 1.48e-05;
    rho     rho [1 -3 0 0 0 0] 1;
}

// Old Water
phase2
{
    transportModel Newtonian;
    nu      nu [0 2 -1 0 0 0] 1e-6;
    rho     rho [1 -3 0 0 0 0] 1000;
}

// New Water
phase3
{
    transportModel Newtonian;
    nu      nu [0 2 -1 0 0 0] 1e-6;
    rho     rho [1 -3 0 0 0 0] 1000;
}

// Surface tension coefficients
sigma12    sigma12 [1 0 -2 0 0 0] 0.05;
sigma13    sigma13 [1 0 -2 0 0 0] 0.05;

// Diffusivity between miscible phases
D23       D23 [0 2 -1 0 0 0] 3e-09;

// ***** //

```

```

// *** constant/turbulenceProperties *** //

turbulenceModel kEpsilon;
turbulence off;

laminarCoeffs
{
}

kEpsilonCoeffs
{
    Cmu Cmu [0 0 0 0 0 0] 0.09;
    C1 C1 [0 0 0 0 0 0] 1.44;
    C2 C2 [0 0 0 0 0 0] 1.92;
    alphaEps alphaEps [0 0 0 0 0 0] 0.76923;
}

// ***** //

// *** system/decomposeParDict***** //

numberOfSubdomains 16;

method    scotch; //automatic decomposition

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

hierarchicalCoeffs
{
    n      ( 1 1 1 );
    delta  0.001;
    order  xyz;
}

manualCoeffs
{
    dataFile  "";
}

distributed no;

roots      ( );

// ***** //

```

```
// *** system/controlDict*** //
```

```
application    interMixingFoam;
```

```
startFrom      latestTime;
```

```
startTime      0;
```

```
stopAt         endTime;
```

```
endTime        3600;
```

```
deltaT         0.001;
```

```
writeControl   runTime;
```

```
writeInterval  0.1;
```

```
purgeWrite     0;
```

```
writeFormat    ascii;
```

```
writePrecision 6;
```

```
writeCompression compressed;
```

```
timeFormat     general;
```

```
timePrecision  6;
```

```
runTimeModifiable yes;
```

```
adjustTimeStep no;
```

```
maxCo          0.5;
```

```
maxAlphaCo     0.5;
```

```
maxDeltaT      0.001;
```

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

```
// *** system/fvSchemes *** //
```

```
ddtSchemes
```

```
{  
    default    Euler;  
}
```

```
gradSchemes
```

```
{  
    default    Gauss linear;  
}
```

```
divSchemes
```

```
{  
    div(rho*phi,U) Gauss limitedLinearV 1;  
    div(phi,alpha) Gauss vanLeer;  
    div(phiIrb,alpha) Gauss interfaceCompression;  
    div((muEff*dev(T(grad(U)))) Gauss linear;  
    div((nuEff*dev(T(grad(U)))) Gauss linear;  
}
```

```
laplacianSchemes
```

```
{  
    default    Gauss linear corrected;  
}
```

```
interpolationSchemes
```

```
{  
    default    linear;  
}
```

```
snGradSchemes
```

```
{  
    default    corrected;  
}
```

```
fluxRequired
```

```
{  
    default    no;  
    p_rgh;  
    pcorr;  
    "alpha.>";  
}
```

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

```
// *** system/fvSolutions*** //
```

```
solvers  
{  
  "alpha."  
  {  
    solver      smoothSolver;  
    smoother    GaussSeidel;  
    tolerance    1e-06;  
    relTol      0;  
    nSweeps     1;  
  }  
}
```

```
pcorr  
{  
  solver      PCG;  
  preconditioner DIC;  
  tolerance    1e-10;  
  relTol      0;  
}
```

```
p_rgh  
{  
  solver      PCG;  
  preconditioner DIC;  
  tolerance    1e-07;  
  relTol      0.05;  
}
```

