Filtration of natural organic matter and micro pollutants

Niels Horsman (4480864)
Faculty of Applied Mathematics

Final project
Applied Mathematics
TU Delft

Delft, July 31, 2019
Prof.dr.ir. C. Vuik

1 Abstract

This study discusses the filtration of micro-pollutants and natural organic matter from water by a granular activated carbon filter. To determine the efficiency of a filter, simulations are run to predict it's efficiency. This study will describe a demo application and it's underlying model that are used to determine the efficiency of a filter over time. The model is able to predict the adsorption of pollutants by the filter. But the model requires a high computational power. A solution to this problem is to look for a better time-integration method. This study will look into the filtration process and usage of time integration-methods on the model.

The following research question will be posed: What numerical method is most valid for the simulation of the filtration process by a GAC filter?

Contents

2 Introduction

Every year an enormous amount of clean water is needed for both consumption and agriculture. Our water usage will only increase over time, due to the increasing industrialization, population growth and a greater energy demand [1]. Water scarcity affects every continent and is listed as the largest global risk in terms of potential impact over the next decade by the World Economic Forum [2]. Therefore it is useful to look into the efficiency of water purification.

In 2015 the Netherlands used roughly 1100 million m^3 water [3]. A sizable part of this water originates from the surface water (395 million m^3). This surface water contains pollutants. Every year about 17 tonne pesticides and 140 tonne medicines enter the surface water [4]. The water also contains certain natural organic matter (NOM), such as proteins, lipids, and carbohydrates. These micro-pollutants and NOM will have to be removed from the water in order for it to become clean.

A way to remove pollutants from water is by using a granular activated carbon filter (GAC). This filter will attract the pollutants from the water and hold on to them. In this way the pollutants are removed from the water. The more pollutants are held by the GAC filter the less efficient it becomes. Thus eventually the activated carbon grains will have to be replaced.

When determining the efficiency of a GAC filter it is interesting to look into how long a filter can remain efficient before it becomes too full. The efficiency can be measured by using experimental data. However obtaining this data proves to be a problem. A filter can remain operational for approximately 1-3 years [5]. Testing for such a long time period is an obstacle. Also water can contain lots of different combinations of pollutants. Every solution behaves differently. Doing tests with every possible solution is impossible. Therefore simulations are run to determine the efficiency of a GAC filter.

To determine the efficiency of a GAC filter a demo-application has been constructed by D. Vries et al. [4] commissioned by KWR. This application makes use of a model provided by van der Aa et al. [6]. The application is able to predict the removal of pollutants by a GAC filter. But the application has room for improvement. The current numerical methods that are used to solve the system require a high computational power. Moreover the application doesn't provide a reliability interval.

This study will examine various numerical methods that can be used to predict the efficiency of a GAC filter.

3 Filtration model

In order for water to become drinkable, certain undesirable particles have to be removed. These particles include natural organic matter (NOM) and certain micro-pollutants. Purifying water from these substances can be done with the help of filters. A widely used type of filter is the granular activated carbon (GAC) filter. This filter contains activated carbon which will adsorb the NOM and the micro-pollutants and thus remove them from the water. The NOM and micro-pollutants will be adsorbed simultaneously. While the NOM will also be removed by bio-degradation. The simultaneous adsorption of the two compounds will cause a competition between the adsorption of NOM and micro-pollutants. Thus the NOM and micro pollutants will hinder each others adsorption. This section will describe the filtration process and the model that is used to simulate the removal of pollutants from water by a GAC filter.

3.1 Filtration process

In this subsection the filtration process will be explained at the hand of a schematic overview shown in Figure 1. When filtering dissolved substances from water, the contaminated water is led through a GAC filter. The water will flow past the activated carbon grains by convection. The activated carbon grains are surrounded by a liquid film. Water containing a surface concentration c_i of substance i will enter the liquid film as a result of reverse osmosis. The diffusion will cause a concentration $c_{s,i}$ of substance i at the surface of the film. When reaching the surface of the liquid film the dissolved substances will be attracted by the activated carbon and transported into the pore.

When a dissolved component i reaches the pore it will be attracted by the activated carbon. This attraction will cause the component to be pulled against the surface of the pore. Component i is now no longer dissolved in the water. The solid phase concentration of component i at the entrance of the pore is named $q_{s,i}$. Due to the diffusion over the surface of the pore component i will go further into the pore. This will increase the dissolved and solid concentrations ($c_{p,i}$ and $q_{p,i}$) inside the pore. The relation between the dissolved and solid concentrations is an equilibrium which dependents on the adsorption isotherm.

When water reaches the surface of the liquid film layer, it will carry some biomass. This biomass will grow inside the film and degrade over time. The biomass will block the pores and hinder the adsorption of the GAC.

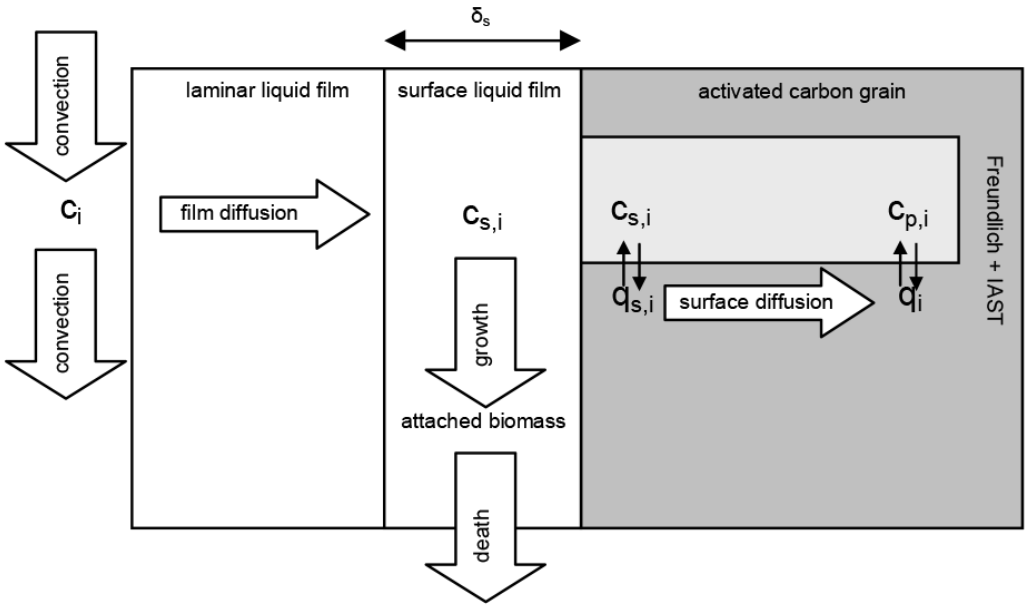

Figure 1: Schematic overview of a GAC filter [6].

3.2 Activated carbon

Activated carbon (AC) is a material that consists of a collection of carbon atoms (carbon) containing pores. The AC possess some very useful properties which can be used to separate substances from water or gas. Therefore AC can be used as a mean to filtrate pollutants from water. Most information in the subsection is derived from the book "Activated Carbon" written by H. March et al. [7].

To create AC, one must activate the carbon. The activation of carbon means the selective gasification of carbon atoms. By exposing the carbon to carbon dioxide at temperatures of around 800-900 °C, some carbon atoms will leave the carbon and will turn into a gaseous state. This process will cause the formation of pores in the carbon. The activation process is a highly complicated process. It is possible to create different pore structures. For example the size and the amount of pores in a carbon structure can be varied.

The pores in the AC provide it with an extremely large surface. This gives AC the ability to hold on to a great amount of pollutants. But these pollutants will first have to reach the pores. Fortunately the pores posses another useful property. The pores exerts intense van der Waals forces. These forces will attract substances to the pore, where they will remain. Thus the AC will adsorb the substances. A schematic display of substances getting adsorbed by a pore is visible in Figure 2.

However the adsorption process is more complicated than just described. The adsorption of a certain substance by the AC is mainly dependent on the adsorption isotherm. In the literature the isotherm describes the relation between the adsorbate and the adsorbent. This is not to be confused with other definitions of the isotherm

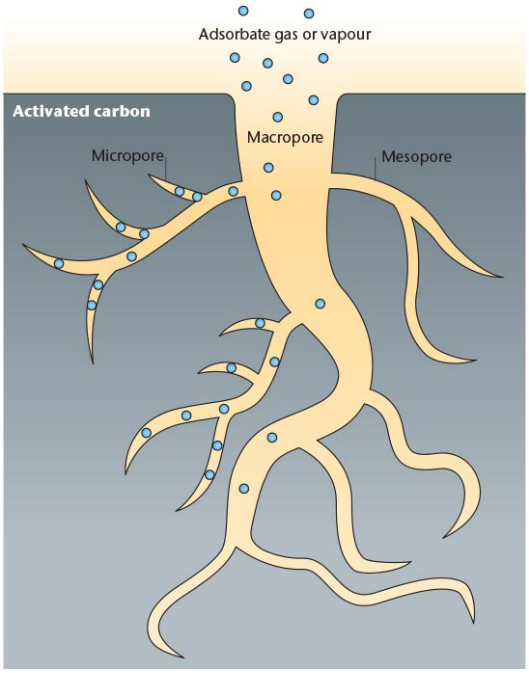

Figure 2: A schematic picture pollutants entering a pore of the activated carbon [8].

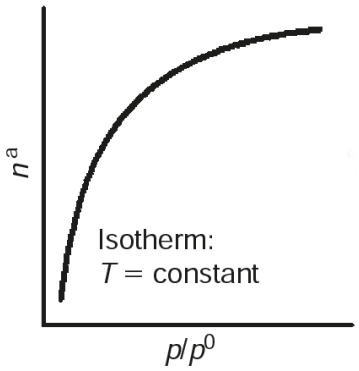

Figure 3: An example of an isotherm curve [7].

that are used in thermodynamics. The isotherm is the curve of the amounts of adsorbed substance ($mmol\ g^{-1}$) against the relative pressure ($\frac{p}{p^0}$). Note that the adsorption isotherm is non-linear. When looking at the adsorption of a mixture of several components, the isotherm of the mixture is dependent on the isotherms of the individual components. Therefore the isotherm of the mixture is highly non-linear. An example of the isotherm curve is given in Figure 3.

3.3 Model description

A model to describe the removal of micro-pollutant's and NOM from water has been established by van der Aa et al. [6]. This study focused on the removal of atrazine, a commonly used pesticide, and the adsorption and bio degradation of biomass. The model is based on the schematic overview that is discussed in subsection 3.1. The model is displayed in the equations 3.1 to 3.7.

Before explaining the model first some properties and assumptions will be given. The model is able to simulate the adsorption of atrazine and NOM, while distinguishing between three types of NOM. These are; non-adsorbable non-biodegradable NOM, adsorbable non-biodegradable NOM and adsorbable biodegradable NOM. The model uses some simplifications. Suspended biomass, attachment/detachment of bacteria and internal pore diffusion are neglected. Biomass death rate is kept constant, while dead biomass is assumed to be washed out completely during backwashing.

Now the model will be displayed hereunder followed by some explanation of the individual equations. Not every parameter/variable will be explained in the subsection.

Their explanation can be found in the appendix in table 1.

$$\frac{\partial c_i}{\partial t} = -\frac{v}{\epsilon}\frac{\partial c_i}{\partial x} - \frac{6(1-\epsilon)\beta_{f,i}}{\epsilon d_{part}}(c_i - c_{s,i}) \tag{3.1}$$

$$\frac{dc_{s,i}}{dt} = \frac{\beta_{f,i}}{\delta_s}(c_i - c_{s,i}) - \frac{\rho_{filt}\beta_{s,i}}{\delta_s(1-\epsilon)}(q_{s,i} - \overline{q_i}) - \frac{\mu_{spec}}{Y_i}X_s\frac{\rho_{filt}d_{part}}{6\delta_s(1-\epsilon)} \tag{3.2}$$

$$\frac{dq_i}{dt} = \frac{6\beta_{s,i}}{d_{part}}(q_{s,i} - \overline{q_i}) \tag{3.3}$$

$$\frac{dX_s}{dt} = (\mu_{spec} - b)X_s \tag{3.4}$$

$$\beta_s, i = 10\frac{D_{s,i}}{d_{part}} \tag{3.5}$$

$$q_i = K_{F,i}{c_{p,i}}^{n_i} \tag{3.6}$$

$$\mu_{spec} = min\Big(\Big(\tfrac{c_{s,i}}{Ks,i+c_{s,1}} \cdot \tfrac{e^{-(\frac{60-T}{44})}}{e^{-(\frac{45}{44})^2}} \cdot \mu_{max,15°C}\Big), \Big(\tfrac{cs,2}{Ks,i+c_{s,2}} \cdot \tfrac{e^{-(\frac{60-T}{44})}}{e^{-(\frac{45}{44})^2}} \cdot \mu_{max,15°C}\Big), etc\Big) \tag{3.7}$$

The most significant equations are the first three equations. These give us the expressions of the derivatives of the concentrations of a certain pollutant at different stages of the filtration process. With the help of these equations and initial values for the concentrations, one can use a time-stepping method to predict future concentrations. This is of course crucial when establishing a simulation of a filtration process.

In Equation 3.4 an expression is given for the derivative of X_s. This is the activity of the biomass that is attached to the carbon grain. The presence of biomass negatively influences the adsorption of pollutants. The effect of biomass on the concentration of a pollutant in the liquid film ($c_{s,i}$) is visible in the last term of equation 3.2. In Equation 3.5 the expression for the term $\beta_{s,i}$ is given. $\beta_{s,i}$ is the surface diffusion mass transfer coefficient of component i. This impacts the film diffusion and therefore the concentrations in the bulk liquid (c_i) and the liquid film ($c_{s,i}$). This is clear when looking at Equations 3.1 and 3.2.

Equation 3.6 represents the equilibrium of dissolved and solid phase concentration of component i in the pore. It is based on the adsorption equilibrium according to Freundlich. In this equation the $K_{F,i}$ is the mass based Freundlich constant of component i. The term n_i is the adsorption isotherm which is also mentioned as the Freundlich exponent. A study by M.A.S. Niandou et al.[9] displays the Freundlich parameters for different filters. Equation 3.7 gives a way to compute the specific growth rate of the biomass.

