UNIVERSITEIT GENT

Faculteit Bio-ingenieurswetenschappen

Academiejaar 2011-2012

Spatio-temporal modeling of filter cake formation in membrane bioreactors

Michael Ghijs
Promotors: Prof. dr. ir. Ingmar Nopens & dr. ir. Jan Baetens
Tutor: ir. Wouter Naessens

Masterproef voorgedragen tot het behalen van de graad van
Master in de bio-ingenieurswetenschappen: Milieutechnologie

Ghent, June 6, 2014

The promoters,

Prof. dr. ir. Ingmar Nopens

dr. ir. Jan Baetens

The tutor,

The author,

ir. Wouter Naessens

Michael Ghijs

Dankwoord

Dit laatste jaar van mijn educationele carrière is er een geweest met een hoge werklast, niet in het minst door al het werk dat in een masterproef kruipt. Ik mag me echter gelukkig prijzen dat ik dit werk bij de vakgroepen van KERMIT en BIOMATH mocht uitvoeren: de gang van de 'Wiskundige modellering, Statistiek en Bioinformatica' bleek namelijk een gemoedelijke en stimulerende werkomgeving. Hiervoor wil ik alle leden van deze vakgroepen bedanken, en in het bijzonder de crew van 'the simulation lab', die voor een leuke sfeer zorgden en altijd klaar stonden voor hulp bij enige thesisproblemen. Ook nog een speciale vermelding voor Daan, dankzij wie ik OpenFOAM onder de knie gekregen heb, en omdat hij het thesiswerk op tijd en stond aanvulde met momenten van humoristische ontspanning. Ook wil ik Wouter nog bedanken voor de technische ondersteuning (zeker die in hectische tijden) en omdat hij een toffe buur was om naast te thesissen. Dat laatste geldt trouwens alle thesisstudenten van de vakgroep, zij konden zelfs van het tegen de deadline werken een 'aangename' tijd maken.

Ik wil ook van harte mijn beide promotoren bedanken; aan Ingmar voor het verschaffen van duidelijke richtlijnen voor het verloop van de thesis, en voor de goede raad en opbouwende kritiek; aan Jan voor het met raad en daad bijstaan tijdens deze masterproef, het verschaffen van kennis die anders veel minder vlot bekomen zou worden, en voor het vele werk dat ook hij in deze thesis gestoken heeft.

En last but definitely not least, bedank ik van harte mijn begeleider, Wouter. Eerst en vooral voor het aanbieden van een boeiend en innovatief thesisonderwerp, maar evenzeer voor de voortreffelijke begeleiding in de uitvoering ervan, aangezien zijn grondige kennis van het onderwerp, alsook zijn rustige en aangename persoonlijkheid een grote steun boden doorheen het hele thesisverhaal.

Bedankt aan allen!

Summary

The object of this dissertation is to determine and to implement a valid framework for a model that accurately describes filter cake formation in membrane bioreactors on a microscopic level, including the heterogeneous structure observed in reality.

First an introduction is given, in order to emphasize the importance of this research line of modeling, and the choice for the chosen Euler-Lagrangian framework is motivated in the literature study. This is followed by a detailed description of the theoretical basis of the model: a force balance that accounts for all the forces exerted on a particle by its surrounding fluid, with the profile of the latter being simulated by solving the Navier-Stokes equations. Subsequently, the implementation of this basis is described, wherein it was accounted for stability issues and flow regime-related correction factors. Also, a collision detection algorithm was constructed in order to model attachment to the membrane or an already formed filter cake. For this, a parameter was introduced that accounts for the influence of a particle's momentum regarding its probability of actually adhering to the cake layer.

Next, a study on a benchmark and a set of derived scenarios is performed in order to assess the influence on the simulated filter cake output of the model. The comparison of the simulated filter cakes was based on statistics such as their average thickness, standard deviation and accumulated number of particles. It is concluded from this results that the aforementioned parameter has a large influence on the nature of the filter cakes generated by the model, contrary to the flux, which, within the range of fluxes found in MBR, has only a small influence on the fouling regime. The results also emphasized that the model is still far from complete and needs some adjustments and extensions in order to fully serve its purpose. These are elaborately documented in the last chapter.

Samenvatting

Het doel van deze uiteenzetting is het bepalen en implementeren van een deuglijk kader voor een model dat een beschrijving zal geven van filterkoekvorming in membraanbioreactoren, die accuraat is tot op microscopisch niveau zodat de spatiaal heterogene structuur kan geobserveerd worden.

In het inleidend hoofdstuk in deze thesis wordt uitgelegd waarom deze manier om membraanvervuiling te modelleren een substantiële bijdrage kan leveren tot het begrip van de onderliggende processen van dit fenomeen. Vervolgens wordt de uiteindelijke keuze voor het Euler-Langrange-raamwerk gemotiveerd in de literatuurstudie. Dit wordt opgevolgd door een gedetaileerde beschrijving van de theoretische basis van het model: een krachtenbalans over het partikel waarin alle krachten die op een partikel, ondergedompeld in een vloeistof, inwerken. De vloeistofstroming wordt gesimuleerd door het oplossen van de Navier-Stokes-vergelijkingen. Vervolgens wordt de implementatie van deze theorie uit de doeken gedaan, waarbij gecorrigeerd moest worden voor het stromingsregime en voor het bewaren van de modelstabiliteit. Daarnaast werd ook nog een collisiedetectie-algoritme opgesteld dat de depositie van partikels op het membraan (of de reeds aanwezige filterkoek) moet modelleren. Hiervoor werd een parameter ingevoerd om de invloed van het moment van een partikel op zijn aanhechtingskans in rekenschap te brengen.

In het volgende deel worden een benchmark en een aantal gerelateerde scenario's bestudeerd, met als doel de invloed van de verschillende parameters op de modeloutput in kaart te brengen. Het vergelijken van deze gevalstudies gebeurde op basis van filterkoekstatistieken zoals de gemiddelde dikte, de standaardafwijking van de koekdikte en het aantal partikels dat in de koek vervat is. Uit de resultaten kon besloten worden dat de voorgenoemde aanhechtingsparameter en grote invloed heeft, en dat de flux over het membraan daarentegen slechts een kleine invloed toont op het vervuilingsregime. Ze benadrukten echter ook dat het model nog een lange weg te gaan heeft, om aan het doel van een fysisch accurate beschrijving van filterkoekvorming tegemoet te komen. De modelextensies en aanpassingen die het daarvoor nodig heeft worden uitgewerkt in het laatste hoofdstuk.

Contents

List of Symbols

A_{pcs}	projected cross-sectional area of the particle
c	grid cell
C_D	drag coefficient
D	tube inner diameter
d_{eq}	volume-equivalent diameter
d_p	particle diameter
Δt	discrete time step
Δx	resolution along the x-direction
Δy	resolution along the y-direction
Δz	resolution along the z-direction
$\Delta \ell$	cake thickness
ϵ	porosity
η_f	fluid dynamic viscosity
$\mathbf{F}_{am}$	added mass force
$\mathbf{F}_{Arch}$	Archimedes force
$\mathbf{F}_{body}$	body force
$\mathbf{F}_{drag}$	drag force
$\mathbf{F}_g$	gravity force
$\mathbf{F}_{hist}$	history force
$\mathbf{F}_{hydr}$	hydrodynamic force
$\mathbf{F}_{lift}$	lift force
$\mathbf{F}_p$	pressure force

$\mathbf{F}_{surf}$	force on the surface of an object
$\mathbf{g}$	gravitational acceleration vector
J	flux
k	parameter for determining the adhesion probability
κ	fluid velocity gradient
K_B	Basset kernel
K_{win}	window kernel
L	tube length
∇^2	Laplacian
m_p	particle mass
μ_f	fluid kinematic viscosity
n_p	local number of adhered particles
ω_f	curl of fluid velocity field
p	pressure
$\mathbf{p}_i$	position vector of particle i
ψ	kinematic pressure
R_c	filtration resistance of the filter cake
Re	Reynolds number
Re_p	particle Reynolds number
ρ_f	fluid density
ρ_p	particle density
R_m	filtration resistance of the membrane
R_p	additional filtration resistance of the membrane due to pore blocking

S	surface over volume ratio
T	temperature
t	time
t_{win}	time window for the calculation of the history force
$\mathbf{U}_c$	continuous phase velocity vector
$\mathbf{U}_p$	particle velocity vector
$\mathbf{U}_r$	particle relative velocity vector
$\mathbf{v}_i$	velocity vector of particle i
V_p	particle volume
$\mathbf{X}_p$	particle position vector

List of Abbreviations

ANN	artificial neural networks
BbM	biomass-based modeling
CA	cellular automaton
CAS	conventional activated sludge
CFD	computational fluid dynamics
CLSM	confocal laser scanning microscopy
COD	chemical oxygen demand
eEPS	extracted extracellular polymeric substances
EPS	extracellular polymeric substances
EPSc	carbohydrate content of extracellular polymeric substances
EPSp	protein content ofextracellular polymeric substances
FVM	finite volume method
HRT	hydraulic retention time
IbM	individual-based modeling
LMH	liters per square meter per hour
MBR	membrane bioreactor

PCA	principal component analysis
PDE	partial differential equation
PSD	particle size distribution
RIS	resistance-in-series models
SMP	soluble microbial products
SRT	sludge retention time
TMP	trans-membrane pressure
TSS	total suspended solids

List of Figures

List of Tables

CHAPTER 1
Problem statement, research objectives and outline

1.1 Introduction

Activated sludge systems have been around for a while, rendering good service in wastewater treatment, improving water quality at a low chemical cost. The application of a biological process to remove polluting substances, however, also requires that the biological agent, i.e. the sludge itself, still needs to be separated from the clean water. For this, several technologies are available, of which the traditional sedimentation tank and the membrane bioreactor (MBR) are the most widespread. The first is based on the gravitational settling of solids, whereas the second employs a permselective membrane to retain the biomass from the effluent. And while the configuration with a final settler remains the most commonly found practice, as it is even termed the conventional activated sludge system (CAS), the MBR's popularity has been ever increasing, with the MBR market showing a compound annual growth rate of 10.5 % (Kraume and Drews, 2010).

The core principle of this promising system is that the mixed liquor is sucked through the membrane, where anything too large to fit through its pores is retained. Pore sizes are usually situated between 0.01 and 0.4 μm, which fits the membranes into the ultra- and microfiltration classes, so that the produced permeate consistently achieves a high quality. Next to its robustness, the most prominent feature of MBRs is their compactness; dependent on the configuration, submerged or sidestream, the membrane is placed either inside or outside the biological reactor itself. Yet even if it is found in an adjacent tank, the membrane module takes up much less space than a sedimentation tank. Another trait is that the settleability of the mixed liquor is no longer a concern. This also opposes the constraint of the relatively low concentrations of sludge that are allowed and limiting in the conventional system. Due to hindered settling, meaning that the settling suffers from interactions among the sinking particles and from compression, sludge concentrations exceeding 6 g TSS/L are rarely found in CAS-systems. In MBRs this limit only depends on filtration characteristics, making that these systems can involve concentrations between 8 and 15 g TSS/L for municipal and up to 40 g TSS/L for industrial wastewater treatment (Rosenberger and Kraume, 2003). TSS represents the total suspended solids, it represents the suspension or solids concentration within the mixed liquor. Therefore, in retrofitting cases, i.e. in plants where the sedimentation tank setup has been replaced by a membrane module, the capacity of the treatment plant is substantially increased because of the higher wastewater loadings that it is able to process.

Other advantages of an MBR system in comparison to a CAS plant relate to its sludge retention time (SRT), which is not coupled to the hydraulic retention time (HRT). The latter is determined by the fact that a certain COD removal efficiency must be reached and that the wastewater needs to be processed adequately fast. The presence of difficult biodegradable COD or the use of slow-growing microorganisms could therefore impose some extra design and operating criteria. In MBR, however, removal efficiencies are governed by the membrane type and persistent COD is retained and broken down at the pace it requires. And since the HRT is disconnected from the SRT here, the sludge age can be regulated without HRT-related concerns. Among the perks following from this degree of freedom is the possibility to adjust the sludge age to benefit the slow-growing autotrophs, which in turn enhances the nitrogen removal rate.

Yet all of these features are coupled to a higher operational cost, a cost that just about comes down to the price tag of fouling abatement. Because even if the MBR is perceived as a reliable system, with the capability of handling varying influent loads without showing any sign of that in its consistent and high quality output characteristics, a look behind the scenes reveals an enduring and energy-consuming toil to maintain its steady outflow. Besides the liquid-solid separation being driven by pressure instead of gravitation, an MBR also needs frequent cleaning efforts of the membrane surface to ensure a sustainable operation. Indeed, the flux decline or pressure increase, dependent on the operational regime that is in effect, constant-pressure or constant-flux operation, would become unacceptable in a matter of minutes since the pressure acts on all suspended material as a driving force towards the membrane, giving rise to layers of organic material on the membrane surface, referred to as fouling. For its nature of hindering the filtration process, fouling needs to be kept under control, a process that requires continuous efforts; next to backwashing with clean permeate also aeration of the membrane and timely chemical cleaning are brought into play to keep the membrane in good shape. But, evidently, all these measures also give rise to a high energy demand. This manifests itself as a substantial share of the treatment cost per m^3 of wastewater treated, as more than 80 % of the operational expenses is energy-related (Judd, 2010). With an average energy consumption between 0.8 and $1.2\,\mathrm{kWh\,m^{-3}}$, this seems only natural, and certainly so when compared to the CAS ($\approx 0.5\,\mathrm{kWh\,m^{-3}}$) (Fenu et al., 2010).

Still, even with these fouling remediation actions in place, the membrane will deteriorate by irrecoverable fouling, necessitating membrane replacement after a certain amount of time;

some plants hold out for more than 10 years without this measure, but there are also a lot of them that do not even make it until 4 years (Wozniak, 2010). Considering that the cost of the membrane makes up about half of the total capital cost of the whole installation (Verrecht et al., 2010), the statement that fouling is the Achilles' heel of the MBR gains even more strength.

1.2 Problem Statement: insight in fouling is turbid

Commercial MBRs have been around since the late 1960s, yet to this day the dynamics of the most crucial process of the plant, i.e. filtration and the accompanying inevitable fouling build-up, are still not fully understood. The design of these installations has always been based on empirical knowledge rather than in-depth understanding of the underlying processes. In addition, also operational strategies are still not cost effective, since dynamics of sludge and influent are not considered in practice and operational guidelines of membrane suppliers are extremely conservative. More insight in the fouling process details would contribute to a less conservative and more dynamic operation, ultimately leading to cost savings and further market breakthrough.

Not that there has not been any attention given to the fouling problem though. Approximately 30 % of all MBR literature deals with the subject(Yang et al., 2006), containing a slew of pilot-, bench- and lab-scale studies, but, unfortunately, less full-scale assessments. A lot of effort has also been put into the search for models that accurately simulate the fouling phenomenon, but none have truly passed the test of being practically applicable, i.e. validation. At least not in a way that fouling rates and optimal backflush frequencies can be predicted, let alone for forecasting of the cake architecture. As particles deposit, it is expected that they influence the flow regime over and through the membrane. On top of that, the matter that is being withheld on the filter is largely of biological nature, with the ability to produce chemical components that facilitate adhesion to substrates and biofilm growth. This brings forth a lot of factors to bear in mind with filter cake formation modeling in MBR, which is spatially very heterogeneous in nature. Hence, the puzzle that is MBR-fouling remediation is a complex one with many small pieces, rooted in different scientific disciplines. A lot of work has been done though, to define theoretical mechanisms upon which predictions about MBR operation could be made. Numerous models have been composed to relate different operational MBR parameters to each other, but so far no universally valid framework has come out of that abundance. These models, which will be reviewed in section 2.2.1, describe the influence of the filter cake on the flux as a Resistance-In-Series problem. They are thus built upon simplified representations of microscale phenomena, trying to find relations between the high-level MBR operational parameters. As they attempt to accurately describe the MBR operation and cake formation without complete knowledge of the exact process, these models do not carry the burden of being too complex, though they need to be calibrated with lots of data from different sites. This makes it hard to come up with a universal model, since a lot of variation in MBR function properties and conditions has to be accounted for. Moreover, only large-scale information can result from these models, as they lump the process into equations that ignore any spatio-temporal variability within the MBR. There is a high risk associated with lumping complex phenomena in easy-to-digest equations, being that the model will be calibrated for one dataset, but will need recalibration for another, while truly, a different mechanism might have become active. Therefore, it is up to microscale models to investigate these local spatio-temporal phenomena.

Aside from the RIS-approach there are other kinds of models (section 2.2.2) that resort even more to the analysis of large amounts of data, gathered over many different MBR-plants. In these cases, Artificial Neural Networks (ANN) and Principal Component Analysis (PCA) are used. Research in this domain is developing quickly, but has not yet evolved to mature models for practical applications.

In short, R&D of MBR fouling up to now has been largely restricted to large-scale modeling, neglecting smaller-scale spatio-temporal distributions, supplying models for operational and design purposes, while the pursuit for more knowledge about the underlying process has been endeavored far less frequently.

1.3 Objectives of this research

In this master dissertation, an attempt is made to fill the gap in the filter cake modeling landscape and to contribute to a framework that allows a precise simulation of fouling layer build-up in MBRs. This model will incorporate the process at the basis of the cake formation phenomenon - i.e. the deposition of particles on the membrane - in a physically accurate way, as the trajectories of the particles themselves will be modeled. The model output will comprise a simulation of a cake architecture under a set of MBR operational conditions, which can be chosen so that they represent the environment outside of the frame that is being modeled. Sludge concentrations, flow properties, imposed fluxes, ... will all be defined by the user. This allows to incorporate the cake filtration model into a simulation of the fluid dynamics of a whole MBR, which is the long term goal of this research line of fouling modeling.

Ideally, these results will shed more light upon the processes that are of major importance in the genesis of an MBR filtration layer, and might lead to paths of more effective fouling fighting. An ambitious target, but nonetheless one very much worth taking a shot at.

1.4 Outline : the roadmap through this dissertation

The following chapter consists of the literature study, which documents the current knowledge about the fouling process and assesses the inspirations for the chosen model framework.

Chapter 3 elaborates on this framework and bundles the theoretical knowledge base for the model, which is then applied in Chapter 4, as the model implementation is disclosed. The conceived filter cake model is applied to a case study in Chapter 5 and the results are discussed. Finally, the discussion of the model performance is presented in Chapter 6.

CHAPTER 2

Literature Review

In order to create a reliable model that neatly portrays the development of a fouling layer on the surface of an MBR membrane, a thorough and profound knowledge base of the involved processes needs to be incorporated, encompassing a lot of different, interacting parameters, of biological and physical nature. The first part of this chapter is dedicated to this, by assessing what is already known about membrane fouling mechanisms. Next, an overview of the current filter cake modeling landscape is given in order to illustrate the conceived model's place in it. Finally, the inspirations for the model structure are delineated.

2.1 The fouling process

Before elaborating on the fouling process, first, a lexiconical introduction is given of the terms that are used to characterize membrane operation. Most important is the permeate flux J, which is a measure for the flow of treated water that is produced per unit of membrane surface [LMH], and also represents the flow velocity orthogonally through the membrane [$\mathrm{m\,s^{-1}}$]. This flux is determined by the transmembrane pressure (TMP or Δp, in Pa), which stands for the pressure difference over the membrane, i.e. between the feed and the permeate. The relation between the aforementioned parameters is described by Darcy's law (see Eq. (2.1)), a correlation that also includes the viscosity of the fluid that is filtered (μ_f, in $\mathrm{kg\,m^{-1}s^{-1}}$) and the resistance of the membrane (R_m, in $\mathrm{m^{-1}}$). The latter also represents the inverse of the membrane permeability, which is a measure of how well the permeate flow is conducted through the membrane. Darcy's law basically dictates that, in order to attain a certain flux over a membrane with an inherent resistance, sufficient pressure needs to be applied on the feed side.

$$\Delta p = J R_m \mu_f . \tag{2.1}$$

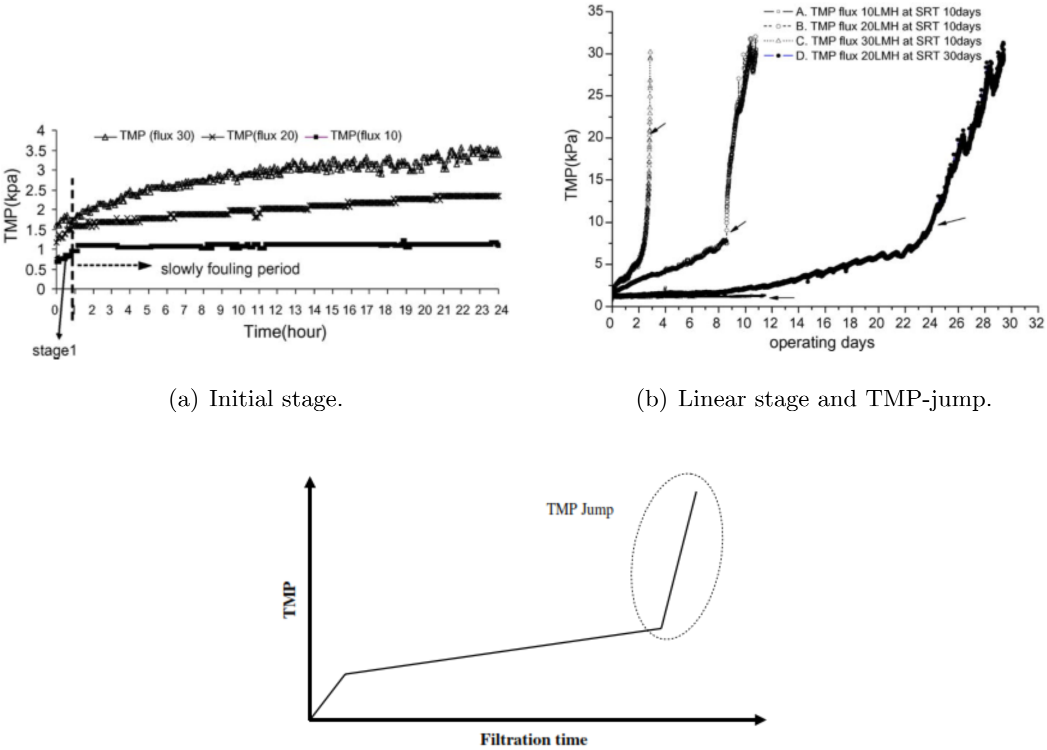
But this equation only holds until fouling emerges. Fouling of membranes can be described as the covering of the membrane surface - external and internal - by deposits that adhere and accumulate during its operation. This phenomenon results in a loss of permeability, meaning that the resistance term in Eq. (2.1) does not solely depend on the resistance of the membrane itself but is now raised with additional resistances put up by fouling. Ergo, when a constant flux is desired during exploitation, the imposed pressure also needs to be increased in proportion to the fouling rate. If not, a dwindling flux will be observed due to the extra hindrance opposed by fouling. In an MBR, this decline is not entirely linear, but can be subdivided into different fouling regimes. When shown from a constant-flux point of view, plotting the TMP that needs to be applied to achieve this operational mode yields the most known profile within the field of MBR (Figure 2.1).