```
p_rghFinal  
{  
  solver      PCG;  
  preconditioner DIC;  
  tolerance    1e-07;  
  relTol      0;  
}
```

```
U  
{  
  solver      PBICG;  
  preconditioner DILU;  
  tolerance    1e-06;  
  relTol      0;  
}
```

```
PIMPLE  
{  
  momentumPredictor no;  
  nCorrectors 3;  
  nNonOrthogonalCorrectors 0;  
  nAlphaCorr 1;  
  nAlphaSubCycles 2;  
  cAlpha 1;  
}
```

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

```
// *** system/setFieldsDict *** //
```

```
defaultFieldValues  
(  
  volScalarFieldValue alpha1 1  
  volScalarFieldValue alpha2 0  
  volScalarFieldValue alpha3 0  
  volVectorFieldValue U (0 0 0)  
);
```

```
regions  
(  
  boxToCell //alpha2  
  {  
    box (-25 -25 0) (25 25 9);  
    fieldValues  
    (  
      volScalarFieldValue alpha1 0  
      volScalarFieldValue alpha2 1  
      volScalarFieldValue alpha3 0  
    );  
  }  
)
```

```
  boxToCell //alpha3  
  {  
    box (-25 -25 -2) (25 25 0);  
    fieldValues  
    (  
      volScalarFieldValue alpha1 0  
      volScalarFieldValue alpha2 0  
      volScalarFieldValue alpha3 1  
    );  
  }  
);
```

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

```

// *** system/snappyHexMeshDict* ***
// *****
//Dictionary for the tool snappyHexMesh
//read left column before (2 pages) followed by the
right column
//addLayers + advanced setting + mesh controls quality
were not used and lets as default
//full example version online on tutorial/mesh/snappy

// Which of the steps to run
castellatedMesh true;
snap true;
addLayers false;

// Geometry. Definition of all surfaces. All surfaces are
of class
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell intersecting
it
// - to specify refinement for any mesh cell
inside/outside/near
// - to 'snap' the mesh boundary to the surface
geometry
{
    tank5.stl
    {
        type triSurfaceMesh;
        name cylinder;
    }

    //- Refine a bit extra around the small centre hole
// refineHole//commented by Julien can be canceled
// {
//     type searchableSphere;
//     centre (0 0 -0.012);
//     radius 0.003;
// }
};

// Settings for the castellatedMesh generation.
castellatedMeshControls
{
    // Refinement parameters
    // ~~~~~

    // If local number of cells is >= maxLocalCells on any
processor
// switches from from refinement followed by
balancing
// (current method) to (weighted) balancing before
refinement.
    maxLocalCells 100000;

    // Overall cell limit (approximately). Refinement will
stop immediately
}

{
    cylinder
    {
        // Surface-wise min and max refinement level
        level (2 2);
    }
}

resolveFeatureAngle 30;

// Region-wise refinement
// ~~~~~

// Specifies refinement level for cells in relation to a
surface. One of
// three modes
// - distance. 'levels' specifies per distance to the
surface the
// wanted refinement level. The distances need to
be specified in
// descending order.
// - inside. 'levels' is only one entry and only the level
is used. All
// cells inside the surface get refined up to the level.
The surface
// needs to be closed for this to be possible.
// - outside. Same but cells outside.

refinementRegions
{
    // refineHole // commented by Julien
    // {
    //     mode inside;
    //     levels ((1E15 3));
    // }
}

// Mesh selection
// ~~~~~

// After refinement patches get added for all
refinementSurfaces and
// all cells intersecting the surfaces get put into these
patches. The
// section reachable from the locationInMesh is kept.
// NOTE: This point should never be on a face, always
inside a cell, even
// after refinement.
// This is an outside point locationInMesh (-0.033 -
0.033 0.0033);
    locationInMesh (5 5 10); // Inside point

// Whether any faceZones (as specified in the
refinementSurfaces)
// are only on the boundary of corresponding
cellZones or also allow
// free-standing zone faces. Not used if there are no

```