Note that the derivative of the solid phase concentration of a component i at the start of the pore ($q_{s,i}$) is not given in any of the Equations 3.1 to 3.7. The value of $q_{s,i}$ is determined by a combination of the adsorption equilibrium according to Freundlich and the Ideal Adsorbed Solution Theory (IAST). IAST describes the adsorption of competing components in gasses, but is also applicable for liquids. It gives an elegant solution to model the adsorption of multiple components while only needing a single isotherm.

The value of $q_{s,i}$ can be computed with the help of Equations 3.8 to 3.11. Instead of the actual concentrations that are used in the previous equations the following equations use molar concentrations. So $c_{p,i,M}$ is the molar liquid phase concentration of component i and $q_{i,M}$ is the molar solid phase concentration of component i.

$$\sum_{i=1}^{N} z_i = \sum_{i=1}^{N} \frac{c_{p,1,M}}{\left(\frac{\Theta \cdot n_i}{K_{F,i,M}}\right)^{\frac{1}{n_i}}} = \sum_{i=1}^{N} \frac{q_{i,M}}{\Theta \cdot n_i} = 1 \tag{3.8}$$

$$q_{t,M} = \left(\sum_{i=1}^{N} \frac{z_i}{\Theta \cdot n_i}\right)^{-1} \tag{3.9}$$

$$q_{i,M} = z_i \cdot q_{t,M} \tag{3.10}$$

$$q_{i,M} = K_{F,i} \cdot MW_i^{n_i - 1} \cdot c_{p,i,M}^{\ n_i} \tag{3.11}$$

With Equation 3.8 the mol fraction of component i (z_i) can be derived. This describes the ratio between $q_{i,M}$ and $c_{p,i,M}$. By filling in z_i in Equation 3.9 the total molar solid phase concentration ($q_{t,M}$) can be computed. Now with the help of Equations 3.10 and 3.11 $q_{i,M}$ can be computed.

4 Problem description

The goal of this research is of course to determine the efficiency of filtration of pollutants from water by a GAC filter by using simulations. However when using the model provided by van der Aa et al.[6], some problems occur.

The rapport by D. Vries et al.[4] described a demo application that could predict the efficiency of a GAC filter, when filtering water containing certain pollutants. The application makes use of the model by van der Aa et al.[6].

The model is able to predict the efficiency of the filtration of up to four pollutants at the same time. The prediction is made by using time-stepping methods with the equations from the model by van der Aa et al.[6]. The calculation time needed to make these time-step takes relatively long. It takes 1-2 minutes to predict the filtration efficiency of 1 pollutant. To shorten this computation time one can look to better divide the workload by parallelization. However it is also useful to look for improvements in the time-step integration method.

A possible cause that could explain the long computation time is the presence of the adsorption isotherm. As earlier explained in section 3.2 the isotherm is highly non-linear causing a very stiff system, which will take small time-steps and high computational power to solve. It is therefore very interesting to look into the effect of the isotherm on the system and experiment with different numerical methods. Perhaps it is possible to find a faster more reliable method.

Another big downside of the demo application is the lack of a confidence interval. The model provides a prediction of the efficiency without telling what the reliability of this prediction is. Computing the confidence interval proved to take a lot of computational power and was therefore omitted from the application[4]. When experimenting with several numerical methods it would be interesting to compute the confidence interval to see how reliable the solutions are. Even if this would take too long to include in an application in the future.

When determining the effectiveness of several numerical methods on the model it is advisable to first experiment with a simplified version of the model. This study will first examine a simplified version of the model given in 3.3 in Section 5. Later expanded versions of the model will be examined in Sections 6 and 8

Thus the model that is used by the demo application is able to predict the removal of several pollutants from water. However the model requires a long computation time and the application doesn't provide a confidence interval. The long computation time could be shortened by looking for a more efficient numerical method. Several numerical methods will first be tested on the simplified model. Later experiments can be performed with a more complicated version.

5 Simple model

This section will introduce a simplified version of the model given in Section 3.3. By using time-integration methods on the simple model it is possible to simulate the adsorption of pollutants. A program that is able to simulate the adsorption will be constructed in python. The results given by this program and the stability of the numerical methods will then be examined.

5.1 Simple filtration model

The model that will be used in this section is displayed in Equations 5.1 to 5.3.

$$\frac{\partial c}{\partial t} = \frac{1}{\epsilon}\frac{\partial c}{\partial x} - \frac{1-\epsilon}{\epsilon}(q_s - q)\,, \qquad 0 \leq x \leq 1 \tag{5.1}$$

$$\frac{dq}{dt} = q_s - q \tag{5.2}$$

$$q_s = Kc^p \tag{5.3}$$

This model is a simplification of the model displayed in Section 3.3. In the simplified version the surface liquid film layer is ignored and the effect of biomass is neglected. Therefore the terms $c_{s,i}$, X_s and μ_{spec} are no longer needed. The terms $\frac{6\beta_{f,i}}{d_part}$, $\frac{6\beta_{s,i}}{d_part}$ and v are set to 1.

First the model will be used to predict the adsorption of only one component. The concentration of this component in the bulk liquid is expressed as c. ϵ is the particle diameter of the component. q_s and q are the solid phase concentrations of the component at the surface of the pore and inside the pore. When computing q_s in Equation 5.3 only the adsorption equilibrium of Freundlich is used. Instead of combining it with IAST. Also the interval $[0, 1]$ is introduced, this is the interval which contains all values of x.

The boundary conditions of the model are displayed in Equation 5.4. These boundary conditions represent an incoming water stream with a certain pollutant with concentration 1 for every time t. At the start of the filtration process the concentration of the component alongside the carbon grain and inside the pore are set at 0.

$$\begin{aligned} c(1,t) &= 1 \\ c(x,0) &= 0, \qquad 0 \leq x < 1 \\ q(x,0) &= 0 \end{aligned} \tag{5.4}$$

5.2 Numerical methods

Before applying a numerical method to the simple model, a spatial and time discretization must be chosen.
In this case the interval on which x exists is $[0, 1]$. This interval will be divided into N grid-points. With the first grid-point located on $x = 0$ and the last grid-point located just before $x = 1$. A grid-point in $x = 1$ is not necessary since the concentration is already given. Therefore Δx will be equal to $\frac{1}{N}$. Grid-point i will be located at $x_i = (i-1)\Delta x$.
The interval on which the time t exists is $[0, \infty)$. A Δt of certain size must be chosen to make time-steps. A small Δt will lead to more accurate results but longer computation time. With Δt chosen, t_n will be equal to $t = n\Delta t$.

Some notation

$$
\begin{aligned}
c_i(t),\ q_i(t),\ q_{s,i}(t) &= \text{the value of } c,\ q \text{ and } q_s \text{ on grid-point } x_i \text{ at time } t \\
c_i,\ q_i,\ q_{s,i} &= \text{the value of } c,\ q \text{ and } q_s \text{ on grid-point } x_i \text{ at an arbitrary time} \\
\underline{c}(t) &= \begin{pmatrix} c_1(t) \\ c_2(t) \\ \vdots \\ c_N(t) \end{pmatrix} \\
\underline{c}_n &= \underline{c}(t_n) \\
\underline{c} &= \text{the vector containing all values of } c_i \text{ at an arbitrary time}
\end{aligned}
\tag{5.5}
$$

The vector notation used for $\underline{c}(t)$ will also be used for $\underline{q}(t)$ and $\underline{q_s}(t)$.

Now a numerical method must be established to compute the values of c, q and q_s on the grid-points through time. First the term $\frac{\partial c}{\partial x}$ in Equation 5.1 must be approximated by a final difference method. In this case an upwind finite difference scheme is chosen. So for every grid-point the following approximation is made:

$$
\frac{\partial c}{\partial x} \approx \frac{c_{i+1} - c_i}{\Delta x} \tag{5.6}
$$

Now the derivative of the vector $\underline{c}$ is easily approximated by the following matrix multiplication.

$$
\frac{\partial \underline{c}}{\partial x} = \frac{1}{\Delta x} \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & \emptyset & \\ & \emptyset & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \underline{c} + \frac{1}{\Delta x} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} = A\underline{c} + a \tag{5.7}
$$

With the help of the approximation made by the upwind finite difference scheme and the differential equations from the model, it is possible to estimate the derivatives of $\underline{c}$, $\underline{q}$ and $\underline{q_s}$.

$$\frac{\partial \underline{c}}{\partial t} \approx \frac{1}{\epsilon}(A\underline{c} + a) - \frac{1-\epsilon}{\epsilon}(K\underline{c}^p - \underline{q}) \tag{5.8}$$

$$\frac{d\underline{q}}{dt} \approx K\underline{c}^p - \underline{q} \tag{5.9}$$

$$\underline{q_s} \approx K\underline{c}^p \tag{5.10}$$

With the help of the estimations for the derivatives time-steps can be made. To perform these time-steps a time-integration method must be chosen. Both an implicit and an explicit method will be used. As explicit method the Forward Euler method will be applied. For the implicit method, the Backward Euler method will be used. The explicit method uses the derivative of the of the current moment in time to make a time-step. This can cause the approximation of the concentration on the next moment in time to become slightly negative. Since in reality a concentration cannot be negative, this could lead to very large errors. Therefore it is expected that the implicit method will provide better results.

Note that from now on the concentrations $\underline{c}$, $\underline{q}$ and $\underline{q_s}$ represent the discretization of the real values. It is customary to use a different notation for the discretization. However to keep it simple the same notation will be used.

Forward Euler

When using the Forward Euler method the time-steps for $\underline{c}$ and $\underline{q}$ are made in the following way. Note that no time-steps are needed for $\underline{q_s}$ since this can be directly computed with the help of $\underline{c}$

$$\begin{aligned} \underline{c}_{n+1} &= \underline{c}_n + \Delta t \underline{c}'_n \\ \underline{q}_{n+1} &= \underline{q}_n + \Delta t \underline{q}'_n \end{aligned} \tag{5.11}$$

So $\underline{c}'_n$ and $\underline{q}'_n$ need to be computed and inserted into the equation. This can be done with the help of Equations 5.8 and 5.9.

$$\begin{aligned} \underline{c}_{n+1} &= \underline{c}_n + \Delta t\big[\frac{1}{\epsilon}(A\underline{c}_n + a) - \frac{1-\epsilon}{\epsilon}(K\underline{c}_n^p - \underline{q}_n)\big] \\ \underline{q}_{n+1} &= \underline{q}_n + \Delta t\big[K\underline{c}_n^p - \underline{q}_n\big] \end{aligned} \tag{5.12}$$

Backward Euler

The time-steps for the Backward Euler method are similar to the Forward Euler method.

$$\underline{c}_{n+1} = \underline{c}_n + \Delta t \underline{c}'_{n+1} \tag{5.13}$$
$$\underline{q}_{n+1} = \underline{q}_n + \Delta t \underline{q}'_{n+1} \tag{5.14}$$

Again the Equations 5.8 and 5.9 are used to approximate the derivatives.

$$\begin{aligned} \underline{c}_{n+1} &= \underline{c}_n + \Delta t\big[\frac{1}{\epsilon}(A\underline{c}_{n+1} + a) - \frac{1-\epsilon}{\epsilon}(K\underline{c}^p_{n+1} - \underline{q}_{n+1})\big] \\ \underline{q}_{n+1} &= \underline{q}_n + \Delta t\big[K\underline{c}^p_{n+1} - \underline{q}_{n+1}\big] \end{aligned} \tag{5.15}$$

If $\underline{c}_{n+1}$ is computed it is possible to make the time-step necessary to compute $\underline{q}_{n+1}$. However computing $\underline{c}_{n+1}$ is more complicated due to the non-linear term. To make this time-step we first rewrite Equation 5.13.

$$(I - \frac{\Delta t}{\epsilon}A)\underline{c}_{n+1} - \underline{c}_n - \frac{\Delta t}{\epsilon}a - \frac{\Delta t(1-\epsilon)}{(1+\Delta t)\epsilon}\underline{q}_n - \underline{c}^p_{n+1}K\Delta t\Big(\frac{\Delta t(1-\epsilon)}{(1+\Delta t)\epsilon} - \frac{(1-\epsilon)}{\epsilon}\Big) = 0 \tag{5.16}$$

So $\underline{c}_{n+1}$ is the zero of the following formula:

$$f(\underline{x}) = (I - \frac{\Delta t}{\epsilon}A)\underline{x} - \underline{c}_n - \frac{\Delta t}{\epsilon}a - \frac{\Delta t(1-\epsilon)}{(1+\Delta t)\epsilon}\underline{q}_n - \underline{x}^p K\Delta t\Big(\frac{\Delta t(1-\epsilon)}{(1+\Delta t)\epsilon} - \frac{(1-\epsilon)}{\epsilon}\Big) \tag{5.17}$$

So to find $\underline{c}_{n+1}$, we have to find the zero of $f(\underline{x})$. To do this the Newton-Raphson method will be used. This method goes as follows:

$$\underline{x}_{i+1} = \underline{x}_i - J_f(\underline{x}_i)^{-1} f(\underline{x}_i) \tag{5.18}$$

In our case the following holds true:

$$\underline{x_0} = \underline{c}_n \tag{5.19}$$
$$J_f(\underline{x}_i)^{-1} = J^{-1} \tag{5.20}$$
$$i = j \Rightarrow J_{ij} = (I - \frac{\Delta t}{\epsilon}A)_{ij} - p\underline{x}_i^{p-1}K\Delta t\Big(\frac{\Delta t(1-\epsilon)}{(1+\Delta t)\epsilon} - \frac{(1-\epsilon)}{\epsilon}\Big) \tag{5.21}$$
$$i \neq j \Rightarrow J_{ij} = (I - \frac{\Delta t}{\epsilon}A)_{ij} \tag{5.22}$$

The method will iterate until $f(x_i) \approx 0$.

Richardson's extrapolation

The program should also be able to give a numerical estimate of the error that is made. To estimate the error Richardson's extrapolation is used. With the help of this method the estimated error and order of the error can be determined at certain grid-points. Only the error in the computed concentration c will be determined. Let $c(\Delta t, \Delta x)$ be the estimated solution given by the numerical method at a certain time t. The Richardson extrapolation gives the following:

$$2^p = \frac{c(2\Delta t, 2\Delta x) - c(4\Delta t, 4\Delta x)}{c(\Delta t, \Delta x) - c(2\Delta t, 2\Delta x)} \tag{5.23}$$
$$e_p(\Delta t, \Delta x)^p = \frac{c(\Delta t, \Delta x) - c(2\Delta t, 2\Delta x)}{(\Delta x)^p(2^p - 1)} \tag{5.24}$$

5.3 Program implementation

With the help of the numerical methods established in Section 5.2, it is now possible to write a program that can predict the adsorption of a pollutant by a GAC filter. The program is written in the program language Python. This section will shortly discuss some properties of the program that was written. The code itself can be downloaded by following the link in the Appendix A. Note that the code contains comments which contain a more detailed explanation then given here.