Along this profile, three different stages can be distinguished, the first of which is a conditioning stage characterized by the initial adsorption of macromolecules. These compounds have been of large interest in the search for major fouling causes, so their nature and influence on the TMP profile will be dealt with in the remainder (Section 2.1.1). In this initial fouling stage however, their role consists of pore blocking and facilitating the adhesion of biomass due to their adsorption on the membrane. This process starts as soon as the membrane is put into contact with the mixed liquor from the reactor and even takes place at zero flux, as it is mainly driven by chemical and physical interactions between the components and the membrane. The result is a short yet precipitous decrease of permeability. After that, a less steep fouling regime is witnessed, consisting of a gradual deposition and cake layer buildup. In this stage, the TMP increases roughly linearly, though in the end it suddenly increases exponentially, a phenomenon termed the TMP-jump.

The latter is referred to as an abrupt escalation of the filtration resistance. The process behind this event has not yet been fully determined, although a few explanations have been proposed. A main suspect is cake layer compaction, which is induced by the pressure forces on the cake. Namely, by acting as an additional filtration barrier, the filter cake increases the TMP necessary to achieve a constant flux. Calculating this pressure drop (Δp) over the cake layer can be done with the Carman-Kozeny equation (Eq. (2.2)).

$$\Delta p = \frac{5\mu_f S^2(1-\epsilon)^2 J}{\epsilon^3}\Delta\ell \tag{2.2}$$

This relation indicates that the pressure drop Δp over a cake layer with depth $\Delta\ell$ $[\mathrm{m}]$ rises with increasing specific surface area (S, equals the ratio of the surface over the volume of a particle $[\mathrm{m}^{-1}]$), increasing flux (J) and decreasing cake porosity (ϵ, $[-]$). By applying this pressure difference, one also exerts a force over the cake layer, which squeezes the cake, possibly causing a further decline in porosity. This, in turn, elevates the force on the cake, which intensifies cake compaction and so on. This process is self-supporting and drives the membrane towards breakage. Another aspect of compaction is that it will first occur in the part of the cake adjacent to the membrane. The mechanism is visualized in Figure 2.2



(a) Initial stage.

(b) Linear stage and TMP-jump.

(c) Schematic TMP-profile.

Figure 2.1: Examples of the TMP rising pattern. 2.1(a) and 2.1(b) depict TMP-curves at different constant fluxes, respectively of the first 24 h and the whole duration of the experiments in Zhang et al. (2006). 2.1(c) gives a schematic representation of the TMP-profile (Meng et al., 2009)

and explains that the force that is applied on one layer is passed onto the underlying layer (following the direction of the permeate flow), augmented with the pressure force that is needed to obtain the permeate flux through this layer. As such, the pressure on a cake layer equals the sum of the pressure drops of all layers above it, making the bottom or membrane-neighboring layer the one that endures the largest pressure force. As a consequence, this layer will be the first to collapse.

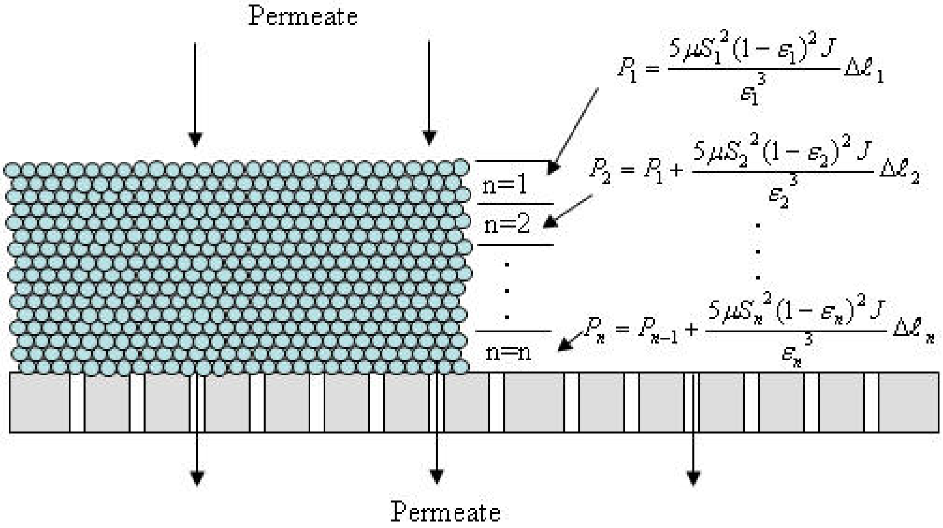

Permeate
$P_1 = \frac{5\mu S_1^2(1-\varepsilon_1)^2 J}{\varepsilon_1^3}\Delta\ell_1$
n=1
$P_2 = P_1 + \frac{5\mu S_2^2(1-\varepsilon_2)^2 J}{\varepsilon_2^3}\Delta\ell_2$
n=2
$P_n = P_{n-1} + \frac{5\mu S_n^2(1-\varepsilon_n)^2 J}{\varepsilon_n^3}\Delta\ell_n$
n=n
Permeate

Figure 2.2: Cumulative buildup of pressure in a cake layer along the direction of the flux. Source: onlinembrsite

Due to the self-accelerating nature of the process, cake compaction is currently the best fitting hypothesis to the TMP-jump phenomenon, but there are a few other theories. One of them indicates pore loss as a main cause for the sudden TMP-rise, pointing at the fact that as the pores of the membrane get blocked by foulants, the remaining pores need to deal with a higher local permeate load since a constant flux is being maintained through a smaller number of pores (Ognier et al., 2004). Along with this increased water flow through the remaining unblocked pores comes an elevated fouling rate and blocking probability, which speeds up the loss of functioning pores - a vicious circle ensues. The pore loss model has its equivalent in the area loss model, where the local loss of permeability is attributed macroscopically by fouled membrane area instead of microscopically by blocked pores (Cho and Fane, 2002). Another rationale attributes the TMP-ramp to quorum sensing (Yeon et al., 2009), but this is out of scope for this dissertation.

2.1.1 Concepts within the field of MBR

Critical flux

Since fouling is flux-related, a theory has arisen that, in an MBR, there exists a ‘critical flux’ below which permeability does not decline over time. This idea has a mathematical ground in the force balance of a particle, with deposition not occurring when drag forces do not exceed dispersive forces. As the case of a constant, effortless flux has never been observed in an MBR in reality, the original austere condition of ’no fouling’ has been softened, so that the critical flux is defined as the highest flux at which either (a) the TMP-curve (see Figure 2.1) remains horizontal (within experimental accuracy, that is), (b) the slope of the TMP-curve is smaller than an arbitrary value, (c) the average TMP and the flux are linearly correlated or (d) a reversible deposit is created or detected, by e.g. direct observation through the membrane (DOTM). Besides, the value for this flux differs within each MBR configuration. In recent years, a similar and more rightly-termed concept has been circulating in MBR-literature, namely the sustainable flux, to indicate the - experimentally determined - highest flux at which fouling is kept below an economically acceptable value for an extended amount of time. It is still a vague term though, since there is no convention about the acceptable amount of fouling or the duration of this regime.

Extracellular polymeric substances

In MBR literature, fouling is often correlated with the concentration of extracellular polymeric substances (EPS) in the mixed liquor. This is a general term to refer to any endemic macromolecule, regardless of its chemical class, that is found at or outside the microbial cell surface. After their secretion by the biomass, these compounds can end up in the extracellular space either by being shed by their manufacturers or by their release during cell lysis. EPS are pivotal in the aggregation of bacterial cells into flocs and biofilms, as well as adhesion to other surfaces, such as a membrane. Moreover, EPS also form a protective barrier around the bacteria and retain water, making that these polymeric biomolecules can form a highly hydrated gel matrix in which the microbial cells are embedded. Finally, biofilm growth on the membrane is again stimulated as the attachment of bioflocs locally provides a major nutrient source for this process. EPS are typically differentiated into compounds residing in or at the cell wall and those that float solubilized in the mixed liquor. The cell wall-adhering fraction is actually commonly referred to as ‘EPS’ in literature, though a more suitable term would be ’extracted EPS’ or eEPS, referring to the lab methodology. Yet it is the fluid-borne fraction that is thought to impact fouling in MBR most substantially. These compounds are called soluble microbial products, or SMP, and refer to EPS that end up in the supernatant after centrifugation of the MBR sludge, as the term is meant to refer to the free soluble fraction of EPS. The comparison of different EPS solutions is often done by a characterization on the basis of relative protein (ESPp) and carbohydrate content (ESPc). The latter fraction is believed to be more critical in fouling because of the hydrophility of the carbohydrates, in contrast to the generally hydrophobical behavior of proteins. Concentration ranges are found to be $11 - 120\,\mathrm{mg/L}$ for ESPp and $7 - 40\,\mathrm{mg/L}$ for ESPc (Judd, 2010).

The deposition of EPS on the membrane surface leads to the buildup of a covering gel layer that facilitates attachment of bacterial cells. In the beginning stage of this development, the first EPS tie to the membrane material due to interaction by forces of physical (adsorption), chemical (covalent bonds) or electrostatic nature (Van der Waals-forces), of which the magnitudes depend on the roughness characteristics of the membrane surface. As illustrated in Figure 2.3, local accumulation of EPS brings along the formation of a stronger matrix by cross-linking of the compounds, becoming a gel layer rich in nutrients with active groups, creating an optimal environment for bacteria to anchor themselves in (Tansel et al., 2006).

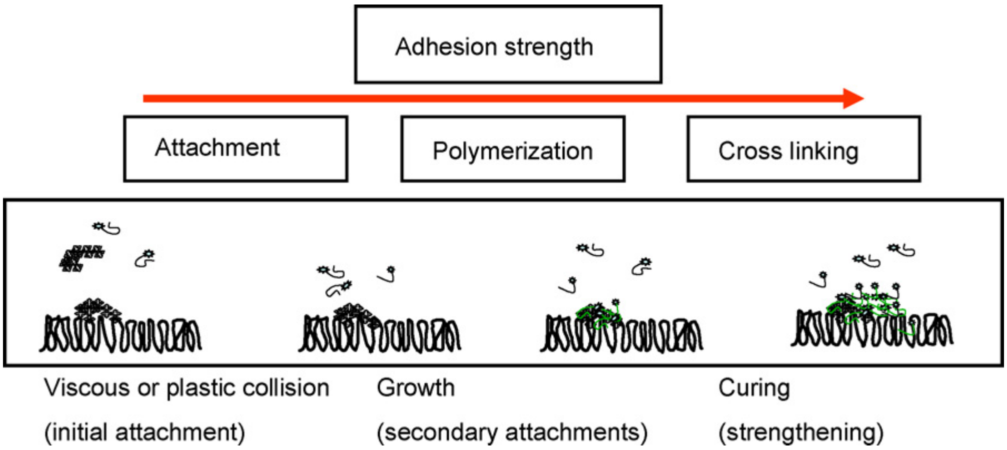


Figure 2.3: Progression of EPS deposition on a membrane. (Tansel et al., 2006)

Whether the EPS-concentration determines the fouling propensity, and to which degree it does, has not yet been determined solidly, but many researchers have reported positive correlations between the two variables (Judd, 2010). These are, however, based on lab trails, in which higher EPS concentrations appear than in a full-scale MBR. The statement that EPS do not significantly influence fouling in fully operational MBR installations has among others been backed by Drews et al. (2008).

Regarding the production of EPS, modeling has also caught up, providing a line of three approaches, each one adding more complexity to their precursor (Menniti and Morgenroth, 2010).

2.2 Fouling modeling up until now

2.2.1 Mechanistic models

The most prominent approach to fouling modeling fits a **Resistance-In-Series** problem (RIS) into Darcy's law (see Eq. (2.1)). The clean membrane resistance R_m, which is provided by the membrane manufacturer, but can also be determined by applying the aforementioned law in an ultrapure water filtration trial, is combined with other resistance terms that arise from membrane fouling in order to form a total filtration resistance.

The additional impedances are conventionally attributed to the resistance of the cake layer (R_c) and pore blocking (R_p), all in $[\mathrm{m}^{-1}]$. For these supplementary resistance components, separate models have been developed that are either based on the exact mechanism of the respective processes or involve a semi-empirical relation. Variations within this genre of filtration modeling mostly decompose one of the two fouling-related resistances according to different mechanisms on a smaller scale, such as in Naessens et al. (2012a). One important observation is also that, generally, these models do not incorporate resistances that might be caused by scaling or concentration polarization, as these are considered negligible.

Although there already exists an abundance of RIS models and their development is still prevalent in MBR research, they arise a few practical issues. First of all, because every resistance term is mimicked by a separate model, each of which involving a few parameters, overfitting might become an issue. Incorporating too many parameters namely requires a larger calibration effort. On the other hand, certain processes that have been neglected by a RIS model can also thwart calibration.

Furthermore, overcoming the step between calibration and validation is one of the hardest parts for fouling models, as the operational conditions can differ substantially between full- and lab-scale MBRs. Therefore, these models need to be validated with data from a number of different full scale MBR plants, but, apart from an extensive data-gathering, most of the RIS models don't even undergo this final evaluation, which severely limits their applicability in cases different from the one in which they were calibrated.

2.2.2 Data-driven models

In comparison to the mechanistic models, data-driven approaches only make up a small segment of the filtration modeling landscape, though different techniques are being researched side-by-side here. One of these is inspired by biological principles, finding its origin in the mathematical representation of the brain, namely Artificial Neural Networks (ANN).

Other methods can be classified as advanced regression techniques, of which the most notable one is Principle Component Analysis (PCA). It is principally a way of dealing with large data sets - a great feature in MBR research - as it applies a data transformation, so that new variables are created according to the directions of largest variance in the output variables (e.g. the flux). These variables are linear combinations of the initial variables and form the principle components, which are thus also carrying information about the greatest fouling causes. These techniques are very promising in fouling control in MBRs and may contribute substantially to future advances in this field (Naessens et al., 2012a).

2.3 Modeling inspirations

In this section, the model paradigms that served as inspirations for the filter cake model are listed and discussed. The first part considers the framework that will actually be used, and in the second, the recent history of biofilm modeling is given, which shows many similarities with the discrete approach that is aimed for.

2.3.1 Multiphase flow modeling

The numerical modeling of multiphase flows is based on the discretisation of the space under consideration in a lattice of cells, where the Navier-Stokes (NS) equations are applied to describe the motion of the fluid. According to whether or not all of the phases are modeled this way, two different paradigms can be distinguished, namely the Euler-Euler and Euler-Lagrange approach (Naessens et al., 2012b). Recalling that the Euler way describes the fluid flow through a fixed location, like a cell in the grid, the Euler-Euler approach comes down to treating all the phases as interpenetrating continua, with volume fractions summing to 1 in each cell. The Lagrange method, on the other hand, involves tracking individual particles through space and time, so the Euler-Lagrange technique follows the freely moving particles the Lagrangian way in a fluid that is modeled as an Eulerian continuum. The force balance on a particle is then also determined by the fluid motion in the grid cell corresponding to its position. With a simpler implementation and results that give a more realistic representation of the dispersed phase, this last method is preferential to the Euler-Euler approach, although high numbers of particles modeled this way can easily result in a model that is too intensive from a computational point of view.

2.3.2 Biofilm models

A biofilm can be described as an assemblage of biological cells that occupies an interface. Supposing that this interface consists of the surface of the membrane of an MBR, it is not hard to acknowledge the fundamental similarities between a biofilm and an MBR filter cake, as also the essential building block of both is embodied by a microbial cell. The main difference though, is that the largest contributor to biofilm growth is cell division, while on the time scale of filter cake buildup in MBRs, which is mainly controlled by the backwash frequency, this may be neglected. Instead, particle deposition from the bulk dictates the volume and architecture of the filtration cake.

Virtual biofilms were also simulated as mechanistic models, with early models that were composed to predict substrate utilization, initially even without accounting for effects of mass transfer within the film (Atkinson et al., 1963; Atkinson and Daoud, 1970). These efforts resulted in the conceptualization of biofilms attached to a flat plate with infinite length and width (Williamson and McCarty, 1976). Such in silico biofilms are composed of layers of active and inactive mass - their border was marked by a certain biofilm depth - and also the effects of incomplete mixing on substrate transfer is mimicked. Yet, these models lack any expression of the interdependence between substrate consumption and biofilm mass. This issue is resolved with the subsequent development of models that also simulate biofilm growth (Kissel et al., 1984; Wanner and Gujer, 1985, 1986). With these developments, biofilm modeling reached somewhat of a turning point, still accurately predicting phenomena of interest to industrial systems (e.g. substrate utilization rate) but additionally providing insight into biofilm accumulation. In fact, the latter models have effectively been used as research tools, in order to assess the effects of different processes like poor antibiotic penetration and quorum sensing (Roberts and Stewart, 2004; Nilsson et al., 2001).

The approach that enabled this added value, along with the advances in computational resources, was pseudo-steady-state modeling, the concept of separating fast from slow processes. As microbial growth and decay occur at a much slower pace than substrate uptake and diffusion within the biofilm, pseudo-steady-state allowed for modeling the first two processes using steady-state substrate concentration profiles. The work of Kissel et al. (1984), the first to implement this idea, therefore provides the conceptual basis of all biofilm models that were formulated beyond that era.

Use of cellular automata

Since nearly two decades, the modeling of biofilms has advanced even more by paying attention to its spatial heterogeneity rather than representing it as a slab of uniform thickness, as is the case in RIS-models (Section 2.2.1). The groundwork for this new movement was laid in Picioreanu (1996) and consists of a cellular automaton (CA) model for both substrate and biomass layers. This paradigm involves updating every cell within a given tesselation at discrete time steps on the basis of the discrete states of its neighboring cells (Figure 2.4). In the biomass component of the model of Picioreanu (1996), a cell can only take two states, being one if biomass is present and zero otherwise, as an occupied cell in the grid resembles a microbial cell.

In what follows, a brief overview will be given of the biofilm growth model of Picioreanu (1996). As already mentioned in the previous paragraph, there are two model layers present, which are interlinked. The first one is to account for the substrate concentration throughout the modeled space and the second layer simulates the spreading and growth of the biomass,

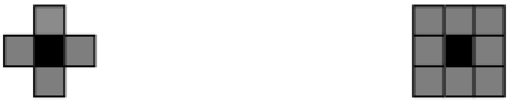


Figure 2.4: Examples of a neighborhood in CAs: von Neumann (left) and Moore (right).

the latter being modeled using a cellular automaton. Globally, this method concerns finding 'donor cells' that dissipate substance to a neighboring acceptor cell. The acceptor is selected from the neighborhood of the donor cell, according to a chosen set of rules, for example according to the largest concentration difference.

Modeling of microbial growth by means of a CA involves tackling similar questions as in the case of substance diffusion, namely which cell will divide at a certain moment and in which grid cell its daughter cell will emerge. The former is typically tackled by considering the cell's access to substrate. For the latter though, it can be tricky to find a place for the daughter cell, as it may happen that there is no free cell in the neighborhood of the mother cell. But this isn't a major issue in the case of filtration cakes and won't be discussed here any further. When a suitable spot is available and a daughter cell emerges, the substrate layer is also affected, as substrate consumption happens simultaneously.

Finally, two other processes are of importance in the establishment of a biofilm: attachment and detachment of individual cells. For the simulation of microbial cells drifting in from the bulk, random walkers are generated in the free space of the lattice (2.5). When they occupy a place adjacent to the biofilm, the walkers can join the cluster by a certain probability in function of different factors, such as the walker density. Detachment, on the other hand, is also mimicked by random walkers in Picioreanu (1996), as only erosion is modeled. Therefore, a flux of random walkers is generated from a point of the square lattice, and when such a drifter reaches the surface of the biofilm, the touched cell is removed from the aggregate with a certain probability.