```

// upon reaching this number so a refinement level
might not complete.
// Note that this is the number of cells before
removing the part which
// is not 'visible' from the keepPoint. The final
number of cells might
// actually be a lot less.
maxGlobalCells 2000000;

// The surface refinement loop might spend lots of
iterations refining just a
// few cells. This setting will cause refinement to stop
if <= minimumRefine
// are selected for refinement. Note: it will at least do
one iteration
// (unless the number of cells to refine is 0)
minRefinementCells 0;

// Number of buffer layers between different levels.
// 1 means normal 2:1 refinement restriction, larger
means slower
// refinement.
nCellsBetweenLevels 1;

// Explicit feature edge refinement
// ~~~~~

// Specifies a level for any cell intersected by its
edges.
// This is a featureEdgeMesh, read from
constant/triSurface for now.
features
(
// {
//   file "flange.eMesh"; // Commented by Julien
//   level 0;
// }
);

// Surface based refinement
// ~~~~~

// Specifies two levels for every surface. The first is
the minimum level,
// every cell intersecting a surface gets refined up to
the minimum level.
// The second level is the maximum level. Cells that
'see' multiple
// intersections where the intersections make an
// angle > resolveFeatureAngle get refined up to the
maximum level.

refinementSurfaces
faceZones.
allowFreeStandingZoneFaces true;
}

// Settings for the snapping.
snapControls
{
// - Number of patch smoothing iterations before
finding correspondence
// to surface
nSmoothPatch 3;

// - Relative distance for points to be attracted by
surface feature point
// or edge. True distance is this factor times local
// maximum edge length.
tolerance 1.0;

// - Number of mesh displacement relaxation
iterations.
nSolverIter 300;

// - Maximum number of snapping relaxation
iterations. Should stop
// before upon reaching a correct mesh.
nRelaxIter 5;

// Feature snapping

// - Number of feature edge snapping iterations.
// Leave out altogether to disable.
nFeatureSnapIter 10;

// - Detect (geometric) features by sampling the
surface
implicitFeatureSnap true;

// - Use castellatedMeshControls::features
explicitFeatureSnap false;

// - Detect features between multiple surfaces
// (only for explicitFeatureSnap, default = false)
multiRegionFeatureSnap false;
}

[... Delete parts]
// ***** //

```

Appendix III- Submitting script for the cluster

```
#!/bin/sh

# Number of nodes needed
#PBS -l nodes=2:ppn=8          #note that one must use one single core for decomposePar and reconstructPar

# Estimated length of job
#PBS -l walltime=24:00:00

# Request that regular output (stdout) and
# terminal output (stderr) go to the same file
#PBS -j oe

# Send mail when job finishes
#PBS -m bea
#PBS -M xxx@student.lu.se

# Make sure that the needed software is available
#Load modules
. use_modules
module add gcc/4.6.2
module add openmpi/1.4.3/gcc/4.6.2
module add openfoam/2.1.0
. ${FOAM_INST_DIR}/OpenFOAM-2.1.0/etc/bashrc

##Go to the directory where the job was submitted
cd $PBS_O_WORKDIR #decommented by Julien : The software needs this command to find where one put the
directory to be executed
#
##Copy all files and subdirectories to the local directory
#mpiexec -pernode cp -pr * $PBS_O_LOCAL
#
##Go to the local directory
#cd $PBS_O_LOCAL #docom
#
# decommented by Julien: then the case will be run directly on the aforementioned directory
#results will be saved there also / and not being lost in the LOCAL directory in case of abortion

#Run the program
mpiexec interMixingFoam -parallel > log.interMixingFoam
#mpiexec decomposePar > log.decomposePar #use this line to decompose your project on the cluster: use only
one node
#mpiexec reconstructPar -latestTime > log.reconstruction # to reconstruct the very last time of your project

#JJ
##Copy any new files and directories to the submission directory
#mpiexec -pernode cp -pur $PBS_O_LOCAL/* $PBS_O_WORKDIR
#JJ
```