Simulation of Wastewater Treatment Plants Modeled by a System of Nonlinear Ordinary and Partial Differential Equations

Gustav Mauritsson

Master's thesis
2013:E62

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Centrum Scientiarum Mathematicarum

Simulation of Wastewater Treatment Plants Modeled by a System of Nonlinear Ordinary and Partial Differential Equations

Gustav Mauritsson
Faculty of Engineering, Lund University
Advisor: Stefan Diehl
Co-advisor: Sebastian Farås

November 14, 2013

Abstract

Wastewater treatment consists of mechanical, chemical and biological purification. This master thesis concerns the biological part of the wastewater treatment called the *activated sludge process* (ASP). Two different mathematical models, one simplified and one complete, of the ASP are investigated. The models contain systems of nonlinear partial and ordinary differential equations. The nonlinearities in the equations give rise to discontinuous solutions, known as *shock waves*, which complicate the numerical analysis of the equations. The aim of the thesis is to implement the models in MATLAB and investigate how to solve these equations most efficiently with respect to accuracy and speed. Several time discretization schemes including built-in routines in MATLAB will be compared. The results show that a certain semi-implicit method seems to be the most efficient way to solve these equations numerically. Higher order fixed time step methods such as Runge-Kutta methods of order 2 and 4 are not suitable and perform even worse than the very simple Euler method of order 1.

Populärvetenskaplig sammanfattning

I ett vattenreningsverk nyttjas både mekanisk, kemisk och biologisk rening. Detta examensarbete berör matematiska modeller som ställts upp för att kunna beskriva den biologiska reningsprocessen. Den biologiska reningen i ett vattenreningsverk nyttjar gravitationen för att separera partiklar från inkommande avloppsvatten. I vattnet som kommer in till reningsverket finns lösta näringsämnen, *substrat*, såsom kväve och syre, samt partikulärt material, *biomassa*, bestående av bakterier och mikroorganismer som konsumerar och bryter ner substraten. Den biologiska reningen, kallat den *aktiva slamprocessen*, av ett vattenreningsverk består typiskt av ett antal *reaktortankar* samt en *sedimenteringstank*. Avloppsvatten flödar in i reaktortank nummer 1 och sedan vidare in till reaktortank nummer 2 och så vidare. I reaktortankarna förbrukas substraten av biomassan, som i sin tur nybildas och dör. Slutligen flödar vattnet in i sedimenteringtanken. Denna tank består av ett inlopp och två utlopp. Till inloppet kommer vatten från den sista i serien av reaktortankar. I sedimenteringstanken får den partikulära biomassan sjunka under gravitationens inverkan. I övre delen av av tanken uppstår då en zon med renat vatten där ett av utloppen sitter. I botten av tanken bildas en zon med slam och här sitter det andra utloppet. Det mesta av detta slam återförs till reaktortank nummer 1 medan en liten del bortförs från verket för att exempelvis användas i gödningsmedel.

För att kunna beskriva den biologiska reningsprocessen krävs rigorösa matematiska modeller som tar hänsyn till fysikaliska och biologiska effekter. Modelleringen ger upphov till ett olinjärt system av ordinära och partiella differentialekvationer. Dessa ekvationer kan inte lösas exakt, annat än i vissa specifika fall. Istället måste de behandlas med numeriska metoder, som med hjälp av beräkningsprogram ger approximativa lösningar. Lösningarna till ekvationerna innehåller ofta diskontinuiteter, eller *chockvågor*, där lösningen abrupt byter värde. Denna egenskap försvårar den numeriska behandlingen.

Detta examensarbete syftar till att undersöka två matematiska modeller för den biologiska reningen. Den ena modellen är en förenklad modell som endast innehåller en typ av substrat, en typ av biomassa och en reaktortank. En fullständig modell innehållandes sju substrat, sex biomassor och fem reaktortankar i serie kommer också att undersökas. Modellerna implementeras i datorprogrammet MATLAB som är ett vanligt förekommande program vid matematiska och tekniska beräkningar. Den numeriska metoden skall önskvärt reproducera den exakta lösningen med ett så litet fel som möjligt samtidigt som den kräver lite beräkningstid. Därför kommer mycket av undersökningarna i arbetet mäta hur effektiva olika lösningsmetoder är med avseende på hastighet och exakthet. Resultaten kan vara av intresse för vattenreningsbranschen som vägledning för vilka lösningsmetoder som ger effektivast simuleringar av den biologiska reningen.

Contents

1 Introduction

Wastewater treatment is a process of purifying wastewater from different contaminants. The wastewater treatment contains several different processes, namely physical (mechanical), chemical and biological, in order to remove respective contaminants. The aim of this master thesis is to simulate different models of the biological treatment process, the activated sludge process (ASP). This process uses a series of biological reactor tanks where incoming sewage, the *substrate*, mainly consisting of organic material and nutrients, are consumed and decomposed by microorganisms, the *biomass*, (Diehl, 2012). The substrate is soluble and thus dissolved in the water while the biomass is particulate material. Both the substrate and the biomass contain several different components. Apart from the series of biological reactor tanks, the ASP also consists of a sedimentation tank, also known as the *settler*. The settler contains one inlet and two outlets. Water consisting of activated sludge, both biomass and substrate, flow from the biological reactor to the inlet of the settler. In the settler the flocculated biomass settles slowly under the force of gravity. The top end of the settler will normally be free from flocculated biomass and here one of the outlets are placed, from which purified water flows. In the bottom end of the settler the second outlet is placed. The sludge flowing out from here are partly recycled, i.e. taken back to the biological reactor and partly removed from the plant, for instance being used as fertilizer. The greater part is recycled since one wants to keep the waste sludge as small as possible for economical and environmental reasons (Diehl, 2012).

This master thesis will firstly consider a simplified model of a wastewater treatment plant, consisting of only one reactor tank, one substrate component and one biomass component. It will also consider an extended, more realistic model, with five reactor tanks, seven substrate components and six biomass components. The aim is to implement the two models using MATLAB and investigate the efficiency, with respect to speed and accuracy, of several different time discretization schemes for the nonlinear system of differential equations involved in the mathematical models.

The mathematical models of the ASP are derived from the conservation law which will be discussed in Section 2. This section also includes relevant mathematical background and concepts necessary for modeling the process and interpret the solutions. Section 3 contains the modeling equations as well constitutive relations modeling physical effects that are present in the process. To be able to solve the system of nonlinear partial and ordinary differential equations (PDE:s and ODE:s) the mathematical model has to be discretized in both space and time. Section 4 will go through this procedure in a rather detailed way. In Section 5 simulations of some different scenarios and results of the efficiency tests are presented. In Section 6 there will be a brief discussion of the results and a concluding summary.

2 Mathematical background

2.1 The conservation law

It is very common that physical phenomena obey the conservation law. The change of the total amount of some physical entity in a certain region of space equals the flux into the region through the boundary minus the flux out through the boundary. If the region contains sinks and sources these also contribute to the change of the amount. Conservation laws are used to model a great variety of physical phenomena such as gas and fluid dynamics, traffic flows and sedimentation of solid particles in a liquid, as is the case in the wastewater treatment plant (Diehl, 1996).

To exemplify this one may consider one-dimensional traffic flow. Let the x-axis be situated along the road and let $u(x,t)$ denote the density of cars [number of cars/m] at the point x at the time point t. The number of cars per unit time passing a point x at time t is the flow rate of cars, denoted by f. If (x_1, x_2) is an arbitrary interval of the x-axis, the conservation law may be written in integral form as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2} u(x,t)\,\mathrm{d}x = f|_{x=x_1} - f|_{x=x_2}, \tag{2.1}$$

where the right hand side can be written as

$$f|_{x=x_1} - f|_{x=x_2} = -\int_{x_1}^{x_2} \frac{\partial f}{\partial x}\,\mathrm{d}x.$$

Assuming that the concentration $u(x,t) \in C^1$ allows for the derivative to be put inside the integral in the left hand side of (2.1) and then one gets

$$\int_{x_1}^{x_2} \left(\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x}\right)\mathrm{d}x = 0.$$

Since the interval is chosen arbitrarily and since the integrand is assumed to be continuous it follows that

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \iff u_t + f_x = 0. \tag{2.2}$$

The partial differential equation (2.2) is called the *continuity equation* and is the differential form of the conservation law.

A common assumption on the flux function f is that it depends on the unknown only:

$$f = f\big(u(x,t)\big), \tag{2.3}$$

where $f(u)$ is assumed to be a smooth function. Using this allows one to rewrite equation (2.2) as

$$u_t + f(u)_x = 0 \iff u_t + f'(u)u_x = 0. \tag{2.4}$$

In the case of traffic flow one may assume that the car speed v depends only on the local concentration of cars at a given point x at time t. Thus one can model the speed as $v = v(u) = v_0(1 - \frac{u}{u_{\max}})$, where v_0 is the free speed of a car, that is the limit speed of the road, and $u_{\max}$ is the maximum concentration of cars. With this choice of the speed function, the total flux function becomes

$$f(u) = v(u)u = v_0\left(1 - \frac{u}{u_{\max}}\right)u, \quad 0 \le u \le u_{\max}. \tag{2.5}$$

In order to solve equation (2.4) one must give some initial concentration distibution at the time $t = 0$. Then one gets the initial value problem

$$\begin{cases} u_t + f(u)_x = 0, & x \in \mathbb{R}, \ \ t > 0 \\ u(x,0) = u_0(x), & x \in \mathbb{R}. \end{cases} \tag{2.6}$$

For some further details on this topic, the reader is referred to Diehl (1996).

2.2 Characteristics

Consider a level curve $x = x(t)$ in the $x - t$ plane, that is

$$u(x(t), t) = \text{constant} = U_0.$$

Differentiating with respect to t yields

$$u_x x'(t) + u_t = 0,$$

and using $u_t = -f'(u)u_x$ with $u(x(t), t) = U_0$ then gives

$$u_x\big(x'(t) - f'(U_0)\big) = 0.$$

In general this implies that $x'(t) = f'(U_0)$ must hold which means that the level curve is a straight line in the $x - t$ plane with the slope $\frac{1}{f'(U_0)}$. $f'(U_0)$ is called the *signal speed* since it is the propagation speed of a wavefront or disturbance. The straight lines that are the level curves are called *characteristics*. Given some initial data one may now construct a solution in implicit form to the initial value problem (2.6):

$$\begin{cases} x = f'\big(u_0(x_0)\big)t + x_0 \\ u = u_0(x_0). \end{cases} \tag{2.7}$$

In Diehl (1996) some examples of this procedure is provided.

2.3 Shock waves and entropy condition

If f is a nonlinear function of u it may happen that the characteristic lines intersect. In this case the procedure described above breaks down and it is not possible to define a continuous solution after a time at which characteristics intersect. Even when $u_0(x) \in C^1$, discontinuous solutions may appear after a finite time. In order to obtain a solution $u(x,t) \in C^1$, the first part of (2.7) needs to be solved for $x_0 = x_0(x,t)$ and then substituted into the second part of equation (2.7). The implicit function theorem states that it is possible if

$$\frac{\mathrm{d}x}{\mathrm{d}x_0} = f''\big(u(x_0)\big)u_0'(x_0)t + 1 \neq 0. \tag{2.8}$$

This holds for small $t > 0$ and thus if $u_0(x)$ is smooth then there is a smooth solution $u(x,t)$ for small $t > 0$. The smallest time for which $\frac{\mathrm{d}x}{\mathrm{d}x_0} = 0$ is called the *critical time* and this is the point where the discontinuity appears.

To handle the discontinuity one reformulates the conservation law given by (2.4). Let $\varphi = \varphi(x,t) \in C_0^1$ be a test function with compact support, multiply this by (2.4) and integrate over the entire $x-$axis:

$$\begin{aligned}
&\int_{-\infty}^{\infty} u_t\varphi \,\mathrm{d}x + \int_{-\infty}^{\infty} f(u)_x\varphi \,\mathrm{d}x = 0,\\
&\int_{-\infty}^{\infty} u_t\varphi \,\mathrm{d}x + \underbrace{\big[f(u)\varphi\big]_{-\infty}^{\infty}}_{=0} - \int_{-\infty}^{\infty} f(u)\varphi_x \,\mathrm{d}x = 0.
\end{aligned}$$

Now one integrates over the $t-$axis:

$$\begin{aligned}
&\int_0^{\infty}\int_{-\infty}^{\infty} u_t\varphi \,\mathrm{d}x\,\mathrm{d}t - \int_0^{\infty}\int_{-\infty}^{\infty} f(u)\varphi_x \,\mathrm{d}x\,\mathrm{d}t = 0,\\
&\int_{-\infty}^{\infty} [u\varphi]_{t=0}^{\infty} \,\mathrm{d}x - \int_0^{\infty}\int_{-\infty}^{\infty} u\varphi_t \,\mathrm{d}x\,\mathrm{d}t - \int_0^{\infty}\int_{-\infty}^{\infty} f(u)\varphi_x \,\mathrm{d}x\,\mathrm{d}t = 0,
\end{aligned}$$

and one finally gets

$$\int_0^{\infty}\int_{-\infty}^{\infty} (u\varphi_t + f(u)\varphi_x) \,\mathrm{d}x\,\mathrm{d}t + \int_{-\infty}^{\infty} u(x,0)\varphi(x,0) \,\mathrm{d}x = 0, \quad \forall\ \varphi \in C_0^1. \tag{2.9}$$

Any function u that satisfies (2.9) is called a *weak* solution of the conservation law (2.4).