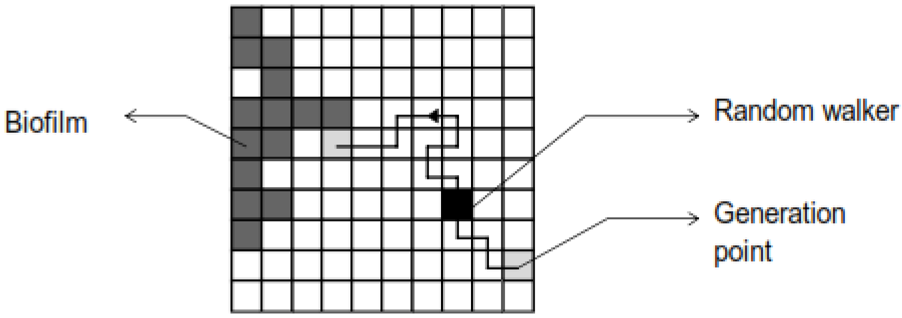
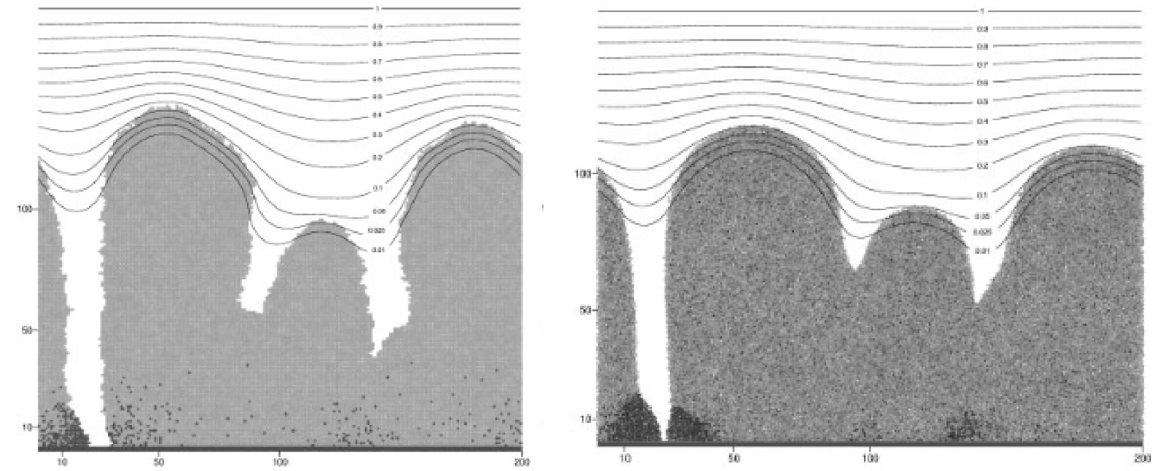


Figure 2.5: Attachment procedure in Picioreanu (1996)

Individual-based modeling

A few years after the emergence of CA-based models, IbMs were proposed to simulate biofilm dynamics , which is short for individual-based (or agent-based) modeling, while by that time Picioreanu's approach had been termed biomass-based modeling (BbM). While the substrate and biomass component of the latter are still present in IbM, the difference lies in the fact that the biomass is not distributed in a discrete grid but bacteria are now represented by (spherical) cells in a continuous space. Also, for modeling the movements of the microbes, the CA-approach is abandoned, so in IbM, these cells shove each other in order to avoid overlap. This goes as follows: for each cell the vector sum of all positive overlap radii is checked, after which the cell is shifted in the opposite direction of that vector. This way, the spreading of the microbes is directionally unconstrained, and this is probably the main reason why IbM are able to produce more rounded and confluent biofilm structures than BbM. The modeling of substrate diffusion, on the other hand, still happens through a discretized grid, but also jettisoned CA rules, this time in favor of mass balances. In Kreft et al. (2001) for example, these are applied as diffusion-reaction equations - partial differential equations that only consider diffusional transport and conversion of the substance - in an implicit difference scheme.

IbM are frequently used to model multispecies biofilms, an objective for which it lends itself perfectly. Namely, IbM allows individual variability of the organisms to be incorporated. When an entity changes its position, also its fixed and variable properties, like genome or state of differentiation, travel along. In order to pour this into an accessible model, the cells along with their characteristics are typically implemented as objects (Kreft et al., 2001). Another interesting feature, one that eases the computational burden, is the sorting of bacteria by location with the help of a tree data structure, wherein each leaf is occupied by one bacterium. As a synthesis of this paragraph, results of Kreft et al. (2001) are shown in Fig.2.6 of a simulated biofilm with the use of BbM and IbM.



(a)

(b)

Figure 2.6: Images of BbM 2.6(a) and IbM 2.6(b) biofilms (with contours of oxygen concentration) containing ammonia (light grey) and nitrite oxidizers (dark grey). (Kreft et al., 2001)

CHAPTER 3

Model development

The general framework of the filter cake model can, as mentioned before, be divided into to two model layers: the Eulerian modeling of the liquid in the system and the Lagrangian modeling of the solid particles in it - referred to as the disperse phase (Figure 3.1).

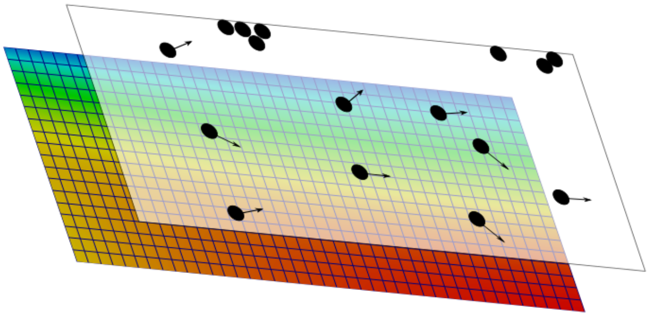


Figure 3.1: Schematic representation of the model layer structure.

The generation of the flow profile is performed only once at the beginning of the simulation. It then serves as a static basis for modeling the disperse phase, which is modeled in discrete time.

In this chapter, first the assumptions of the model are laid out, as such providing the conceptual framework of the model. In the remainder, a section is dedicated to each of the two model layers.

3.1 Assumptions

The aim of the filter cake model is to represent the fouling process in MBRs as realistic as possible, yet for achieving a certain computational efficiency, and for not making the model overly complex, a few assumptions have been made. These assumptions revolve around the nature of the particles, the system in which they are modeled, etc. In this section, all the assumptions for the model, as it is conceived within the scope of this work, are listed, in order to lay out the conceptual basis of the model. A detailed discussion of the assumptions can be found in Section **??**.

The assumption that the filter cake model is practically based upon is that **all sludge particles are rigid, perfect spheres**. This is necessary since the formulas of the force components have all been derived for spheres in a flow field. Also, **all sludge particles are of the same size** and the particle diameter d_p is a constant. Particle size distributions for MBR sludge are available in literature (Wisniewski and Grasmick, 1998), and can normally be implemented with little effort, but this would also require a revision of the way the build-up of the filter cake itself is simulated. **Free moving particles do not collide with each other** in the filter cake model. The only collision detection in the model is the one for adhesion to the membrane or the filter cake. Also these collisions are, in case the particle does not stick to the filter cake, completely elastic, since all particles are considered as rigid spheres.

The system is basically **modeled in two dimensions**. The modeled space does comprise a certain volume, but the depth of the volume is always taken as the particle diameter d_p. Roughly stated, the framework is a slice of any chosen three-dimensional system, with depth d_p. Only the particle movements and fluid velocities in the x- and y-direction are modeled, and it is assumed that those in the z-direction do not have an influence on the simulation results as they cancel one another out. Formation of filter cake on the membrane does not alter the local flux or the flow profile.

The **flow profile of the continuous phase is only computed once** at the beginning of the simulation. It serves as a static basis for modeling the particle movements. The continuous phase has the properties of water at $p = 1\,\mathrm{atm}$ and $T = 20\,^\circ\mathrm{C}$, so the influence of the sludge on the fluid viscosity is neglected.

In relation to the cake build-up, the assumption is made that particles adhere with a probability that is inversely proportional to their velocity (see Section 4.1.3). Particles that have adhered are immobilized in the filter cake model. They remain at their collision position on the filter cake, regardless of the local fluid velocity. They cannot detach from the filter cake either. This imperfection can partly be compensated by the aforementioned adhesion probability.

3.2 Mathematization of filtration cake formation

For an accurate description of a particle trajectory in a fluid, the particle's velocity needs to be determined at every time step. As the position vector of the particle, $\mathbf{X}_p(t)$ $[\mathrm{m}]$, is determined by its center-of-mass, the translational velocity of the particle $\mathbf{U}_p(t)$ $[\mathrm{m\,s^{-1}}]$ is given by :

$$\mathbf{U}_p(t) = \frac{d\mathbf{X}_p(t)}{dt}. \tag{3.1}$$

In force balance equations though, the particle's velocity is expressed relatively with respect to the fluid velocity, i.e. $\mathbf{U}_r = \mathbf{U}_p - \mathbf{U}_c$, with $\mathbf{U}_r$ $[\mathrm{m\,s^{-1}}]$ the relative velocity of the particle and $\mathbf{U}_c$ $[\mathrm{m\,s^{-1}}]$ the characteristic velocity of the continuous phase, for example of the surrounding liquid.

The particle's velocity is governed by Newton's second law (Wörner, 2003):

$$m_p \frac{d}{dt}\mathbf{U}_p(t) = \mathbf{F}_{surf} + \mathbf{F}_{body}, \tag{3.2}$$

where it is assumed that the particle mass m_p [kg] remains constant. In this equation, $\mathbf{F}_{body}$ embodies the forces that act throughout the volume of a body, such as gravity and electromagnetic forces, while the resulting component of the contact forces that are exerted on the particle's surface, is represented by $\mathbf{F}_{surf}$. For a particle immersed in a liquid, $\mathbf{F}_{surf}$ consists of the following components (see Eq. (3.3)) (Wörner, 2003):

- $\mathbf{F}_{Arch}$: the Archimedes force, which is the upward force exerted on a body submerged in a fluid, equaling the weight of the volume of fluid that is displaced by the submerged body.
- $\mathbf{F}_p$: the force resulting from the pressure gradient ∇p,
- $\mathbf{F}_{hydr}$: the hydrodynamic force, which can in turn be split into :
  - $\mathbf{F}_{drag}$: the drag force, the most important hydrodynamic component, with a direction opposite to the particle's motion vector, as it represents a resistance experienced by the particle due to the presence of the liquid (Stokes, 1851),
  - $\mathbf{F}_{am}$: the added mass force, a force required to accelerate the fluid surrounding the particle (Parmar et al., 2011),
  - $\mathbf{F}_{hist}$: the history force or Basset force, which describes the force due to the lagging boundary layer formation with changing velocity (Vojir and Michaelides, 1994). It depends on the acceleration history of the particle, hence its common name,
  - $\mathbf{F}_{lift}$: the lift force, which distinguishes itself as it has a direction perpendicular to the one of the velocity vector of the particle, whereas the other hydrodynamic forces are directionally opposite to it (Saffman, 1965). This force encompasses phenomena like shear lift, as it comprises the inertia effects in the viscous flow around the particle.

All together, $\mathbf{F}_{surf}$ can be written as :

$$\begin{aligned}\mathbf{F}_{surf} &= \mathbf{F}_{Arch} + \mathbf{F}_p + \mathbf{F}_{hydr} \\ &= \mathbf{F}_{Arch} + \mathbf{F}_p + \mathbf{F}_{drag} + \mathbf{F}_{am} + \mathbf{F}_{hist} + \mathbf{F}_{lift}\end{aligned} \tag{3.3}$$

The direction of the latter four force components in Eq. (3.3) is determined by the direction of $\mathbf{U}_r$ (and for $\mathbf{F}_{lift}$ also by the fluid velocity gradient perpendicular to $\mathbf{U}_r$). In Figure 3.2 all of the force vectors that work on a particle immersed in a fluid are set out for the case of a negative relative velocity $\mathbf{U}_r$.

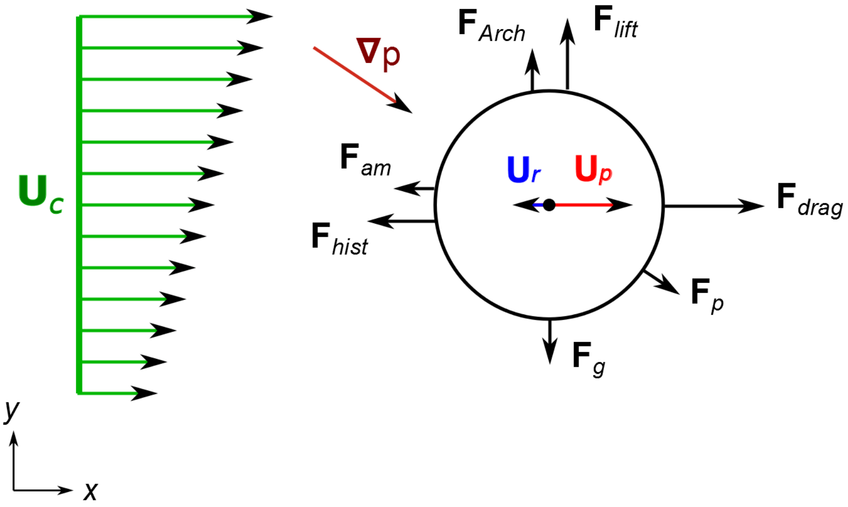


Figure 3.2: Schematic representation of forces that act on a particle immersed in a fluid. The particle has a negative relative velocity $\mathbf{U}_r$ and is being accelerated in the positive x-direction ($\frac{d\mathbf{U}_r}{dt} > 0$).

The formulas for calculating the different forces are summarized in Table 3.1 and the parameter descriptions can be found in Table 3.2. Any peculiarities to these equations are discussed here, such as the calculation of the fluid velocity gradient κ, which is formulated as

$$\kappa_i = \frac{d\mathbf{U}_{c,j}}{di}$$

for direction i and j in a Cartesian coordinate system.

The value of the drag coefficient C_D [-] is a function of the particle Reynolds number Re_p [-], which varies depending on the flow regime around the particle. Re_p is calculated by (Wörner, 2003)

$$Re_p = \frac{\rho_f d_{eq} U_r}{\mu_f}, \tag{3.4}$$

Table 3.1: Summary of particle forces and their formulae.

Force component	Formula
$\mathbf{F}_g$	$m_p\mathbf{g}$
$\mathbf{F}_{Arch}$	$-\rho_f V_p\mathbf{g}$
$\mathbf{F}_p$	$-V_p\nabla p$
$\mathbf{F}_{drag}$	$-\frac{1}{2}\rho_f A_{pcs} C_D \mathbf{U}_r\lvert\mathbf{U}_r\rvert$
$\mathbf{F}_{am}$	$-\frac{1}{2}\rho_f V_p \frac{d\mathbf{U}_r}{dt}$
$\mathbf{F}_{hist}$	$-\frac{3}{2}\sqrt{\pi\eta_f\rho_f}d_p^2\int_0^t \frac{d\mathbf{U}_r(\tau)/d\tau}{\sqrt{t-\tau}}d\tau$
$\mathbf{F}_{lift}$	$-1.615\rho_f d_p^2\mathbf{U}_r\sqrt{\mu_f\lvert\kappa\rvert}\,\mathrm{sgn}\,(\kappa)$

with d_{eq} [m] the volume-equivalent diameter of the particle, U_r [m s^{-1}] the magnitude of the relative velocity and μ_f [Pa s] the kinematic viscosity of the fluid. For the case of a rigid spherical particle, its influence on the calculation of C_D can be formalized as (Wörner, 2003):

$$C_D = \begin{cases} 24/Re_p & \text{if } Re_p < 1, \\ \frac{24}{Re_p}(1 + 0.15Re_p^{0.687}) & \text{if } 1 < Re_p < 1000, \\ 0.44 & \text{if } 1000 < Re_p < 3 \times 10^5. \end{cases} \tag{3.5}$$

Table 3.2: Lexicon of the parameters in Table 3.1.

Parameter	Description	Unit
A_{pcs}	projected cross-sectional area of the particle normal to the direction of $\mathbf{U}_r$	m^2
C_D	drag coefficient	-
d_p	particle diameter	m
$\mathbf{g}$	gravitational acceleration vector	m s^{-2}
∇p	external pressure gradient	Pa m^{-1}
V_p	particle volume	m^3
η_f	fluid dynamic viscosity	Pa s
κ	fluid velocity gradient	s^{-1}
μ_f	fluid kinematic viscosity	m^2 s^{-1}
ρ_f	density of the fluid	kg m^{-3}

When Re_p exceeds the critical value of 3×10^5, C_D suddenly decreases because the boundary layer becomes turbulent. The drag force calculation in the Stokes region ($Re_p < 1$) is equivalent to Stokes' Law (Stokes, 1851):

$$\mathbf{F}_{drag} = -3\pi\eta d_p \mathbf{U}_r. \tag{3.6}$$

This equation can be applied when the following conditions are met :

- the fluid is incompressible,
- the flow pattern around the particle is laminar, i.e. $Re_p < 1$,
- there are no other particles in the neighborhood that could alter the flow pattern,
- the particle is rigid and perfectly spherical with a smooth surface.

The formula for the lift force $\mathbf{F}_{lift}$, as shown in Table 3.1, was determined by Saffman (1965). This formula is also only applicable in case $Re_p < 1$. More turbulence means that the magnitude of the lift force needs to be multiplied with the Saffman-Mei correction factors (Mei, 1992), so that the correct lift force component $\mathbf{F}'_{lift}$ is given by :

$$\frac{\mathbf{F}_{lift}}{\mathbf{F}'_{lift}} = \begin{cases} \left(1 - 0.3314\sqrt{\beta}\right) \cdot e^{-0.1Re_p} + 0.3314\sqrt{\beta} & \text{if } \mathrm{Re_p} < 40, \\ 0.0524 \cdot \sqrt{\beta Re_p} & \text{if } 40 \leq \mathrm{Re_p} < 1000, \end{cases} \tag{3.7}$$

with

$$\beta = \frac{Re_\omega}{2Re_p}, \tag{3.8}$$

$$Re_\omega = \frac{\rho_f \omega_f d_p^2}{\eta}, \tag{3.9}$$

$$\omega_f = \left|\nabla \times \mathbf{U}_c\right|, \tag{3.10}$$

where ω_f stands for the curl of the velocity field of the surrounding fluid.

The acceleration of the particle can be derived from Newton's second law Eq. (3.2), using the overall force balance. However, in order to describe particle motion through a fluid accurately, it must also be corrected for the presence of the nearby boundaries of the fluid, such as a wall. This is exactly what Faxéns correction comes down to. It brings into account the reduction in particle velocity due to the effect of the fluid being constrained between the particle and the nearby boundary (Faxén, 1922). With the Faxén correction factor in place, the Maxey-Riley equation for the overall force balance on a particle in a fluid is given as (Maxey and Riley, 1983):

$$\begin{aligned} m_p \frac{d}{dt}\mathbf{U}_p(t) = &-3\pi\eta d_p \left(\mathbf{U}_r - \frac{1}{24} d_p^2 \nabla^2 \mathbf{U}_c\right) + \left(m_p - V_p \rho_f\right) \mathbf{g} \\ &- \frac{1}{2} V_p \rho_f \left(\frac{d\mathbf{U}_r}{dt} - \frac{1}{40} d_p^2 \frac{d}{dt}\left(\nabla^2 \mathbf{U}_c\right)\right) \\ &- V_p \nabla p - \frac{3}{2}\sqrt{\pi\eta\rho_f} d_p^2 \int_0^t \frac{d\mathbf{u}(\tau)/d\tau}{\sqrt{t-\tau}} d\tau \\ &- 1.615 \rho_f d_p^2 \left(\mathbf{U}_r - \frac{1}{24} d_p^2 \nabla^2 \mathbf{U}_c\right) \sqrt{\mu_f \left|\kappa\right|} \operatorname{sgn}\left(\kappa\right). \end{aligned} \tag{3.11}$$

in which $\mathbf{u} = \mathbf{U}_r - \frac{1}{24} d_p^2 \nabla^2 \mathbf{U}_c$ in the history term.

3.3 Continuous phase

One of the two model layers describes the flow profile of the mixed liquor within the system. This velocity field forms the basis of the motion of the sludge particles as their acceleration is almost completely governed by the forces that are exerted by the surrounding fluid on the particles' surfaces. Within the framework of this dissertation, the flow profile is generated only once at the beginning of a filter cake simulation, to serve as a static basis for the particle motion throughout the whole simulation. Otherwise, the coupling between the two layers should be made explicit and this is considered beyond the scope of this work. Consequently, it is assumed that the filter cake build-up on the membrane does not alter the motion of the continuous phase, which is plausible if the cake volume is negligible compared to the volume of the entire system.

Essentially the continuous phase is modeled the Eulerian way. In contrast to the Lagrangian way, a fixed set of points in space is considered which are connected to a static reference grid. The Eulerian method then consists of describing the movement status (velocity) and acceleration of the fluid parcels that pass through the fixed points, a fluid parcel being an elementary amount of fluid with a constant mass. This requires solving the momentum, mass and energy conservation equations for each fluid parcel, which is a set of partial differential equations (PDE) that is better known as the Navier-Stokes equations.