The program is based on Equations 5.1, 5.2 and 5.3. These equations contain some variable, for which a value must be chosen. This value depends on the kind of pollutant which has to be absorbed and the filter. Since we are only interested in the examination of the model the choice of these values is not of great significance. For ϵ the same values is chosen as in the paper by van der Aa et al [6].

$$\epsilon = 0.40$$
$$K = 0.004087$$

Other variables such as Δt, Δx and p will be varied to see the effects on the simulation.

The program itself is fairly simple. The main code can be found in the file *Simple.py*. This file contains functions that are able to simulate the adsorption of a pollutant until a certain time by using the Forward or Backward Euler method. Note that the Backward Euler method behaves differently depending on the input value of n. If n equals 1 the Newton-Raphson method will be skipped since there is no non-linear term. The code also contains a function that is able to perform the Richardson extrapolation.

The file *SimpleFunctions.py* contains some less significant functions that are needed for the program. These are put into a separate file to keep the main code accessible. These functions are for example the power function of a vector, the creation of the upwind matrix and functions to plot a concentration.

5.4 Results and discussion

This section will discuss the results generated by the Python program that was used to approximate the concentration of a pollutant through time. The results will be discussed and a conclusion on the correctness of the solution will be made.

Before running the program some variables had to be chosen. These are; Δx, Δt, p (the isotherm) and the final time (T) at which we want to approximate the concentrations. When running the program it was found that $\Delta x = 1/20$ and $\Delta t = 0.001$ provided fast and reliable results. This will be later confirmed in the results discussing the errors estimated by the Richardson extrapolation. T and p will be varied.

Figures 4 to 6 show the approximated concentrations of c, q and q_s at certain time-steps. The value for p was set to 1. This value of p will result in a linear problem. The approximations were given by the Forward Euler method. The results of the Backward Euler method are not shown because they are very similar. These results can be found by following the link in the appendix A. The downloadable files in the link also contains videos that show the concentration through time.

When looking at the concentrations through time the solution behaves entirely as expected. The concentrations of c, q and q_s increase gradually from the right-hand side. This is due to the income stream of pollutants from $x = 1$. There are no spikes in the solution and therefore it appears to be correct. Note that the concentration of c at the end of the filter ($x = 0$) is almost 1, this is equal to the incoming concentration. This means almost no pollutants were filtered from the bulk liquid. Although in a realistic situation this should be impossible, the improbable results are likely caused by the simplicity of the used model. At this stage the stability of the model is more interesting than the realism of the solution.

When varying the value of p the impact on the solution seems to be minimal. The concentrations behave the same as in the Figures 4 to 6, for both the Forward Euler method as well as the Backward Euler method.

However, when examining the estimated error and order of the error given by the Richardson extrapolation the impact of the parameter p on the system becomes clearly visible. The stability of the model is tested for the following values of p; $\frac{1}{2}$, 1 and 2. The order of the errors are displayed in Figures 13 and 14. The average order of the errors of the Backward Euler method are stable. For any value of p the order is close to 1. The average order of the Forward Euler method vary for different values of p. The orders for the concentration q can become relatively low when using values of p which are not equal to 1.

The estimated errors of concentration c are displayed in Figures 7 to 12. The estimated errors of the Backward Euler method are all very comparable. A smaller value for Δt will result in a more accurate result and the errors are equal for the chosen values of p. The errors for the Forward Euler behave differently. In Figures 7 and 9 the error increases with a smaller value for Δt. For $p = \frac{1}{2}$ the error increases when using $\Delta t = 10^{-4}$ instead of $\Delta t = 10^{-3}$, as is displayed in Figure 11. The estimated errors also change for every chosen value of p. The errors seem to be

approaching a limit and therefore the method appears to be stable. To test this more runs with smaller time-steps should have to be made. The errors for the Backward Euler method approach the limit from above and the errors for the Forward Euler method approach the limit from below. Thus a lower Δt will result in a higher accuracy for the Backward Euler method and a lower accuracy for the Forward Euler method. The fact that the error approaches the same limit for both methods indicates that the error is mainly caused by the spatial discretization.

The computation time needed to complete the simulation is displayed in Figure 15 and 16. The Forward Euler method is faster than the Backward Euler method. However the difference in computation time is relatively small. Therefore the difference will not be critical in determining which method is favorable.

The system seems to be stable for both the Forward Euler method as well as the Backward Euler method. The error will approach the same limit for both errors. However where the Backward Euler method will have roughly the same orders of errors for varying values of p, these will fluctuate for the Forward Euler method. Therefore it is advised to use the Backward Euler method.

The results produced by the simple model are stable but not very realistic. The next step is to expand the model and add certain neglected variables to see if the solution becomes closer to the reality.

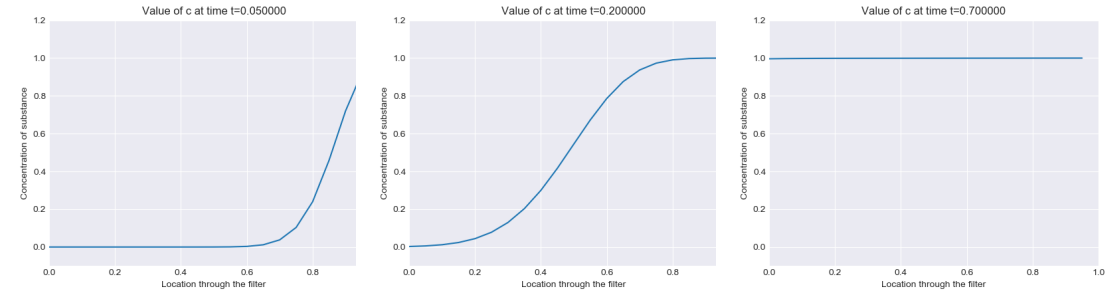

Figure 4: The concentration of c displayed at three different time-steps, t=0.05, t=0.2 and t=0.7

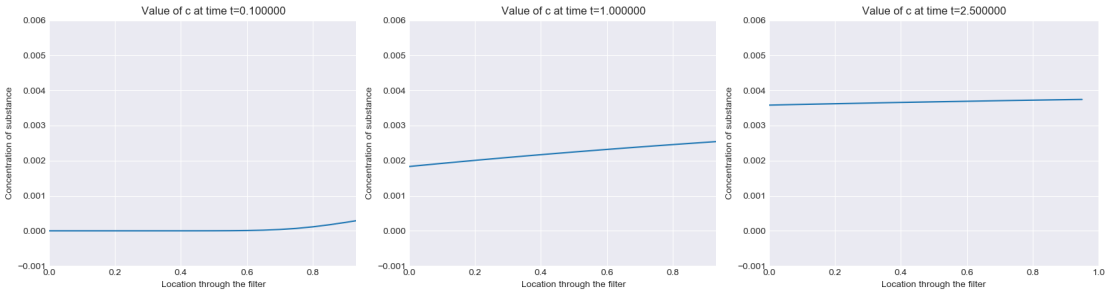

Figure 5: The concentration of q displayed at three different time-steps, t=0.05, t=0.2 and t=0.6

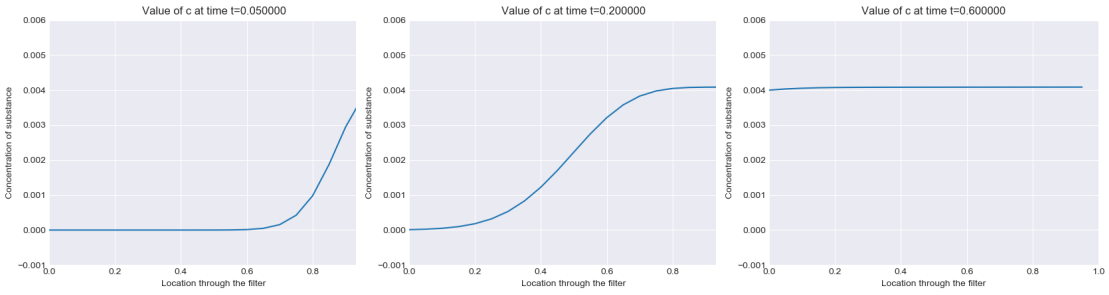

Figure 6: The concentration of qs displayed at three different time-steps, t=0.1, t=1 and t=2.5

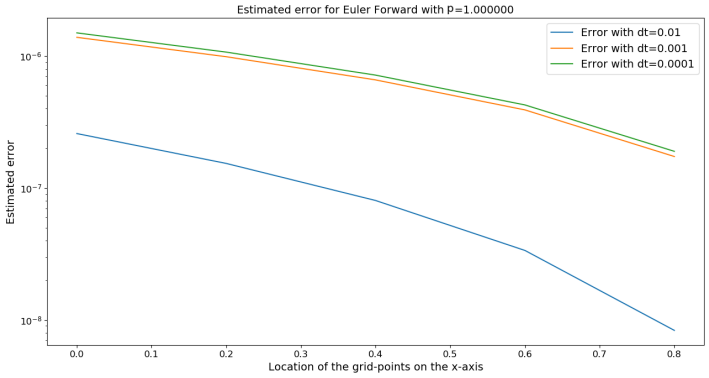

Figure 7: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 1$.

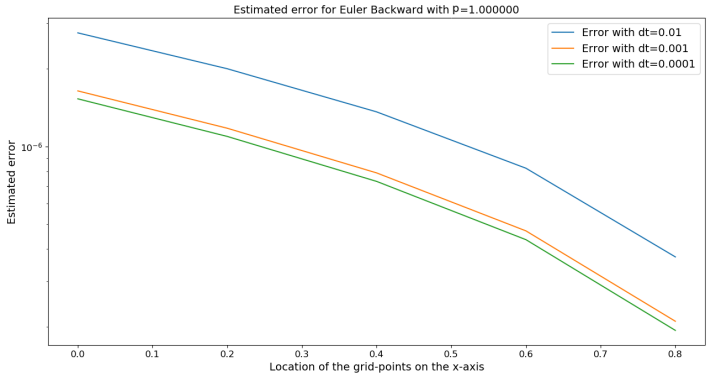

Figure 8: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 1$.

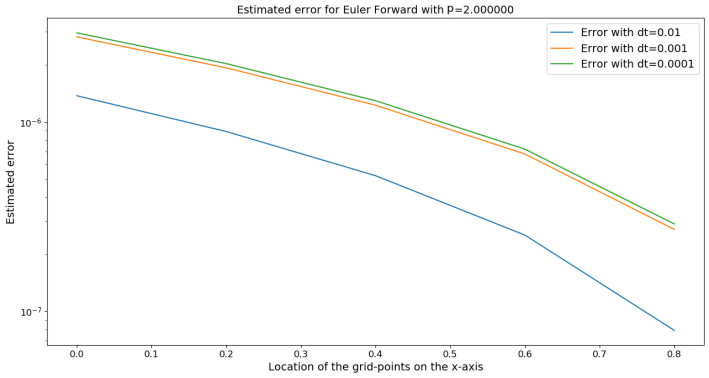

Figure 9: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 2$.

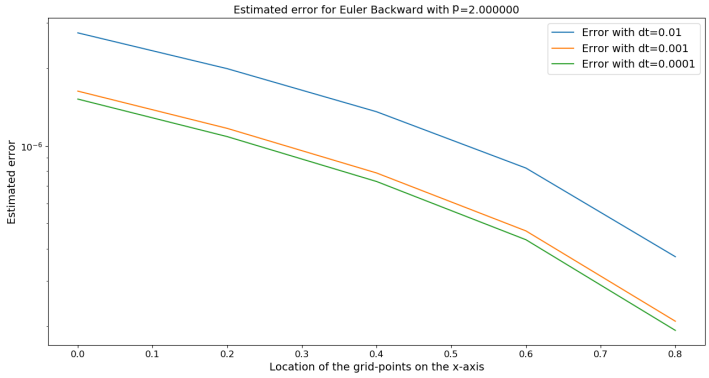

Figure 10: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 2$.

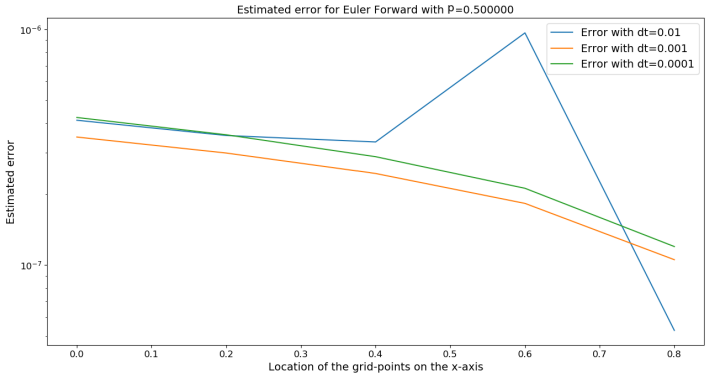

Figure 11: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 0.5$.

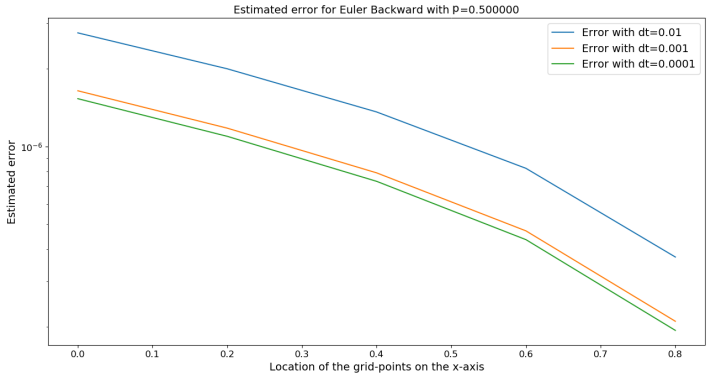

Figure 12: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 0.5$.