The conservation law also contains information about the movement of the discontinuity. If u is a piecewise C^1 solution, $x = x(t) \in C^1$ is a curve in the $x - t$ plane,

along which u is discontinuous, (a, b) is an interval parallel with the x-axis, such that the curve $x(t)$ intersects the interval at a time t and if $u^{\pm} = u(x(t) \pm 0, t)$ are the values of the solution to the left and right of the discontinuity curve, then (2.1) gives for the interval (a, b) :

$$\begin{aligned} f\big(u(a,t)\big) - f\big(u(b,t)\big) &= \frac{\mathrm{d}}{\mathrm{d}t}\int_a^b u\,\mathrm{d}x = \frac{\mathrm{d}}{\mathrm{d}t}\Big(\int_a^{x(t)} u\,\mathrm{d}x + \int_{x(t)}^b u\,\mathrm{d}x\Big) = \\ &= \int_a^{x(t)} u_t\,\mathrm{d}x + u^- x'(t) + \int_{x(t)}^b u_t\,\mathrm{d}x - u^+ x'(t) = \\ &= [u_t = -f_x] = f\big(u(a,t)\big) - f\big(u(b,t)\big) + \\ &+ f(u^+) - f(u^-) - (u^+ - u^-)x'(t) \end{aligned}$$

and thus the speed of the discontinuity satisfies

$$x'(t) = \frac{f(u^+) - f(u^-)}{u^+ - u^-}. \tag{2.10}$$

This equation is called the *jump condition* or the *Rankine-Hugoniot condition*. If $u(x, t)$ is a piecewise smooth function satisfying $u(x, 0) = u_0(x)$, then $u(x, t)$ is a weak solution of (2.9) if and only if the conservation law is satisfied at every point where $u \in C^1$ and (2.10) is satisfied at discontinuities (Diehl, 1996).

There is a problem involved with the weak solutions. For some given initial data one may obtain several solutions that fulfills both (2.9) and (2.10) and thus are valid weak solutions, see Diehl (1996) for details on this. In order to select a unique solution that is physically relevant, one must impose some extra condition, the *entropy condition*. Instead of (2.4) consider the viscous equation

$$u_t + f(u)_x = \varepsilon u_{xx}, \quad \varepsilon > 0. \tag{2.11}$$

By letting ε be small one obtains approximately the same solutions as for (2.4) but the shocks will be somewhat smoothed. Consider now a solution u to (2.4) which consists of one single shock moving with speed $x'(t) =: s$ and left and right limits u^- and u^+. A shock is allowed only if it satisfies the *viscous profile condition* which states that for given constants u^-, u^+ and

$$s = \frac{f(u^+) - f(u^-)}{u^+ - u^-},$$

there exists a traveling wave solution, or viscous profile

$$u(x, t) = v(\xi), \quad \text{with} \quad \xi = \frac{x - st}{\varepsilon}, \tag{2.12}$$

of equation (2.11) with $v(\xi) \to u^{\pm}$ as $\xi \to \pm\infty$. Thus if $\varepsilon \to 0^+$ the traveling wave converges to the expected shock with speed s. Now combining (2.11) and (2.12) yields $v'' = f(v)_\xi - sv'$ and after an integration one gets

$$v' = f(v) - sv + C, \tag{2.13}$$

for some constant C. Now using (2.10) and letting $\xi \to \pm\infty$ yields that $C = -f(u^-) + su^- = -f(u^+) + su^+$ and thus $v(\xi)$ satisfies the ordinary differential equation

$$v' = f(v) - f(u^-) - s(v - u^-) = f(v) - f(u^+) - s(v - u^+), \tag{2.14}$$

where one defines

$$f(v) - f(u^-) - s(v - u^-) =: \psi(v). \tag{2.15}$$

Assume now that $u^- > u^+$. If $v'(\xi_0) = \psi\big(v(\xi_0)\big) = 0$ for some ξ_0, then $v(\xi) = v(\xi_0)$ is the unique solution of (2.14). Since $v(\xi) \to u^{\pm}$ as $\xi \to \pm\infty$, v is either strictly increasing or strictly decreasing. The only possibility is thus that $v'(\xi) = \psi\big(v(\xi)\big) < 0$, $\forall\ \xi \in \mathbb{R}$. Hence $\psi(v) < 0$ for $v \in (u^+, u^-)$ or by using (2.15)

$$s < \frac{f(v) - f(u^-)}{v - u^-}, \quad \text{for all } v \text{ strictly between } u^- \text{ and } u^+. \tag{2.16}$$

Equation (2.16) also holds if $u^- < u^+$ and is a necessary condition for an admissible, that is a physically relevant, shock. The condition

$$s \leq \frac{f(v) - f(u^-)}{v - u^-}, \quad \text{for all } v \text{ between } u^- \text{ and } u^+ \tag{2.17}$$

is a sufficient condition for uniqueness of a weak solution to the problem (2.6). Now by letting $v \to u^-$ or $v \to u^+$ one gets from the definition of the derivative that $f'(u^-) \geq s$ and analogously one can show that $f'(u^+) \leq s$, thus the entropy condition implies

$$f'(u^-) \geq x'(t) \geq f'(u^+). \tag{2.18}$$

For some further details on the derivation of the entropy condition the reader is referred to Diehl (1996).

3 Modeling of the process

3.1 Simplified model

The activated sludge process, Figure 1, consists of two different tanks, the biological reactor tank and the sedimentation tank. Wastewater is flowing into the biological reactor with the volumetric flow rate Q [m^3/s]. The wastewater contains only one type of soluble organic material and nutrients (substrate) which is consumed and decomposed by only one type of biomass. The concentration of the substrate is denoted by S [kg/m^3] and the concentration of the biomass is denoted by X [kg/m^3]. The control parameters r and w govern the amount of sludge going back into the reactor tank and the amount of waste sludge.

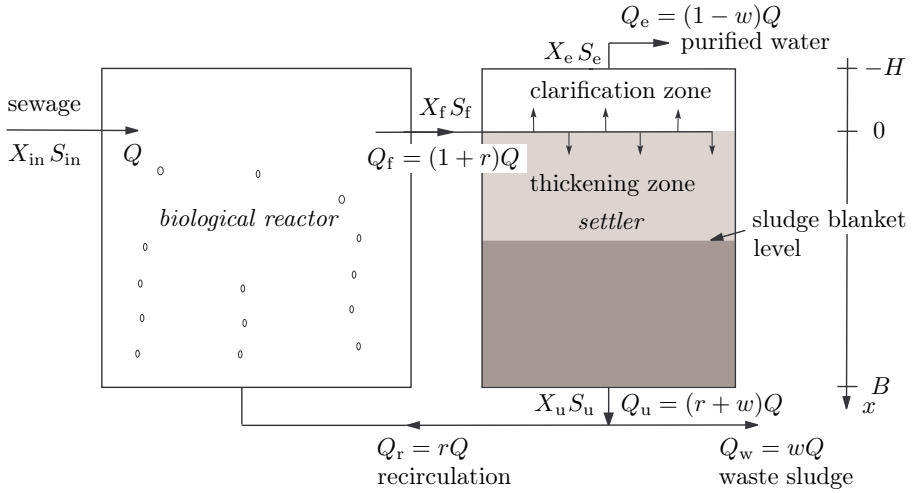

Figure 1: The activated sludge process consists of two tanks, a biological reactor and a sedimentation tank. The indices stand for f = feed, e = effluent, u = underflow, r = recycle, w = waste and in = influent.

The model equations provided by Diehl (2012) are

$$V\frac{\mathrm{d}S_\mathrm{f}}{\mathrm{d}t} = QS_\mathrm{in} + rQS_\mathrm{u} - (1+r)QS_\mathrm{f} - V\frac{\mu(S_\mathrm{f})}{Y}X_\mathrm{f}, \tag{3.1}$$

$$V\frac{\mathrm{d}X_\mathrm{f}}{\mathrm{d}t} = QX_\mathrm{in} + rQX_\mathrm{u} - (1+r)QX_\mathrm{f} + V(\mu(S_\mathrm{f}) - b)X_\mathrm{f}, \tag{3.2}$$

$$A\frac{\partial S}{\partial t} + A\frac{\partial}{\partial x}\big(F^\mathrm{s}(S,x,r,w,Q,t)\big) = (1+r)QS_\mathrm{f}\delta(x), \tag{3.3}$$

$$\begin{aligned} A\frac{\partial X}{\partial t} + A\frac{\partial}{\partial x}\big(F(X,x,r,w,Q,t)\big) &= \\ = A\frac{\partial}{\partial x}\left(\Big(\gamma(x)d_\mathrm{comp}(X) + d_\mathrm{disp}\big(x,Q_\mathrm{f}(t)\big)\Big)\frac{\partial X}{\partial x}\right) &+ (1+r)QX_\mathrm{f}\delta(x). \end{aligned} \tag{3.4}$$

Here V [m^3] denotes the volume of the biological reactor, A is the cross-sectional area of the settler, Y is a positive, dimensionless constant which relates the usage of substrate to growth of biomass, b [s^{-1}] is the death rate of the biomass and μ [s^{-1}] is a function modeling the growth rate of the biomass. The functions $F^{\mathrm{s}}(S, x, r, w, Q, t)$ and $F(X, x, r, w, Q, t)$ are the convective fluxes of S and X respectively. They are given by

$$F^{\mathrm{s}}(S, x, r, w, Q, t) := \begin{cases} -\frac{(1-w)Q(t)}{A} S, & \text{for } x < 0, \\ \frac{(r+w)Q(t)}{A} S, & \text{for } x > 0, \end{cases} \tag{3.5}$$

$$F(X, x, r, w, Q.t) := \begin{cases} -\frac{(1-w)Q(t)}{A} X, & \text{for } x < -H, \\ f_{\mathrm{bk}}(X) - \frac{(1-w)Q(t)}{A} X, & \text{for } -H < x < 0, \\ f_{\mathrm{bk}}(X) + \frac{(r+w)Q(t)}{A} X, & \text{for } 0 < x < B, \\ \frac{(r+w)Q(t)}{A} X, & \text{for } x > B, \end{cases} \tag{3.6}$$

where $f_{\mathrm{bk}}(X)$ is the Kynch batch flux density function

$$f_{\mathrm{bk}}(X) := X v_{\mathrm{hs}}(X), \tag{3.7}$$

where $v_{\mathrm{hs}}(X)$ [m/s] is the hindered settling velocity, see Section 3.2. The discontinuous function

$$\gamma(x) := \begin{cases} 1 & \text{for } -H \leq x \leq B, \\ 0 & \text{for } x < -H \text{ or } x > B, \end{cases} \tag{3.8}$$

makes sure that compression only occur within the settler tank. The functions $d_{\mathrm{comp}}(X)$ and $d_{\mathrm{disp}}\big(x, Q_{\mathrm{f}}(t)\big)$ are the compression and diffusion functions, see Section 3.2. The substrate is completely dissolved in the water and is thus not subject to any compression and diffusion. $\delta(x)$ is the Dirac delta function.

3.2 Constitutive relations

When modeling the process of wastewater treatment assumptions on the constitutive relations, that is the physical relations between quantities, have to be made. The modeling or calibration of the constitutive relations involves the solving of an inverse problem and is a subject of its own. The inverse problem is solved by finding optimal values of the parameters in the constitutive relations, such that the solution to the system of equations fit as well as possible with measured data. The inverse problem for the activated sludge process seems to be ill conditioned, implying difficulties in finding unique optimal parameter values. There is thus plenty of problems when modeling the constitutive relations and they go beyond the frame of this thesis.

The growth rate μ of the biomass can be modeled using the Monod relation, then

$$\mu(S) = \hat{\mu} \frac{S}{K + S}, \tag{3.9}$$

where $\hat{\mu}$ [s^{-1}] is the maximal growth rate and K [kg/m^3] is the half saturation constant.

For the hindered settling velocity $v_{hs}(X)$ several suggestions has been made. Here the expression suggested by Vesilind (1968),

$$v_{hs}(X) = v_0 e^{-r_V X}, \tag{3.10}$$

will be used. v_0 [m/s] is the settling velocity of a single particle and $r_V > 0$ [m^3/kg] is a parameter. Combining this expression with the expression for the Kynch batch flux density function (3.7) one gets

$$f_{bk}(X) = v_0 X e^{-r_V X}. \tag{3.11}$$

The expression for the compression function $d_{comp}(X)$ is given by

$$d_{comp}(X) = \frac{\rho_s}{(\rho_s - \rho_f)g} v_{hs}(X) \sigma_e'(X), \tag{3.12}$$

where ρ_f and ρ_s [kg/m^3] are the fluid mass density and the solid mass density with $\rho_s > \rho_f$, g [m/s^2] is the acceleration of gravity and $\sigma_e(X)$ [N/m^2] is the effective solid stress function. For the effective solid stress function several suggestions has been made as well, a particular simple one is

$$\sigma_e(X) = \begin{cases} 0 & \text{for } X < X_c, \\ a(X - X_c) & \text{for } X \geq X_c, \end{cases} \tag{3.13}$$

where a [m^2/s^2] is a positive parameter and X_c is the critical concentration for which compression effects start to take place. This choice implies that

$$\sigma_e'(X) = \begin{cases} 0 & \text{for } X < X_c, \\ a & \text{for } X \geq X_c, \end{cases} \tag{3.14}$$

and thus

$$d_{comp}(X) = \begin{cases} 0 & \text{for } X < X_c, \\ \frac{\rho_s a v_0 e^{-r_V X}}{(\rho_s - \rho_f)g} & \text{for } X \geq X_c, \end{cases} \tag{3.15}$$

The dispersion function $d_{disp}\big(x, Q_f(t)\big)$ models turbulence and mixing phenomena that occurs near the feed inlet. Thus this function should be zero at a certain distance away from the feed inlet. One can set

$$d_{disp}\big(x, Q_f(t)\big) = \frac{Q_f(t)}{A} L\big(x, Q_f(t)\big), \tag{3.16}$$

where L is some continuous function that vanishes at some certain distance away from the feed inlet. Since dispersion only takes place inside the tank, the dispersion function must satisfy

$$d_{disp}\big(x, Q_f(t)\big) \begin{cases} = 0 & \text{for } x \leq -H \text{ and } x \geq B, \\ \geq 0 & \text{for } -H < x < B. \end{cases} \tag{3.17}$$

Now several different choices of L can be made as long as (3.17) is fulfilled. In this paper L is chosen such that

$$d_{\text{disp}}\big(x, Q_{\text{f}}(t)\big) = \begin{cases} \alpha_1 Q_{\text{f}} \exp\left(\frac{-x^2/(\alpha_2 Q_{\text{f}})^2}{1-|x|/(\alpha_2 Q_{\text{f}})}\right) & \text{for } |x| < \alpha_2 Q_{\text{f}}, \\ 0 & \text{for } |x| \geq \alpha_2 Q_{\text{f}}. \end{cases} \tag{3.18}$$

α_1 [m^{-1}] and α_2 [s/m^2] are parameters where (3.17) requires that

$$\alpha_2 < \frac{\min(H, B)}{\max\limits_{t \geq 0} Q_{\text{f}}(t)}.$$

3.3 Extended model

The extended model of the process contains five different biological reactor tanks, seven different soluble substrates and six types of organic biomass. The model used here is the Activated Sludge Model no. 1 (ASM1) described by Copp (2002). Figure 2 shows an overview of the wastewater treatment plant. Table 1 contains all the different state variables that are included in the extended model.