3.3.1 Components of the Navier-Stokes equations

The first equation that is incorporated in the Navier-Stokes theorem is the continuity equation, which describes the conservation of mass of the fluid with density ρ $[\mathrm{kg\,m^{-3}}]$ in a given static volume Ω $[\mathrm{m^3}]$.

$$\frac{\partial\rho}{\partial t} + \nabla(\rho\mathbf{U}) = 0 \tag{3.12}$$

$\mathbf{U}$ $[\mathrm{m\,s^{-1}}]$ depicts the fluid velocity vector in the volume. Since no mass can be created, Eq. (3.12) simply dictates that the accumulation within the volume Ω equals the difference between the mass entering and leaving it through its surfaces. When applied to incompressible fluids (i.e. fluids of which the compressibility of the fluid parcels is negligible), Eq. (3.12) is simplified since ρ is then a constant ($\frac{\partial\rho}{\partial t} = 0$ and $\nabla(\rho\mathbf{U})$ simplifies to $\rho\nabla\mathbf{U}$), so that

$$\nabla\mathbf{U} = 0 \tag{3.13}$$

Next we have the Euler equation, which is represented in vector form in Eq. (3.14),

$$\frac{1}{\rho}\nabla p = \mathbf{S} - \frac{d\mathbf{U}}{dt} \tag{3.14}$$

where ∇p $[\mathrm{Pam^{-1}}]$ is the pressure gradient, $\mathbf{S}$ $[\mathrm{m\,s^{-2}}]$ the acceleration vector of the external forces and $\frac{d\mathbf{U}}{dt}$ $[\mathrm{m\,s^{-2}}]$ the resulting acceleration vector. These equations are based upon the conservation of mass, momentum and energy, but are only valid for inviscid fluids. For viscous flow the Euler equations need to be expanded with a term that brings into account shear stress and non-elastic pressure. The resulting PDE is the Cauchy momentum equation Eq. (3.15), in which this term is represented by Φ $[\mathrm{m\,s^{-2}}]$:

$$\frac{1}{\rho}\nabla p = \mathbf{S} - \frac{d\mathbf{U}}{dt} - \Phi. \tag{3.15}$$

In the case of Newtonian fluids, Φ can be written in terms of the dynamic viscosity η $[\mathrm{Pa\,s}]$ and the fluid velocity $\mathbf{U}$, for Newton showed that for many fluids, such as water, stress and the rate of strain are almost linearly related (Batchelor, 2000). Newtonian fluids can therefore be modeled by a coefficient peculiar to each fluid, namely the viscosity. Replacing Φ for these Newtonian fluids accordingly leads to the Navier-Stokes equations Eq. (3.16) :

$$\frac{1}{\rho}\nabla p = \mathbf{S} - \frac{d\mathbf{U}}{dt} + \mu_f\nabla^2\mathbf{U}. \tag{3.16}$$

This equation, in which the kinematic viscosity $\mu_f = \frac{\eta}{\rho}$ is used, applies to three-dimensional flow of incompressible Newtonian fluids.

3.3.2 Computational fluid dynamics

For obtaining the flow profile in a certain space in three dimensions, we relied on computational fluid dynamics or CFD, a fluid flow simulation approach that solves PDEs numerically, typically by means of finite element or finite volume methods. In what follows, the procedure of a CFD simulation is outlined.

In the first step, a geometry that represents the modeled space is defined, which is discretized subsequently into three-dimensional cells. In other words, a mesh is created for the space in which the fluid is modeled.

Next, all the equations that are required to describe the involved processes are specified - in this case, the Navier-Stokes equations. Finally, the boundary conditions along the surfaces of the fluid volume are defined. The two types of boundary conditions that are used most frequently, are Dirichlet and Neumann boundary conditions. The *Dirichlet* type indicates that the value of the dependent variable on the boundary is given, whereas a *Neumann* boundary condition prescribes the gradient of the dependent variable normal to the boundary. Then the fluid flow is simulated by iteratively solving the specified equations as a steady state or transient. A *Steady state* solution is obtained by neglecting the cross-terms and higher order terms that are related to time are largely ignored, while a *transient* solution also takes these into account. Roughly speaking, marching a transient solution further in time will, if the boundary conditions do not change, converge to the steady state solution.

The most common discretization approach in CFD is the *Finite Volume Method* (FVM), as it has an advantage in memory usage and solution speed (Patankar, 1980). In this method, the values of the variables are calculated at discrete points (called nodes) that are surrounded by a small volume. In other words, these values are determined in the center points of the cells of the mesh, and they represent the volume averaged values in these cells. Next, the PDEs are discretized according to the FVM-scheme, in which the equations are first recast in a conservative form, yielding volume integrals for the conservation components and surface integrals for the divergence terms, i.e.

$$\frac{\partial}{\partial t} \int \int \int \mathbf{Q} dV + \int \int \mathbf{F} dA = 0 \tag{3.17}$$

in which $\mathbf{Q}$ is the vector of conserved variables, $\mathbf{F}$ the vector of fluxes, V the volume of the finite volume element (i.e. the mesh cell) and A the surface area of the finite volume element. The conservative aspect lies in the fact that the flux entering a grid cell is equal to the one leaving the adjacent cell. So, in short, the PDEs are discretized so that they can be solved for the discrete volumes in the mesh, and the retrieved values actually represent the volume averaged outcomes of the calculations.

CHAPTER 4
Model implementation

4.1 Disperse phase

The overall model for mimicking filter cake formation, will give a representation of the motion of sludge particles that is physically accurate up to some level, since a few assumptions had to be made, which were required either for restricting the computing time or the total complexity of the model (see Section 3.1) . The following section details the Lagrangian side of the filter cake model, which tracks the position of each particle p_i at every consecutive time step as well as the adhesion of particles on a membrane and those leaving the area under consideration.

4.1.1 Overview

The filter cake model is simulated in MATLAB (MathWorks, Natick, MA, USA). The model makes use of five different scripts (m-files), of which one is the main script *(Model.m)* that calls the other scripts, directly or indirectly. The overall scheme is given in Figure 4.1 and the function names and descriptions are, in order of appearance :

- `readFoamFiles` : this function searches the files in ASCII-coding that contain the information about the pressure and velocity fields, imports those into MATLAB matrices and saves them into the MATLAB workspace.
- `newPosition` : within this function the new positions of the particles $\mathbf{p}_i(t)$ are calculated on the basis of the ones at the previous time step $\mathbf{p}_i(t - \Delta t)$, the particles' velocity vectors $\mathbf{v}_i(t)$ and the pressure and velocity fields.
- `adhesion` : this is the function that performs the adhesion check, i.e. whether the particles have trajectories intersecting with the border of the filtration cake, by adjusting the adhering particles' new position $\mathbf{p}_i(t)$, their status, and the thickness of the filter cake.

- `addNewParticles` : stands for the addition of incoming particles at the inlet(s) of the grid.

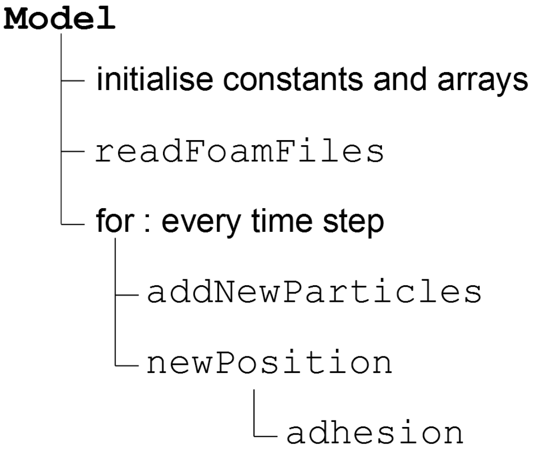


Figure 4.1: Model scheme.

In the main script of the particle model, first all parameters related to the particle characteristics and simulation properties are saved under their respective structure variables and the pressure and velocity profiles are read into matrices with `readFoamFiles`. Also the array in which the (current) positions of the particles (under the form of x- and y-coordinates) will be registered, is initialized and is of sufficiently large size to contain every particle that will pass through the modeled space during part of the simulation. In addition to the coordinates, also the status of each particle is kept track of in this array. There are four different states which are denoted by an integer :

- 0 : the particle has not yet been initialized
- 1 : the particle is freely moving within the system
- 2 : indicates an adhered particle
- 3 : the particle has left the system

The array that is created to contain the particle velocities (also in the x- and y-direction) has an extra time dimension, since for each time step both the previous as well as the current velocities need to be known as input for the force balance Eq. (3.11). The same holds for the relative particle velocities and the computation of the history force even requires that the values of the relative velocities are withheld for a certain time window (cfr. Eq. (3.11)). The next step is that the initial number of particles present in the system is calculated out of the TSS concentration and the volume under consideration. Their initial positions within the space are random, and their initial velocity values are set equal to the local values of the governing velocity.

Then a loop is entered that calculates the new positions - with inherent adhesion check - of the particles, and ads the new particles that have entered the modeled space, for every consecutive time step. The first is done by calling `newPosition`, only for those that are still moving freely within the modeled space. Subsequently, the function `addNewParticles` is called to mimic the introduction of the particles that enter the system. It encompasses the addition of new particle entries with a status that indicates that they are freely moving (i.e. a status set as 1). It overwrites the first number of particle entries with a status 0. Analogously to the particles that are initially present, their velocity components are set equal to the velocities of the governing continuous phase. Their number might not be discrete, so this is floored or ceiled with a probability that is equal to the decimals of the real amount. Removal of off-grid particles is also done within the loop, though only every 1,000 time steps, in order to decrease the computational burden.

4.1.2 Notes on calculating the force balance

At the heart of the computation of the new positions is the calculation of the resulting force on each particle, which is separately done for the x- and y-direction. The force balance has been implemented semi-implicitly for stabilization purposes, since instability was introduced by the drag force, the added mass force and the history force in their explicit form. This means that a few terms of the right hand side of the force balance equation are transferred to the left hand side. The actual mechanism is elucidated in the history force subsection (see further).

Drag force

The drag force $\mathbf{F}_{drag}$, calculated using either Stokes' or Newton's law, has a direction opposed to the relative velocity. It represents the force exerted by the surrounding fluid on the particle as a consequence of the difference in velocities between these two phases. When no other forces would be in play on a particle immersed in a fluid, the drag force would minimize the difference between the fluid and the particle velocity, correcting its motion to the streamlines of the continuous phase.

Keeping this principle in mind, it is easy to explain the stability issues with this force component when it is applied with discrete time. Namely, when its correcting action during a time step is an overshoot, this influences the calculation of the drag force in the next time step, which leads to an oscillating particle trajectory. Whether this profile is converging or not does not matter, since each calculated particle velocity is translated into a particle movement. These movements are not physically accurate and would bring distortion into the filter cake model. Therefore, the influence of the drag force on the particle velocity should nicely converge without showing oscillations. In other words, it should not be overly correcting during one time step.

What has been explained above basically comes down to the following. A change in particle velocity, resulting from the drag force, may not exceed the current difference between the local velocity and the particle velocity itself (i.e. the relative velocity of the particle). This way, the particle velocity will gradually align with the local velocity of the surrounding fluid. Mathematically, it thus boils down to the following :

$$\begin{aligned} |\mathbf{U}_{drag}| &\leq |\mathbf{U}_r| \quad \text{or} \\ \left|\frac{\mathbf{F}_{drag}}{m_p}\Delta t\right| &\leq |\mathbf{U}_r| \end{aligned} \tag{4.1}$$

This equation represents stability the criterion for the calculation of the drag force with discrete time steps of length Δt. It imposes a maximum value for Δt [s] for a given particle diameter d_p and mass density ρ_p, which becomes clear after rewriting Eq. (4.1) for the Stokes regime :

$$3\pi\eta d_p\,|\mathbf{U}_r|\,\frac{\Delta t}{m_p} \leq |\mathbf{U}_r|\,,$$

such that :

$$\Delta t \leq \frac{\rho_p d_p^2}{18\eta_f}. \tag{4.2}$$

The equation above indicates that the maximum time step size will principally be constrained by the smallest encountered particle diameter, as $\Delta t_{max} \sim d_p^2$. In Figure 4.2, this is visualized for $\rho_p = 1003\,\mathrm{kg\,m^{-3}}$.

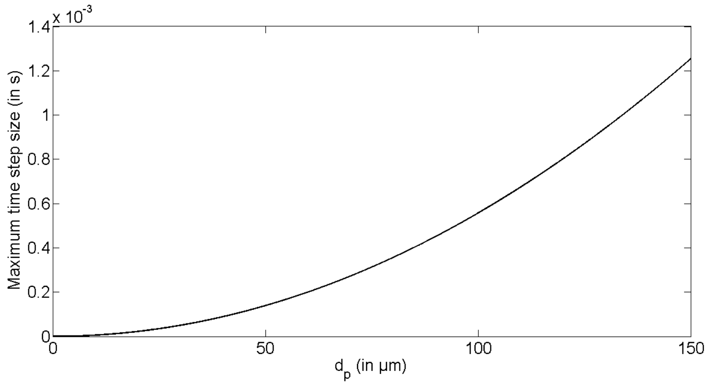


Figure 4.2: Maximum time step size for Stokes drag stability in function of d_p with $\rho_p = 1003\,\mathrm{kg\,m^{-3}}$.

Here, it may be concluded that, the drag force puts a constraint on the maximum time step that can be taken in the filter cake model. In the model, the drag force is calculated with the Faxén correction Eq. (3.11).

Lift force

The direction of the lift force is always perpendicular to the particle velocity vector $\mathbf{U}_p$. In which direction it points, depends on both the velocity gradient of the surrounding fluid κ, as well as the particle's relative velocity $\mathbf{U}_r$. This force results from the pressure difference in the fluid that is generated by the particle's influence on its surrounding streamlines in the fluid.

The discrete calculation of κ in the lift force formula (as found in Table 3.1) is accomplished through the central difference scheme. Within a given grid cell $c(j,k)$, the value for κ in, for example, the y-direction gives :

$$\frac{d}{dy}U_{c,x}^{j,k} \approx \frac{U_{c,x}^{j,k+1} - U_{c,x}^{j,k-1}}{2\Delta y}$$

Loosely speaking, this means that the gradient in the y-direction of the fluid velocity in the x-direction will result in a force in the y-direction, and vice versa. Whether the particle will then actually experience a lift force towards the region with the greatest velocity depends on its relative velocity, since the lift force becomes negative if the latter is negative. Finally, the lift force also needs to be corrected for the flow profile around it, as described in Eq. (3.7).

Added mass force

The added or virtual mass force $\mathbf{F}_{am}$ accounts for the displacement of the fluid due to the moving particle, which is simply accounted for by describing it as the motion of half a particle volume of fluid (see its formula in Table 3.1). Also, the Faxén correction is different than for the other force components, and the time derivative is needed. This means that the model needs to keep track of the Faxén corrected added mass velocities at t and $t - \Delta t$.

Finally, calculation of the added mass force in discrete time also brings along stability issues. These are dealt with by making its calculation partially implicit, which is explained in the remainder of this section since the history force is treated in a similar way.

History force

The history force accounts for the lagging of the boundary layer formation in the force balance. As a consequence of the definite integral in its formula, it is computationally the most demanding force component (Van Hinsberg et al., 2011), as it requires that the relative velocities at the previous time steps are known. Another difficulty is that the history term cannot be approximated analytically, since its integrand is space-dependent and cannot be described by a linear function (Michaelides, 1992). A numerical approach should be followed to compute the integral, one that reduces the history of the particle velocities to a certain time window t_{win} (Dorgan and Loth, 2007). This can be illustrated by rewriting form of the

Basset history force:

$$\mathbf{F}_{hist} = -\frac{3}{2}\sqrt{\pi\eta\rho_f}d_p^2\int_0^t K_B(t-\tau)\mathbf{h}(\tau)d\tau, \tag{4.3}$$

in which

$$\mathbf{h}(t) = \frac{d\mathbf{u}(t)}{dt}, \quad \mathbf{u}(t) = \mathbf{U}_r - \frac{1}{24}d_p^2\nabla^2\mathbf{U}_c \quad \text{and} \quad K_B(t) = \frac{1}{\sqrt{t}}.$$

with $K_B(t)$ referred to as the Basset kernel. In the approach of Dorgan and Loth (2007) this kernel is replaced by a window kernel $K_{win}(t)$, evaluating the integral over a finite time window from $t - t_{win}$ up to t. This kernel is thus defined as

$$K_{win}(t) = \begin{cases} K_B(t) & \text{for } t \geq t_{win}, \\ 0 & \text{for } t < t_{win}. \end{cases}$$

In our the particle model, the numerical approximation in Van Hinsberg et al. (2011) is used, in which the time interval $[t - t_{win}, t]$ is discretized into $N+1$ equidistant points according to $\tau_n = t - n\Delta t$, for $n = 0, 1, 2, ..., N$ with $\Delta t = t_{win}/N$. The window-approximated history force $\mathbf{F}_{hist,win}$ can then be written as :

$$\begin{aligned}\mathbf{F}_{hist,win} \approx & -\frac{4}{3}c_B\mathbf{h}_0\sqrt{\Delta t} - c_B\mathbf{h}_N\frac{\sqrt{\Delta t}\left(N-\frac{4}{3}\right)}{(N-1)\sqrt{N-1}-\left(N-\frac{3}{2}\right)\sqrt{N}} \\ & - c_B\sqrt{\Delta t}\sum_{n=1}^{N-1}\mathbf{h}_n\left(\frac{\left(n-\frac{4}{3}\right)}{(n+1)\sqrt{n+1}+\left(n+\frac{3}{2}\right)\sqrt{n}} + \frac{\left(n-\frac{4}{3}\right)}{(n-1)\sqrt{n-1}+\left(n-\frac{3}{2}\right)\sqrt{n}}\right),\end{aligned} \tag{4.4}$$

with

$$c_B = \frac{3}{2}d_p^2\sqrt{\pi\eta\rho_f}.$$

Here, we set $N = 8$ and $\Delta t = 0.0001$ s such that the window spans 0.0008 s. This is relatively short, but still it is the most CPU-time consuming component of the force balance (Figure 4.3). Thus, $N+1$ or 9 Faxén-corrected relative velocities need to be saved in the workspace for calculating the N derivatives in time. The latter is achieved by using a backwards difference scheme. The coefficients from Eq. (4.4) are constant, so these are calculated only once. Note that this behavior originates from the fact that all the particles have the same diameter d_p. In Van Hinsberg et al. (2011) it also mentioned that explicit solution of the history force shows poor stability and that it requires extremely small time steps to reach a stable solution. The paper therefore offers an alternative method for circumventing these stability issues, making the calculation partially implicit. which consists of bringing part of the history force term (namely $\frac{d\mathbf{U}_p}{dt}(t)$) to the left hand side of Eq. (3.11). The added mass term is treated similarly, which results in the following equation for the motion of particles :

$$\left(m_p + \rho_f V_p + \frac{4}{3}c_B\sqrt{\Delta t}\right)\frac{d\mathbf{U}_p}{dt}(t) = \mathbf{F}_g + \mathbf{F}_{Arch} + \mathbf{F}_p + \mathbf{F}_{drag} + \mathbf{F}'_{am} + \mathbf{F}'_{hist} + \mathbf{F}_{lift}, \tag{4.5}$$

with

$$\mathbf{F}'_{am} = \frac{1}{2}\rho_f V_p \left(\frac{d\mathbf{U}_c}{dt} + \frac{1}{40} d_p^2 \frac{d}{dt} \left(\nabla^2 \mathbf{U}_c \right) \right) \text{ and}$$

$$\mathbf{F}'_{hist} = \mathbf{F}_{hist} - \frac{4}{3} c_B \sqrt{\Delta t} \frac{d\mathbf{U}_p}{dt}(t).$$

Other forces and the Faxén correction

The contributions of $\mathbf{F}_g$ and $\mathbf{F}_{Arch}$ are combined in the implementation, as $(m_p - \rho_f V_p)\,\mathbf{g}$. Naturally, these terms only have an influence in the force balance in the y-direction. If the particles all have the same size, these terms are constants.

Calculating the contribution of the pressure force $\mathbf{F}_p$ requires knowledge of the pressure gradient ∇p:

$$\nabla p(x, y) = \frac{\partial p}{\partial x} + \frac{\partial p}{\partial y},$$

which is discretized according to the central difference scheme :

$$\nabla p_{j,k} = \frac{p_{j+1,k} - p_{j-1,k}}{2\Delta x} + \frac{p_{j,k+1} - p_{j,k-1}}{2\Delta y}.$$

Herein, Δx and Δy represent the discretization step in the x- and y-direction respectively. This gradient, multiplied with the particle volume brings into account the force on the particle due to the pressure gradient in the fluid.

Taking into account the Faxén correction requires calculating the Laplacian of the fluid velocity $\nabla^2 \mathbf{U}_c$, which is performed discretely with a central difference scheme. The value for the Laplacian in grid cell $c(j, k)$ is thus given by :

$$\nabla^2 \mathbf{U}_c^{j,k} \approx \frac{\mathbf{U}_c^{j-1,k} + \mathbf{U}_c^{j,k-1} + \mathbf{U}_c^{j+1,k} + \mathbf{U}_c^{j,k+1} - \mathbf{U}_c^{j,k}}{\Delta x \Delta y}.$$

For both $\mathbf{F}_{am}$ as $\mathbf{F}_{hist}$ the time derivatives of the Faxén corrected relative velocities (in a different form for each of these forces, see Eq. (3.11)) are required. For this, the values of these velocities in the previous time step are also withheld in the workspace.