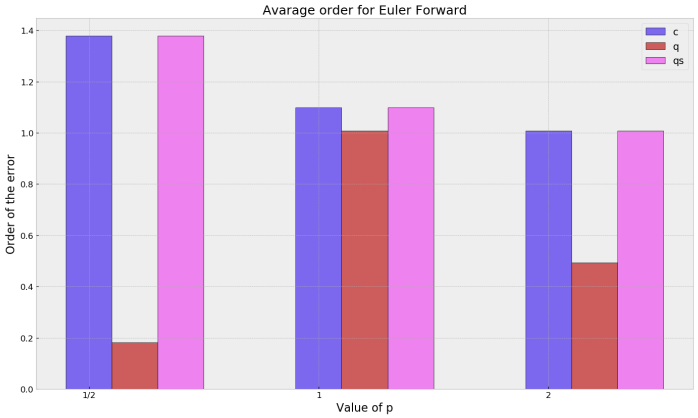

Figure 13: The order of the error when using the Forward Euler method.

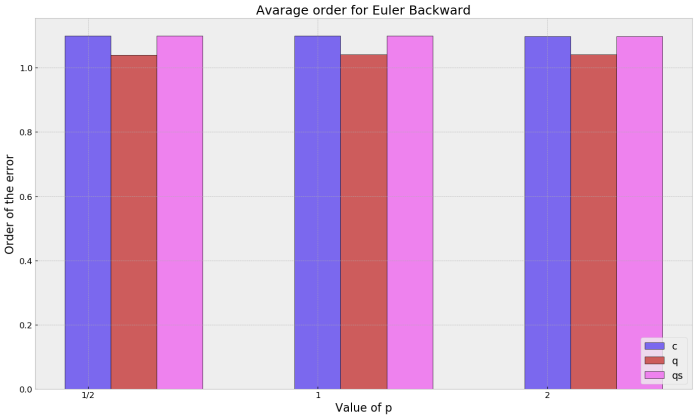

Figure 14: The order of the error when using the Backward Euler method.

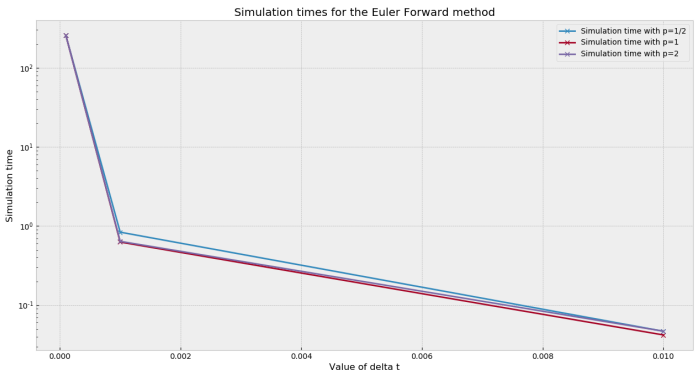

Figure 15: Simulation time needed when using the Forward Euler method.

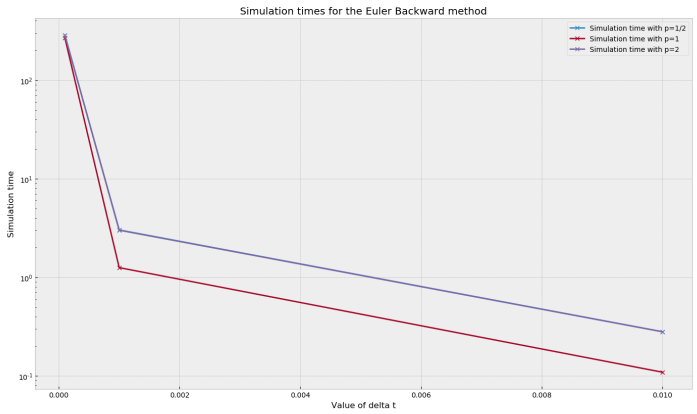

Figure 16: Simulation time needed when using the Backward Euler method.

6 Extended model

The previous section described the implementation of the simple model to simulate the adsorption of a pollutant. This simulation proved to give a stable solution however the result was not very realistic. In this section the simple model given in Section 5.1 will be extended.

6.1 Extended filtration model

The model that will be used in this section is displayed in Equations 6.1 to 6.4.

$$\begin{aligned}
\frac{\partial c}{\partial t} &= \frac{1}{\epsilon}\frac{\partial c}{\partial x} - \frac{1-\epsilon}{\epsilon}(c - c_s)\,, \qquad 0 \leq x \leq 1 && (6.1)\\
\frac{dc_s}{dt} &= c - c_s - \frac{1}{1-\epsilon}(q_s - q) && (6.2)\\
\frac{dq}{dt} &= q_s - q && (6.3)\\
q_s &= Kc_s^p && (6.4)
\end{aligned}$$

The main difference between the model displayed in Equations 6.1 to 6.4 and the simple version displayed in Section 5.1, is the addition of c_s. This is the concentration of the pollutant at the surface of the liquid film. Adding this concentration to the model gives us a new derivative displayed in Equation 6.2. The introduction of concentration c_s also changes the derivatives of c and q_s in the simple model (Equations 5.1 and 5.3).

The boundary conditions will be very similar to the ones used in Section 5.1, only now c_s will also be set to zero.

$$\begin{aligned}
c(1,t) &= 1\\
c(x,0) &= 0, \qquad 0 \leq x < 1\\
c_s(x,0) &= 0\\
q(x,0) &= 0
\end{aligned} \tag{6.5}$$

6.2 Numerical methods

When applying a numerical method, a spatial and time discretization must be chosen. The extended model will use the same spatial and time discretization as the simple model (5.2). Thus, $x_i = (i-1)\Delta x$ and $t_n = n\Delta t$.

The notation that will be used in this section will be almost identical to the notation used in section 5.2. Only now the notations will also be used for concentration c_s.

A numerical method must be chosen to compute the the values of c, c_s, q and q_s through time at every grid-point. This can be done by approximating the derivatives of c, c_s and q. To compute the derivative of $\underline{c}$ with the help of Equation 6.1 the term $\frac{\partial \underline{c}}{\partial x}$ must be approximated. This is done with the upwind finite difference scheme:

$$\frac{\partial \underline{c}}{\partial x} = \frac{1}{\Delta x}\begin{pmatrix} -1 & 1 & & \\ & -1 & 1 & \emptyset \\ & \emptyset & \ddots & \ddots \\ & & & -1 \quad 1 \end{pmatrix} \underline{c} + \frac{1}{\Delta x}\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} = A\underline{c} + a \tag{6.6}$$

With the help of the upwind finite difference scheme approximations can be made for the derivatives of $\underline{c}$, $\underline{c_s}$, $\underline{q}$ and $\underline{q_s}$.

$$\frac{\partial \underline{c}}{\partial t} \approx \frac{1}{\epsilon}(A\underline{c} + a) - \frac{1-\epsilon}{\epsilon}(\underline{c} - \underline{c_s}) \tag{6.7}$$

$$\frac{d\underline{c_s}}{dt} \approx \underline{c} - \underline{c_s} - \frac{1}{1-\epsilon}(K\underline{c_s}^p - \underline{q}) \tag{6.8}$$

$$\frac{d\underline{q}}{dt} \approx K\underline{c_s}^p - \underline{q} \tag{6.9}$$

$$\underline{q_s} \approx K\underline{c_s}^p \tag{6.10}$$

With the help of the estimations for the derivatives, time-steps can be made. These time-steps will be made with the Forward- and Backward Euler method. Note that from now on the concentrations $\underline{c}$, $\underline{c_s}$, $\underline{q}$ and $\underline{q_s}$ represent the discretization of the real values. It is customary to use a different notation for the discretization. However to keep it simple the same notation will be used.

Forward Euler

When using the Forward Euler method the time-steps for $\underline{c}$, $\underline{c_s}$ and $\underline{q}$ are made in the following way. Note that no time-steps are needed for $\underline{q_s}$ since this can be directly computed with the help of $\underline{c_s}$.

$$\begin{aligned} \underline{c}_{n+1} &= \underline{c}_n + \Delta t \underline{c}'_n \\ \underline{c_s}_{n+1} &= \underline{c_s} + \Delta t \underline{c_s}'_n \\ \underline{q}_{n+1} &= \underline{q}_n + \Delta t \underline{q}'_n \end{aligned} \tag{6.11}$$

So $\underline{c}'_n$, $\underline{c_s}'_n$ and $\underline{q}'_n$ need to be computed and inserted into the equation. This can be

done with the help of Equations 6.7, 6.8 and 6.9.

$$\begin{aligned}
\underline{c}_{n+1} &= \underline{c}_n + \Delta t\big[\frac{1}{\epsilon}(A\underline{c}_n + a) - \frac{1-\epsilon}{\epsilon}(\underline{c}_n - \underline{c_s}_n)\big] \\
\underline{c_s}_{n+1} &= \underline{c_s}_n + \Delta t\big[\underline{c}_n - \underline{c_s}_n - \frac{1}{1-\epsilon}(K\underline{c_s}_n^{\,p} - \underline{q}_n\big] \\
\underline{q}_{n+1} &= \underline{q}_n + \Delta t\big[K\underline{c}_n^p - \underline{q}_n\big]
\end{aligned} \tag{6.12}$$

Backward Euler

The time-steps for the Backward Euler method are similar to the Forward Euler method.

$$\begin{aligned}
\underline{c}_{n+1} &= \underline{c}_n + \Delta t\underline{c}'_{n+1} \\
\underline{c_s}_{n+1} &= \underline{c_s}_n + \Delta t\underline{c_s}_{n+1} \\
\underline{q}_{n+1} &= \underline{q}_n + \Delta t\underline{q}'_{n+1}
\end{aligned} \tag{6.13}$$

Again the Equations 6.7, 6.8 and 6.9 are used to approximate the derivatives.

$$\underline{c}_{n+1} = \underline{c}_n + \Delta t\big[\frac{1}{\epsilon}(A\underline{c}_{n+1} + a) - \frac{1-\epsilon}{\epsilon}(\underline{c}_{n+1} - \underline{c_s}_{n+1})\big] \tag{6.14}$$

$$\underline{c_s}_{n+1} = \underline{c_s}_n + \Delta t\big[\underline{c}_{n+1} - \underline{c_s}_{n+1} - \frac{1}{1-\epsilon}(K\underline{c_s}_{n+1}^{\,p} - \underline{q}_{n+1}\big] \tag{6.15}$$

$$\underline{q}_{n+1} = \underline{q}_n + \Delta t\big[K\underline{c_s}_{n+1}^{\,p} - \underline{q}_{n+1}\big] \tag{6.16}$$

Note that in order to compute $\underline{c}_{n+1}$, $\underline{c_s}_{n+1}$ is needed. Moreover to compute $\underline{c_s}_{n+1}$, $\underline{c}_{n+1}$ is needed. This makes the time-steps more complicated to solve than in the simple model. First Equation 6.16 will be rewritten.

$$\underline{q}_{n+1} = \frac{1}{1+\Delta t}(\underline{q}_n + \Delta tK\underline{c_s}_{n+1}^{\,p}) \tag{6.17}$$

Now Equation 6.15 will be rewritten by inserting the expression for $\underline{q}_{n+1}$ found in Equation 6.17.

$$\underline{c_s}_{n+1} = \underline{c_s}_n + \Delta t\big[\underline{c}_{n+1} - \underline{c_s}_{n+1} - \frac{1}{1-\epsilon}(K\underline{c_s}_{n+1}^{\,p} - \frac{1}{1+\Delta t}(\underline{q}_n + \Delta tK\underline{c_s}_{n+1}^{\,p})\big] \tag{6.18}$$

$$\Rightarrow\ \underline{c_s}_{n+1} = \underline{c_s}_n + \Delta t\underline{c}_{n+1} - \Delta t\underline{c_s}_{n+1}\frac{\Delta t}{1-\epsilon}K(1 - \frac{\Delta t}{1+\Delta t})\underline{c_s}_{n+1}^{\,p} - \frac{\Delta t}{(1-\epsilon)(1+\Delta t)}\underline{q}_n \tag{6.19}$$

Now Equations 6.14 and 6.19 will be rewritten by moving all terms to the left hand side.

$$\big(I - \frac{\Delta t}{\epsilon}A + \Delta t\frac{1-\epsilon}{\epsilon}I\big)\underline{c}_{n+1} - \frac{\Delta t(1-\epsilon)}{\epsilon}\underline{c_s}_{n+1} - \underline{c}_n\frac{\Delta t}{\epsilon}a = 0 \tag{6.20}$$

$$-\Delta tI\underline{c}_{n+1} + (1+\Delta t)I\underline{c_s}_{n+1} - \underline{c_s}_n + \frac{\Delta t}{1-\epsilon}K(1 - \frac{\Delta t}{1+\Delta t})\underline{c_s}_{n+1}^{\,p} - \frac{\Delta t}{(1-\epsilon)(1+\Delta t)}\underline{q}_n = 0 \tag{6.21}$$

In the simple model the next step would be to find function to which $\underline{c}_{n+1}$ and $\underline{c_s}_{n+1}$ are the zeros, so the Newton-Raphson method can be used. But this is in this case

not an option. To find a function on which the Newton-Raphson method can be used, the following vector is introduced.

$$\underline{c^*} = \begin{pmatrix} \underline{c}_{n+1} \\ \underline{c_s}_{n+1} \end{pmatrix} = \begin{pmatrix} \underline{c^*}_1 \\ \underline{c^*}_2 \end{pmatrix} \tag{6.22}$$

Now Equations 6.20 and 6.21 can be combined.

$$\begin{pmatrix} I - \frac{\Delta t}{\epsilon}A + \Delta t\frac{1-\epsilon}{\epsilon}I & -\Delta t\frac{1-\epsilon}{\epsilon}I \\ -\Delta t I & (1+\Delta t)I \end{pmatrix} \underline{c^*} - \begin{pmatrix} \underline{c}_n \\ \underline{c_s}_n \end{pmatrix} - \begin{pmatrix} \frac{\frac{\Delta t}{\epsilon}a}{\frac{\Delta t}{(1-\epsilon)(1+\Delta t)}\underline{q}_n} \end{pmatrix} + \frac{\Delta t}{1-\epsilon}K(1 - \frac{1}{1+\Delta t}) \begin{pmatrix} 0 \\ \underline{c^*}_2 \end{pmatrix}^p = 0 \tag{6.23}$$