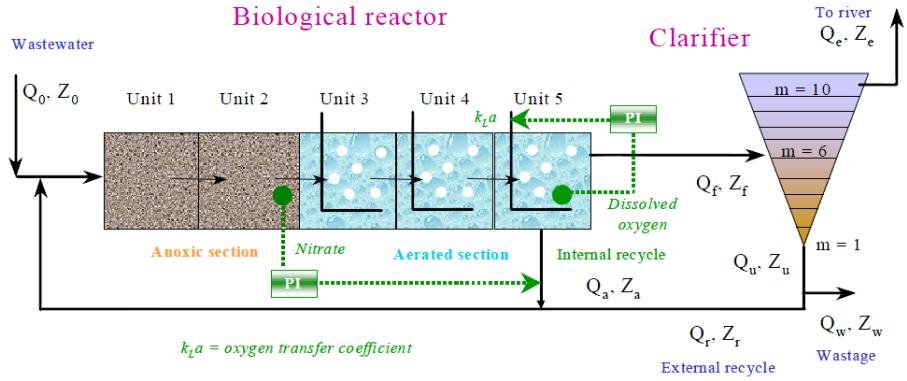

Figure 2: Overview of the wastewater treatment in the extended model. The figure is taken from Alex et al. (2008).

Let Q_k, Z_k and r_k denote the flow rate, concentration and conversion rate of the state variable Z in unit k with the volume V_k. Then the general mass balance in the reactor tanks is described by the following ODE:s:

For k=1:

$$\frac{\text{d}Z_1}{\text{d}t} = \frac{1}{V_1}(Q_{\text{a}} Z_{\text{a}} + Q_{\text{r}} Z_{\text{r}} + Q_0 Z_0 + r_1 V_1 - Q_1 Z_1), \tag{3.19}$$

$$Q_1 = Q_{\text{a}} + Q_{\text{r}} + Q_0,$$

Definition	Notation
Soluble inert organic matter	S_{I}
Readily biodegradable substrate	S_{S}
Particulate inert organic matter	X_{I}
Slowly biodegradable substrate	X_{S}
Active heterotrophic biomass	$X_{\mathrm{B,H}}$
Active autotrophic biomass	$X_{\mathrm{B,A}}$
Particulate products arising from biomass decay	X_{P}
Oxygen	S_{O}
Nitrate and nitrite nitrogen	S_{NO}
$NH4^+$ + NH_3 nitrogen	S_{NH}
Soluble biodegradable organic nitrogen	S_{ND}
Particulate biodegradable	X_{ND}
Alkalinity	S_{ALK}

Table 1: List of state variables with definitions and notations.

For k=2, 3, 4, 5:

$$\frac{\mathrm{d}Z_k}{\mathrm{d}t} = \frac{1}{V_k}(Q_{k-1}Z_{k-1} + r_k V_k - Q_k Z_k), \tag{3.20}$$
$$Q_k = Q_{k-1},$$

with the oxygen being a special case

$$\frac{\mathrm{d}S_{\mathrm{O},k}}{\mathrm{d}t} = \frac{1}{V_k}(Q_{k-1}S_{\mathrm{O},k-1} + r_k V_k + (K_{\mathrm{L}}a)_k V_k(S_{\mathrm{O}}^* - S_{\mathrm{O},k}) - Q_k S_{\mathrm{O},k}), \tag{3.21}$$

where S_{O}^* [kg/m^3] is a saturation concentration for oxygen and $K_L a$ [s^{-1}] is an oxygen transfer coefficient which can be manipulated to control the concentration of the dissolved oxygen. The parameter Q_a is the internal recycle flow rate from unit 5 to unit 1 and is a control parameter.

The conversion rates, following the order in Table 1, for the respective state variables are:

$$r_1 = 0, \tag{3.22}$$
$$r_2 = -\frac{1}{Y_{\mathrm{H}}}\rho_1 - \frac{1}{Y_{\mathrm{H}}}\rho_2 + \rho_7, \tag{3.23}$$
$$r_3 = 0, \tag{3.24}$$
$$r_4 = (1 - f_{\mathrm{P}})\rho_4 + (1 - f_{\mathrm{P}})\rho_5 - \rho_7, \tag{3.25}$$
$$r_5 = \rho_1 + \rho_2 - \rho_4, \tag{3.26}$$
$$r_6 = \rho_3 - \rho_5, \tag{3.27}$$
$$r_7 = f_{\mathrm{P}}\rho_4 + f_{\mathrm{P}}\rho_5, \tag{3.28}$$
$$r_8 = -\frac{1 - Y_{\mathrm{H}}}{Y_{\mathrm{H}}}\rho_1 - \frac{4.57 - Y_{\mathrm{A}}}{Y_{\mathrm{A}}}\rho_3, \tag{3.29}$$
$$r_9 = -\frac{1 - Y_{\mathrm{H}}}{2.86Y_{\mathrm{H}}}\rho_2 + \frac{1}{Y_{\mathrm{A}}}\rho_3, \tag{3.30}$$
$$r_{10} = -i_{\mathrm{XB}}\rho_1 - i_{\mathrm{XB}}\rho_2 - \left(i_{\mathrm{XB}} + \frac{1}{Y_{\mathrm{A}}}\right)\rho_3 + \rho_6, \tag{3.31}$$
$$r_{11} = -\rho_6 + \rho_8, \tag{3.32}$$
$$r_{12} = (i_{\mathrm{XB}} - f_{\mathrm{P}}i_{\mathrm{XP}})\rho_4 + (i_{\mathrm{XB}} - f_{\mathrm{P}}i_{\mathrm{XP}})\rho_5 - \rho_8, \tag{3.33}$$
$$r_{13} = -\frac{i_{\mathrm{XB}}}{14}\rho_1 + \left(\frac{1 - Y_{\mathrm{H}}}{14 \cdot 2.86Y_{\mathrm{H}}} - \frac{i_{\mathrm{XB}}}{14}\right)\rho_2 - \left(\frac{i_{\mathrm{XB}}}{14} + \frac{1}{7Y_{\mathrm{A}}}\right)\rho_3 + \frac{1}{14}\rho_6. \tag{3.34}$$

The biological behavior of the system is described by eight basic processes,

$$\rho_1 = \mu_{\mathrm{H}}\left(\frac{S_{\mathrm{S}}}{K_{\mathrm{S}} + S_{\mathrm{S}}}\right)\left(\frac{S_{\mathrm{O}}}{K_{\mathrm{O,H}} + S_{\mathrm{O}}}\right)X_{\mathrm{B,H}}, \tag{3.35}$$
$$\rho_2 = \mu_{\mathrm{H}}\left(\frac{S_{\mathrm{S}}}{K_{\mathrm{S}} + S_{\mathrm{S}}}\right)\left(\frac{K_{\mathrm{O,H}}}{K_{\mathrm{O,H}} + S_{\mathrm{O}}}\right)\left(\frac{S_{\mathrm{NO}}}{K_{\mathrm{NO}} + S_{\mathrm{NO}}}\right)\eta_{\mathrm{g}}X_{\mathrm{B,H}}, \tag{3.36}$$
$$\rho_3 = \mu_{\mathrm{A}}\left(\frac{S_{\mathrm{NH}}}{K_{NH} + S_{\mathrm{NH}}}\right)\left(\frac{S_{\mathrm{O}}}{K_{O,A} + S_{\mathrm{O}}}\right)X_{\mathrm{B,A}}, \tag{3.37}$$
$$\rho_4 = b_{\mathrm{H}}X_{\mathrm{B,H}}, \tag{3.38}$$
$$\rho_5 = b_{\mathrm{A}}X_{\mathrm{B,A}}, \tag{3.39}$$
$$\rho_6 = k_{\mathrm{A}}S_{\mathrm{ND}}X_{\mathrm{B,H}}, \tag{3.40}$$
$$\rho_7 = k_{\mathrm{h}}\frac{X_{\mathrm{S}}/X_{\mathrm{B,H}}}{K_X + X_{\mathrm{S}}/X_{\mathrm{B,H}}}\left[\left(\frac{S_{\mathrm{O}}}{K_{\mathrm{O,H}} + S_{\mathrm{O}}}\right) + \eta_{\mathrm{h}}\left(\frac{K_{\mathrm{O,H}}}{K_{\mathrm{O,H}} + S_{\mathrm{O}}}\right)\left(\frac{S_{\mathrm{NO}}}{K_{\mathrm{NO}} + S_{\mathrm{NO}}}\right)\right]X_{\mathrm{B,H}}, \tag{3.41}$$
$$\rho_8 = k_{\mathrm{h}}\frac{X_{\mathrm{S}}/X_{\mathrm{B,H}}}{K_X + X_{\mathrm{S}}/X_{\mathrm{B,H}}}\left[\left(\frac{S_{\mathrm{O}}}{K_{\mathrm{O,H}} + S_{\mathrm{O}}}\right) + \eta_{\mathrm{h}}\left(\frac{K_{\mathrm{O,H}}}{K_{\mathrm{O,H}} + S_{\mathrm{O}}}\right)\left(\frac{S_{\mathrm{NO}}}{K_{\mathrm{NO}} + S_{\mathrm{NO}}}\right)\right]X_{\mathrm{B,H}}(X_{\mathrm{ND}}/X_{\mathrm{S}}). \tag{3.42}$$

Parameter	Unit
Y_{A}	dimensionless
Y_{H}	dimensionless
f_{P}	dimensionless
i_{XB}	dimensionless
i_{XP}	dimensionless
μ_{H}	s^{-1}
K_{S}	$\mathrm{kg/m}^3$
$K_{\mathrm{O,H}}$	$\mathrm{kg/m}^3$
K_{NO}	$\mathrm{kg/m}^3$
b_{H}	s^{-1}
η_{g}	dimensionless
η_{h}	dimensionless
k_{h}	s^{-1}
K_{X}	dimensionless
μ_{A}	s^{-1}
K_{NH}	$\mathrm{kg/m}^3$
b_{A}	s^{-1}
$K_{\mathrm{O,A}}$	$\mathrm{kg/m}^3$
k_{A}	$\mathrm{m}^3/(\mathrm{kgs})$

Table 2: List of parameters in ASM1 with their SI-units.

Table 2 contains the units of the different parameters appearing in the equations for the basic processes and conversion rates. For explanations of the parameters the reader is referred to Copp (2002).

All different substrates are treated separately in the settler tank and thus seven different PDE:s have to be solved, one for each substrate. The six different biomass components however are modeled as being clustered together in the settler tank, thus only contributing with one more PDE. The different substrate components and the clustered biomass obey Equations (3.3) and (3.4) respectively. The concentration of biomass leaving tank 5 and flowing into the settler tank is:

$$\begin{aligned} X_{\mathrm{f}} &= \frac{1}{fr_{\mathrm{COD-SS}}}(X_{\mathrm{S},5} + X_{\mathrm{P},5} + X_{\mathrm{I},5} + X_{\mathrm{B,H},5} + X_{\mathrm{B,A},5}) = \\ &= [fr_{\mathrm{COD-SS}} = 4/3] = \frac{3}{4}(X_{\mathrm{S},5} + X_{\mathrm{P},5} + X_{\mathrm{I},5} + X_{\mathrm{B,H},5} + X_{\mathrm{B,A},5}). \end{aligned} \tag{3.43}$$

Similar equations hold when calculating the concentrations of biomass X_{e} leaving at the effluent level and X_{u} leaving at the underflow level. Furthermore for simplicity, the distribution of the different biomass components is assumed to remain constant across the settler:

$$\frac{X_{\mathrm{S},5}}{X_{\mathrm{f}}} = \frac{X_{\mathrm{S,u}}}{X_{\mathrm{u}}}. \tag{3.44}$$

This is a pretty crude assumption since it does not take into account the time it takes for the particles to sink in the settler. In a stationary situation the assumption is however correct. Similar equations hold for the other biomass components $X_{\mathrm{P,u}}, X_{\mathrm{I,u}}, X_{\mathrm{B,H,u}}, X_{\mathrm{B,A,u}}$ and $X_{\mathrm{ND,u}}$.