To conclude, the CPU times of the force components are given in terms of percentage in Figure 4.3.

4.1.3 Adhesion to the cake layer

Checking whether particles adhere to the membrane, or to already formed filter cake, is also done by the function `newPosition`. This check is only performed for those particles that are found in the proximity of the filter cake. How large this proximity is defined, is in relation to the maximum distance that a particle can travel between two time steps (in other words: during Δt). As a next step, the old and newly calculated positions, $\mathbf{p}_i(t - \Delta t)$ and $\mathbf{p}_i(t)$, of the particles near the cake layer, as well as their velocities $\mathbf{v}_i(t)$, are given to

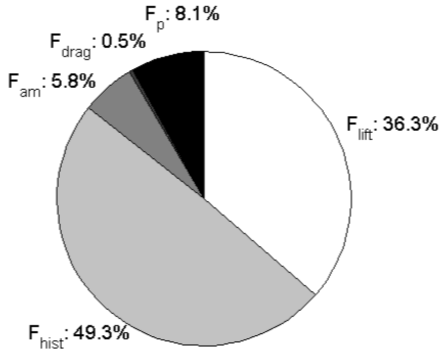


Figure 4.3: CPU time in terms of percentage of the overall force balance calculation.

the `adhesion` function. This function contains several logical arrays to check whether the particle's trajectory path intersects with the boundary of the filter cake. It also determines whether particles stick to the cake or bounce off.

Implementation of filter cake formation

Let us first of all take a look at how the boundary of the filtration cake is kept track of. For starters, it is represented in a discrete form, by an array that contains the local cake thickness, referred to as `wall`. This is done with a specified resolution that should be sufficiently low to give a realistic representation and build-up mechanism of the cake. The local cake thickness across a clean membrane is initially the same over the entire membrane, but it is not initialized as zero. Instead, the initial values in the `wall` array are set to the particle radius, which makes it compatible with the fact that the particle trajectory vectors $(\mathbf{p}_i(t - \Delta t) - \mathbf{p}_i(t))$ represent the movements by the particles' centers-of-mass. If such a trajectory would intersect with the actual cake boundary, it would already be halfway in the cake.
Analogously, when a particle settles, it creates an augmentation of the cake thickness at its settling position. The adjustment of the wall thickness is done within a few steps (Figure 4.4). This will be explained in this paragraph, but might be a greater help understanding the principle. Firstly, the augmentation consists of the upper half of a circle with the particle diameter d_p as radius. This is similar to the principle of initializing the cake thickness as the particle radius, in order to achieve accurate collisions. The cake elevation caused by a settling particle is then set to the value in `wall` at its settling position, augmented with the semicircle. This is then compared to the current values in `wall` at which the augmentation would take place, and for each position in `wall` that is under consideration, the maximum of the original wall and the elevation is withheld.

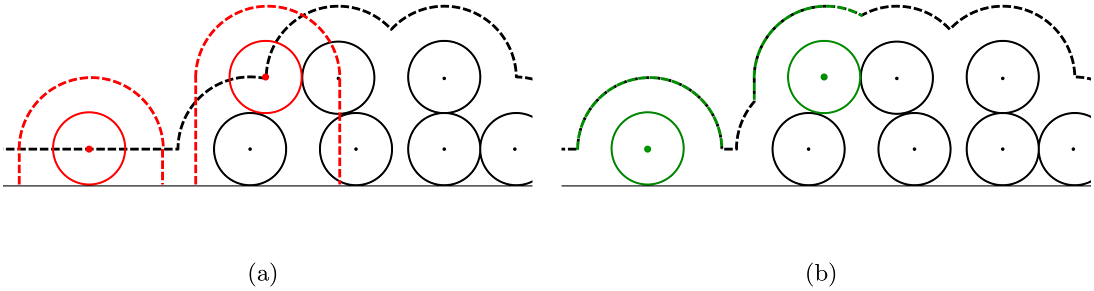


Figure 4.4: Newly settled particles (red) and their augmentation 4.4(a), the final wall adjustments 4.4(b).

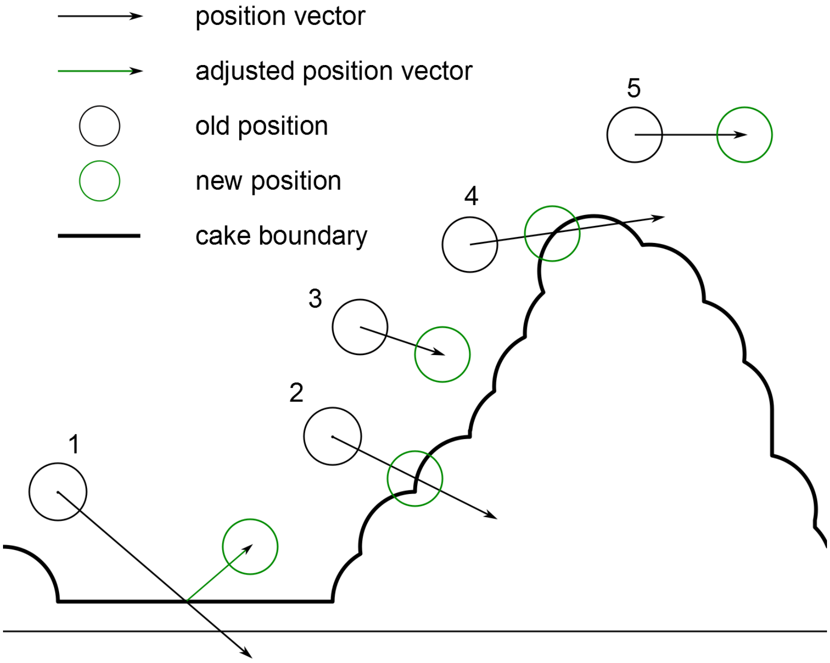


Figure 4.5: Adhesion procedure for different particles.

The adhesion check is illustrated in Figure 4.5. Particle 1's position vector intersects with the cake boundary, but due to its larger velocity it has a smaller probability of settling. Particle 2 has a lower velocity and therefore a higher probability of settling, which it does at the intersection point of its position vector and the cake boundary. Particle 3 is on its way of touching the filter cake, but does not in the current time step. Particle 4's position vector intersects twice with the cake boundary, but it can only touch at the first intersection point. The movement path of particle 5 is not affected by the cake boundary.

For the adhesion check of a moving particle, its position vector is also discretized, with the same resolution as the one in the `wall` array. Then for every element in the position vector it is checked whether its value (perpendicular to the membrane, for example the y-coordinates if the membrane is positioned horizontally) is still above the corresponding value in `wall`. The first point where the value in the discrete position vector is below or equal to the discrete local cake thickness is taken as the particle's adhesion point.

Adhesion probability

It would be very unrealistic if every particle that comes into contact with the filter cake, also settles on it. Essentially, it may be assumed that it becomes more likely that a particle attaches to the cake as its momentum decreases. To incorporate this feature into the adhesion check, the `adhesion` function also accounts for the fact that a particle p_i will only settle with a probability $P(v_i)$. For determining this probability, an exponential function of the particle velocity v_i is assumed:

$$P(v_i) = e^{-kv_i}, \tag{4.6}$$

in which k $[\mathrm{s\,m^{-1}}]$ is a parameter that needs to be determined experimentally. The values of $P(v_i)$ for different values of k are shown in Figure 4.6 for a particle velocity range deemed relevant in MBR systems.

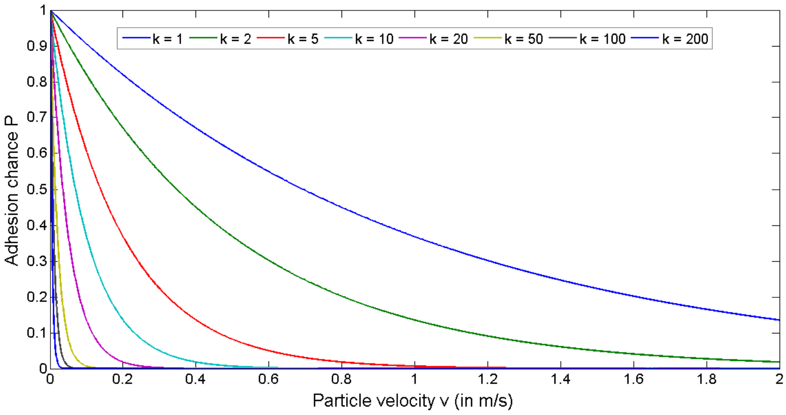


Figure 4.6: Adhesion probability P for different values of k, for particle velocities v between 0 and $2\,\mathrm{m\,s^{-1}}$.

4.2 Continuous phase

The CFD simulations were performed in **OpenFOAM** ®(Open Field Operation and Manipulation, OpenCFD Ltd. (ESI Group), Bracknell, UK), an open source CFD package, directly written in C++ for the Unix/Linux operating system, containing numerous solvers and other

CFD utilities such as mesh generators and data visualization tools. The placement of nodes in the mesh and the incorporation of boundary conditions is done according to a collocated grid, meaning that all calculated variables are located at the centers of the grid cells.

The used solver module is the `icoFoam` package, which solves the Navier-Stokes equations for laminar flow of incompressible, Newtonian fluids - in such a way that it is befit for simulating the water flow inside a chosen configuration. For turbulent flow, the `simpleFoam` solver can be used. The discretization method is by default the Euler method. These solvers arrive at a steady state solution by generating solutions until a certain stability criterion is satisfied. The latter is defined as the difference between the solution of the last time step with the previous one. When this difference is below the threshold specified by the user, it is presumed that a steady state solution is reached.

In addition to the specification of the solver, a mesh along with the boundary conditions for every variable, as well as their initial values should be specified. A mesh is generated with `blockMesh`, a function in OpenFOAM that requires an input file that specifies, among other things, the dimensions, the number of cells and the faces of the geometric grid.

The initial values of the variables along with their boundary conditions are specified in a separate file for every variable. In the case of a fixed value at the boundary φ_b, a face gradient $(\nabla\varphi)_f$ over the boundary face is calculated using the boundary value and the cell center value in the adjacent cell, according to :

$$\mathbf{A}_f \cdot (\nabla\varphi)_f = |A_f| \frac{\varphi_b - \varphi_C}{|\mathbf{d}|},$$

with $\mathbf{A}_f$ $[\mathrm{m}^2]$ the face area vector, $|A_f|$ $[\mathrm{m}^2]$ the face area magnitude, φ_C the cell center value for the considered variable and $\mathbf{d}$ $[\mathrm{m}]$ the length vector between the center of the cell of interest C and the position of the boundary node φ_b. Similarly, the fixed gradient boundary condition g_b represents the inner product of the gradient and unit normal to the boundary, or :

$$g_b = \left(\frac{\mathbf{A}}{|\mathbf{A}|} \cdot \nabla\varphi\right)_f$$

When the value on a boundary face φ_f is required, it can be retrieved by interpolation :

$$\varphi_f = \varphi_C + |\mathbf{d}|\, g_b$$

CHAPTER 5

Case studies

Although the model is not yet completely realistic, it is useful to already investigate the results. The model is still a rough approximation of a real filter cake build-up, so any pattern in the output might have some significance, or it might point out certain flaws in the current version of the model. All in all, it will provide insight in whether the basis of the filter cake model, as conceived in this master thesis, is a valid one.

In this chapter, the first section features a description of the chosen system as well as its modeling specifications (time step, boundary conditions, etc.). In the next step, a benchmark situation is defined with a fixed set of (realistic) parameters. Finally, the environmental conditions, i.e. model parameters, will be varied and the response of the model will be assessed through the macroscopic properties of the simulated filter cakes.

5.1 Setup

The membrane configuration for which the filter cake model is applied, is the tubular membrane. This is a tube that is constructed of a sturdy micro-porous material, where the mixed liquor is fed through the inside and the permeate is obtained at the outside (Figure 5.1(b)). These tubes are mostly installed in parallel in a tubular membrane module (Figure 5.1(a)). Tubular membranes have a reputation of being very robust in such a way that they can handle high fluxes and cross-flow velocities.

The dimensions of the tubular membrane in which filter cake formation is simulated, are an inner diameter D of 8 mm and a length L of 30 cm. Tubular membranes are usually longer, in the order of magnitude of a couple of meters, but for restricting the computational burden, only the first 30 cm is considered here. The flux over the membranes is set to 36 LMH or $1 \times 10^{-5}\,\mathrm{m\,s^{-1}}$ and the cross-flow velocity to $1\,\mathrm{m\,s^{-1}}$. As the filter cake model is still restricted to two dimensions, a two dimensional representation of the tube is found by taking a slice of the tube, as shown in Figure 5.2. This slice cuts through the center of the tube and has a thickness of d_p, so that just one particle fits in the depth of the grid.

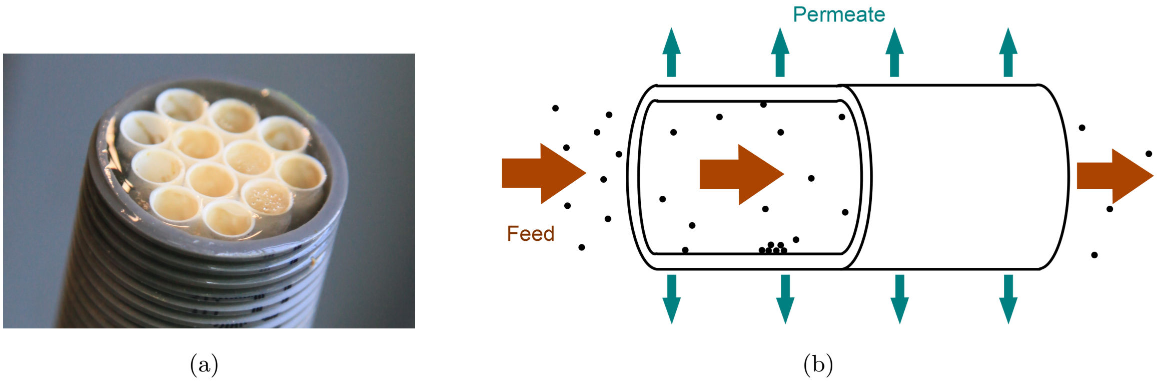


Figure 5.1: A tubular membrane module 5.1(a) and a schematic of a tubular membrane 5.1(b).

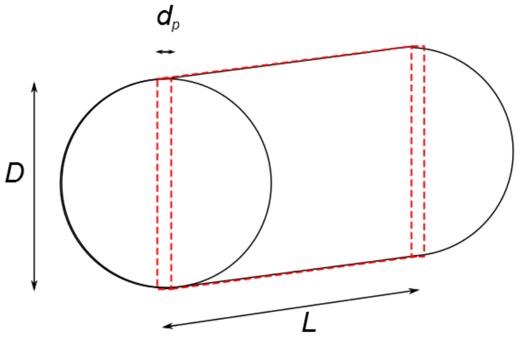


Figure 5.2: Schematic representation of the modeled system.

As particle diameter d_p, a value of 50 µm is chosen, a value which is in compliance with the granulometric distributions found in (Wisniewski and Grasmick, 1998). According to the stability criterion in Eq. (4.2), the time step Δt may at most be 1.3931×10^{-4} s, but for convenience a Δt of 0.0001 s is chosen. A sludge concentration of 10 g TSS/L is taken for the simulation, which is within the range of MBRs in municipal wastewater treatment (Rosenberger and Kraume, 2003).

5.2 Benchmark

In this section, the setup from the previous section is simulated. The results of both modeling layers are discussed, as the patterns in the resulting filter cakes are determined by the pressure and velocity profile of the continuous phase. Therefore, the latter will be discussed first, and thereafter the results of the filter cake model.

5.2.1 Continuous phase

The first step in the simulation is the generation of the flow profile in the system in OpenFOAM. For this, a mesh is defined with a resolution of 0.2 mm in all directions, generating a grid of $1500 \times 40 = 60,000$ cells. All the boundary faces normal to the x- and y-direction are denoted the `patch`-type, indicating that there is a certain flux through every face. The surfaces normal to the z-direction are set as `empty`; no calculations need to be performed in this dimension. The boundary conditions of the inlet and the upper and lower membrane consist of the Dirichlet-type, and equal $1\,\mathrm{m\,s^{-1}}$ (the cross-flow velocity), $1 \times 10^{-5}\,\mathrm{m\,s^{-1}}$ and $-1 \times 10^{-5}\,\mathrm{m\,s^{-1}}$ (the upward and downward flux over the respective membranes).

The type of flow regime that will be observed in the tube is determined by the Reynolds number Re :

$$Re = \frac{\rho v D}{\eta}. \tag{5.1}$$

It is the Reynolds number for the flow in a pipe, so the influence of the flux over the walls of the pipe on the flow regime is considered negligible. For this simulation the values in the equation above are $\rho = 1000\,\mathrm{kg\,m^{-3}}$ and $\eta = 1 \times 10^{-3}\,\mathrm{Pa\,s}$ (properties of water at $p = 1\,\mathrm{atm}$ and T =20 °C), cross-flow velocity $v = 1\,\mathrm{m\,s^{-1}}$ and tube inner diameter $D = 8\,\mathrm{mm}$. This results in $Re = 8000$, which indicates a turbulent flow regime in the tube (Massey and Ward-Smith, 1989), implying that the OpenFOAM solver for turbulent, incompressible Newtonian fluid flows needs to be used, i.e. `simpleFoam`. It searches a steady-state solution for the given situation.

The results of this steady-state simulation, expressed in terms of the simulated pressure and velocity fields, are shown in Figures 5.3 to 5.5. Keep in mind that pressure is given as kinematic pressure ψ (in $\mathrm{m^2\,s^{-2}}$), which relates to the absolute pressure p as follows :

$$\psi = \frac{p}{\rho_f}.$$

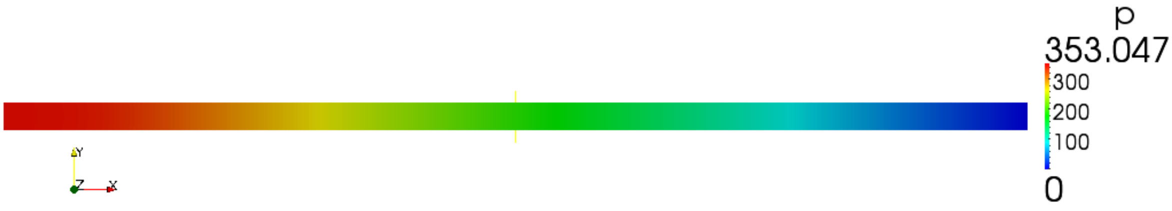


Figure 5.3: Kinematic pressure field in the tubular membrane (in $\mathrm{m^2\,s^{-2}}$).

Although the flow regime is turbulent, the pressure profile in the tube is very smooth. Along the center of the tube the pressure shows a linear drop from 353,047 Pa at the entrance to 0 Pa at the outlet as prescribed by the boundary condition. Note that the inlet pressure would

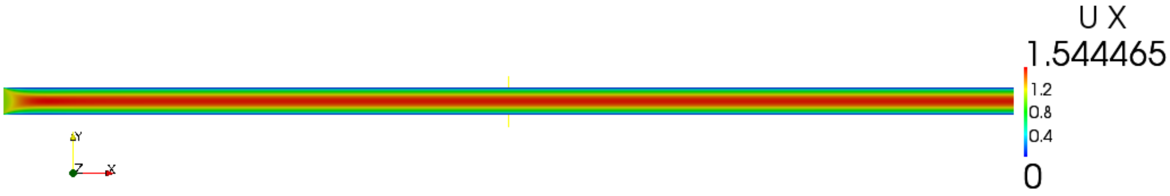


Figure 5.4: x-component of the fluid velocity in the tubular membrane (in $\mathrm{m\,s^{-1}}$).

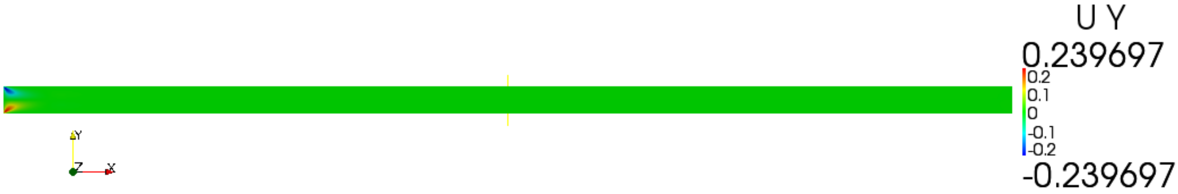
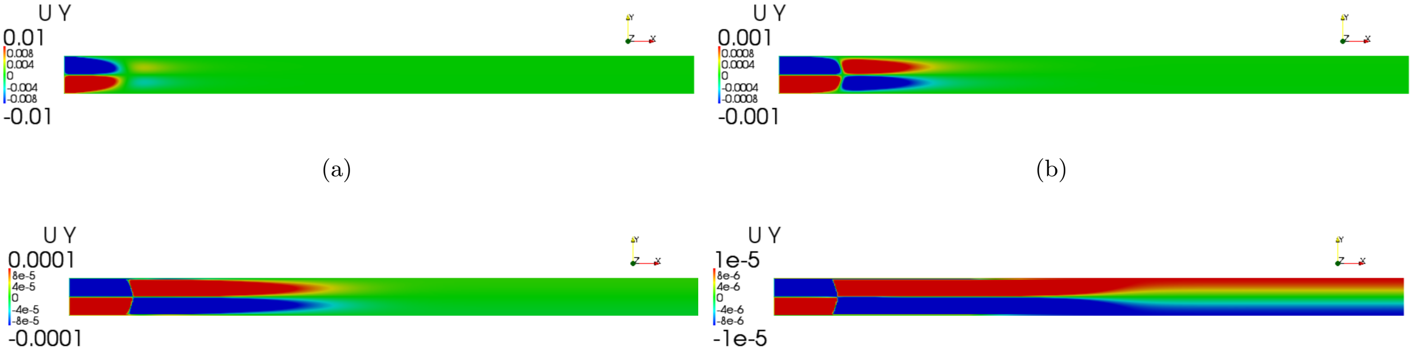


Figure 5.5: y-component of the fluid velocity in the tubular membrane (in $\mathrm{m\,s^{-1}}$).

be higher if a tube of 1 m length would have been simulated. Besides, there is practically no variation in pressure in the y-direction, so the only influence of the pressure force $\mathbf{F}_p$ will be that it pushes the particles towards the exit of the tube.