So $\underline{c^*}$ is the zero of the following formula:

$$f(\underline{x}) = \begin{pmatrix} I - \frac{\Delta t}{\epsilon}A + \Delta t\frac{1-\epsilon}{\epsilon}I & -\Delta t\frac{1-\epsilon}{\epsilon}I \\ -\Delta t I & (1+\Delta t)I \end{pmatrix} \underline{x} - \begin{pmatrix} \underline{c}_n \\ \underline{c_s}_n \end{pmatrix} - \begin{pmatrix} \frac{\frac{\Delta t}{\epsilon}a}{\frac{\Delta t}{(1-\epsilon)(1+\Delta t)}\underline{q}_n} \end{pmatrix} + \frac{\Delta t}{1-\epsilon}K(1 - \frac{1}{1+\Delta t}) \begin{pmatrix} 0 \\ \underline{x}_2 \end{pmatrix}^p \tag{6.24}$$

To find $\underline{c^*}$, we have to find the zero of $f(\underline{x})$. To do this the Newton-Raphson method will be used. This method goes as follows:

$$\underline{x}_{i+1} = \underline{x}_i - J_f(\underline{x}_i)^{-1} f(\underline{x}_i) \tag{6.25}$$

In our case the following holds true:

$$\underline{x_i} = \begin{pmatrix} \underline{c}_i \\ \underline{c_s}_i \end{pmatrix} = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix} \tag{6.26}$$

$$J_f(\underline{x}_i)^{-1} = J^{-1} \tag{6.27}$$

$$i \neq j \Rightarrow J_{ij} = (I - \frac{\Delta t}{\epsilon}A)_{ij} \tag{6.28}$$

$$i = j \Rightarrow J_{ij} = \begin{pmatrix} I - \frac{\Delta t}{\epsilon}A + \Delta t\frac{1-\epsilon}{\epsilon}I & -\Delta t\frac{1-\epsilon}{\epsilon}I \\ -\Delta t I & (1+\Delta t)I \end{pmatrix}_{ij} + \frac{\Delta t}{1-\epsilon}K\big(1 - \frac{1}{1+\Delta t}\big)p \begin{pmatrix} 0 \\ \underline{x}_2 \end{pmatrix}_j^{p-1} \tag{6.29}$$

The method will iterate until $f(x_i) \approx 0$. With $\underline{c^*}$ known, $\underline{c}_{n+1}$ and $\underline{c_s}_{n+1}$ are easily computed. Now that $\underline{c}_{n+1}$ is approximated, $\underline{q}_{n+1}$ is also easily computed by filling in Equation 6.17.

6.3 Program implementation

Now that the numerical methods for the extended model are established, a Python program can be constructed. The created program can be found in the Appendix A. Note that the code contains comments which gives a a more detailed explanation then given here.

The main code can be found in the file *Extended.py*. The code greatly resembles the program discussed in Section 5.3. The program is able to simulate the concentrations of c, c_s, q and q_s through time. The same values for ϵ and K are used.

$$\epsilon = 0.40$$
$$K = 0.004087$$

Other variables such as Δt, Δx and p will again be varied to see the effects on the simulation.

Some less significant functions that are needed can be found in the file *Extended-Functions.py*. These are put into a separate file to keep the main code accessible. These functions are for example the power function of a vector, the creation of the upwind matrix and functions to plot a concentration.

7 Results and Discussion

This section will discuss the results generated by the Python program that was used to approximate the concentration of a pollutant through time. The results will be discussed and a conclusion on the correctness of the solution will be made.

When using the program certain values for Δx, Δt, p (the isotherm) and the final time at which we want to approximate the concentrations (T) have to be chosen. The same values will be used as in the simple model. So $\Delta x = \frac{1}{20}$ and $\Delta t = 0.001$. p and T will be varied.

Figures 17 to 20 show the approximated concentrations of c, c_s q and q_s at certain time-steps. The value for p was set to 1. This value of p will result in a linear problem. The approximations were given by the Forward Euler method. The results of the Backward Euler method are not shown because they are very similar. These results can be found by following the link in the appendix A.

The concentrations through time behave very similar to the concentrations of the simple model. However, now the concentration of c at $x = 0$ will increase to 1 at a significantly slower rate. Thus at the exit of the filter a part of the pollutant has been filtered from the bulk liquid. Through time the concentration of c_s, q and q_s will slowly rise to a stable limit. So the filter is slowly filling up through time. The fuller the filter becomes the higher the concentration of c at the exit. The concentration of c is still relatively high. This is likely due to the unrealistic value of certain parameters.

The impact of p on the system seems to be minimal when looking at the concentrations through time, just as was observed in the simple model. The varying of p does have an effect of the estimated errors and the orders.

The order of the errors are displayed in Figures 27 and 28. The orders of the Backward Euler method are equal for every chosen value of p. The orders of the Forward Euler method change when varying the value p. The orders of the Forward Euler method are higher than the orders of the Backward Euler method. However, this is for the chosen values of p. Perhaps a different value of p will result in lower orders. This has to be investigated further.

The estimated errors are displayed in Figures 21 to 26. The errors of the Forward and Backward Euler method converge to the same error. The Backward Euler method approaches this limit from above and the Forward Euler method approaches this limit from below.

The computation time needed to complete the simulation is displayed in Figures 29 and 30. The Forward Euler method is faster than the Backward Euler method. The computation time is roughly ten times larger when using the Backward Euler method. The big difference in the computation time could be caused by the allowed error in the Newton-Raphson method. Choosing a larger allowed error will lead to a faster simulation.

The system seems to be stable for both the Forward Euler method as the Backward Euler method. The orders for the Backward Euler method are stable for any value

of p. When varying p the orders of the Forward Euler method fluctuate a bit. For a value of $p = \frac{1}{2}$ the orders are higher than for the other values of p. It is likely that different values for p will also change the orders for the Forward Euler method. Perhaps a certain value of p will result in an order lower than one. This has to be investigated further. Choosing between the Backward and Forward Euler method is a trade-off between speed and reliability. The Forward Euler method is likely to produce a fast and accurate result. However for certain values of p the order could result in unwanted values for the orders. Which could lead to less accurate results.

The results produced by the expanded model are closer to reality than the simple model. Now the liquid will leave the filter with a lower concentration of the pollutant until the filter is full. The next step is to expand the model and add certain overlooked variables to see if the solution becomes closer to the reality.

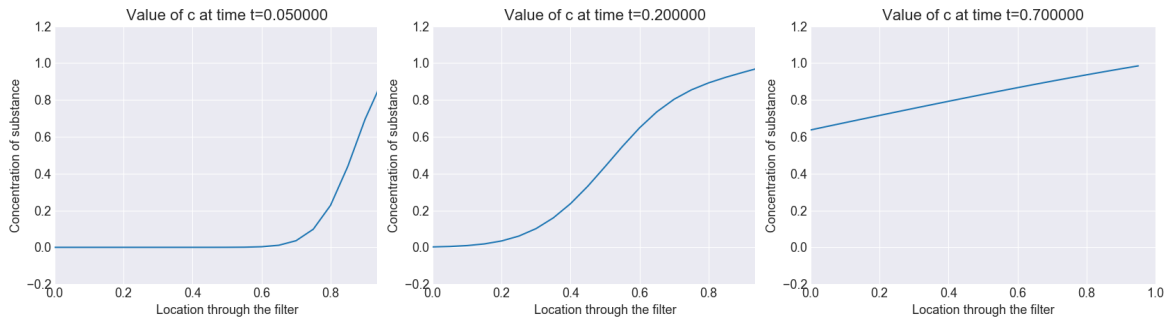

Figure 17: The concentration of c displayed at three different time-steps, t=0.05, t=0.2 and t=0.7

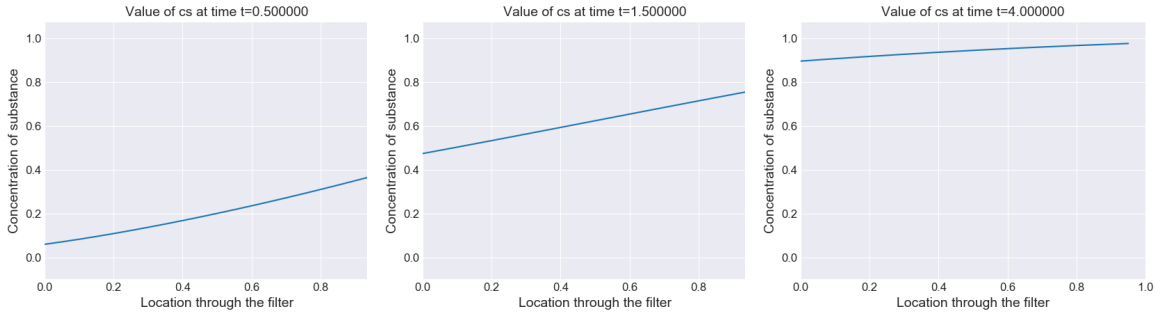

Figure 18: The concentration of c_s displayed at three different time-steps, t=0.5, t=1.5 and t=4

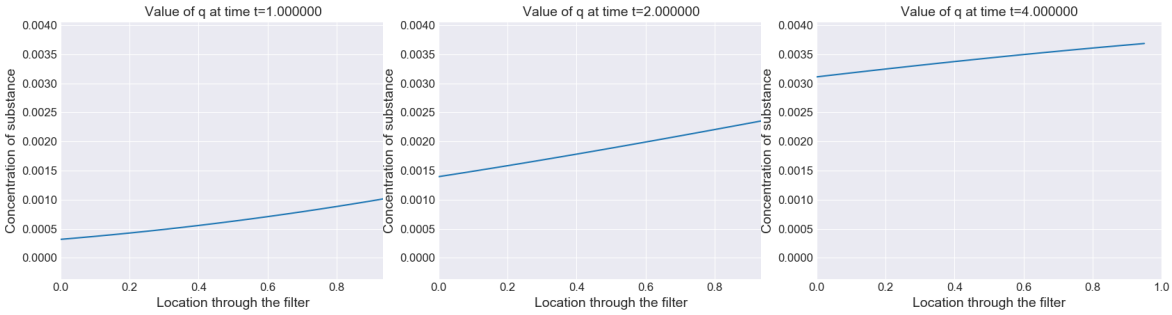

Figure 19: The concentration of q displayed at three different time-steps, t=1, t=2 and t=4

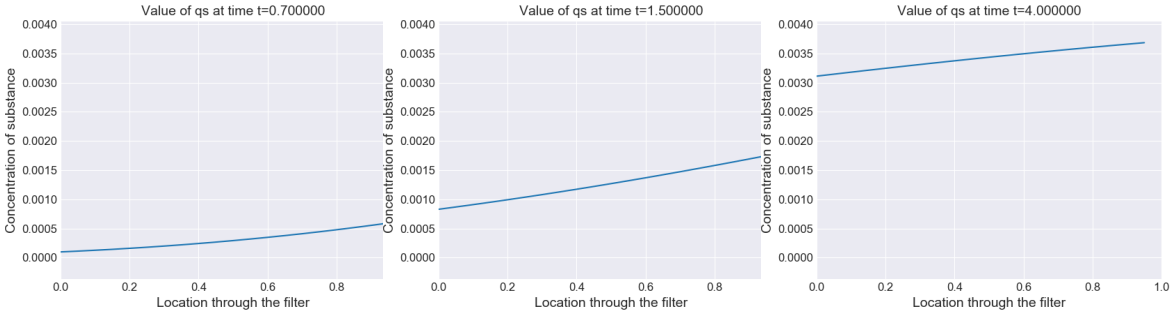

Figure 20: The concentration of qs displayed at three different time-steps, t=0.7, t=1.5 and t=4

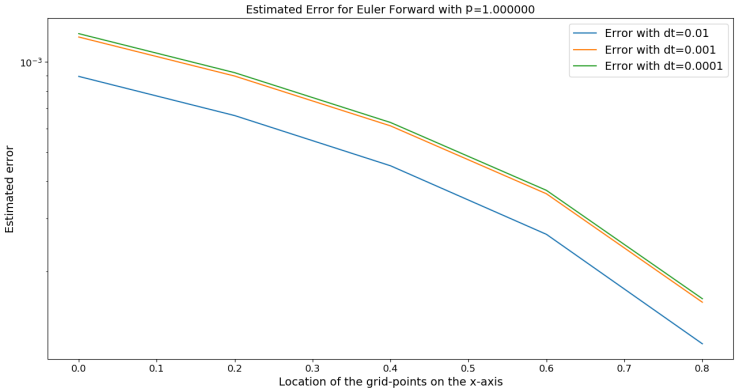

Figure 21: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 1$.

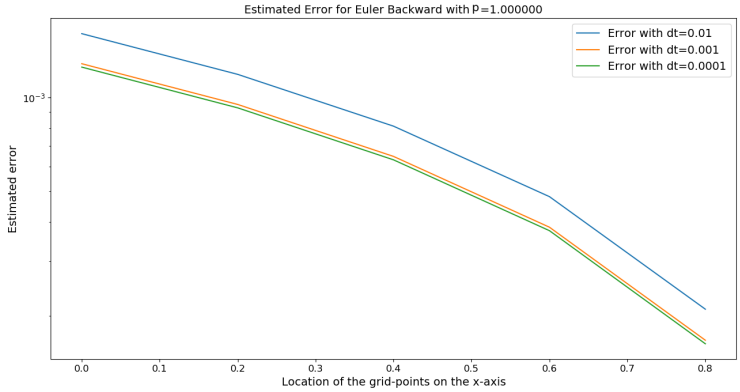

Figure 22: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 1$.

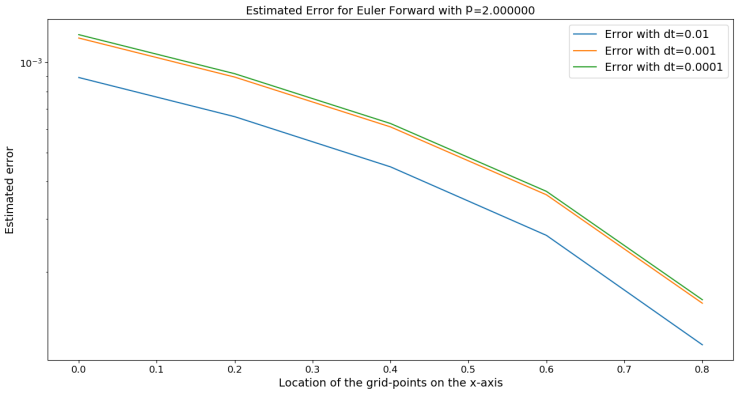

Figure 23: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 2$.