4 Discretization of the mathematical model

4.1 The conservation law on integrated form

Equations (3.3) and (3.4) are of the form

$$\frac{\partial Z}{\partial t} + \frac{\partial}{\partial x}F(Z,x,t) = \frac{\partial}{\partial x}\left(\Big(\gamma(x)d_{\text{comp}}(Z) + d_{\text{disp}}\big(x, Q_{\text{f}}(t)\big)\Big)\frac{\partial Z}{\partial x}\right) + \frac{Q_{\text{f}}(t)Z_{\text{f}}(t)}{A}\delta(x). \tag{4.1}$$

Here Z denotes the concentration of any substrate or biomass. In both models only one type of biomass occurs in the settler. Meanwhile there are not only one but seven types of substrates in the settler using the extended model compared to the simplified one. Of course the compression and dispersion terms vanish for all substrates since these are dissolved in the water. In order to solve equation (4.1) one needs to establish a reliable numerical method. Since $F(Z,x,t)$ is a discontinuous function of x and $d_{\text{comp}}(Z)$ vanishes for a range of concentration values it is not possible to solve (4.1) by standard methods. The solution $Z = Z(x,t)$ of (4.1) may be discontinuous and thus cannot be interpreted in the pointwise sense. Instead (4.1) needs to be reformulated on an integrated form which allows for a derivation of a nonlinear system of ODE:s (Bürger et al., 2012b). The integrated form of the equation does not involve the partial derivatives $\frac{\partial Z}{\partial t}$ and $\frac{\partial Z}{\partial x}$, which are not well defined for a discontinuous function Z. The total flux Φ is defined as

$$\Phi\left(Z, \frac{\partial Z}{\partial x}, x, t\right) = F(Z,x,t) - \Big(\gamma(x)d_{\text{comp}}(Z) + d_{\text{disp}}\big(z, Q_{\text{f}}(t)\big)\Big)\frac{\partial Z}{\partial x}, \tag{4.2}$$

and (4.1) may then be rewritten as

$$A\frac{\partial Z}{\partial t} = -A\frac{\partial \Phi}{\partial x} + Q_{\text{f}}(t)Z_{\text{f}}(t)\delta(x), \tag{4.3}$$

Now consider an arbitrary interval (x_1, x_2) on the x-axis and integrate (4.3) over x to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2} AZ(x,t)\,\mathrm{d}x = A(\Phi|_{x=x_1} - \Phi|_{x=x_2}) + \int_{x_1}^{x_2} Q_{\text{f}}(t)Z_{\text{f}}(t)\delta(x)\,\mathrm{d}x. \tag{4.4}$$

Equation (4.4) is in fact just the conservation law of mass stating that the rate of increase in (x_1, x_2) equals the flux in minus the flux out $(\Phi|_{x=x_1} - \Phi|_{x=x_2})$ plus the production inside the interval (the integral term).

In order to discretize the model, the x-axis is divided into N internal layers where each layer has the depth $\Delta x = (B+H)/N$. B is the depth of the thickening zone and H is the height of the clarification zone. The boundary points of the layers are then located at $x_j := j\Delta x - H, \quad j = 0, \ldots, N$. Layer j is defined as the interval

$[x_{j-1}, x_j]$. The feed layer, that is the layer which contains the feed inlet ($x = 0$) is located in $(x_{j_\mathrm{f}-1}, x_{j_\mathrm{f}}]$ where $j_\mathrm{f} := \lceil H/\Delta x \rceil$. In order to obtain a correct numerical implementation an additional four layers are added to the computational domain, two of these layers are located in the effluent zone and two layers are located in the underflow zone. In all, the computational domain thus consists of $N + 4$ layers of depth Δx with the boundaries located at the points $x_j,\ j = -2, -1, \ldots, N + 2$. Figure 3 shows an illustration of the subdivision of the settler into layers.

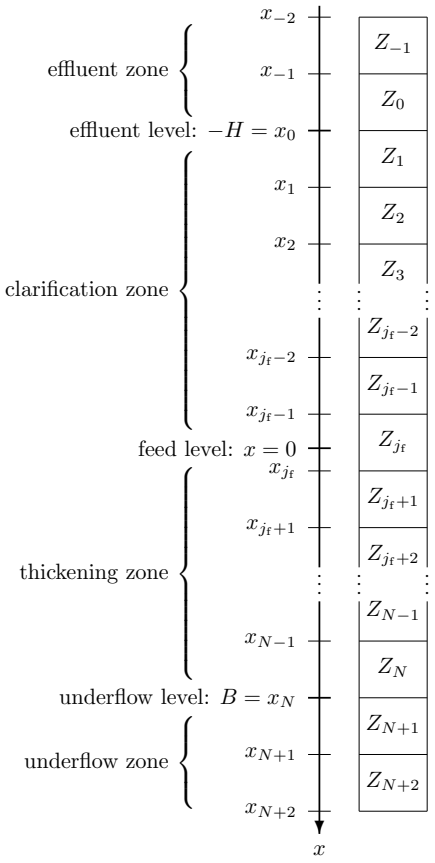

Figure 3: The sedimentation tank is subdivided into layers in order to discretize the space.

The average $Z_j = Z_j(t)$ of the exact solution Z over layer j at time t is defined as

$$Z_j(t) := \frac{1}{\Delta x} \int_{x_{j-1}}^{x_j} Z(x,t)\,\mathrm{d}x. \tag{4.5}$$

The primitive of d_comp is defined as

$$D(Z) := \int_{Z_\mathrm{c}}^{Z} d_\mathrm{comp}(s)\,\mathrm{d}s, \tag{4.6}$$

and thus

$$d_{\text{comp}}(Z)\frac{\partial Z}{\partial x} = \frac{\partial}{\partial x}D(Z).$$

Now by defining

$$J_{\text{disp}}(x,t) := d_{\text{disp}}\big(x, Q_{\text{f}}(t)\big)\frac{\partial Z}{\partial x}, \tag{4.7}$$

$$J_{\text{comp}}(x,t) := \gamma(x)\frac{\partial D(Z)}{\partial x}, \tag{4.8}$$

(4.2) can be written as

$$\Phi = F - J_{\text{disp}} - J_{\text{comp}}. \tag{4.9}$$

Now using equations (4.5)-(4.9), (4.4) can be rewritten as

$$\begin{aligned}\frac{\mathrm{d}Z_j}{\mathrm{d}t} =& -\frac{F(Z(x_j,t),x_j,t) - F(Z(x_{j-1},t),x_{j-1},t)}{\Delta x} + \frac{J_{\text{disp}}(x_j,t) - J_{\text{disp}}(x_{j-1},t)}{\Delta x} + \\ &+ \frac{J_{\text{comp}}(x_j,t) - J_{\text{comp}}(x_{j-1},t)}{\Delta x} + \frac{1}{\Delta x}\int_{x_{j-1}}^{x_j}\frac{Q_{\text{f}}(t)Z_{\text{f}}(t)}{A}\delta(x)\,\mathrm{d}x,\end{aligned} \tag{4.10}$$

which holds for all layers $j = -1, 0, 1, \ldots, N+2$.

It should be noted that (4.10) is indeed an exact form of Equation (4.4) for each layer j. What appears to be finite difference quotients in the right hand side of (4.10) are in fact not, the expressions follow from the conservation law. Equation (4.10) is now a discretized version of Equation (4.1) rewritten as a system of ordinary differential equations, depending on the time t. This system of equations can now be solved using some time discretization scheme.

4.2 Approximations

Equation (4.10) consists of several terms that need to be approximated to be able to be solved.

4.2.1 The convective flux

The convective flux $F(Z(x_j,t),x_j,t)$ at a certain position x_j is assumed to only depend on the adjacent layer concentrations $Z_j(t)$ and $Z_{j+1}(t)$, that is,

$$F_j^{\text{num}}(Z_j(t), Z_{j+1}(t), t) \approx F(Z(x_j,t),x_j,t).$$

The convective flux function for the substrate given by (3.5) is linear in S and thus can be easily approximated by

$$F_j^{\text{s,num}} = F_j^{\text{s,num}}(S_j, S_{j+1}, t) = \begin{cases} -\frac{(1-w)Q}{A}S_{j+1}, & \text{for } x_j < 0, \\ \frac{(r+w)Q}{A}S_j, & \text{for } x_j > 0, \end{cases}$$

which written out for each layer is

$$F_j^{\mathrm{s,num}} = \begin{cases} -\frac{(1-w)Q}{A} S_{j+1}, & \text{for } j = -2, -1, \ldots, j_{\mathrm{f}} - 1, \\ \frac{(r+w)Q}{A} S_j, & \text{for } j = j_{\mathrm{f}}, j_{\mathrm{f}} + 1, \ldots, N + 2. \end{cases} \tag{4.11}$$

The approximation of the convective flux function for the biomass given by (3.6) is slightly more complicated as it involves an approximation of the nonlinear Kynch batch flux density function (3.7). A good approximation choice in terms of simulation speed is to use the Godunov numerical flux $G_j = G_j(X_j, X_{j+1})$ on f_{bk} as an approximation of $f_{\mathrm{bk}}\big(X(x_j, t)\big)$, that is,

$$f_{\mathrm{bk}}\big(X(x_j, t)\big) \approx G_j = \begin{cases} \min\limits_{X_j \le X \le X_{j+1}} f_{\mathrm{bk}}(X), & \text{if } X_j \le X_{j+1}, \\ \max\limits_{X_j \ge X \ge X_{j+1}} f_{\mathrm{bk}}(X), & \text{if } X_j > X_{j+1}. \end{cases} \tag{4.12}$$

Now the approximated flux can be written as

$$F_j^{\mathrm{num}} = F_j^{\mathrm{num}}(X_j, X_{j+1}, t) = \begin{cases} -\frac{Q_{\mathrm{e}}(t)}{A} X_{j+1}, & \text{for } x < -H, \\ -\frac{Q_{\mathrm{e}}(t)}{A} X_{j+1} + G_j, & \text{for } -H < x < 0, \\ \frac{Q_{\mathrm{u}}(t)}{A} X_j + G_j, & \text{for } 0 < x < B, \\ \frac{Q_{\mathrm{u}}(t)}{A} X_j, & \text{for } x > B, \end{cases}$$

which written out for each layer is

$$F_j^{\mathrm{num}} := \begin{cases} -\frac{Q_{\mathrm{e}}(t)}{A} X_{j+1}, & \text{for } j = -2, -1, \\ -\frac{Q_{\mathrm{e}}(t)}{A} X_{j+1} + G_j, & \text{for } j = 0, 1, \ldots, j_{\mathrm{f}} - 1, \\ \frac{Q_{\mathrm{u}}(t)}{A} X_j + G_j, & \text{for } j = j_{\mathrm{f}}, j_{\mathrm{f}} + 1, \ldots, N, \\ \frac{Q_{\mathrm{u}}(t)}{A} X_j, & \text{for } j = N + 1, N + 2. \end{cases} \tag{4.13}$$

In Bürger et al. (2012a) it is investigated whether the use of the Godunov numerical flux leads to convergence to a unique solution. With another approximation of the convective flux, the Engquist-Osher numerical flux, and the use of an explicit Euler time integrator, convergence to a unique solution is guaranteed. Simulations provided in the article indicate that the Godunov method and an explicit Euler time integrator yields the same result as the Engquist-Osher method. It is thus assumed that the use of the Godunov numerical flux leads to convergence to a unique solution, although it remains to be proven. The reason for working with the Godunov method instead of the Engquist-Osher method is that the former is easier to implement. For further details on this topic see Bürger et al. (2012a).

4.2.2 The dispersion and compression fluxes

The dispersion flux given by (4.7) can be approximated by a finite difference quotient

$$J_{\mathrm{disp}}(x_j, t) \approx J_{\mathrm{disp},j}^{\mathrm{num}} := d_{\mathrm{disp},j} \frac{X_{j+1} - X_j}{\Delta x}. \tag{4.14}$$

In the same way the compression flux (4.8) can be approximated by

$$J_{\mathrm{comp}}(x_j,t) \approx J^{\mathrm{num}}_{\mathrm{comp},j} := \gamma(x_j)\frac{D^{\mathrm{num}}_{j+1} - D^{\mathrm{num}}_{j}}{\Delta x}, \tag{4.15}$$

where

$$D^{\mathrm{num}}_j := D(X_j) = \int_{X_{\mathrm{c}}}^{X_j} d_{\mathrm{comp}}(s)\,\mathrm{d}s. \tag{4.16}$$

With the particular choice of the effective solid stress function (3.13) the compression function is given by (3.15) and the primitive can be computed exactly. Trivially for, $0 \le X_j < X_{\mathrm{c}},\ D^{\mathrm{num}}_j = 0$. For $X_j \ge X_{\mathrm{c}}$,

$$D^{\mathrm{num}}_j = D(X_j) = \int_{X_{\mathrm{c}}}^{X_j} d_{\mathrm{comp}}(s)\,\mathrm{d}s = \frac{\rho_{\mathrm{s}} a v_0}{(\rho_{\mathrm{s}}-\rho_{\mathrm{f}})r_{\mathrm{V}} g}\big(\mathrm{e}^{-r_{\mathrm{V}}X_{\mathrm{c}}} - \mathrm{e}^{-r_{\mathrm{V}}X_j}\big),$$

so to conclude

$$D^{\mathrm{num}}_j = \begin{cases} 0 & \text{for } X_j < X_{\mathrm{c}}, \\ \frac{\rho_{\mathrm{s}} a v_0}{(\rho_{\mathrm{s}}-\rho_{\mathrm{f}})r_{\mathrm{V}} g}\big(\mathrm{e}^{-r_{\mathrm{V}}X_{\mathrm{c}}} - \mathrm{e}^{-r_{\mathrm{V}}X_j}\big) & \text{for } X_j \ge X_{\mathrm{c}}. \end{cases} \tag{4.17}$$

Now the exact conservation law (4.10) for the biomass can be rewritten by an approximate method-of-lines formula

$$\begin{aligned}\frac{\mathrm{d}X_j}{\mathrm{d}t} =& -\frac{F^{\mathrm{num}}_j - F^{\mathrm{num}}_{j-1}}{\Delta x} + \frac{1}{\Delta x}\Big(J^{\mathrm{num}}_{\mathrm{disp},j} - J^{\mathrm{num}}_{\mathrm{disp},j-1} + J^{\mathrm{num}}_{\mathrm{comp},j} - J^{\mathrm{num}}_{\mathrm{comp},j-1}\Big) + \\ &+ \frac{Q_{\mathrm{f}}X_{\mathrm{f}}}{A\Delta x}\delta_{j,j_{\mathrm{f}}}, \quad j = -1, 0, \ldots, N+2,\end{aligned} \tag{4.18}$$

and analogously for the substrate

$$\frac{\mathrm{d}S_j}{\mathrm{d}t} = -\frac{F^{\mathrm{s,num}}_j - F^{\mathrm{s,num}}_{j-1}}{\Delta x} + \frac{Q_{\mathrm{f}}S_{\mathrm{f}}}{A\Delta x}\delta_{j,j_{\mathrm{f}}}, \quad j = -1, 0, \ldots, N+2, \tag{4.19}$$

where $\delta_{j,j_{\mathrm{f}}} = 1$ if $j = j_{\mathrm{f}}$ and $\delta_{j,j_{\mathrm{f}}} = 0$ otherwise.