When reviewing the velocity profile, it is obvious that the simulated flow field is actually not turbulent, but instead displays a parabolic, laminar profile. This is a consequence of using the steady-state solver, which outputs an average value of the flow profile, so that it may seem laminar. But since in this trial no statements are made about the difference between filter cakes in laminar and turbulent flow regimes, this profile is nonetheless used. As it remains static and different properties in the particle motion model are simulated on the same flow field, and as it is not attempted to calibrate the model, the incorrectness of the continuous phase has no influence on the conclusions with relation to the discrete model. It only serves as a basis for the particle motion and to verify whether phenomena occurring in this flow field also translate to the resulting cake formation.

The phenomena that will be observed in relation to the velocity profile in the x-direction will depend on adhesion parameter k. Since the x-component of the fluid velocity is generally much larger than the y-component, and as the particle velocities depend on the fluid velocity, the absolute particle velocities $|\mathbf{U}_p|$ will be almost entirely determined by the former component. The absolute velocity of particle i, $\left|\mathbf{U}_p^i\right|$, determines the probability of it sticking to the filter cake once it collides with it. This means that in the region near the membranes, where $\mathbf{U}_{c,x}$ approaches zero, the particles will have a greater probability of adhering to the filter cake. This effect, however, can be neutralized and even overcome if parameter k is badly chosen. Because, even if the probability of settling is lower towards the center of the tube, also more



(a)

(b)

(c)

(d)

Figure 5.6: y-component of the fluid velocity in the first 15 cm of the tube, for scales on a different magnitude.

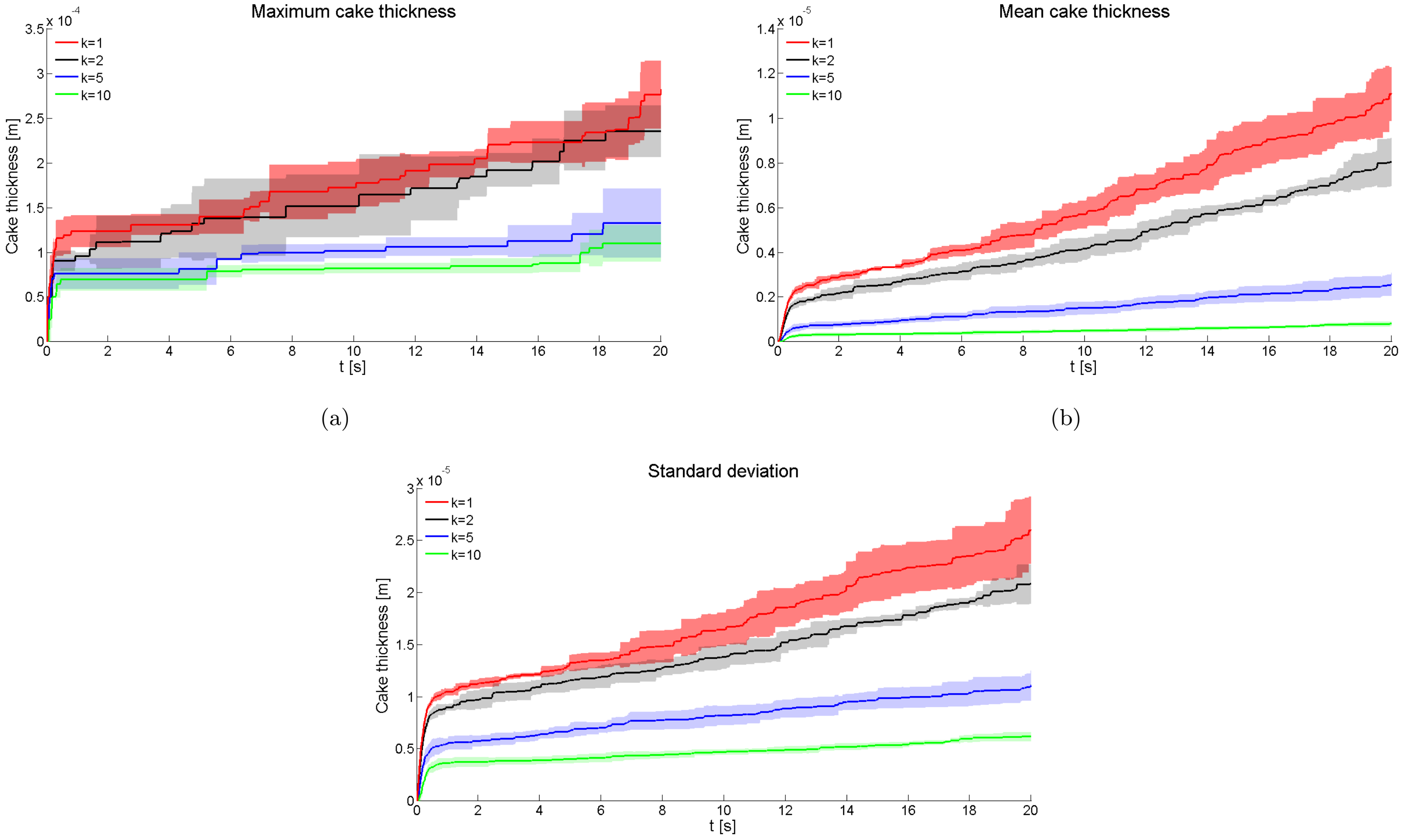
collisions will occur in this region, which, if k is too high, will lead to an exponential growth of the filter cake in this simulation. The better the determination of k, the more linear this cake growth will be. This phenomenon would also be visible in the architecture of the cake, as a lower k would cause the higher peaks to grow too rapidly and consequently block part of the incoming particles. A better choice of k will result in a more evenly distributed cake thickness.

For $\mathbf{U}_{c,y}$ two different figures are shown (Figure 5.5 and 5.6) due to the different magnitudes that occur. Clearly visible are the entry effects due to the uniform boundary condition at the inlet, which disappear 9 cm into the tube. It is expected that especially the stronger effects at the beginning of the tube will affect the particle motion and thereby the geography of the cake. After the inlet phenomena a stable profile develops, with a linear gradient for $\mathbf{U}_{c,y}$ in the y-direction, ranging from 1×10^{-5} to $-1 \times 10^{-5}\,\mathrm{m\,s^{-1}}$ (Figure 5.6(d)), the boundary conditions that indicate the membrane fluxes. It is presumed that this gradient will not have a major effect on the particle motion, since the mean velocity the tube ($1\,\mathrm{m\,s^{-1}}$) is five magnitudes higher.

5.2.2 Disperse phase

The velocity and pressure profiles are now used as a basis for the filter cake model. Initially, there are no particles present in the tube and incoming particles enter at the tube inlet ($x = 0$). The first simulations are applied to find an appropriate benchmark, i.e. to choose a value for the adhesion parameter k. Comparing the results from different outputs will be based on a few macroscopic characteristics of the filter cakes, which will be studied as a function of the simulated time. These include - along the axial direction - the maximum cake thickness, the average cake thickness, the number of adhered particles and the standard deviation of the cake thickness. Simulation times are set within a range of 15 to 20 s.

In order to compare the results across different sets of model parameters, a benchmark value for the adhesion parameter k is chosen. Its influence can be assessed by studying the simulated filter cake characteristics over time in Figures 5.7 and 5.8, respectively for the bottom and top membrane in the system. These represent the average and standard deviation for four runs per k-value.



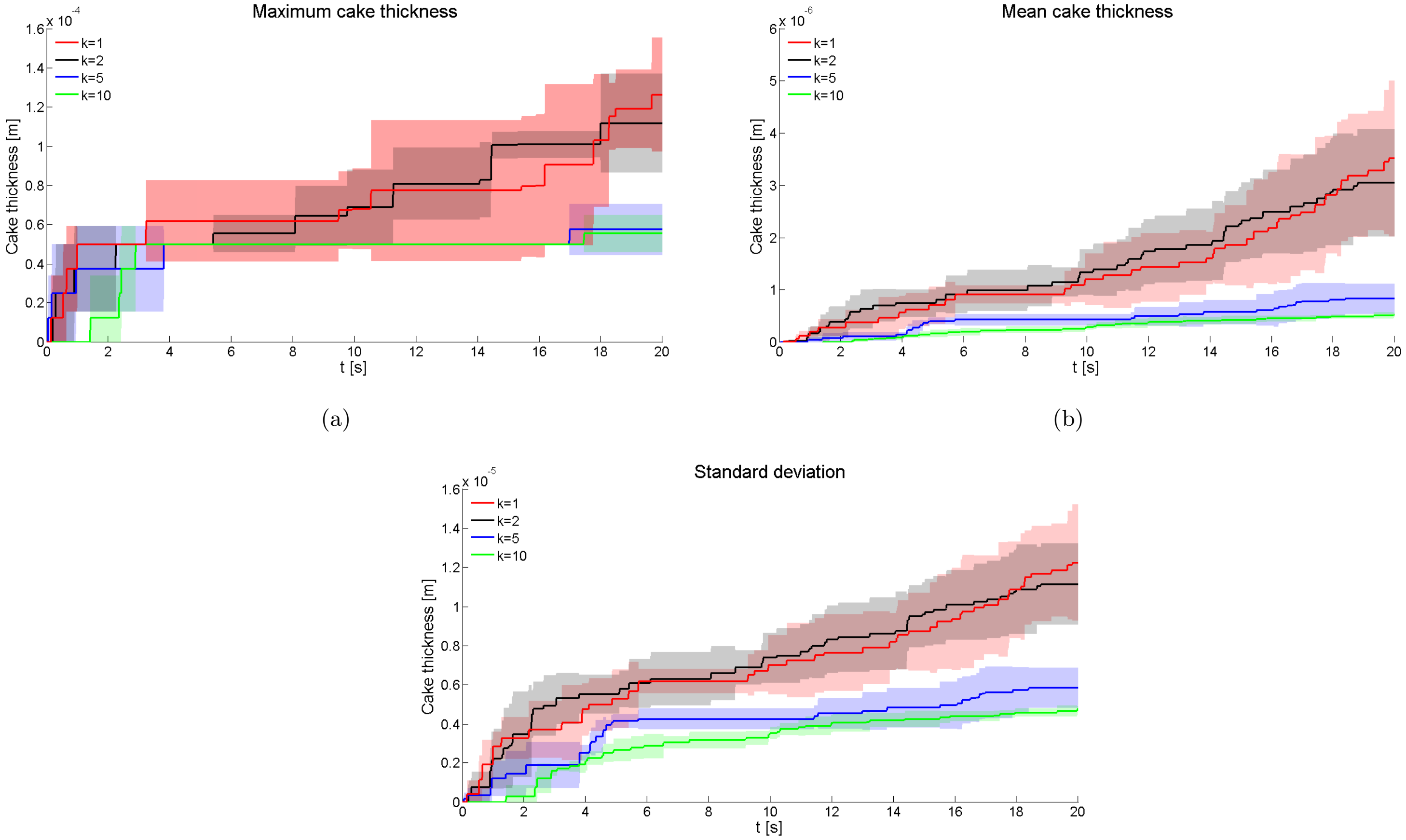
(a)

(b)

(c)

Figure 5.7: Average characteristics and standard deviation of the filter cake formed on the lower membrane versus time, for different adhesion parameters k.

The most prominent observation in the filter cake growth pattern on the lower membrane, is the steep ascent at the start of the simulation. Irrespective of k, as for every profile, the subsequent more gradual slope sets in after about 0.3 s. This is probably due to the inlet phenomena, which force a portion of the incoming particles into certain regions. If these particles slowed down sufficiently, they can deposit. When the cake grows in this region,



(a)

(b)

(c)

Figure 5.8: Average characteristics and standard deviation of the filter cake formed on the upper membrane versus time, for different adhesion parameters k.

collisions with the cake will occur at higher particle velocities, and deposition will occur less frequently, so that the steep cake growth is slowed down. This theory is corroborated by the curve of the maximum cake thickness, which rapidly ascends to values that are higher than the particle diameter (so this can not be explained by the attachment of one particle). At a certain height, which depends on k, the curve becomes less steep, indicating that the odds of adhering become much smaller. The attachment of many particles at the beginning of the simulation must be confined to a specific region on the membrane since the number of deposited particles is too small to cover a large part of the membrane and block other particles from adhering (Figure 5.7(b)). After this initial stage, the cake growth is almost linear and presumably occurs nearly homogeneously across the entire membrane, with a slight increase towards the end of the tube. This is further confirmed by the resulting filter cakes in Figure 5.9. These are presumably the particles that have already collided with the filter cake in the first part of the tube but did not adhere. After that these particles remained near the membrane and settled in the rear of the tube.

The profiles for different k-values are similar though quantitatively different, which is in line with the expectations. These discrepancies between two consecutive k values are the most pronounced between $k = 2$ and $k = 5$. Keeping Figure 4.2 in mind, this is natural, since the range of particle velocities in this case will be more affected by a change from $k = 2$ to $k = 5$ than from for example $k = 1$ to $k = 2$.

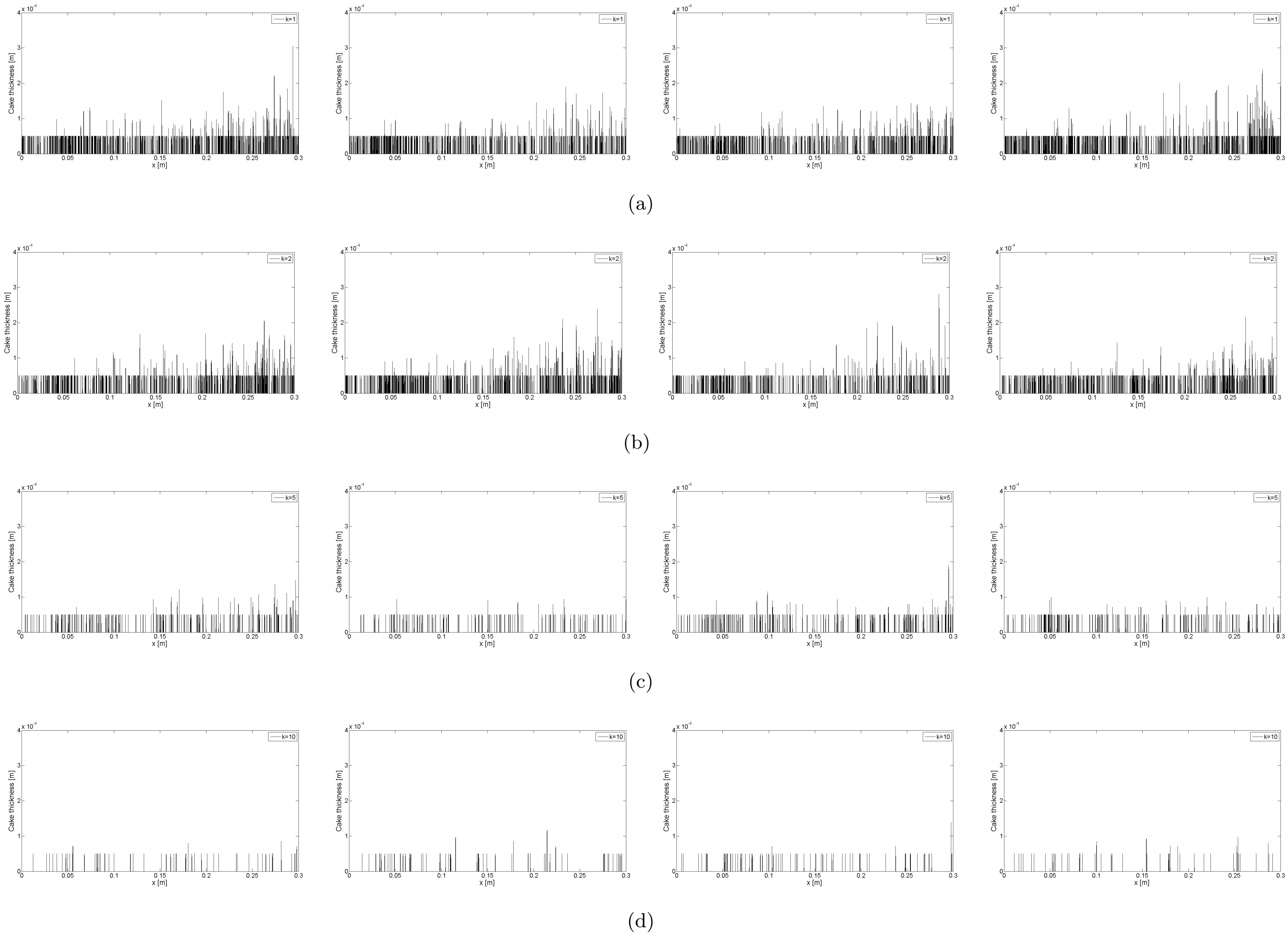
One final trend that can be discerned in Figure 5.7 is that the standard deviation among the runs generally increases as time progresses. This is explained by the fact that the cake catches more particles, which in turn increases the cake height and its influence field. The cake growth actually displays an ever increasing slope (as we will see further on, in Figure 5.11), since the diminished adhesion chance towards the middle of the tube (due to larger particle velocities) is overcompensated by the increased collision frequency. This illustrates the importance of the right choice for k, but also the need for including a detachment module in the model.

By comparing Figure 5.7 and 5.8 it can be further inferred that the upper membrane suffers to a much lesser extent from fouling in this simulation. Therefore the results in Figure 5.8 will be biased by the stochasticity of the model (which resides in the random y-coordinate of the incoming particles), so concluding anything from this figure should be approached with cause. The reason of this disproportion between the upper and lower membrane should not be sought within the balance of $\mathbf{F}_{Arch} - \mathbf{F}_g$, as this only results in an acceleration which is generally three magnitudes lower than the drag force. Since the flow profile is axisymmetric, the discrepancy can only have arisen from a bug in the model.

That the curves for $k = 1$ and $k = 2$ stay within the same range is once again illustrative for the fact that the particle velocity range in this tube is not significantly affected by a choice between these two values. At the other side of the spectrum, where the k values vary between $k = 5$ and $k = 10$, there is not much that can be concluded from the curves. There is a difference, which could be described as the filter cake developing the same way for both scenarios, but only slower for $k = 10$. But due to the lack of a large number of simulation data, which was not possible within the time frame of this thesis, these conclusions cannot be more firmly backed here.

Finally, a look at the final cake geographies (on the lower membrane) depicted in Figure 5.9 indicates that the deposition of particles occurs more frequently towards the rear of the tube. Probably this may again be attributed to the inlet effects, as these result in the particles being launched towards the membranes. As a consequence of the larger velocities in the x-direction, most particles will have traveled further in the tube before their adhesion.

In the remainder of this chapter, different scenarios will be studied for $k = 2$. Not because it seems the most realistic value, on the contrary, it shows too high peaks (Figure 5.7(a)) for a 20 s-old filter cake. But in order to perceive some of the model trends within the simulated time frame, $k = 2$ is best chosen since a sufficient number of particles can deposit in order to



(a)

(b)

(c)

(d)

Figure 5.9: Filter cake outputs on the lower membrane after a simulation time of 20 s for $k = 1$ 5.9(a), $k = 2$ 5.9(b), $k = 5$ 5.9(c) and $k = 10$ 5.9(d).

derive these trends. For the same reason, it is opted to restrict the discussion to the results of cake on the lower membrane. Also, its variance is smaller than that of the $k = 1$-curves, making it preferable to the latter parameter setting. The parameter values of the benchmark are summarized in Table 5.1.

Table 5.1: Benchmark values.