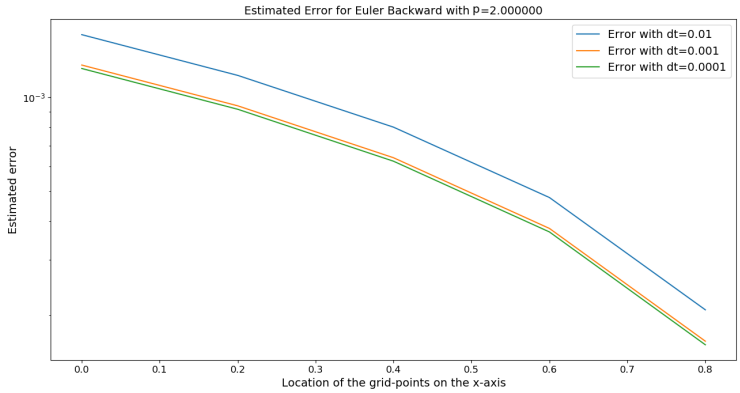

Figure 24: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 2$.

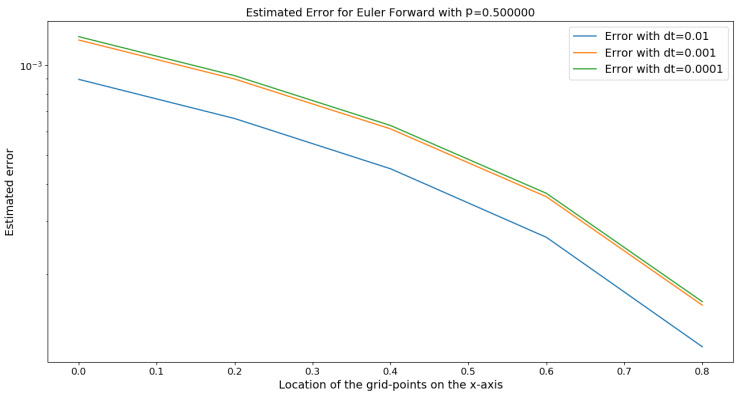

Figure 25: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 0.5$.

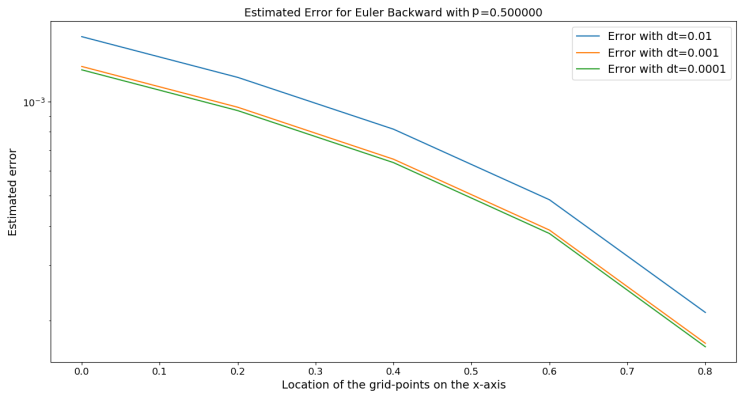

Figure 26: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 0.5$.

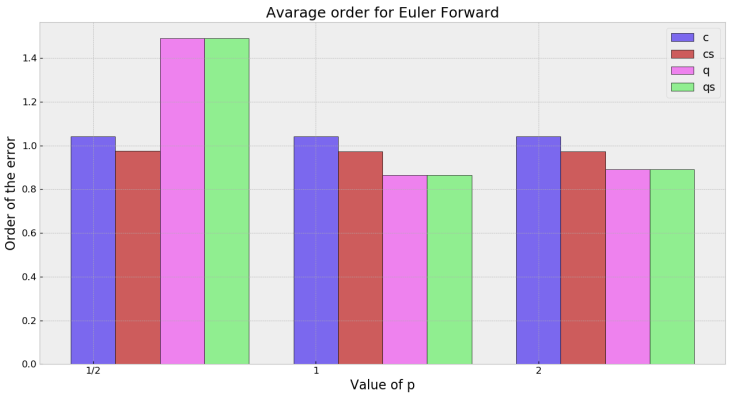

Figure 27: The order of the error when using the Forward Euler method.

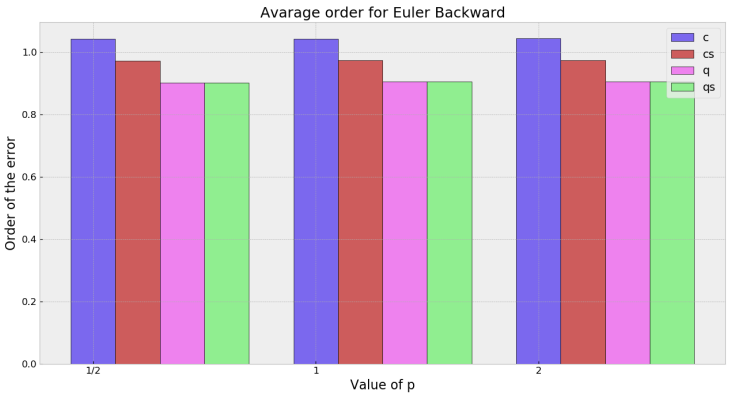

Figure 28: The order of the error when using the Backward Euler method.

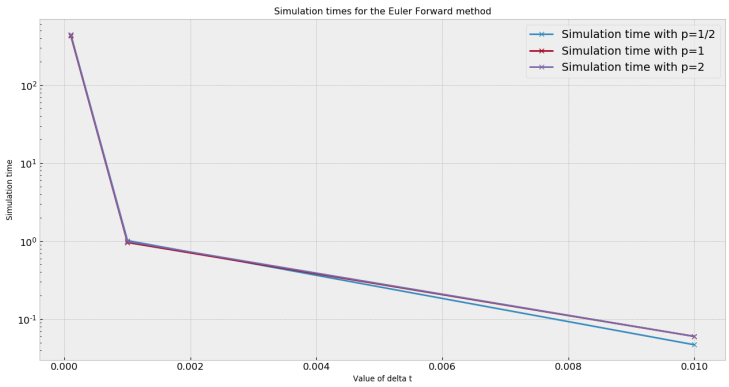

Figure 29: Simulation time needed when using the Forward Euler method.

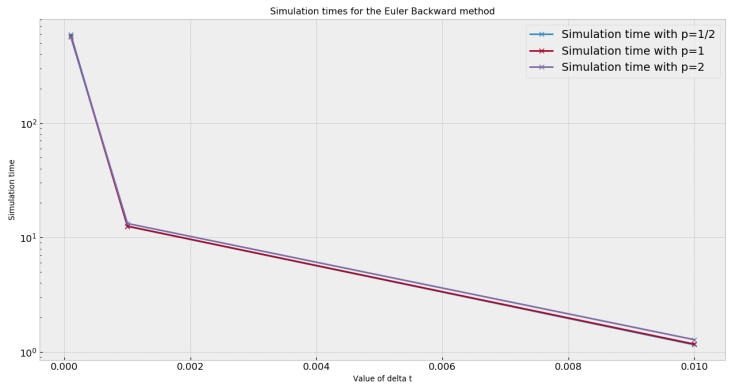

Figure 30: Simulation time needed when using the Backward Euler method.

8 Filling in parameters

The last step that will be made in this research is filling in realistic values for the parameters into the extended model that were neglected earlier. This section will establish the final model with all the parameters filled in. A Python program will be constructed to simulate the concentrations through time. The results generated by the program will be discussed.

8.1 Filtration model

The following model will be used.

$$\frac{\partial c}{\partial t} = \frac{v}{\epsilon}\frac{\partial c}{\partial x} - \frac{6(1-\epsilon)\beta_f}{\epsilon d}(c_s - c)\,, \qquad 0 \leq x \leq L \tag{8.1}$$

$$\frac{dc_s}{dt} = \frac{\beta_f}{\delta}(c - c_s) - \frac{\rho\beta_s}{\delta(1-\epsilon)}(q_s - q) \tag{8.2}$$

$$\frac{dq}{dt} = \frac{6\beta_s}{d}q_s - q \tag{8.3}$$

$$q_s = Kc_s^p \tag{8.4}$$

The meaning of all the newly introduced parameters has been explained in Section 3.3 and can be found in the Appendix B. Note that the notation is slightly different from the model displayed in Section 3.3. The parameters d, ρ and δ are shorter notations of d_{part}, ρ_{filt} and δ_s.

For each of these parameters a certain value will have to be chosen. The values for L and v will later be set such that the simulation will require a low computation time. Setting the water flow and interval length could lead to slow simulations. The Freundlich parameters, K_f and p will be varied based on the values given by papers written by M.A.S. Niandou et al.[9] and Y.B. Kumar et al.[10]. The remaining variables will be chosen based on the study by van der Aa et al.[6]. Below all the chosen values for the parameters are displayed.

$$\begin{aligned} \epsilon &= 0.4 & \beta_f &= 4.9 \cdot 10^{-5} \\ d &= 0.0012 & D_s &= 2,4 \cdot 10^{-15} \\ \delta &= 1 \cdot 10^{-5} & \beta_s &= 10\frac{D_s}{d} \end{aligned}$$

The filtration model will use the following boundary conditions.

$$\begin{aligned} c(L,t) &= 1,79 \cdot 10^{-3} \\ c(x,0) &= 0, \qquad 0 \leq x < L \\ c_s(x,0) &= 0 \\ q(x,0) &= 0 \end{aligned} \tag{8.5}$$

In these boundary conditions $c(L,t)$ represents the incoming concentration of the pollutant. This value is chosen based on the study of van der Aa et al.[6].

8.2 Numerical methods

The same numerical methods will be used as on the extended model 6.2. The only difference in the computations will be the parameters. Therefore the implementation of the numerical methods is rather straightforward and will not be displayed in this report. Note that also the same notation will be used as in Section 6.2.

Forward Euler

The time-steps for the Forward Euler are made in the following way.

$$\begin{aligned}
\underline{c}_{n+1} &= \underline{c}_n + \Delta t\big[\frac{v}{\epsilon}(A\underline{c}_n + a) - \frac{6(1-\epsilon)\beta_f}{\epsilon d}(\underline{c}_n - \underline{c_s}_n)\big] \\
\underline{c_s}_{n+1} &= \underline{c_s}_n + \Delta t\big[\frac{\beta_f}{\delta}(\underline{c}_n - \underline{c_s}_n) - \frac{p\beta_s}{\delta(1-\epsilon)}(K\underline{c_s}_n^p - \underline{q}_n\big] \\
\underline{q}_{n+1} &= \underline{q}_n + \Delta t\big[\frac{6\beta_s}{\delta}(K\underline{c}_n^p - \underline{q}_n)\big]
\end{aligned} \tag{8.6}$$

Backward Euler

To compute $\underline{c}_{n+1}$ and $\underline{c_s}_{n+1}$ the Newton-Raphson method is needed. The function f and the Jacobian matrix that are used are displayed in Equations 8.7 and 8.8.

$$\begin{aligned}
f(\underline{x}) = \begin{pmatrix} I - \frac{\Delta t v}{\epsilon}A + \Delta t\frac{6(1-\epsilon)\beta_f}{\epsilon d}I & -\Delta t\frac{6(1-\epsilon)\beta_f}{\epsilon d}I \\ -\Delta t\frac{\beta_f}{\delta}I & (1+\Delta t\frac{\beta_f}{\delta})I \end{pmatrix}\underline{x} - \begin{pmatrix} \underline{c}_n \\ \underline{c_s}_n \end{pmatrix} - \begin{pmatrix} \frac{\Delta t v}{\epsilon}a \\ \frac{\Delta t p\beta_s}{\delta(1-\epsilon)(1+\Delta t\frac{6\beta_s}{d})}\underline{q}_n \end{pmatrix} \\
+ \frac{\Delta t p\beta_s}{\delta(1-\epsilon)}K(1 - \frac{6\beta_s\Delta t}{1+\Delta t\frac{6\beta_s}{d}})\begin{pmatrix} 0 \\ \underline{x}_2 \end{pmatrix}^p
\end{aligned} \tag{8.7}$$

$$\begin{aligned}
&i \neq j \Rightarrow J_{ij} = (I - \frac{\Delta t}{\epsilon}A)_{ij} \\
&i = j \Rightarrow J_{ij} = \begin{pmatrix} I - \frac{\Delta t v}{\epsilon}A + \Delta t\frac{6(1-\epsilon)\beta_f}{\epsilon d}I & -\Delta t\frac{6(1-\epsilon)\beta_f}{\epsilon d}I \\ -\Delta t\frac{\beta_f}{\delta}I & (1+\Delta t\frac{\beta_f}{\delta})I \end{pmatrix}\underline{x} + \frac{\Delta t p\beta_s}{\delta(1-\epsilon)}K(1 - \frac{6\beta_s\Delta t}{1+\Delta t\frac{6\beta_s}{d}})\begin{pmatrix} 0 \\ \underline{x}_2 \end{pmatrix}^p
\end{aligned} \tag{8.8}$$

With the Newton-Raphson method $\underline{c}_{n+1}$ and $\underline{c_s}_{n+1}$ can be computed. Now $\underline{q}_{n+1}$ can be computed as follows.

$$\underline{q}_{n+1} = \frac{1}{1+\Delta t\frac{6\beta_s}{d}}\big(\underline{q}_n + \Delta t\frac{p\beta_s}{\delta(1-\epsilon)}K\underline{c_s}_{n+1}\big) \tag{8.9}$$

8.3 Program implementation

A Python code is constructed to simulate the adsorption of the concentrations as a function of the time. This code can be be found in the Appendix A. The code is very similar to the code used for the simulations of the Extended model in Section 6.3. The functions are the same for the exception of the introduction of the parameters.

The values of the parameters K and p will be varied. These values will be based of the studies by M.A.S. Niandou et al.[9] and Y.B. Kumar et al.[10]. The paper by Nianou et al. displayd the Freundlich parameters of six different types of carbon filters. Of these six sets of parameters two will be used. The study by Y.B. Kumar et al. gives one set of Freundlich parameters. The following parameters will be used and their results will be compared.

$$\begin{aligned} p &= 0.71 & K &= 4.59 \\ p &= 0.78 & K &= 5.81 \\ p &= 1.162 & K &= 36.430 \end{aligned}$$

The code can be found in the file *Parameters.py*. The code is able to predict the concentrations of c, c_s, q and q_s through time. Some less significant functions can be found in the file *ParametersFunctions.py*. These functions include functions used for plots, the creation of the upwind matrix and functions that output the power of a vector or certain value.