4.3 Time discretization

4.3.1 Explicit Euler method

Equations (4.18) and (4.19) are systems of ordinary differential equations depending only on time t. In order solve them one needs to make use of some time discretization scheme. The simplest one is the explicit Euler method. Select a time step size

$\Delta t > 0$ and define $t_n := n\Delta t,\ n = 0, 1, 2, \ldots$. To ensure stability of the numerical scheme the time step has to be sufficiently short. The time step must satisfy the CFL (Courant-Friedrichs-Lewy) condition given by

$$\begin{aligned}\Delta t \leq \Bigg[& \frac{1}{\Delta x}\left(\max_{0\leq t\leq T} \frac{Q_{\mathrm{f}}(t)}{A} + \max_{0\leq Z\leq Z_{\max}} \left| f'_{\mathrm{bk}}(Z) \right| \right) + \\ & + \frac{2}{(\Delta x)^2}\left(\max_{0\leq Z\leq Z_{\max}} d_{\mathrm{comp}}(Z) + \max_{\substack{-H\leq x\leq B,\\ 0\leq t\leq T}} d_{\mathrm{disp}}\big(x, Q_{\mathrm{f}}(t)\big) \right) \Bigg]^{-1}.\end{aligned} \tag{4.20}$$

Only Equation (4.18) contains second-order derivative terms that arises from the compression and diffusion and thus this equation will be far more restricting to the time step length then Equation (4.19) for which these terms will vanish.

In accordance with (4.5) one defines the layer concentration at time t_n as

$$Z_j^n := Z_j(t_n) = \frac{1}{\Delta x} \int_{x_{j-1}}^{x_j} Z(x, t_n)\,\mathrm{d}x, \quad j = -1, 0, \ldots, N+2, \quad n = 0, 1, 2, \ldots$$

and the time derivative is approximated by

$$\frac{\mathrm{d}Z_j}{\mathrm{d}t} \approx \frac{Z_j^{n+1} - Z_j^n}{\Delta t}. \tag{4.21}$$

Now putting (4.21) into (4.18) and (4.19) one obtains the fully discrete method for the biomass and the substrate respectively

$$\begin{aligned} X_j^{n+1} = X_j^n & - \frac{\Delta t}{\Delta x}\left(F_j^{\mathrm{num},n} - F_{j-1}^{\mathrm{num},n} \right) + \\ & + \frac{\Delta t}{\Delta x}\left(J_{\mathrm{disp},j}^{\mathrm{num},n} - J_{\mathrm{disp},j-1}^{\mathrm{num},n} + J_{\mathrm{comp},j}^{\mathrm{num},n} - J_{\mathrm{comp},j-1}^{\mathrm{num},n} \right) + \frac{\Delta t Q_{\mathrm{f}}(t_n) X_{\mathrm{f}}(t_n)}{A\Delta x}\delta_{j,j_{\mathrm{f}}}, \\ & j = -1, 0, \ldots, N+2, \quad n = 0, 1, 2, \ldots \end{aligned} \tag{4.22}$$

$$\begin{aligned} S_j^{n+1} = S_j^n & - \frac{\Delta t}{\Delta x}\left(F_j^{\mathrm{s,num},n} - F_{j-1}^{s,\mathrm{num},n} \right) + \frac{\Delta t Q_{\mathrm{f}}(t_n) S_{\mathrm{f}}(t_n)}{A\Delta x}\delta_{j,j_{\mathrm{f}}}, \\ & j = -1, 0, \ldots, N+2, \quad n = 0, 1, 2, \ldots \end{aligned} \tag{4.23}$$

In order to solve this system of equations one also has to impose initial conditions on the concentrations, that is one has to prescribe X_j^0 and S_j^0 for $j = -1, 0, \ldots, N+2$. The ordinary differential equations in the models, i.e. Equations (3.1), (3.2), (3.19), (3.20) and (3.21) are discretized in the same way and need also be subject to some initial condition.

4.3.2 Other Runge-Kutta methods

The Runge-Kutta methods, often denoted the RK methods, is a family of solvers for initial value problems. The explicit Euler method is an RK method of convergence order 1. There exists a great variety of RK methods having different orders of convergence. The CFL condition (4.20) is a limitation on the step size using the Euler method. Higher order RK methods may be able to take larger time steps than this, while maintaining stability, thus making up for the extra computations needed in each time step. For most initial value problems higher order Runge Kutta methods are superior to the Euler method both in terms of speed and accuracy. Equations (4.18) and (4.19) are initial value problems that hold for each layer j. The two equations may both, after redefining their respective right hand sides be written as

$$\frac{\mathrm{d}Z_j}{\mathrm{d}t} := f(t, Z_j), \quad Z_j(0) = Z_j^0, \quad t \in [0, T]. \tag{4.24}$$

In the same fashion as for the explicit Euler method one chooses a time step of length Δt, and defines $Z_j^n = Z_j(t_n) = Z_j(n\Delta t)$ for $n = 0, 1, 2, \ldots$. A second order RK method (RK2) to solve (4.24), the so called *midpoint method*, is given by

$$\begin{cases} F_1 = f(t_n, Z_j^n), \\ F_2 = f(t_n + \Delta t/2, Z_j^n + \Delta t F_1/2), \\ Z_j^{n+1} = Z_j^n + \Delta t F_2. \end{cases} \tag{4.25}$$

A fourth order RK method (RK4), also known as the *classical method*, is given by

$$\begin{cases} F_1 = f(t_n, Z_j^n), \\ F_2 = f(t_n + \Delta t/2, Z_j^n + \Delta t F_1/2), \\ F_3 = f(t_n + \Delta t/2, Z_j^n + \Delta t F_2/2), \\ F_4 = f(t_n + \Delta t, Z_j^n + \Delta t F_3), \\ Z_j^{n+1} = Z_j^n + \frac{\Delta t}{6}(F_1 + 2F_2 + 2F_3 + F_4). \end{cases} \tag{4.26}$$

Similar formulas of course hold for Equations (3.1), (3.2), (3.19), (3.20) and (3.21). For more details on RK methods the reader is referred to Edsberg (2008).

4.3.3 Semi-implicit method

When solving (4.18) the greatest restriction on the length of the time step Δt given by the CFL condition (4.20) arises from the second-order derivative terms, i.e. the compression and dispersion terms, since these are multiplied by $\frac{2}{(\Delta x)^2}$ instead of $\frac{1}{\Delta x}$ as is the case for the first-order derivative terms. An idea of increasing the simulation speed is to use a semi-implicit method that uses two different methods to solve the hyperbolic part (containing the first-order derivative terms) and the parabolic

part (containing the second-order derivative terms) of the equation respectively. The hyperbolic part is solved with the explicit Euler method using a step size that fulfills

$$\Delta t \leq \left[\frac{1}{\Delta x} \left(\max_{0 \leq t \leq T} \frac{Q_{\mathrm{f}}(t)}{A} + \max_{0 \leq X \leq X_{\max}} \left| f'_{\mathrm{bk}}(X) \right| \right) \right]^{-1}. \tag{4.27}$$

Now the idea is to use the same time step to solve also the parabolic part of the equation using an implicit method. The cost for taking this larger time step is the need to solve a nonlinear system of equations at every time step. The method is established by first modifying (4.22) so that

$$\begin{aligned} X_j^{n+1} = X_j^n &- \frac{\Delta t}{\Delta x} \left(F_j^{\mathrm{num},n} - F_{j-1}^{\mathrm{num},n} \right) + \\ &+ \frac{\Delta t}{\Delta x} \left(J_{\mathrm{disp},j}^{\mathrm{num},n+1} - J_{\mathrm{disp},j-1}^{\mathrm{num},n+1} + J_{\mathrm{comp},j}^{\mathrm{num},n+1} - J_{\mathrm{comp},j-1}^{\mathrm{num},n+1} \right) + \\ &+ \frac{\Delta t Q_{\mathrm{f}}(t_n) X_{\mathrm{f}}(t_n)}{A \Delta x} \delta_{j,j_{\mathrm{f}}}, \quad j = -1, 0, \ldots, N+2, \quad n = 0, 1, 2, \ldots \end{aligned} \tag{4.28}$$

where the time index has been changed from n to $n+1$ for the second-order derivative terms. Now with X_j^n given, take an explicit Euler step for the hyperbolic part of the equation,

$$X_{j,\mathrm{hyp}}^n := X_j^n - \frac{\Delta t}{\Delta x} \left(F_j^{\mathrm{num},n} - F_{j-1}^{\mathrm{num},n} \right) + \frac{\Delta t Q_{\mathrm{f}}(t_n) X_{\mathrm{f}}(t_n)}{A \Delta x} \delta_{j,j_{\mathrm{f}}}, \tag{4.29}$$

to obtain

$$X_j^{n+1} = X_{j,\mathrm{hyp}}^n + \frac{\Delta t}{\Delta x} \left(J_{\mathrm{disp},j}^{\mathrm{num},n+1} - J_{\mathrm{disp},j-1}^{\mathrm{num},n+1} + J_{\mathrm{comp},j}^{\mathrm{num},n+1} - J_{\mathrm{comp},j-1}^{\mathrm{num},n+1} \right),$$

or equally

$$0 = X_j^{n+1} - X_{j,\mathrm{hyp}}^n - \frac{\Delta t}{\Delta x} \left(J_{\mathrm{disp},j}^{\mathrm{num},n+1} - J_{\mathrm{disp},j-1}^{\mathrm{num},n+1} + J_{\mathrm{comp},j}^{\mathrm{num},n+1} - J_{\mathrm{comp},j-1}^{\mathrm{num},n+1} \right),$$

which, using (4.14)-(4.17), can be written as

$$\begin{aligned} 0 = X_j^{n+1} - X_{j,\mathrm{hyp}}^n - \frac{\Delta t}{(\Delta x)^2} \Big(& d_{\mathrm{disp},j} \big(X_{j+1}^{n+1} - X_j^{n+1} \big) - d_{\mathrm{disp},j-1} \big(X_j^{n+1} - X_{j-1}^{n+1} \big) \\ & + \gamma(x_j) \big(D_{j+1}^{\mathrm{num},n+1} - 2 D_j^{\mathrm{num},n+1} + D_{j-1}^{\mathrm{num},n+1} \big) \Big) =: M_j(\mathbf{X}^{n+1}), \end{aligned} \tag{4.30}$$

where $\mathbf{X}^{n+1}$ is a vector containing all the layer concentrations at the time point $n+1$. Let $\mathbf{M}$ denote the column vector with components M_j and the problem thus reduces to solve the nonlinear system of equations given by

$$\mathbf{M}(\mathbf{X}^{n+1}) = \mathbf{0}. \tag{4.31}$$

The system of equations can for instance be solved by using the Newton-Rhapson method. At each iteration k, the method computes the root of a first order Taylor approximation, at a point $\mathbf{X}_0$, of $\mathbf{M}$ by approximating

$$\mathbf{M}(\mathbf{X}) \approx \mathbf{M}(\mathbf{X}_0) + \mathbf{M}'(\mathbf{X}_0)(\mathbf{X} - \mathbf{X}_0). \tag{4.32}$$

Now the right hand side of (4.32) with $\mathbf{X} = \mathbf{X}^{n+1,k+1}$ and $\mathbf{X}_0 = \mathbf{X}^{n+1,k}$ is put into (4.31) giving the equation

$$\mathbf{M}(\mathbf{X}^{n+1,k}) + \mathbf{M}'(\mathbf{X}^{n+1,k})(\mathbf{X}^{n+1,k+1} - \mathbf{X}^{n+1,k}) = \mathbf{0}, \tag{4.33}$$

where $\mathbf{M}'$ is the Jacobian matrix given by

$$\mathbf{M}'(\mathbf{X}^{n+1}) = \begin{pmatrix} \frac{\partial M_{-1}}{\partial X_{-1}^{n+1}} & \frac{\partial M_{-1}}{\partial X_{0}^{n+1}} & \cdots \\ \frac{\partial M_{0}}{\partial X_{-1}^{n+1}} & \frac{\partial M_{0}}{\partial X_{0}^{n+1}} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}. \tag{4.34}$$