Parameter		Value
D	tube inner diameter	$0.008\,\mathrm{m}$
L	tube length	$0.3\,\mathrm{m}$
$\overline{\mathbf{U}}_c$	cross-flow velocity	$1\,\mathrm{m\,s^{-1}}$
J	membrane flux	$1 \times 10^{-5}\,\mathrm{m\,s^{-1}}$
TSS	total suspended solids concentration	$10\,\mathrm{kg\,m^{-3}}$
d_p	particle diameter	$50\,\mu\mathrm{m}$
ρ_p	particle mass density	$1,003\,\mathrm{kg\,m^{-3}}$
k	adhesion parameter	$2\,\mathrm{s\,m^{-1}}$

5.3 Scenario analysis

TSS

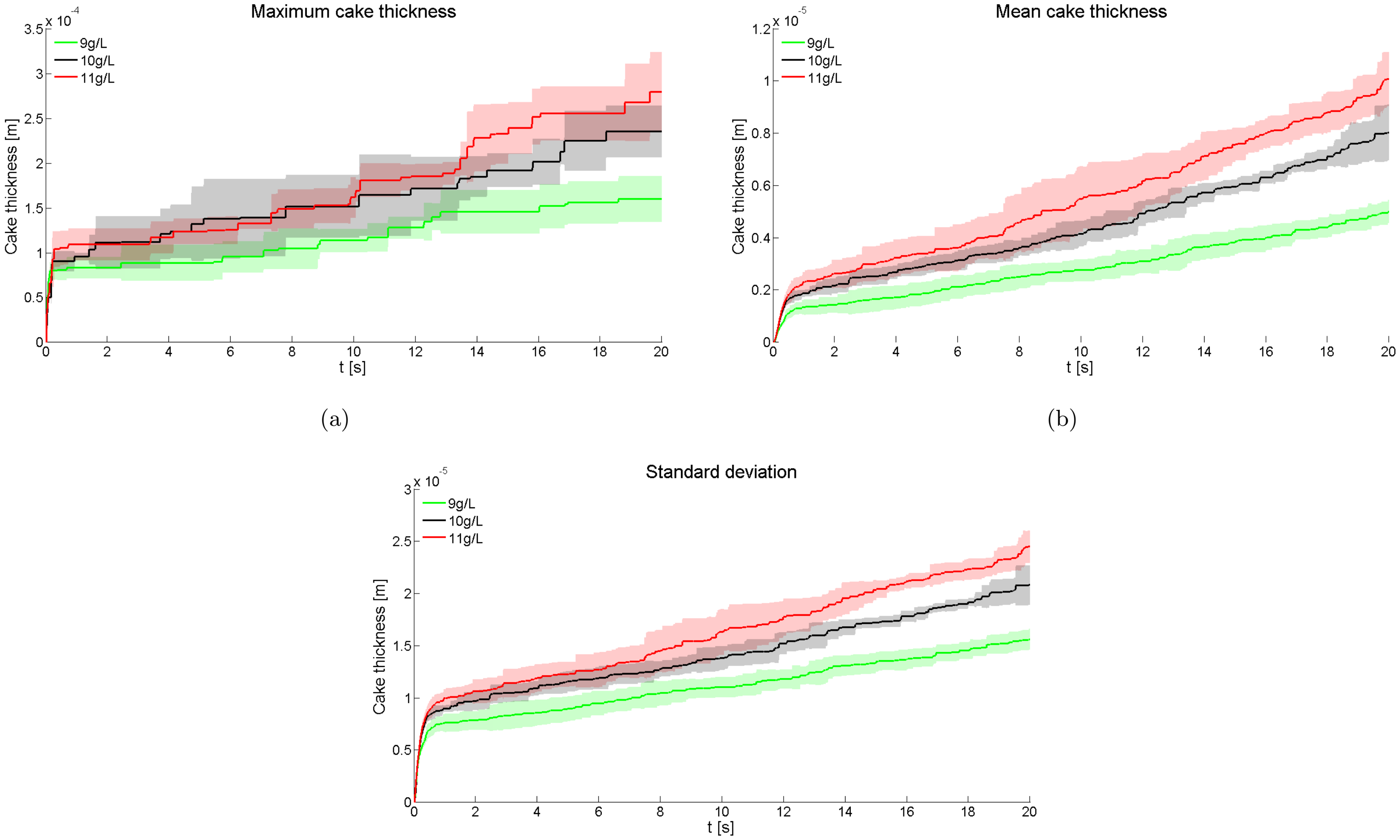
In literature, the correlation between TSS and the fouling propensity is described as complex, with reports of an increasing TSS having both a negative as well as an insignificant impact on the membrane filterability (Judd, 2010). Here, the effect is merely straightforward, as a higher TSS-concentration implies more particles in the system, and therefore a more pronounced filter cake formation (Figure 5.10).

More precisely, the rate of fouling increased upon augmenting the TSS-concentration, as can be derived from the slopes of the curve in Figure 5.10(b). This conclusion is confirmed by the next case, where this trend is very visible.

Particle diameter

In this section, the benchmark simulations are compared to the cases for which d_p =45 µm and d_p =55 µm (Figure 5.11). Given a constant sludge concentration of 10 g TSS/L, this implies that the number of particles present in the system (excluding the deposited ones) is larger if d_p =45 µm (22,568 particles vs. 15,108 in the case of d_p =55 µm).

This case actually serves as an example of the influence of a badly chosen value for k, as well as the need for a force balance on the deposited particles. Firstly, in this case it is perceived that, due to the flow profile, the upper membrane actually experiences a fouling regime with a steeper slope than the lower membrane (Figure 5.12). When too many particles are allowed to attach, such as in this case, this increases the height of the highest peaks to beyond the tube radius (Figure 5.12(a)). Moreover, these peaks are not very wide, as can be derived from the large standard deviation (Figure 5.12(c)). Therefore, this phenomenon is unrealistic, and this depicts exactly the consequence of a bad k-value. The statement that an increased collision frequency on the long term overcompensates for the effect of the adhesion



(a)

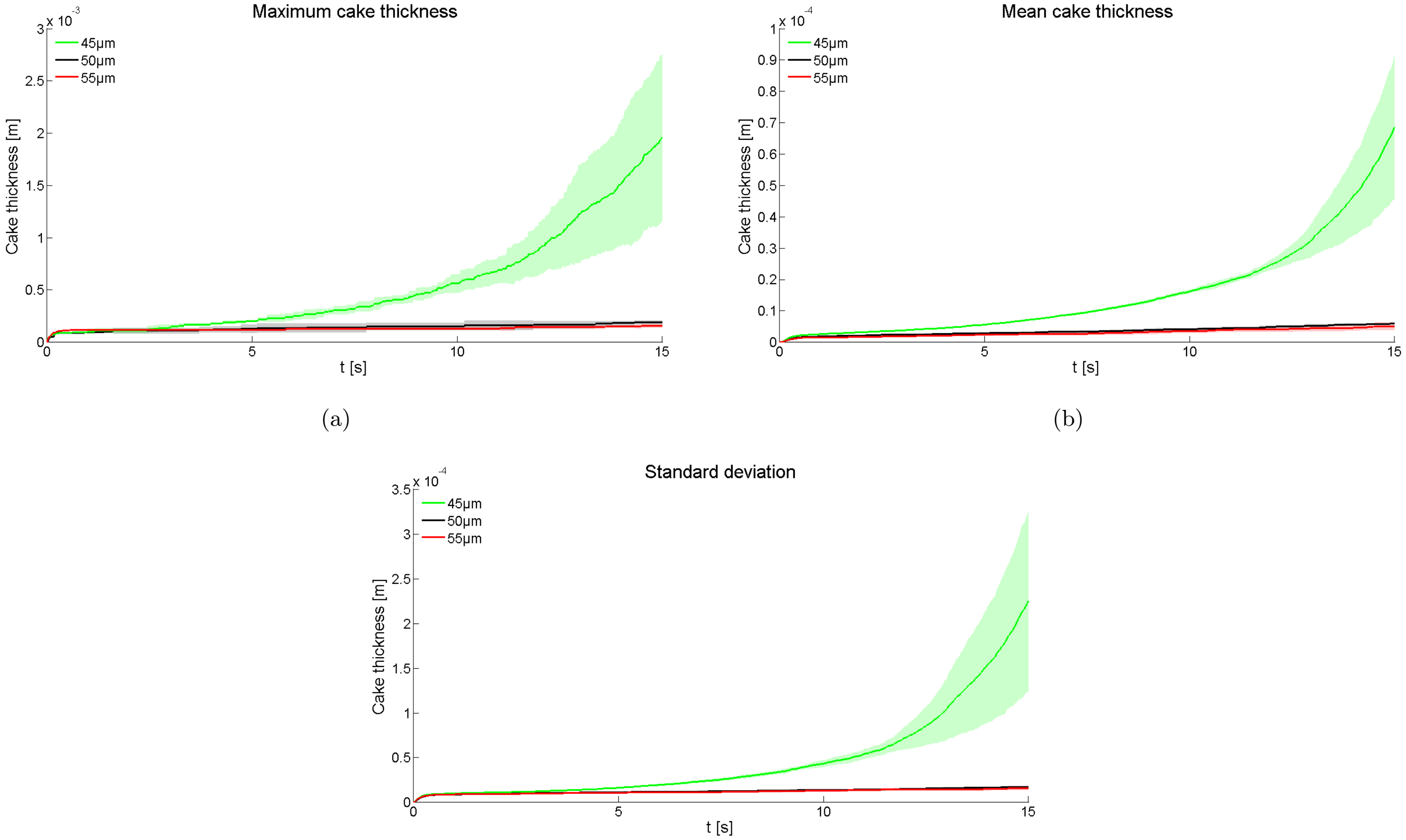
(b)

(c)

Figure 5.10: Average characteristics and standard deviation of the filter cake formed on the lower membrane versus time, for $k = 2$ and for varying TSS-concentrations.

probability is demonstrated by this figure (Figure 5.12(c)). This again emphasizes that the adhesion probability P alone cannot correct for detachment of particles.

One important note here is that the effect of a too large value for k, is actually aggravated by a flaw in the used adhesion mechanism. This is due to the fact that deposited particles can create a vertical wall in the filter cake boundary. This in turn means that a number of particles will be placed much higher than their actual deposition point. This again propagates this vertical border and gives a bias in the resulting filter cakes. This disproportion with the reality has an increasing influence during a simulation, fortunately it can be overcome with the extension of two mechanisms, namely detection of the upper and lower boundary of the filter cake and, most of all, calculating the force balance over the attached particles (see Chapter 6).



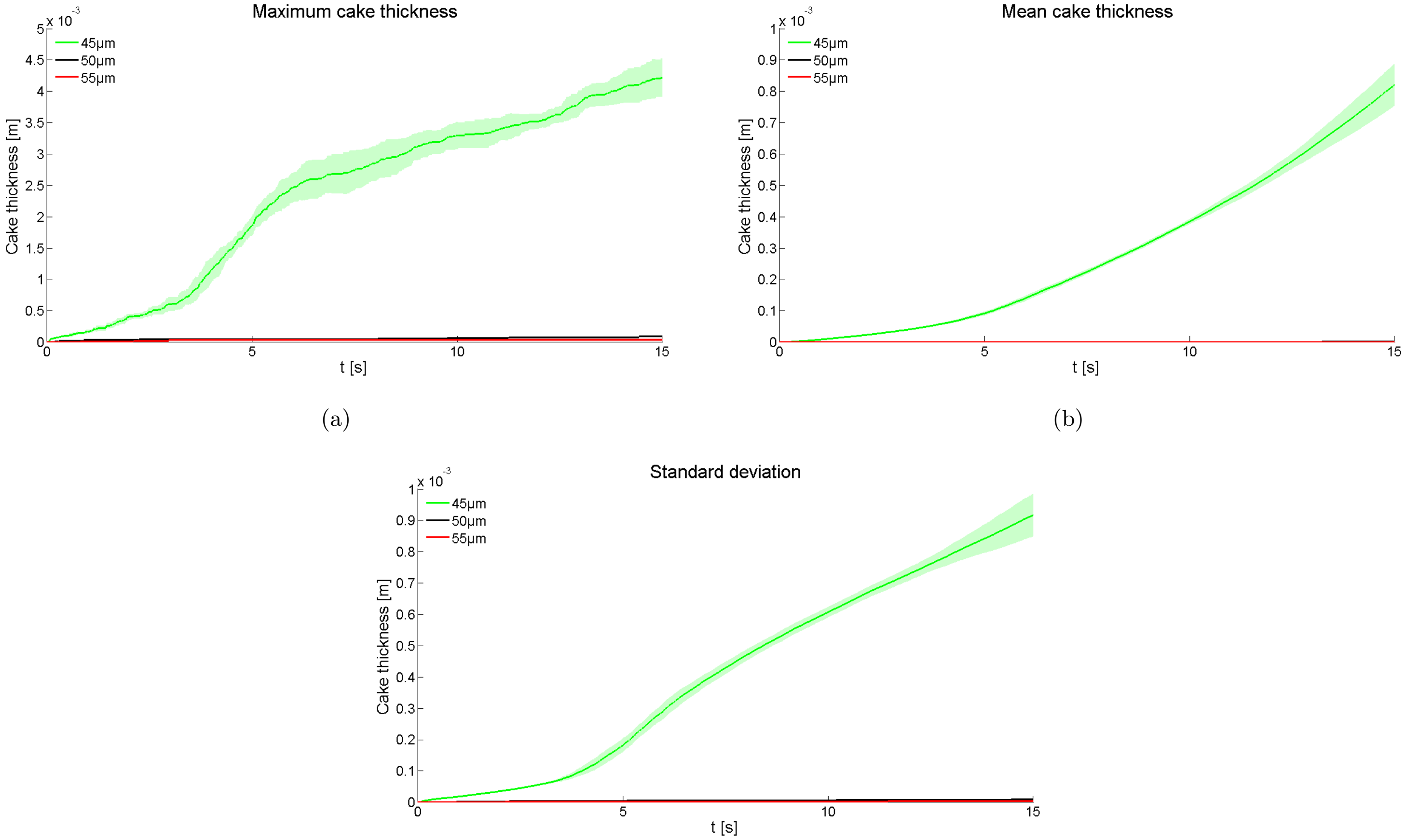
(a)

(b)

(c)

Figure 5.11: Average characteristics and standard deviation of the filter cake formed on the lower membrane versus time, for $k = 2$ and for varying particle diameters d_p.

It can also be discerned from Figure 5.12 that the clearly exaggerated cake growth in the case of $d_p = 45$ µm shows different regimes: it starts off with a gradual cake growth until about 3.5 s, then both the maximum (Figure 5.12(a)) as well as the standard deviation (Figure 5.12(c)) of the cake layer increase steeply until after about 6.5 s, this phenomenon ebbs and cake growth again shows a gradually increasing slope. The steep increase, however, did not manifest itself in the curves of the other cake characteristics, as these simply exhibit an ever increasing fouling rate. The course of cake formation behind these patterns then goes as follows: in the first stage, fouling is spread evenly over the membrane, though after 3.5 s a few peaks are catching a lot of particles, and they shoot up. Their excessive heightening is blocked after 6.5 s, where the fluid velocities are too high to permit a continuation of this particle deposition rate. Meanwhile, however, these high peaks block a large part of the particles in the stream, which also aggravates the fouling regime and explains the curve in Figure 5.12(b).



(a)

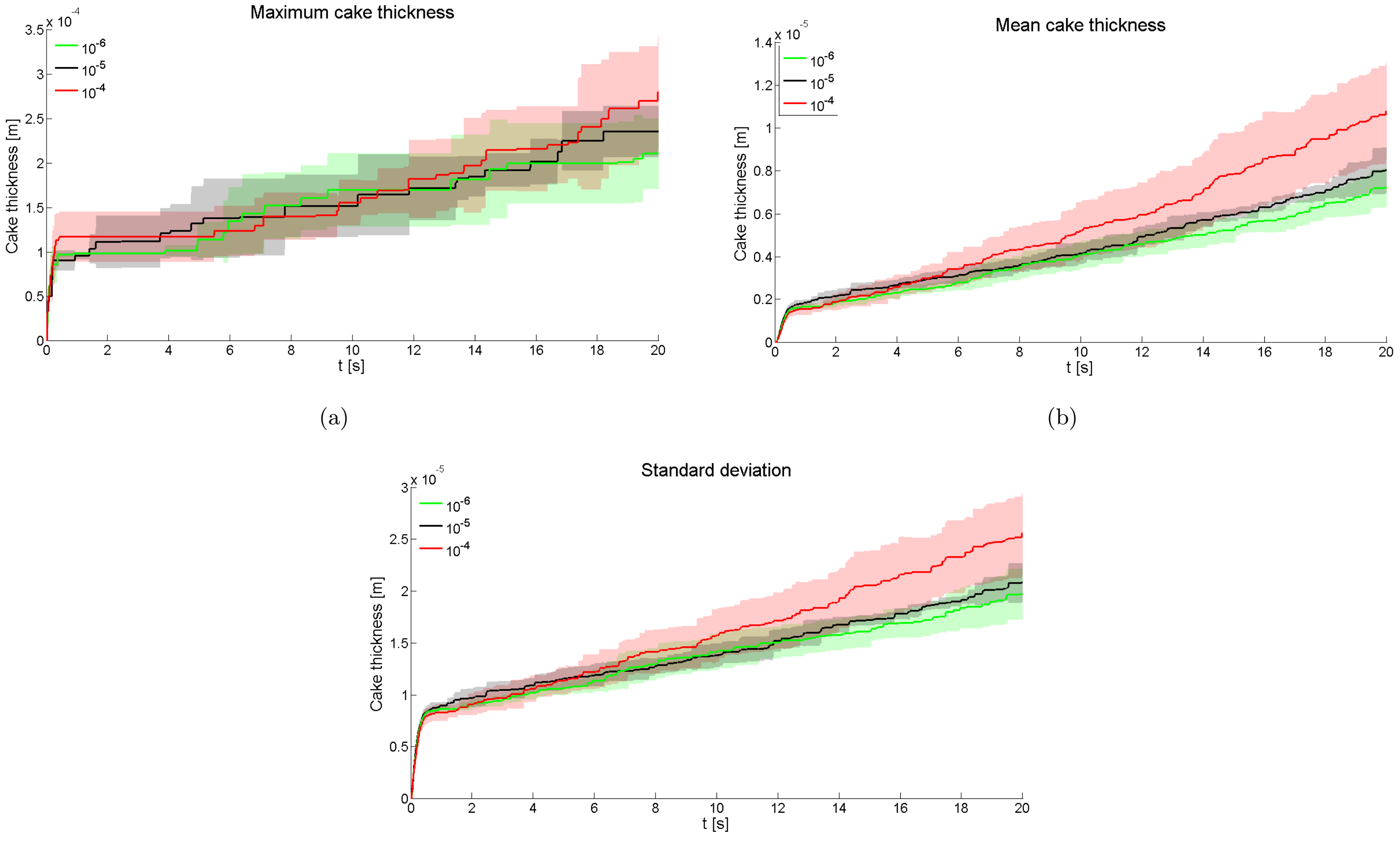
(b)

(c)

Figure 5.12: Average characteristics and standard deviation of the filter cake formed on the upper membrane versus time, for $k = 2$ and for varying particle diameters d_p.

Flux

Next, the influence of a different membrane flux J is assessed. Note that the chosen fluxes, $J = 1 \times 10^{-4}$ and $J = 1 \times 10^{-6}$, rarely occur in MBRs. The conclusion within this paragraph is also that the flux does not have a significant influence on the movement pattern of the particles. Increasing the flux tenfold only shows a minor or almost no increase in fouling rate (Figure 5.13). This indicates the small impact of the flux on the velocity profile. This assumption, however, might also be emblematic for the inaccurate way in which the flow profile is simulated. For a discussion on this, we refer to Chapter 6.



(a)

(b)

(c)

Figure 5.13: Average characteristics and standard deviation of the filter cake formed on the lower membrane versus time, for $k = 2$ and for varying membrane flux J.

CHAPTER 6

Discussion and future work

The cake is a lie.
— *writing on the wall in Portal, Valve Corporation*

In the previous chapter it was already emphasized that from the generated results, conclusions could only be made regarding the model performance. This was due to the low number of data from simulations, since a larger number was not possible within the time frame of this master thesis, and also to a bug in the code resulting in a discrepancy between adhesion on the upper and lower membrane in an axisymmetrical flow profile. Yet, we were able to establish the influence of adhesion parameter k and, to a lesser extent, of the flux and the TSS-concentration.

It must also be said that, with the current framework in place, a foundation is in place for an important research line of fouling modeling, which can substantially contribute to a more thorough understanding of the process involved in filter cake formation. In order to serve that purpose, the model still needs to be expanded and some of the assumptions need to be questioned. This is done in this chapter, as the current assumptions and implementations are discussed and an outline of the future improvements is given. Then, to conclude, some guidelines regarding the calibration of the model are given.

6.1 Regarding filter cake representation

The most prominent imperfection of the current filter cake model is the complete immobilization of the adhered particles. An adamant resistance against any form of shear is, first of all, very unrealistic, but it also has a great influence on the simulation output. Since the particles are not able to roll over each other or push each other away, or sweep each other back into the bulk, the maximum cake thickness will always be an overestimation of any realistic filter cake. This flaw is only partly compensated by the adhesion probability feature. It does correct for the number of particles that settles in regions with a higher fluid velocity, but as the filter cake peaks get higher they also catch more collisions, which might just compensate the adhesion probability. This also emphasizes the importance of making the right choice for parameter k in the adhesion probability.

So the first improvement of the current filter cake model should be the implementation of a force balance over the deposited particles. It is analogous to the one that is now in place for the free particles, in the way that forces also arise from the velocity profile of the surrounding fluid, only in this case, also normal forces need to be determined. This extension of the model is necessary for a realistic cake architecture in which particles are accurately positioned after their attachment, and for modeling detachment. The latter should be done similarly to the adhesion check, with an exponential function describing the probability of detachment, which would be in function of the resulting force on the adhered particle. This probability is introduced in order to account for the fact the filter cake is actually a biofilm, with a extracellular matrix of EPS keeping the bacterial cells together.