9 Results and discussion

In this section the results generated by the Python program will be displayed and discussed. The values for L and v have yet to be chosen. The program will not use realistic values for the exception of L and v. These will still be kept at 1 to speed up the program. Some other parameters which have yet to be chosen will be set as follows: $\Delta x = \frac{1}{20}$ and $\Delta t = 0.001$.

Figures 31 to 33 show the concentrations of c through time for the three chosen sets of Freundlich parameters. For every time-step the concentrations of each set of parameters is almost equal. When comparing the concentration at the end of the filter ($x = 0$) the differences are very small. The following concentrations were computed for $t = 0.7$.

$$\text{Final concentration for } p = 0.71 \text{ and } K = 4.59 : 0.0011113686601762786$$
$$\text{Final concentration for } p = 0.78 \text{ and } K = 5.81 : 0.0011113686601762786$$
$$\text{Final concentration for } p = 1.162 \text{ and } K = 36.430 : 0.0011113853866299005$$

It is expected that the different filter types have a similar efficiency. Perhaps when simulating over a longer time, bigger differences will be observed between the filters.

When observing the final concentrations and the Figures 31 to 33 it seems that the values of p and K have almost no influence on the efficiency of the filter. This is not the case however. Figures 35 to 38 show the concentrations at the end of the filter for the chosen parameter sets, where one parameter of the pair is fixed and the other is being varied. These figures clearly show the effect of the parameters on the system. Figures 35, 37 and 39 show that a lower value of p will lead to a lower concentration at the end of the filter and thus a more effective filter. Figures 34, 36 and 38 show that a higher value of K will lead to a lower concentration at the end of the filter and thus a more effective filter.

The orders of the errors are displayed in Figures 46 to 47. The orders are identical for every value of p for both the Forward Euler method and the Backward Euler method. The errors are all close to 1 indicating a stable numerical method.

The estimated errors are displayed in Figures 40 to 45. Note that in previous chapters the error was also displayed for $dt = 0.01$. When using the program this value for dt cause an error and therefore it is missing from the figures. The estimated errors are the same for the Forward and Backward Euler method. The errors are small for every set of parameters. The errors seem to be very small when compared to the errors found in the simple model and extended model. However, it must be noted that the concentrations of c in this model is also a lot smaller. This is likely the cause of the decrease in error. The errors are still relatively small when compared to the values of the solution. It is recommended that in future research the relative error made in each model is compared.

The simulation time is displayed in Figures 48 and 49. Just as was observed in the simple and extended model, the Backward Euler method requires a longer simulation time. The parameter p has little influence on the simulation time.

The model seems to produce accurate results. The concentration of c decreases the further it gets through the filter. Thus a part of the pollutant will be filtered out of the liquid. When looking at the orders and the estimate errors both the Forward and Backward Euler method it is clear that both methods provide accurate and stable results. It is still unclear why a value of $dt = 0.01$ causes an error. This has to be further investigated. The exiting concentrations are still very high which is not realistic. To obtain a more realistic result it would be advised to use different values for the parameters L and v. A lower water flow and a longer interval will likely result in a lower concentration of c at the end of the filter. However this must be further investigated.

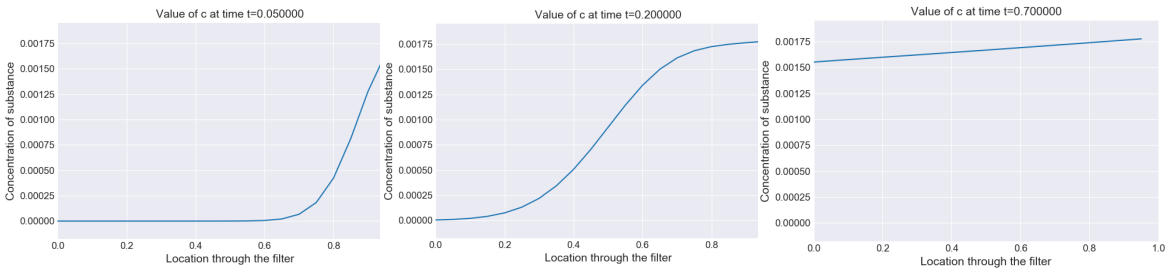

Figure 31: The concentration of c displayed at three different time-steps, t=0.05, t=0.2 and t=0.7 with $p = 0.71$ and $K = 4.59$

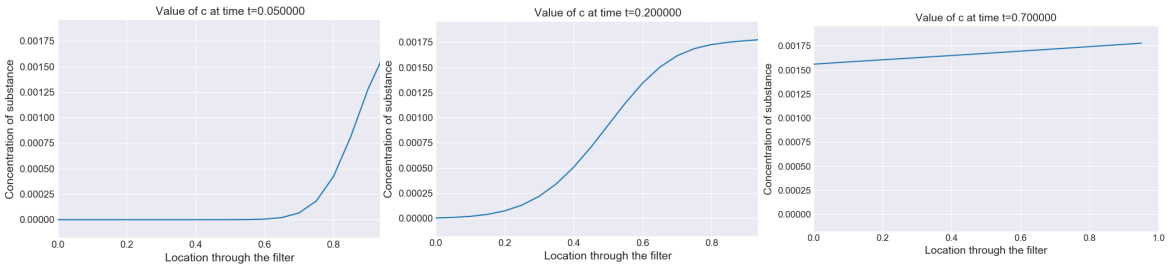

Figure 32: The concentration of c displayed at three different time-steps, t=0.05, t=0.2 and t=0.7 with $p = 0.78$ and $K = 5.81$

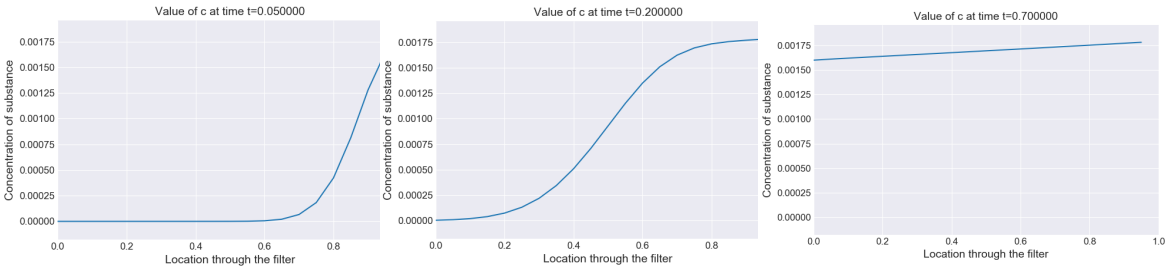

Figure 33: The concentration of c displayed at three different time-steps, t=0.05, t=0.2 and t=0.7 with $p = 1.162$ and $K = 36.430$

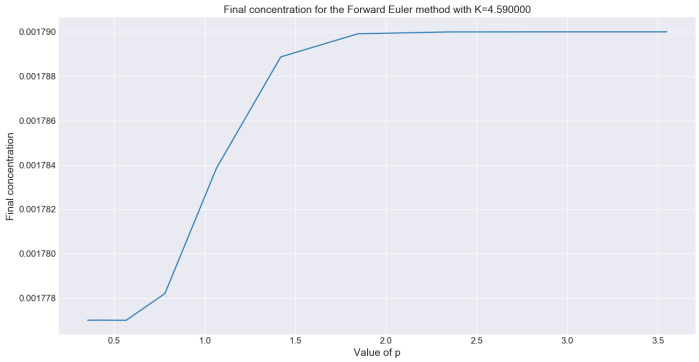

Figure 34: The final concentration of c at the end of the filter computed with the Forward Euler method and a value of $p = 0.71$.

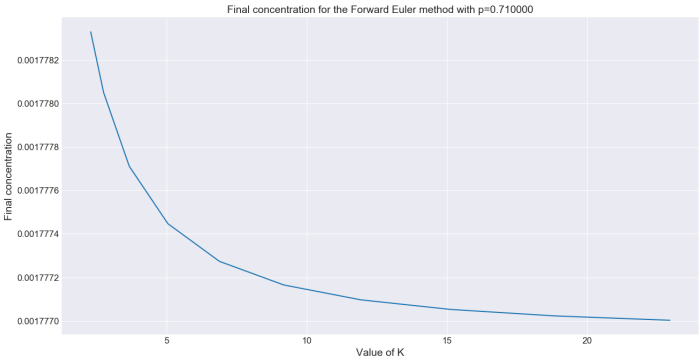

Figure 35: The final concentration of c at the end of the filter computed with the Forward Euler method and a value of $K = 4.59$.

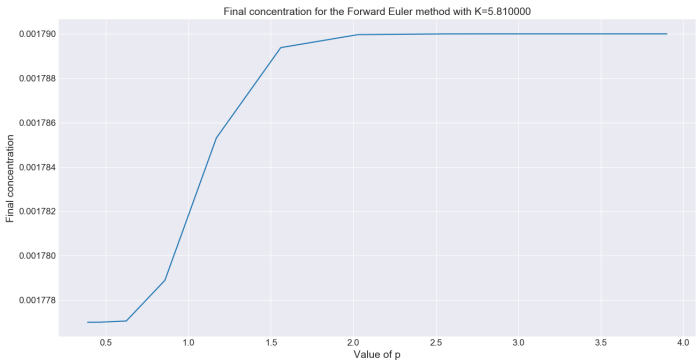

Figure 36: The final concentration of c at the end of the filter computed with the Forward Euler method and a value of $p = 0.78$.

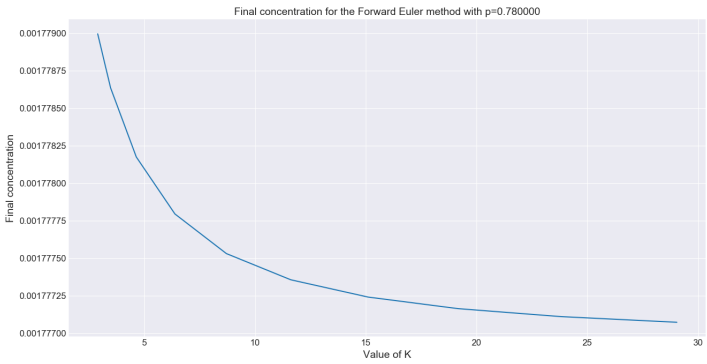

Figure 37: The final concentration of c at the end of the filter computed with the Forward Euler method and a value of $K = 5.81$.

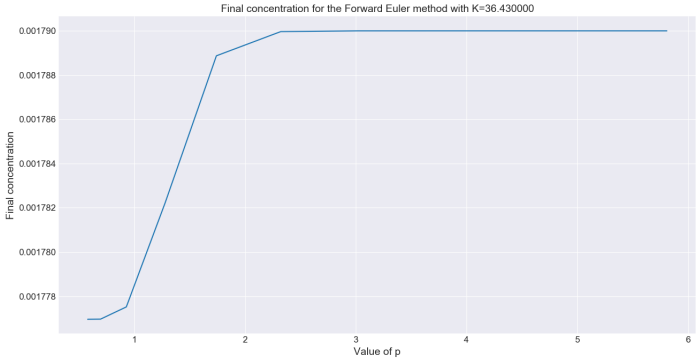

Figure 38: The final concentration of c at the end of the filter computed with the Forward Euler method and a value of $p = 1.162$.

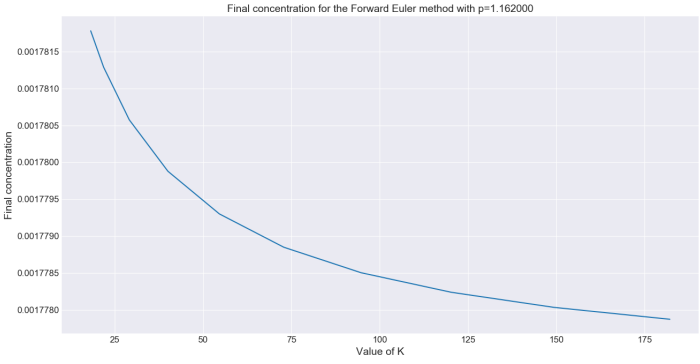

Figure 39: The final concentration of c at the end of the filter computed with the Forward Euler method and a value of $K = 36.43$.

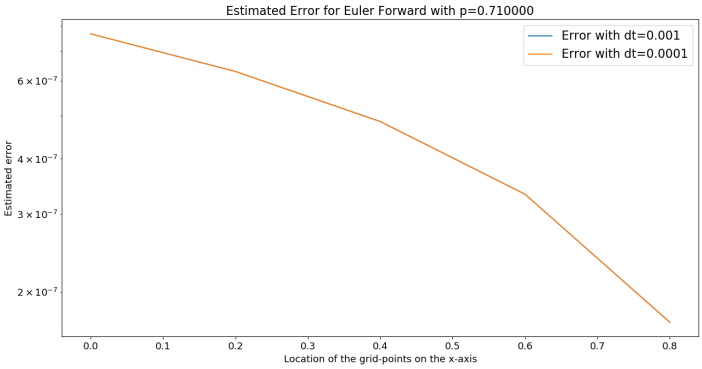

Figure 40: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 0.71$ and $K = 4.59$.

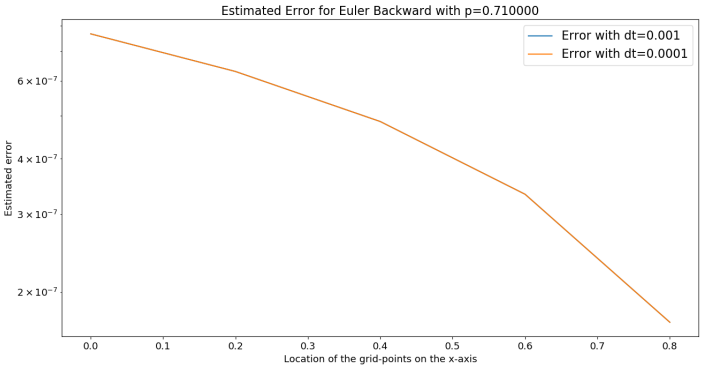

Figure 41: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 0.71$ and $K = 4.59$.