For more details on the Newton-Rhapson method see Edsberg (2008). The two outmost layers, indexed $j = -1$ and $j = N + 2$ are not subject to any compression and dispersion terms and need not to be included here. These layers only contain first-order derivative terms and can be solved with the explicit Euler method directly. Thus one considers only layers $j = 0, 1, \ldots, N+1$ when solving the system. Looking at (4.30) it is evident that the Jacobian matrix has a tridiagonal structure, that is (now excluding the layers at the top as well as the bottom),

$$\mathbf{M}'(\mathbf{X}^{n+1}) = \begin{pmatrix} \frac{\partial M_{0}}{\partial X_{0}^{n+1}} & \frac{\partial M_{0}}{\partial X_{1}^{n+1}} & 0 & 0 & \cdots \\ \frac{\partial M_{1}}{\partial X_{0}^{n+1}} & \frac{\partial M_{1}}{\partial X_{1}^{n+1}} & \frac{\partial M_{1}}{\partial X_{2}^{n+1}} & 0 & \cdots \\ 0 & \frac{\partial M_{2}}{\partial X_{1}^{n+1}} & \ddots & \ddots & \ddots \\ 0 & 0 & \ddots & \ddots & \frac{\partial M_{N}}{\partial X_{N+1}^{n+1}} \\ \vdots & \vdots & \ddots & \frac{\partial M_{N+1}}{\partial X_{N}^{n+1}} & \frac{\partial M_{N+1}}{\partial X_{N+1}^{n+1}} \end{pmatrix}, \tag{4.35}$$

where

$$\frac{\partial M_j}{\partial X_{j-1}^{n+1}} = -\frac{\Delta t}{(\Delta x)^2}\left(d_{\text{disp},j-1} + d_{\text{comp}}\big(X_{j-1}^{n+1}\big)\right), \tag{4.36}$$

$$\frac{\partial M_j}{\partial X_{j}^{n+1}} = 1 + \frac{\Delta t}{(\Delta x)^2}\left(d_{\text{disp},j} + d_{\text{disp},j-1} + 2d_{\text{comp}}\big(X_{j}^{n+1}\big)\right), \tag{4.37}$$

$$\frac{\partial M_j}{\partial X_{j+1}^{n+1}} = -\frac{\Delta t}{(\Delta x)^2}\left(d_{\text{disp},j} + d_{\text{comp}}\big(X_{j+1}^{n+1}\big)\right), \tag{4.38}$$

for the layers $j = 2, 3, \ldots, N-1$ and

$$\frac{\partial M_0}{\partial X_{-1}^{n+1}} = 0, \tag{4.39}$$

$$\frac{\partial M_0}{\partial X_0^{n+1}} = 1 + \frac{\Delta t}{(\Delta x)^2} d_{\text{comp}}\big(X_0^{n+1}\big), \tag{4.40}$$

$$\frac{\partial M_0}{\partial X_1^{n+1}} = -\frac{\Delta t}{(\Delta x)^2} d_{\text{comp}}\big(X_1^{n+1}\big), \tag{4.41}$$

$$\frac{\partial M_1}{\partial X_0^{n+1}} = -\frac{\Delta t}{(\Delta x)^2} d_{\text{comp}}\big(X_0^{n+1}\big), \tag{4.42}$$

$$\frac{\partial M_1}{\partial X_1^{n+1}} = 1 + \frac{\Delta t}{(\Delta x)^2}\Big(d_{\text{disp},1} + 2d_{\text{comp}}\big(X_1^{n+1}\big)\Big), \tag{4.43}$$

$$\frac{\partial M_1}{\partial X_2^{n+1}} = -\frac{\Delta t}{(\Delta x)^2}\Big(d_{\text{disp},1} + d_{\text{comp}}\big(X_2^{n+1}\big)\Big), \tag{4.44}$$

$$\frac{\partial M_N}{\partial X_{N-1}^{n+1}} = -\frac{\Delta t}{(\Delta x)^2}\Big(d_{\text{disp},N-1} + d_{\text{comp}}\big(X_{N-1}^{n+1}\big)\Big), \tag{4.45}$$

$$\frac{\partial M_N}{\partial X_N^{n+1}} = 1 + \frac{\Delta t}{(\Delta x)^2}\Big(d_{\text{disp},N-1} + 2d_{\text{comp}}\big(X_N^{n+1}\big)\Big), \tag{4.46}$$

$$\frac{\partial M_N}{\partial X_{N+1}^{n+1}} = -\frac{\Delta t}{(\Delta x)^2} d_{\text{comp}}\big(X_{N+1}^{n+1}\big), \tag{4.47}$$

$$\frac{\partial M_{N+1}}{\partial X_N^{n+1}} = -\frac{\Delta t}{(\Delta x)^2} d_{\text{comp}}\big(X_N^{n+1}\big), \tag{4.48}$$

$$\frac{\partial M_{N+1}}{\partial X_{N+1}^{n+1}} = 1 + \frac{\Delta t}{(\Delta x)^2} d_{\text{comp}}\big(X_{N+1}^{n+1}\big), \tag{4.49}$$

$$\frac{\partial M_{N+1}}{\partial X_{N+2}^{n+1}} = 0. \tag{4.50}$$

Consider a general tridiagonal matrix

$$\mathbf{A}_n = \begin{pmatrix} a_1 & b_1 & 0 & 0 & \cdots \\ c_1 & a_2 & b_2 & 0 & \cdots \\ 0 & c_2 & \ddots & \ddots & \ddots \\ 0 & 0 & \ddots & \ddots & b_{n-1} \\ \vdots & \vdots & \ddots & c_{n-1} & a_n \end{pmatrix}.$$

The matrix $\mathbf{A}_n$ is said to be strictly diagonally dominant if

$$|a_1| > |b_1|, \quad |a_i| > |b_i| + |c_{i-1}|, \text{ for } i = 2, 3, \ldots, n-1, \quad |a_n| > |c_{n-1}|.$$

According to theorem (8.3.1) in DuChateau and Zachmann (1989), it holds that $\mathbf{A}_n$ is invertible if it is strictly diagonally dominant. This condition thus provides a sufficient condition for the invertibility of a tridiagonal matrix but it is not a

necessary one. For the Jacobian matrix given by (4.35) the sufficient conditions guarantee invertibility if

$$\begin{cases} 1+\frac{\Delta t}{(\Delta x)^2}d_{\text{comp}}\big(X_0^{n+1}\big) > \frac{\Delta t}{(\Delta x)^2}d_{\text{comp}}\big(X_1^{n+1}\big) \\ 1+\frac{2\Delta t}{(\Delta x)^2}d_{\text{comp}}\big(X_j^{n+1}\big) > \frac{\Delta t}{(\Delta x)^2}\Big(d_{\text{comp}}\big(X_{j-1}^{n+1}\big)+d_{\text{comp}}\big(X_{j+1}^{n+1}\big)\Big) \\ 1+\frac{\Delta t}{(\Delta x)^2}d_{\text{comp}}\big(X_{N+1}^{n+1}\big) > \frac{\Delta t}{(\Delta x)^2}d_{\text{comp}}\big(X_N^{n+1}\big) \end{cases} \tag{4.51}$$

are fulfilled for $j = 1, 2, \ldots, N$. From the above conditions one can easily get a condition on the step size Δt that gives a restrictive condition guaranteeing invertibility. From (4.51) it is clear that invertibility is guaranteed if

$$\frac{\Delta t}{(\Delta x)^2}d_{\text{comp}}\big(X_j^{n+1}\big) < \frac{1}{2}$$

for all $j = 0, 1, \ldots, N+1$. This implies

$$\frac{\Delta t}{(\Delta x)^2}\max(d_{\text{comp}}) < \frac{1}{2},$$

which written out becomes

$$\Delta t < \frac{(\Delta x)^2}{2\max(d_{\text{comp}})} = \frac{g(\rho_{\text{s}}-\rho_{\text{f}})(\Delta x)^2}{2\rho_{\text{s}}av_0\text{e}^{-r_{\text{V}}X_{\text{c}}}}. \tag{4.52}$$

This is a restrictive condition because of the factor $(\Delta x)^2$ and it gives the same condition as the one that arises in the CFL condition given by (4.20). Thus this condition is not very useful since the point of the semi-implicit scheme is to be able to take larger time steps. In most physical relevant situation this condition will turn out to be overly restrictive. The distribution of particles exhibit discontinuous behavior below the critical concentration X_{c} but above that concentration it will be continuous. In an article by Bürger et al. (2005) a proof is given regarding the convergence of semi-implicit difference scheme for an initial-boundary value problem for a strongly degenerate parabolic equation. Lemma (3.3) in the article implies that the there exists a unique solution to the semi-implicit scheme above, thus guaranteeing the invertibility of the Jacobian matrix.

Now since the Jacobian matrix is invertible, (4.33) gives the iterative method

$$\mathbf{X}^{n+1,k+1} = \mathbf{X}^{n+1,k} - \big(\mathbf{M}'(\mathbf{X}^{n+1,k})\big)^{-1}\mathbf{M}(\mathbf{X}^{n+1,k}). \tag{4.53}$$

In order for the method to converge one needs to guess an initial solution that is sufficiently close to the actual solution. Assuming that the concentration changes little in each time step one is led to take $\mathbf{X}^{n+1,0} = \mathbf{X}^n$, that is the concentration in the last time step, as the initial guess. The iteration proceeds until a certain termination criteria is fulfilled, for instance the iteration may be terminated if

$$\frac{||\mathbf{X}^{n+1,k+1}-\mathbf{X}^{n+1,k}||}{||\mathbf{X}^{n+1,k}||} < \varepsilon,$$

for some chosen small tolerance ε. This means that the iteration is terminated if the relative change in the solution is small. It will be investigated how this tolerance should be chosen for optimal performance with respect to simulation time and accuracy. It is not necessarily so that a very small tolerance yields a significantly more accurate numerical solution than the case with a higher tolerance but the computations will be more expensive resulting in longer simulation times.

Another way to solve equation system (4.33) is to make use of an **LU**- decomposition of the Jacobian Matrix $\mathbf{M}'$. The expression (4.33) can be written as

$$\mathbf{M}'(\mathbf{X}^{n+1,k})\mathbf{X}^{n+1,k+1} = \mathbf{M}'(\mathbf{X}^{n+1,k})\mathbf{X}^{n+1,k} - \mathbf{M}(\mathbf{X}^{n+1,k}). \tag{4.54}$$

Since the Jacobian matrix is tridiagonal one can make use of the so called Thomas algorithm, see Conte and de Boor (1972) and Thomas (1949), that is a simplified form of Gaussian elimination. The system (4.54) can be written as

$$\begin{pmatrix} a_1 & b_1 & 0 & 0 & \cdots \\ c_1 & a_2 & b_2 & 0 & \cdots \\ 0 & c_2 & \ddots & \ddots & \ddots \\ 0 & 0 & \ddots & \ddots & b_{n-1} \\ \vdots & \vdots & \ddots & c_{n-1} & a_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}.$$

The algorithm first modifies the coefficients and successively eliminates unknown from the equations:

$$b'_i = \begin{cases} \frac{b_i}{a_i} & \text{for } i = 1 \\ \frac{b_i}{a_i - b'_{i-1}c_{i-1}} & \text{for } i = 2, 3, \ldots, n-1 \end{cases}$$

$$d'_i = \begin{cases} \frac{d_i}{a_i} & \text{for } i = 1 \\ \frac{d_i - d'_{i-1}c_{i-1}}{a_i - b'_{i-1}c_{i-1}} & \text{for } i = 2, 3, \ldots, n. \end{cases}$$

Then by a back substitution one obtains the solution

$$\begin{cases} x_n = d'_n \\ x_i = d'_i - b'_i x_{i+1} & \text{for } i = n-1, n-2, \ldots, 1. \end{cases}$$

4.3.4 MATLAB built-in solvers

In MATLAB there are a number of built-in ODE solvers, each with their advantages and disadvantages. All these solvers are using a variable step size contrary to the solvers described above which all use a fixed step size during the entire simulation. With a variable step size the solver takes an initial time step and then estimates the error. If the error is below a given tolerance the step is accepted, otherwise if the

error is above the preset tolerance the step is rejected and the solver will try again using a smaller step size. The disadvantage using a variable step size contrary to a fixed step size is the cost of estimating the error. Also if the equation is very stiff it may happen that the time step approaches zero causing the time integration to get stuck. The obvious advantage is that the solver often can take larger time steps well above the step size limit given by the CFL condition. Three different ODE solvers that are included in MATLAB will be tested, these are ode45, ode23 and ode15s, the latter is suitable for stiff problems. For more information on these solvers one may consult the documentation in MATLAB.

5 Simulations and Results

Throughout the simulations the parameter values are those found in Table 3 if not mentioned otherwise.

5.1 Simplified model

The simulations using the simplified model will only consider the biomass in a stand alone settler tank, thus neglecting the substrate and the reactor tank equations. It is sufficient to only consider the settler tank when investigating the efficiency of the different solvers since the settler equations will use most of the computer power.

5.1.1 Scenario 1

The first scenario is a very simplified situation with a stand alone settler. There are no bulk flows present and thus no dispersion effects take place, also all compression effects are neglected. In this simulation another expression for the Kynch batch flux density function is used. Instead of the expression given by Equation (3.11), the expression

$$f_{\text{bk}} = \begin{cases} v_0 X(1-X), & \text{for } 0 \leq X \leq 1, \\ 0, & \text{for } X > 1, \end{cases}$$

will be used. The reason to use this artificial expression is that it makes it easier to find "worst case scenarios" which means that one is forced to take time steps very close to the CFL condition to achieve a meaningful solution. When testing the efficiency of the different solvers one wishes to use such scenarios. The system is released with an initial sludge blanket at the point $x = 0$. Two different shock waves appear and merge after a short time when the solution reaches a steady state. Figure 4 shows the solution using the Euler method on a space grid of 2430 internal layers.

Parameter	Value
Y_{A}	0.24
Y_{H}	0.67
f_{P}	0.08
i_{XB}	0.08
i_{XP}	0.06
μ_{H}	$1/6$ h^{-1}
K_{S}	0.01 $\mathrm{kg/m}^3$
$K_{\mathrm{O,H}}$	0.0002 $\mathrm{kg/m}^3$
K_{NO}	0.0005$\mathrm{kg/m}^3$
b_{H}	0.0125 h^{-1}
η_{g}	0.8
η_{h}	0.8
k_{h}	0.125 h^{-1}
K_{X}	0.1
μ_{A}	$1/48$ h^{-1}
K_{NH}	0.001 $\mathrm{kg/m}^3$
b_{A}	$1/480$ h^{-1}
$K_{\mathrm{O,A}}$	0.0004 $\mathrm{kg/m}^3$
k_{A}	$25/12$ $\mathrm{m}^3/(\mathrm{kgh})$
S_{O}^*	0.008 $\mathrm{kg/m}^3$
H	1 m
B	3 m
A	1500 m^2
v_0	10 m/h
r_{V}	0.37 m^3/kg
g	9.81 $\mathrm{m/s}^2$
ρ_{s}	1050 $\mathrm{kg/m}^3$
$\Delta\rho$	52 $\mathrm{kg/m}^3$
Q_a	0 $\mathrm{kg/m}^3$
$(K_{\mathrm{L}}a)_1$	0 s^{-1}
$(K_{\mathrm{L}}a)_2$	0 s^{-1}
$(K_{\mathrm{L}}a)_3$	0 s^{-1}
$(K_{\mathrm{L}}a)_4$	0 s^{-1}
$(K_{\mathrm{L}}a)_5$	0 s^{-1}
V_1	1000 m^3
V_2	1000 m^3
V_3	1333 m^3
V_4	1333 m^3
V_5	1333 m^3

Table 3: List of parameter values used in the simulations. Note that some parameters may differ between simulations but if not stated otherwise these values will be used.