Also, the influence of the artifact in the cake layer formation that was mentioned in Section 5.3 would be much less in case a force balance would move the deposited particles into their appropriate positions. Yet, also an adjustment to the current collision detection algorithm could provide a significant improvement. This would consist of a collision detection with the lower boundary of the cake, a boundary that is drawn by the particles that are the closest to the membrane, just as the upper cake boundary is now determined by the particles at the top of the filter cake. Of course, this implies that a number of these boundary detecting arrays should be implemented for a truly accurate representation of the cake's 2D-architecture, especially if complex structures occur. However, it is assumed that, due to the shear rate and pressure regime in an MBR, any complex cake formations, such as the branches formed in biofilms, will not occur very often, so that the proposed cake boundary detection algorithm, combined with a force balance calculation of the adhered particles, suffices.

6.1.1 Particle size distribution

The assumption that all particles in the mixed liquor are spherical and of the same size does not apply to a real MBR, as the biological aggregates grow or break apart according to the shear regime. Particle size distributions (PSD) have been derived for MBR sludge in Wisniewski and Grasmick (1998) and Jiang et al. (2007), in which the mean particle diameter showed great dependency on the flow regime within the reactor. A different PSD will also impact the formation of the filter cake, as the particles will be found differently stacked. A different structure of the filter cake will show itself mainly in its porosity, which is of importance in processes such as filter cake compaction and the permeability decline. Moreover, the denser and more compact cake structure is bound to affect the attachment/detachment equilibrium, as well as the filtration resistance.

It would not be too much trouble to implement a PSD into the filter cake model, but it would require that the maximum time step would be determined by the smallest occurring particle diameter d_p (Eq. (4.2)). Next to that though, the adhesion check needs to be adapted to the fact that different particle diameters occur. This means that either an actual collision detection algorithm for spheres needs to be implemented, or that the adhesion algorithm needs to be tweaked again. Due to the large number of particles present in the simulation (a number that increases with increasing TSS and decreasing mean particle diameter) this method might be computationally intensive, so the modeler would have to search for efficient algorithms for fast collision detection between spheres (Kim et al., 1998).

Yet, simply a revision of the adhesion algorithm could also offer a solution. Instead of detecting the intersection of a particle trajectory with a contour around the filter cake, now the actual cake boundary should be used, and for the collision check the smallest distance between the particle trajectory and this boundary could be checked. If this distance is smaller than the particle radius, then the particle has touched the filter cake.

On a more advanced level, even the effect of different particle geometries could be assessed. Namely, it has been reported that biological cells with rod-like shapes can significantly affect the porosity and the compressibility of the cake, and that this property is thus of importance to cake compaction (Mota et al., 2002). These shapes could then be modeled as aggregates of spheres (Hubbard, 1996).

6.1.2 Adhesion probability

With the adhesion probability, a parameter is brought into the model that requires calibration, namely k. This parameter incorporates the assumption that particles with a lower momentum have a smaller chance of adhering. But, strictly speaking, it has only brought into account the particle velocity. In a system where all particles have the same mass, this is applicable, but, as previously mentioned, when a PSD is used, this property should be adjusted. Indeed, the momentum P of a particle is given by $P = m_p v$, so the velocity term v in Eq. (4.6) should be replaced with P.

This feature of the model comes in very handy, as it incorporates various different system properties, such as the EPS-concentration, the particle charge and the roughness and hydrophobicity of the membrane. The latter though, also implies that different k-values need to be determined for collision with the membrane and with the filter cake separately, and that two parameters have to be calibrated. As for the influence of the EPS-concentration: studies have shown that EPS influence the fouling regime, and Yun et al. (2006) even concludes that the spatial distribution of EPS over the membrane has a significant impact on the membrane filterability. This could be included in the model, and simulated by implementing diffusion equations into OpenFOAM. Yet, this is another extension that would contribute to the complexity of the model (see further), and since the influence of EPS has already been questioned in literature (Drews, 2010), this should not be placed high on the priority list of necessary features in the cake model.

Finally, collisions (with the cake/membrane) are assumed to be totally elastic. This does not hold in reality, as the particles in an MBR largely consist of biological cells - which are no rigid spheres - and as the presence of the fluid will also have an influence. However, the bias resulting from this assumption is considered small, certainly if the time step is taken small enough for the particle motion to be corrected by the fluid velocity profile.

6.2 Regarding the continuous phase

The first assumption that can be challenged is the one of the simulation of the continuous phase being based on the properties of clean water. Next to the components already present in the wastewater that is fed to the MBR, also the presence of EPS in the fluid will determine the actual viscosity of the continuous phase. Empirical correlations have been composed between the TSS and the viscosity of the mixed liquor in Xing et al. (2001). Also, it has been found that the mixed liquor in an MBR does not behave as a Newtonian fluid (Defrance et al., 2000). Implementing the adjusted viscosity was not relevant in this master thesis, as the results could not be compared to experimental filter cakes. But when a calibration is carried out, for example for adhesion parameter k, this needs to be taken into account in the CFD-simulation and the force balance, especially since the high TSS present in MBR can alter this parameter significantly (Xing et al., 2001). So it is recommended that the viscosity of the sludge is measured and applied in the filter cake model.

It must also be said that the boundary conditions in Chapter 5 are too simple to reflect a real tubular membrane, for in reality, not the cross-flow velocity is set but rather the pumping pressure. The boundary conditions at the inlet should therefore be of the Neumann-type for the (x-component of the) velocity and a Dirichlet for the pressure, which then represents this pumping pressure. The boundary conditions for the membranes are also not accurate, as was visible in the pressure profile in Figure 5.3, which should in reality show some difference in the y-direction. It should have been set to the TMP, instead of $p = 0$.

However, this holds only for as long as fouling has no considerable influence on the flux. To correct for filter cake formation on the membrane, the Carman-Kozeny equation (Eq. (2.2)) should be used to calculate the pressure drop over the filter cake. This implies though, that data from the modeling of the disperse phase is transferred to the simulation of the continuous phase, i.e. from MATLAB to OpenFOAM, which will take some programming effort. With this connection in place, data from the MATLAB output, containing the local pressure drop due to the cake, should be transferred to the boundary condition over the membrane in OpenFOAM. This pressure drop can be calculated provided that the local porosity ϵ and thickness $\Delta\ell$ of the filter cake are known; the latter is already retained for the collision detection, and the former can be calculated as :

$$\epsilon = 1 - \frac{n_p V_p}{\Delta\ell \Delta x \Delta z}, \tag{6.1}$$

with n_p the local number of particles, V_p the particle volume and Δx and Δz the respective resolutions along the x- and z-coordinate axes, given that $\Delta\ell$ is taken in the y-direction.

As a final note here, it needs to be mentioned that tubular membranes, when they are applied in an MBR configuration, are usually positioned vertically, instead of horizontally as in the case studies in Chapter 5. This configuration is namely much more facilitating to the application of air scouring as a cleaning mechanism. This has an influence on the pressure boundary condition, which will be higher at the inlet to compensate for the pressure of the water column in the tube. Evidently, this makes only a small difference. Next to that, also the gravity field needs to be accounted for when simulating the flow profile.

Regarding the particle motion

Only if an adequately fast collision detection algorithm would be found, the detection of the impaction of the freely moving spheres in the simulation could be performed. Else the collision detection would severely elevate the computational time due to the vast number of particles present in the bulk of an MBR sludge. It is a different case than the filter cake collision detection, since now all spheres are moving. If not implemented, one not only misses out on the effect of elastic collisions between the particles, but then there is also no way of modeling the formation of aggregates. One way in which this contact detection could be implemented is to only apply it on particles that are found in the same grid cell, which needs to be of an appropriate size in order to not perform the detection excessively. This could be realized by the application of the `unique` function in MATLAB, which selects the unique entries in an array. This function, however, is already being used in the current filter cake model, more specifically in the adhesion detection, where it has shown to be a large contributor to CPU time.

The conclusion is that the assumption of no collisions occurring between particles in the bulk fluid will likely remain in place in the further development of the filter cake model. Yet it can be compensated by, on one side, implementing PSDs that already account for aggregate formation (such as those in Jiang et al. (2007)). This is preferable to adding another degree of complexity to the model and aggravating the ratio of CPU time to real time.

6.3 Advanced extensions

Essentially, the filter cake model can easily be expanded to three dimensions, the only thing to do is to apply the same force balance as in the other Cartesian coordinate axes to the z-direction and expand the gradients and Laplacians to take into account the third dimension.

The continuous phase needs to be simulated in 3D, which is exactly what CFD is all about. The only concern is the number of particles that would be under consideration. Within the case, a slice of the tubular membrane was taken as representative for the whole tube, which allowed for the number of particles that was present in the system (excluding the deposited ones) to be as low as 18,280 (this depends on the TSS-concentration, the particle diameter d_p and its density ρ_p). If the whole tube was considered, it would mean that, for every time step, the force balance would need to be calculated for 2,297,109 particles, and this for the first 30 cm of a tube.
Another feature of the advanced model is the modeling of the aeration of the membrane, although, this would be a great addition to the computational burden of the entire model, as well as it would require some programming effort. So, to conclude, the inclusion of the aforementioned measures depends on the purpose to which the model serves.

6.4 Calibration guidelines

The calibration of the adhesion parameter k - as well as extra parameters that follow from the inclusion of any of the model extensions mentioned in this chapter - requires that experimental filter cakes are compared to the simulated ones. This can be done according to filter cake characteristics: the average cake thickness, the standard deviation of the cake thickness, the coverage of the membrane and macroscopical characteristics of the fouling layer. Measuring these characteristics in an experimentally generated filter cake can be done with various tools, but one seems particularly interesting, namely a stylus profiler.

CHAPTER 7
Conclusion

The purpose of this master thesis was to construct the foundations of a model that simulates filter cake formation in membrane bioreactors with an accuracy that goes down to a microscopic level. The resulting Euler-Lagrangian framework is considered as a valid one, as it allows for an accurate representation of particle motion, based on a fluid velocity profile that is generated in a flexible software package.

The results in this dissertation have proven that the conceived model yields interesting results, despite the underlying assumptions. The conclusions from those results are summarized as follows:

- the force balance on a particle in a fluid has been implemented correctly. The stability issues were resolved, although this resulted in some restrictions.
- the adhesion parameter k has a large share in the resulting fouling rate.
- increasing the TSS-concentration means increasing the number of particles in the system, and proportionally increases the fouling rate.
- lowering the particle diameter resulted in a much higher number of particles, and illustrated that filter cake growth can go out of bounds when k is badly chosen.
- changing the flux within a range that can be found in MBRs does not significantly alter the fouling rate.
- the assumptions that particles adhere according to their momentum, that particles do not influence each others motion can still lead to an accurate depiction of the reality, and can be kept in the model. On the other hand, the assumptions regarding the static flow profile, the equally-sized particles and the immobilization of adhered particles will not hold in a real MBR, and should be revised.
- the ratio of CPU time to simulated time should be in the range of minutes to simulate seconds.

- the attachment algorithm can be improved, and cake growth should be extended with a detachment module.

Still, we were able to assess the influence of a few simulation parameters, of which the most notable one is the adhesion parameter k. The resulting filter cakes have, moreover, emphasized that the latter has a great influence on the model output, as it completely determines the fouling rate.

Another conclusion from the results was that the current version of the model does not allow for simulating dynamic filter cakes. Therefore, the first things that should be dealt with are the calculation of the resulting forces on the deposited particles and, by extension, the possibility of detachment. Also, the adhesion algorithm is up for revision, and should be improved with the proposed mechanisms.

Beyond that, a number of extensions of the model is possible, which should be accessed according to the purpose for which the model will be engaged. If the final model should be used as a research tool, to gain more insight in the fouling process, it should describe its constituting processes thoroughly in detail. In case it is applied to, for example, optimize a certain aeration configuration, it would suffice if some processes would be simplified. It should for every case be assessed how complex the model should be, considering the CPU time required for the simulations.

All in all, this filter cake model could become an important tool in getting a better understanding of the processes in MBR, and in this master thesis a decent framework was developed where it can be built upon.

Bibliography

Atkinson, B., Busch, A. W., and Dawkins, G. S. (1963). Recirculation, reaction kinetics, and effluent quality in a trickling filter flow model. *Journal (Water Pollution Control Federation)*, pages 1307–1317.

Atkinson, B. and Daoud, I. S. (1970). Diffusion effects within microbial films. *Transactions of the Institution of Chemical engineers and the Chemical Engineer*, 48(7-10):T245.

Batchelor, G. K. (2000). *An introduction to fluid dynamics*. Cambridge university press.

Cho, B. and Fane, A. (2002). Fouling transients in nominally sub-critical flux operation of a membrane bioreactor. *Journal of Membrane Science*, 209(2):391–403.

Defrance, L., Jaffrin, M. Y., Gupta, B., Paullier, P., and Geaugey, V. (2000). Contribution of various constituents of activated sludge to membrane bioreactor fouling. *Bioresource Technology*, 73(2):105–112.

Dorgan, A. and Loth, E. (2007). Efficient calculation of the history force at finite Reynolds numbers. *International Journal of Multiphase Flow*, 33(8):833–848.

Drews, A. (2010). Membrane fouling in membrane bioreactorsCharacterisation, contradictions, cause and cures. *Journal of Membrane Science*, 363(1-2):1–28.

Drews, A., Vocks, M., Bracklow, U., Iversen, V., and Kraume, M. (2008). Does fouling in MBRs depend on SMP? *Desalination*, 231(1-3):141–149.

Faxén, H. (1922). Der Widerstand gegen die Bewegung einer starren Kugel in einer z{ä}hen Fl{ü}ssigkeit, die zwischen zwei parallelen ebenen W{ä}nden eingeschlossen ist. *Annalen der Physik*, 373(10):89–119.

Fenu, A., Roels, J., Wambecq, T., De Gussem, K., Thoeye, C., De Gueldre, G., and Van De Steene, B. (2010). Energy audit of a full scale MBR system. *Desalination*, 262(1-3):121–128.

Hubbard, P. (1996). Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics (TOG)*, 15(3):179–210.

Jiang, T., Kennedy, M. D., Yoo, C., Nopens, I., van der Meer, W., Futselaar, H., Schippers, J. C., and Vanrolleghem, P. A. (2007). Controlling submicron particle deposition in a side-stream membrane bioreactor: a theoretical hydrodynamic modelling approach incorporating energy consumption. *Journal of membrane science*, 297(1):141–151.

Judd, S. (2010). *The MBR book: principles and applications of membrane bioreactors for water and wastewater treatment.* Elsevier.

Kim, D.-J., Guibas, L. J., and Shin, S.-Y. (1998). Fast collision detection among multiple moving spheres. *Visualization and Computer Graphics, IEEE Transactions on*, 4(3):230–242.

Kissel, J. C., McCarty, P. L., and Street, R. L. (1984). Numerical simulation of mixed-culture biofilm. *Journal of Environmental Engineering*, 110(2):393–411.

Kraume, M. and Drews, a. (2010). Membrane Bioreactors in Waste Water Treatment - Status and Trends. *Chemical Engineering & Technology*, 33(8):1251–1259.

Kreft, J. U., Picioreanu, C., Wimpenny, J. W., and van Loosdrecht, M. C. (2001). Individual-based modelling of biofilms. *Microbiology (Reading, England)*, 147(Pt 11):2897–912.

Massey, B. S. and Ward-Smith, A. J. (1989). *Mechanics of fluids*, volume 45. Springer.

Maxey, M. R. and Riley, J. J. (1983). Equation of motion for a small rigid sphere in a nonuniform flow. *Physics of Fluids (1958-1988)*, 26(4):883–889.

Mei, R. (1992). An approximate expression for the shear lift force on a spherical particle at finite Reynolds number. *International Journal of Multiphase Flow*, 18(1):145–147.

Meng, F., Chae, S.-R., Drews, A., Kraume, M., Shin, H.-S., and Yang, F. (2009). Recent advances in membrane bioreactors (MBRs): membrane fouling and membrane material. *Water research*, 43(6):1489–512.

Menniti, A. and Morgenroth, E. (2010). Mechanisms of SMP production in membrane bioreactors: choosing an appropriate mathematical model structure. *Water research*, 44(18):5240–51.

Michaelides, E. E. (1992). A novel way of computing the Basset term in unsteady multiphase flow computations. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 4(7):1579–1582.

Mota, M., Teixeira, J. a., and Yelshin, A. (2002). Influence of cell-shape on the cake resistance in dead-end and cross-flow filtrations. *Separation and Purification Technology*, 27(2):137–144.

Naessens, W., Maere, T., and Nopens, I. (2012a). Critical review of membrane bioreactor modelsPart 1: Biokinetic and filtration models. *Bioresource technology*, 122:95–106.

Naessens, W., Maere, T., and Ratkovich, N. (2012b). Critical review of membrane bioreactor modelsPart 2: Hydrodynamic and integrated models. *Bioresource technology*, 122:107–118.

Nilsson, P., Olofsson, A., Fagerlind, M., Fagerström, T., Rice, S., Kjelleberg, S., and Steinberg, P. (2001). Kinetics of the AHL regulatory system in a model biofilm system: How many bacteria constitute a quorum? *Journal of molecular biology*, 309(3):631–640.

Ognier, S., Wisniewski, C., and Grasmick, a. (2004). Membrane bioreactor fouling in subcritical filtration conditions: a local critical flux concept. *Journal of Membrane Science*, 229(1-2):171–177.

Parmar, M., Haselbacher, a., and Balachandar, S. (2011). Generalized Basset-Boussinesq-Oseen Equation for Unsteady Forces on a Sphere in a Compressible Flow. *Physical Review Letters*, 106(8):084501.

Patankar, S. (1980). *Numerical heat transfer and fluid flow.* CRC Press.

Picioreanu, C. (1996). Modelling biofilms with cellular automata. *Final report to European Environmental Research Organisation*, (January).

Roberts, M. E. and Stewart, P. S. (2004). Modeling antibiotic tolerance in biofilms by accounting for nutrient limitation. *Antimicrobial Agents and Chemotherapy*, 48(1):48–52.

Rosenberger, S. and Kraume, M. (2003). Filterability of activated sludge in membrane bioreactors. *Desalination*, 151(2):195–200.

Saffman, P. G. (1965). The lift on a small sphere in a slow shear flow. *Journal of Fluid Mechanics*, 22(02):385–400.

Stokes, G. (1851). *On the effect of the internal friction of fluids on the motion of pendulums.*

Tansel, B., Sager, J., Garland, J., Xu, S., Levine, L., and Bisbee, P. (2006). Deposition of extracellular polymeric substances (EPS) and microtopographical changes on membrane surfaces during intermittent filtration conditions. *Journal of Membrane Science*, 285(1-2):225–231.

Van Hinsberg, M. A. T., ten Thije Boonkkamp, J. H. M., and Clercx, H. J. H. (2011). An efficient, second order method for the approximation of the Basset history force. *Journal of Computational Physics*, 230(4):1465–1478.

Verrecht, B., Maere, T., Nopens, I., Brepols, C., and Judd, S. (2010). The cost of a large-scale hollow fibre MBR. *Water research*, 44(18):5274–83.

Vojir, D. and Michaelides, E. (1994). Effect of the history term on the motion of rigid spheres in a viscous fluid. *International Journal of Multiphase Flow*, 20(3):547–556.

Wanner, O. and Gujer, W. (1985). Competition in biofilms. *Water Science & Technology*, 17(2-3):27–44.

Wanner, O. and Gujer, W. (1986). A multispecies biofilm model. *Biotechnology and bioengineering*, 28(3):314–328.

Williamson, K. and McCarty, P. L. (1976). A model of substrate utilization by bacterial films. *Journal (Water Pollution Control Federation)*, pages 9–24.

Wisniewski, C. and Grasmick, A. (1998). Floc size distribution in a membrane bioreactor and consequences for membrane fouling. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 138(23):403–411.

Wörner, M. (2003). *A compact introduction to the numerical modeling of multiphase flows*. Number November. Forschungszentrum Karlsruhe.

Wozniak, T. (2010). MBR design and operation using MPE-technology (Membrane Performance Enhancer). *Desalination*, 250(2):723–728.

Xing, C., Qian, Y., Wen, X., Wu, W., and Sun, D. (2001). Physical and biological characteristics of a tangential-flow MBR for municipal wastewater treatment. *Journal of Membrane Science*, 191:31–42.

Yang, W., Cicek, N., and Ilg, J. (2006). State-of-the-art of membrane bioreactors: Worldwide research and commercial applications in North America. *Journal of Membrane Science*, 270(1):201–211.

Yeon, K.-M., Cheong, W.-S., Oh, H.-S., Lee, W.-N., Hwang, B.-K., Lee, C.-H., Beyenal, H., and Lewandowski, Z. (2009). Quorum sensing: a new biofouling control paradigm in a membrane bioreactor for advanced wastewater treatment. *Environmental science & technology*, 43(2):380–5.

Yun, M.-A., Yeon, K.-M., Park, J.-S., Lee, C.-H., Chun, J., and Lim, D. J. (2006). Characterization of biofilm structure and its effect on membrane permeability in MBR for dye wastewater treatment. *Water research*, 40(1):45–52.

Zhang, J., Chua, H., Zhou, J., and Fane, A. (2006). Factors affecting the membrane performance in submerged membrane bioreactors. *Journal of Membrane Science*, 284(1-2):54–66.