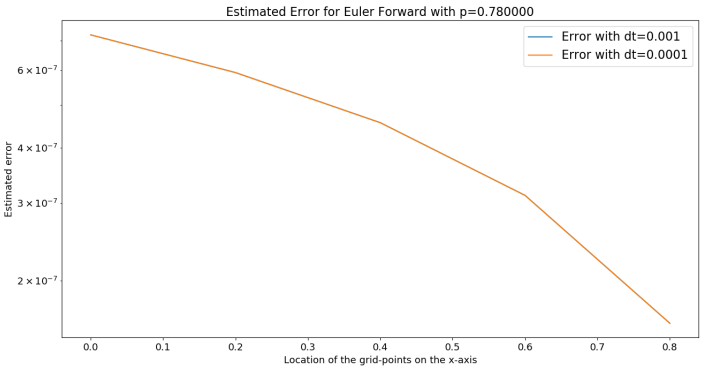

Figure 42: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 0.78$ and $K = 5.81$.

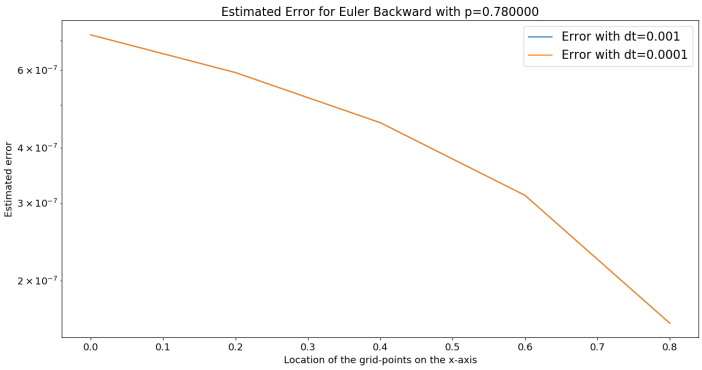

Figure 43: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 0.78$ and $K = 5.81$.

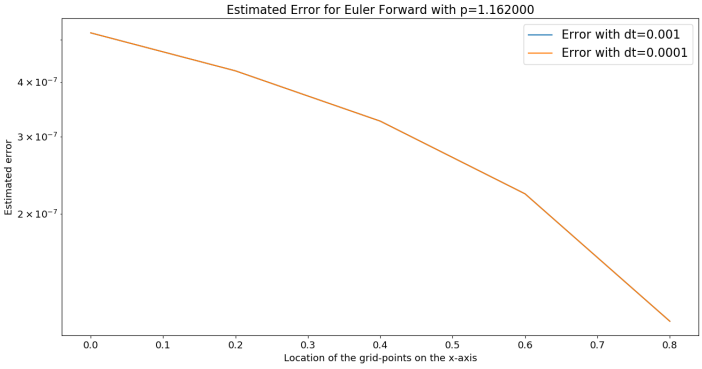

Figure 44: The estimated error of c for the Euler Forward method on certain grid-points determined by the Richardson extrapolation using $p = 1.162$ and $K = 36.430$.

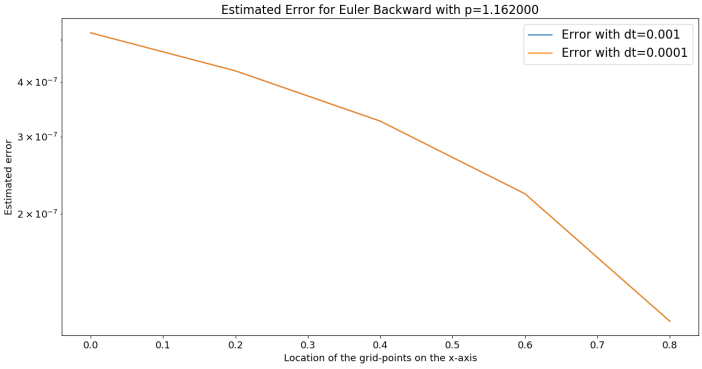

Figure 45: The estimated error of c for the Euler Backward method on certain grid-points determined by the Richardson extrapolation using $p = 1.162$ and $K = 36.430$.

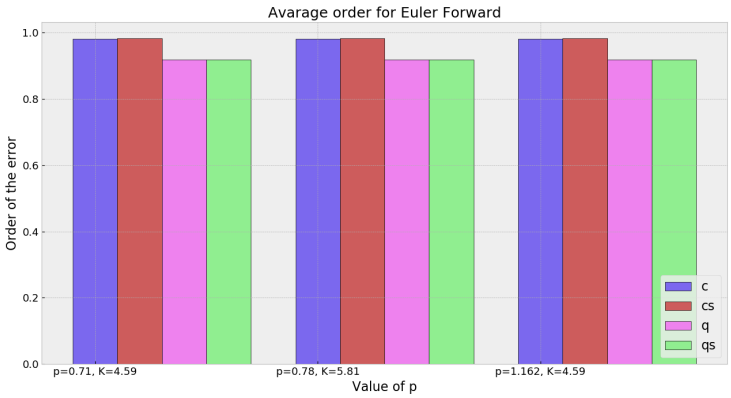

Figure 46: The order of the error when using the Forward Euler method.

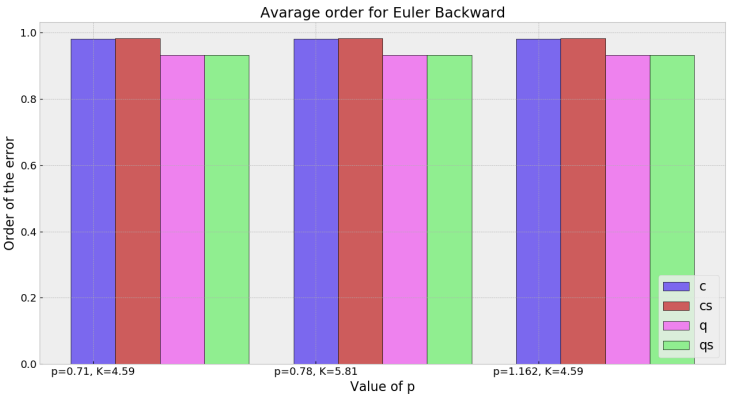

Figure 47: The order of the error when using the Backward Euler method.

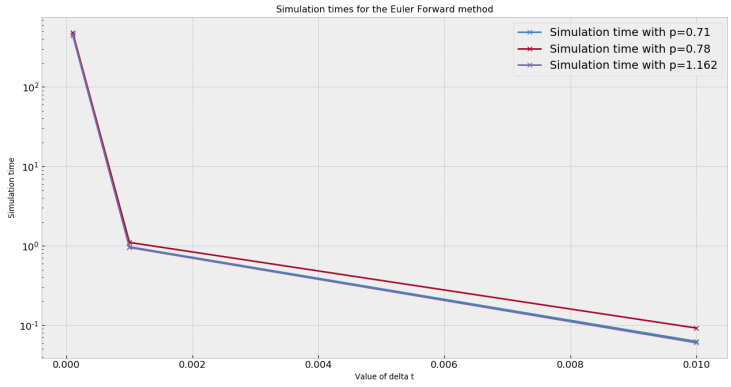

Figure 48: Simulation time needed when using the Forward Euler method.

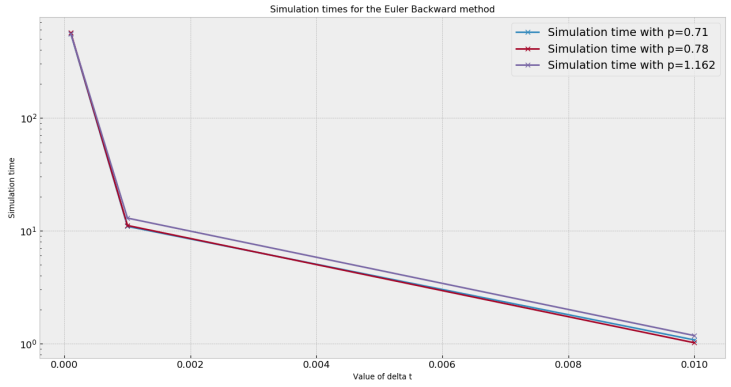

Figure 49: Simulation time needed when using the Backward Euler method.

10 Conclusion and Recommendations

This report established three models and examined the effectiveness of an implicit and an explicit numerical method to approximate the solution of these models.

The simple model and the extended model show similar results. For both models the Backward Euler method required a longer computation time but provided more stable orders. The Forward Euler method also gives accurate results. However, the model has only been tested for certain chosen values of p. For different values it is likely that the Backward Euler method will still be equally accurate. For the Forward Euler method this may not be true due to the fluctuating orders. Therefore it is advised to use the Backward Euler method.

The model including realistic values for the parameters behaves differently than the other two models. The orders and the errors are equal for the Forward and Backward Euler method. In the extended model it was observed that the error will converge to a limit when Δt decreases. For the Forward Euler method the error will approach the limit from below. For the Backward Euler method the error will approach the limit from above. This cannot be observed in this model because the errors for $\Delta t = 0.001$ and $\Delta t = 0.0001$ are equal. It is advised to use experiment with different values for Δt. However, it is likely that the error is mainly determined by spatial discretization. Therefore different values of Δt are expected to give the same result. The orders are also the same for the Forward and Backward Euler method. However, this was the case for the chosen parameter sets. This may not be true for different parameter sets, since the extended model also only shows a difference in orders for $p = \frac{1}{2}$.

Thus for the simple and extended model the conclusion is clear. The Backward Euler method is the method which will result in the most accurate result where the Forward Euler method requires less computation time.

The conclusion for the model with the realistic parameters is less conclusive. The model gives equally accurate results for both numerical methods. However, it is unclear if this will be the case when using smaller values for dt, different parameter sets and realistic values for L and v. For now the Forward Euler method is clearly the better method since this method is equally accurate and requires a lower computation time. However more research needs to be done to confirm this.

References

[1] M. A. Shannon, P. W. Bohn, M. Elimelech, J. G. Georgiadis, B. J. Marinas, and A. M. Mayes, "Science and technology for water purification in the coming decades," in *Nanoscience And Technology: A Collection of Reviews from Nature Journals*, pp. 337–346, World Scientific, 2010.

[2] "Water crises are a top global risk." `https://www.weforum.org/agenda/2015/01/why-world-water-crises-are-a-top-global-risk/`, 2015. World Economic Forum, Accessed: 02-5-2019.

[3] "Productie van drinkwater, 1950-2015." `https://www.clo.nl/indicatoren/nl0045-productie-van-drinkwater`, 2017. World Economic Forum, Accessed: 02-5-2018.

[4] D. Vries, B. Wols, M. Korevaar, and E. Vonk, "Aquapriori: a priori het verwijderingsrendement bepalen," *KWR*, 2017.

[5] "Kwr - predicting the removal performance of activated carbon filters in water treatment." `https://www.swi-wiskunde.nl/swi2019/problems/predicting-the-removal-performance-of-activated-carbon-filters-in-water-treatment/`, 2019. Studiegroep Wiskunde met de Industrie, Accessed: 25-4-2019.

[6] L. van der Aa, L. Rietveld, and J. van Dijk, "Simultaneous removal of natural organic matter and atrazine in biological granular activated carbon filters: model validation," *IWA Specialist Conference on Water and Wastewater Treatment Plants in Towns and Communities of the XXI Century: Technologies,Design Operation.*, 2010. Mocow, Russia.

[7] H. Marsh and F. Rodríguez-Reinoso, *Activated Carbon*. Elsevier, 2006.

[8] P. Lodewyckx, *Adsorption on Activated Carbon: One Underlying Mechanism?* Dordrecht: Springer, 2008.

[9] M. Niandou, S. Luster-Teasley, J. Novak, D. Rehrah, and M. Ahmedna, "Adsorption isotherm parameters of atrazine and metolachlor with pecan shell-based activated carbons," *International Journal of Agricultural Science and Technology*, 2016.

[10] B. Yengkhom, N. Singh, and S. Sing, "Removal of atrazine, metribuzin, metolachlor and alachlor by granular carbon," *Journal of environmental and analytical toxicology*, vol. 3, 2013.

A Link to downloadable files

The link given hereunder will lead to a downloadable file containing the python programs and all results. The file contains a text file which explains all the contents of the file. When viewing this document in PDF format the link is clickable and will lead directly to the site.

Link: https://www.dropbox.com/sh/o6sou7mf5gogsk2/AABdUeui2aDKkHV7XhCPtviba?dl=1.

B Overview of all used parameters

Table 1: Parameters used in the filtration model

Parameters	Description
c_i	Concentration of component i in the water stream (gCm^{-3})
$c_{s,i}$	Concentration of dissolved component i in the surface liquid film (gCm^{-3})
$c_{p,i}$	Concentration of dissolved component i in the pore (gCm^{-3})
$q_{s,i}$	Concentration of the solid phase component i at the surface of the pore (gCm^{-3})
$q_{p,i}$	Concentration of the solid phase component i in the pore (gCm^{-3})
v	Flow rate of the water through the filter (ms^{-1})
x	Distance in the direction of the flow (m)
t	Time (s)
T	Temperature ($°C$)
ϵ	Filter bed porosity (-)
d_{part}	AC particle diameter (m)
$\beta_{f,i}$	Film diffusion mass transfer coefficient of component i (ms^{-1})
$\beta_{s,i}$	Surface diffusion mass transfer coefficient of component i (ms^{-1})
ρ_{filt}	AC mass density of a packed filter bed (gm^{-3})
δ_s	Thickness of the surface liquid film (m)
μ_{spec}	Specific growth rate of the biomass (s^{-1})
X_s	Total activity from the biomass attached onto the AC ($g\ ATP(g\ AC)^{-1}$)
Y_i	Yield of biomass on component i ($gATP(gC)^{-1}$)
b	Death rate of the biomass (s^{-1})
$D_{s,i}$	Surface diffusion coefficient of component i (m^2s^{-1})
$K_{f,i}$	Freundlich constant of component i ($g\ C \cdot (g\ AC)^{-1} \cdot (g\ C\ m^{-3})^{-n}$)
$K_{s,i}$	Monod half velocity constant of component i ($g\ C\ m^{-3}$)
n_i	Freundlich exponent of component i (-)
z_i	The mol fraction of component i adsorbed (-)
Θ	Spreading pressure ($mol \cdot (g\ AC^{-1})$)
$c_{p,i,M}$	Molar liquid phase concentration of component i in the pore ($mol \cdot m^{-3}$)
MW_i	Molar weight of the C-atoms of component i ($gmol^{-1}$)
$q_{i,M}$	Molar solid phase concentration of component i ($mol\ (g\ AC^{-1})$)
$q_{t,M}$	Total molar solid phase concentration of all components ($mol\ (g\ AC^{-1})$)