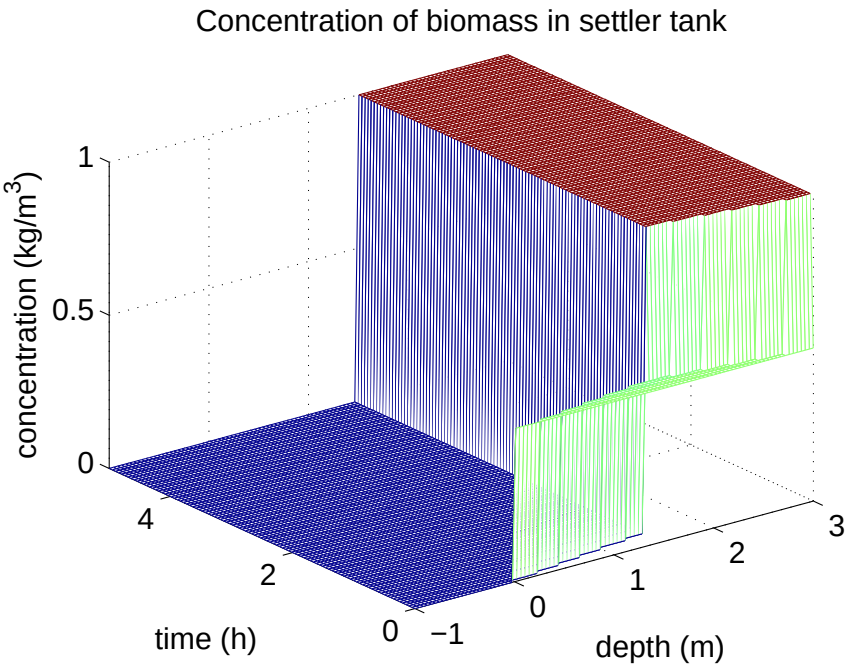

Figure 4: Simulation 1. Reference solution obtained using the Euler method with N=2430 plotted for 100×100 points.

5.1.2 Scenario 2

The second scenario also has no bulk flows and thus no dispersion present. However it takes compression into account contrary to Scenario 1. For the same reason as described under Scenario 1 another expression for the Kynch batch flux density function will be used,

$$f_{\mathrm{bk}} = v_0 X^2 \mathrm{e}^{-r_{\mathrm{V}} X}.$$

The parameters in the compression function given by (4.17) are $a = 0.1$ m^2/s^2 and $X_{\mathrm{c}} = (2 - \sqrt{2})/r_{\mathrm{V}} \approx 1.58$ kg/m^3. Figure 5 contains a simulation using the Euler method with a space grid consisting of 2430 internal layers.

Parameter	Value
a	0.5 $\mathrm{m}^2/\mathrm{s}^2$
X_c	6 kg/m^3
r	0.35
w	0.0155
α_1	0.01 m^{-1}
α_2	0.5638 s/m^2
r_V	0.45 m^3/kg
X_f	3.28 kg/m^3

Table 4: List of additional parameter values used in Scenario 3.

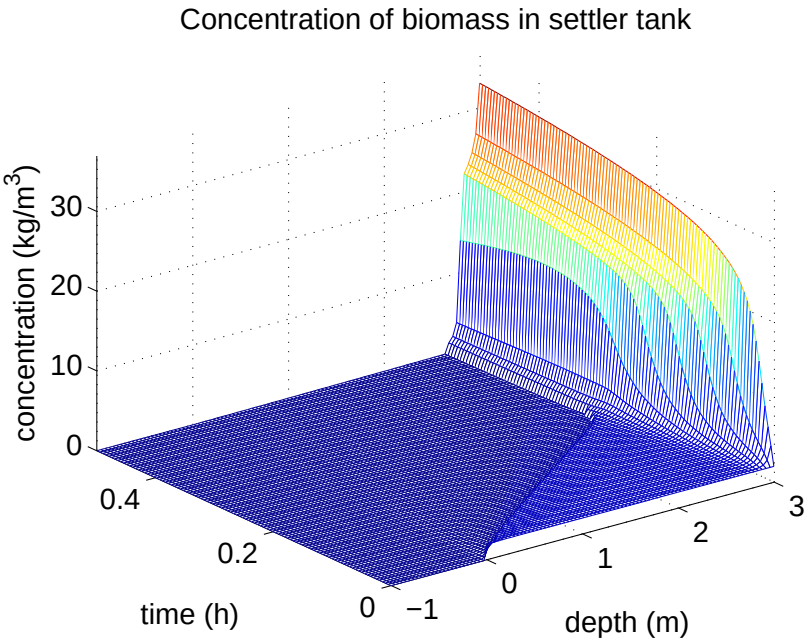

Figure 5: Simulation 2. Reference solution obtained using the Euler method with N=2430 plotted for 100 × 100 points.

5.1.3 Scenario 3

The third simulation is a more realistic scenario containing bulk flows, dispersion and compression. Here the Kynch batch flux density function (3.11) is used. Figure 7 shows the simulation starting from a steady state profile obtained from a simulation with constant influent. The solution is obtained with the Euler method and 810 internal layers. The volumetric inflow is depicted in Figure 6 and additional parameter values are found in Table 4. By the end of the simulation the settler becomes overloaded with sludge going up in the effluent zone.

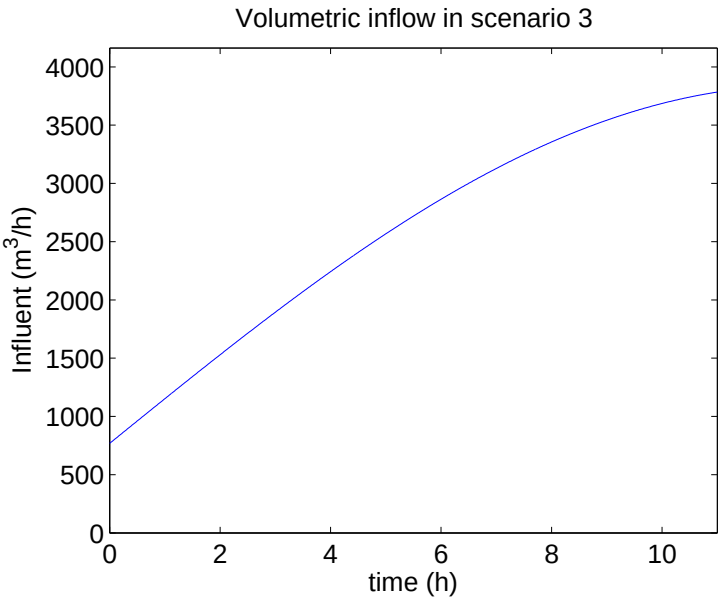

Figure 6: Volumetric inflow in Scenario 3.

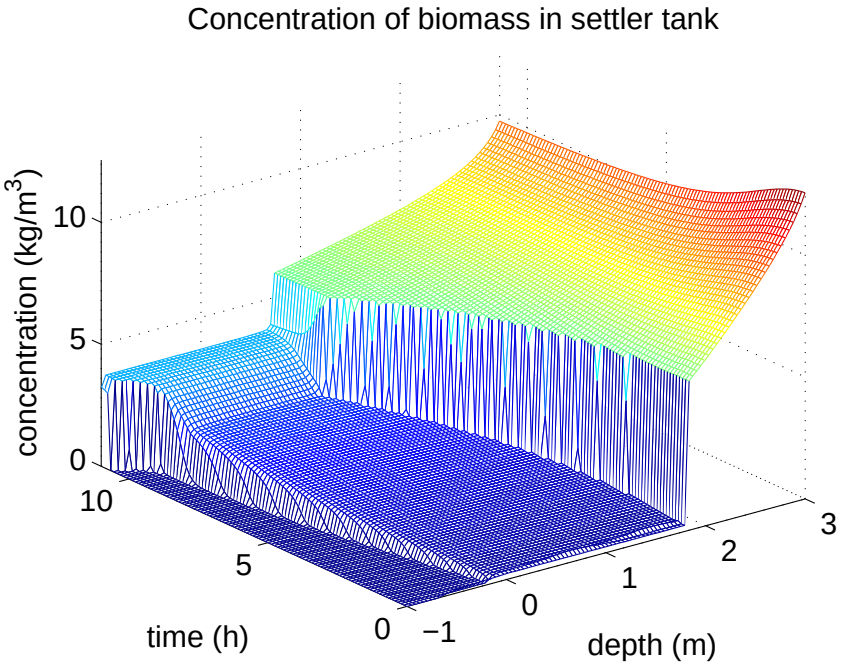

Figure 7: Simulation 3. Reference solution obtained using the Euler method with N=810 plotted for 100×100 points.

Figures 8 and 9 show what may happen when the CFL condition is violated. In this situation the time step length is set to approximately 20 % above the maximum

time step length given by the CFL condition. As the simulation shows the solution blows up after some time.

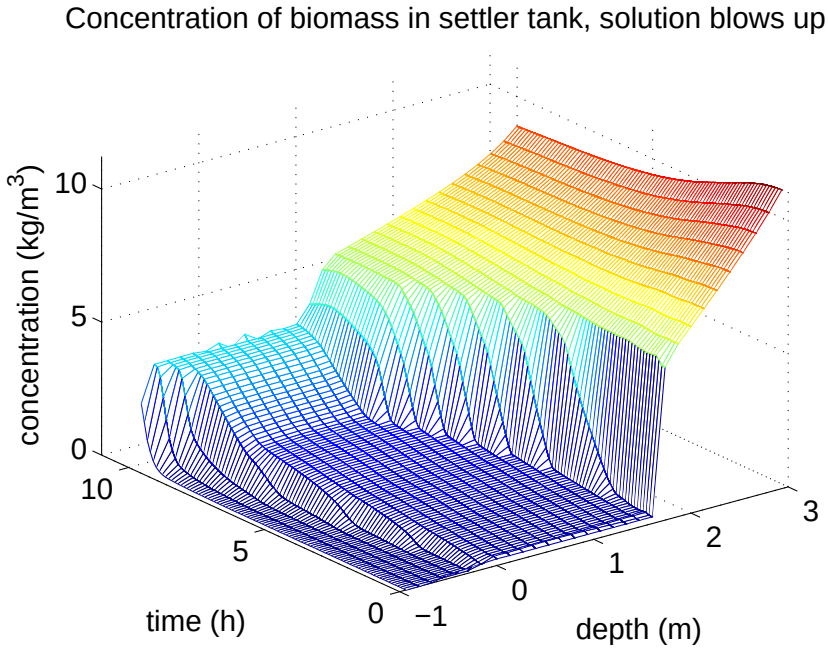

Figure 8: The CFL condition is violated and thus the solution blows up after some time.

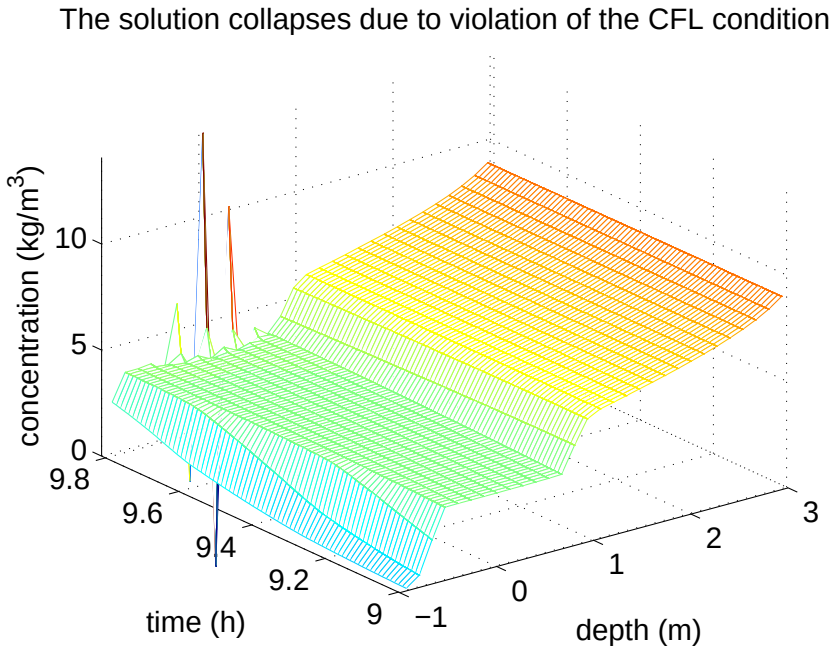

Figure 9: A closer look at when the solution blows up due to the violation of the CFL condition.

5.1.4 Efficiency of different solvers

The most time consuming equation to solve is Equation (3.4), that is the PDE modeling the concentration of the biomass in the settler tank. After establishing the integrated form and the method-of-lines formulas, the equation transforms into the system of ordinary differential equations given by (4.18). A good numerical method should be both fast and accurate and thus it is highly interesting to see how the different initial value problem solvers described in Section 4.3 perform when solving (4.18) numerically. In the following the different scenarios are simulated using different methods that are compared to each other with respect to the CPU time needed to execute the simulation and the relative error when compared to the reference solution for the specific scenario.

The different methods used are the explicit Euler method, the midpoint method (RK2), the classical method (RK4), the semi-implicit method and the MATLAB built-in solvers ode45, ode23 and ode15s. For the semi-implicit method it is investigated whether it is more efficient to use the backslash command in MATLAB or the Thomas algorithm to solve the linear system of equations involved in the method. The backslash command in MATLAB uses Gaussian elimination to solve the system. All the different methods are tested with the number of internal layers being $N = 10 \cdot 3^p$ for $p = 0, 1, 2, 3, 4$. The reference solution is obtained by solving (4.18) with $N = 10 \cdot 3^5 = 2430$ or $N = 10 \cdot 3^4 = 810$ layers using the explicit Euler method. It is assumed that this solution is very close to the exact one since investigations in Bürger et al. (2012a) indicated that the Euler method produces numerical solutions that converge to the exact solution as the grid size tends to zero.

The relative error used by Bürger et al. (2012a) is

$$e_X := \frac{\int_0^T \int_{-B}^{H} |X^N(x,t) - X^{\mathrm{ref}}(x,t)| \,\mathrm{d}x \,\mathrm{d}t}{\int_0^T \int_{-B}^{H} X^{\mathrm{ref}}(x,t) \,\mathrm{d}x \,\mathrm{d}t} \tag{5.1}$$

Here X^N is a piecewise constant representation of the approximate solution using N internal layers. X^{ref} are the reference solution obtained with the explicit Euler method using 2430 internal layers and restricted to the same grid as X^N by taking averages over the layers.

Figure 10 shows an efficiency plot for Scenario 1 with values found in Table 6. The Euler- RK2- and RK4-methods all use a time step length equal to the one given by the CFL condition (4.20), this will also be the case in all other simulations. Both the RK2- and RK4-methods will thus be slower since they perform more computations at every time step than the Euler method. Since these solvers are of higher orders

N	Maximum time step
10	$1.60 \cdot (\Delta t)_{\text{CFL}}$
30	$1.60 \cdot (\Delta t)_{\text{CFL}}$
90	$1.60 \cdot (\Delta t)_{\text{CFL}}$
270	$1.60 \cdot (\Delta t)_{\text{CFL}}$
810	$1.60 \cdot (\Delta t)_{\text{CFL}}$

Table 5: Maximum time steps that RK4 may take in Scenario 1 without causing instability.

than Euler one may expect that they would converge faster to the reference solution but this is not the case. Both RK2 and RK4 may take time steps exceeding the CFL condition while maintaining stability but not large enough time steps to be as fast as the Euler method, see Table 5. As seen in Table 6 the RK4 method takes around 4 times longer than the Euler method to execute the simulation and since the method only manages a factor of 1.6 times the Euler step length, it will be slower. Moreover the RK methods seem to produce a slightly larger error than the Euler method. The MATLAB built-in solvers all perform worse than the Euler method. Since no second order derivative terms are present in this scenario there is no use for the semi-implicit method.

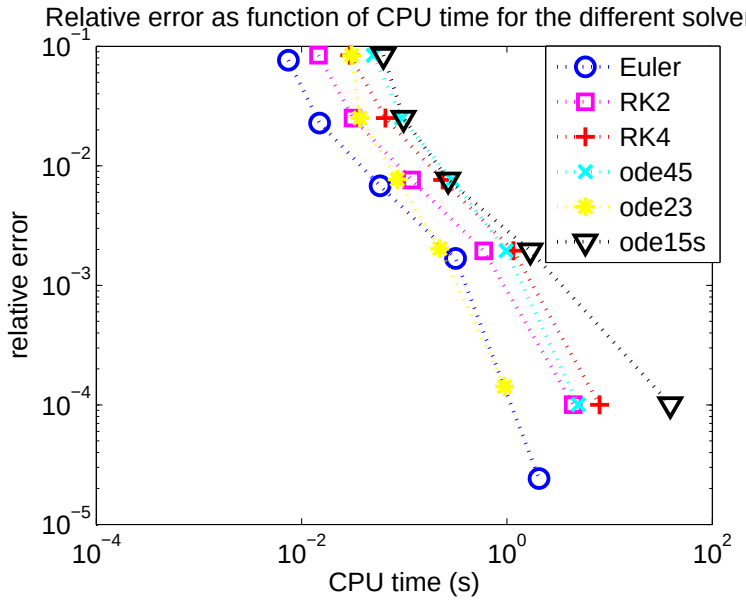

Figure 10: The relative errors for the different methods as function of the CPU time for Scenario 1.

Figure 11 shows an efficiency plot for Scenario 2. The results are similar to those for Scenario 1 with all solvers being rather equal in terms of accuracy but Euler being the fastest of them. The semi-implicit method makes uses of the larger allowed

Method	N	e_X	CPU time [s]
Euler	10	0.0765	0.0074
	30	0.0228	0.0150
	90	0.0068	0.0582
	270	0.0017	0.3161
	810	0.00002	2.0537
RK2	10	0.0843	0.0146
	30	0.0251	0.0317
	90	0.0076	0.1178
	270	0.0019	0.5946
	810	0.0001	4.4307
RK4	10	0.0842	0.0295
	30	0.0251	0.0657
	90	0.0076	0.2365
	270	0.0019	1.1563
	810	0.0001	7.9928
ode45	10	0.0842	0.0500
	30	0.0251	0.0943
	90	0.0076	0.2934
	270	0.0019	0.9829
	810	0.0001	4.9690
ode23	10	0.0842	0.0306
	30	0.0251	0.0370
	90	0.0077	0.0856
	270	0.0020	0.2220
	810	0.0001	0.9465
ode15s	10	0.0844	0.0626
	30	0.0251	0.0985
	90	0.0076	0.2677
	270	0.0019	1.6927
	810	0.0001	38.9931

Table 6: Errors and CPU times in Scenario 1 for the different methods.

N	Maximum time step
10	$1.35 \cdot (\Delta t)_{\text{CFL}}$
30	$1.50 \cdot (\Delta t)_{\text{CFL}}$
90	$1.45 \cdot (\Delta t)_{\text{CFL}}$
270	$1.40 \cdot (\Delta t)_{\text{CFL}}$
810	$1.40 \cdot (\Delta t)_{\text{CFL}}$

Table 7: Maximum time steps that RK4 may take in Scenario 2 without causing instability.

time step (4.27). In this simulation MATLAB's backslash command is used in the semi-implicit algorithm. Figure 12 contains a simulation of Scenario 2 with different tolerances in the Thomas algorithm. This shows that there is no gain in having have a very small tolerance. However a too large tolerance does not allow the Newton-Rhapson method to converge causing an increase in the relative error. For Scenario 2 it seems as a tolerance of $\varepsilon = 10^{-4}$ is the appropriate choice and this tolerance will be used in other scenarios as well. Figure 13 shows an efficiency plot now using the Thomas algorithm in the semi-implicit method with this tolerance. The method clearly improves in speed when using the Thomas algorithm, that makes use of the tridiagonal structure of the Jacobian matrix. It even performs better than the Euler method for 810 internal layers. Table 8 contains values from the simulations. Table 7 contains the maximum possible time steps for the RK4 method without causing instability. They are well below $4 \cdot (\Delta t)_{\text{CFL}}$ needed to achieve similar simulation speed as the Euler method.

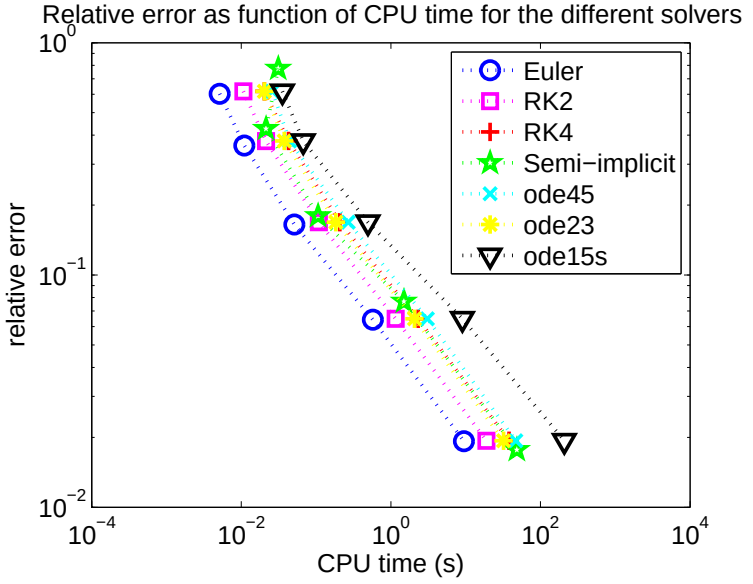

Figure 11: The relative errors for the different methods as function of the CPU time for Scenario 2 with the backslash command in MATLAB used in the semi-implicit method.

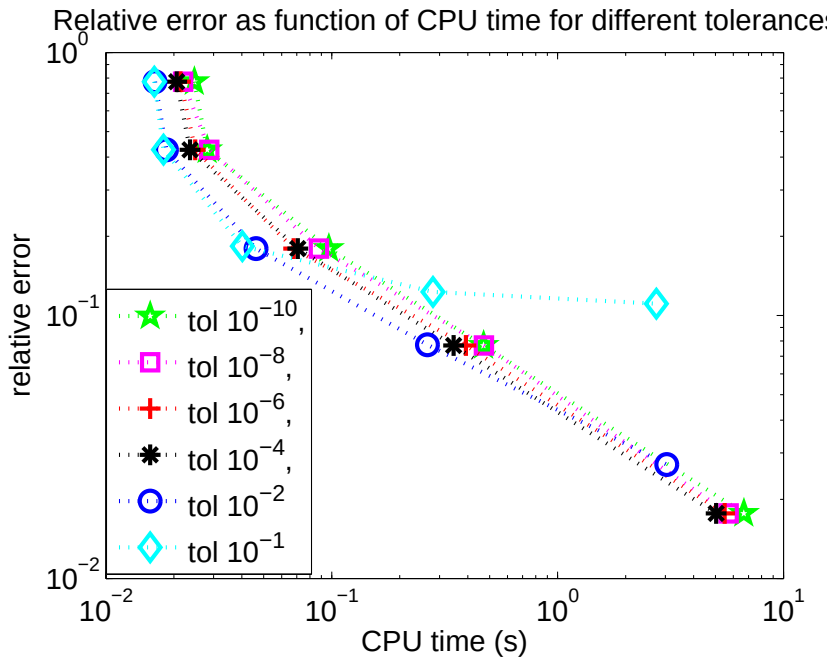

Figure 12: The relative errors for different tolerances in the Thomas algorithm as function of the CPU time for Scenario 2.

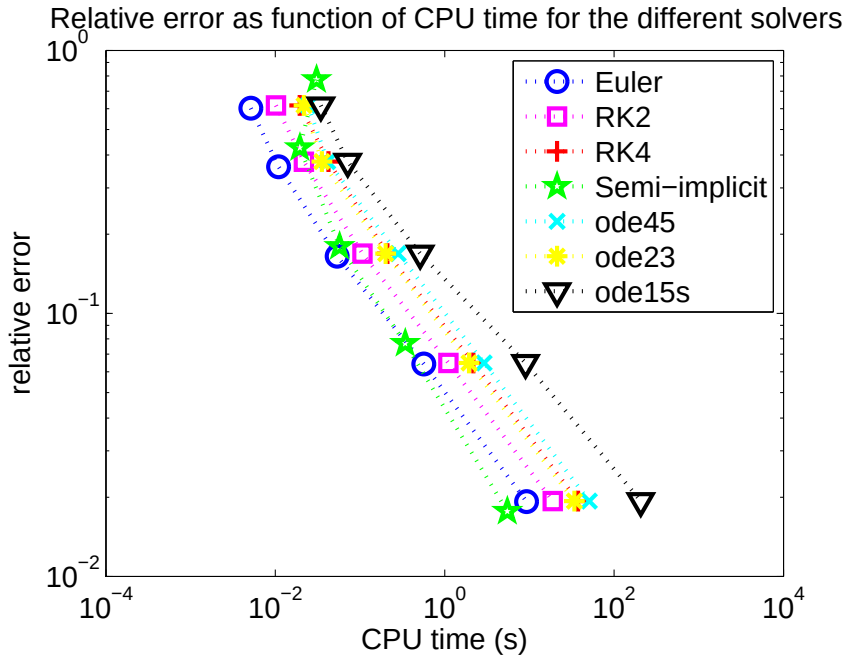

Figure 13: The relative errors for the different methods as function of the CPU time for Scenario 2 with the Thomas algorithm used in the semi-implicit method, tolerance in Newton step 10^{-4}.

Method	N	e_X	CPU time [s]
Euler	10	0.6026	0.0051
	30	0.3608	0.0110
	90	0.1650	0.0514
	270	0.0642	0.5760
	810	0.0192	9.4675
RK2	10	0.6179	0.0107
	30	0.3775	0.0215
	90	0.1687	0.1090
	270	0.0648	1.1616
	810	0.0193	18.8785
RK4	10	0.6177	0.0204
	30	0.3776	0.0428
	90	0.1687	0.2074
	270	0.0648	2.3235
	810	0.0193	38.0906
ode45	10	0.6178	0.0260
	30	0.3775	0.0443
	90	0.1687	0.2689
	270	0.0648	3.0819
	810	0.0193	46.7806
ode23	10	0.6178	0.0204
	30	0.3776	0.0375
	90	0.1687	0.1839
	270	0.0648	2.0669
	810	0.0193	32.1065
ode15s	10	0.6177	0.0355
	30	0.3775	0.0674
	90	0.1687	0.4939
	270	0.0648	9.0495
	810	0.0193	208.2970
Semi-implicit (backslash)	10	0.7740	0.0312
	30	0.4268	0.0214
	90	0.1797	0.1064
	270	0.0769	1.4936
	810	0.0177	48.5172
Semi-implicit (Thomas)	10	0.7740	0.0306
	30	0.4268	0.0196
	90	0.1797	0.0576
	270	0.0769	0.3419
	810	0.0177	5.4920

Table 8: Errors and CPU times in Scenario 2 for the different methods.

N	Maximum time step
10	$2.45 \cdot (\Delta t)_{\text{CFL}}$
30	$1.60 \cdot (\Delta t)_{\text{CFL}}$
90	$1.45 \cdot (\Delta t)_{\text{CFL}}$

Table 9: Maximum time steps that RK4 may take in Scenario 3 without causing instability.

Figure 14 shows the efficiency plot for Scenario 3 with values in Table 10. In this scenario the CFL condition is highly dominated by the dispersion term. Since the CFL condition for the semi-implicit method is unaffected by the the dispersion and compression it may take much larger time steps than the Euler- RK2- and RK4-methods. It is evident that the semi-implicit method is far superior to the other solvers in this scenario. Table 9 contains the maximum possible time steps for the RK4-method while maintaining stability and they are to short for the method to be as fast as the Euler method which is a factor 4 times faster when the methods use the same step size.

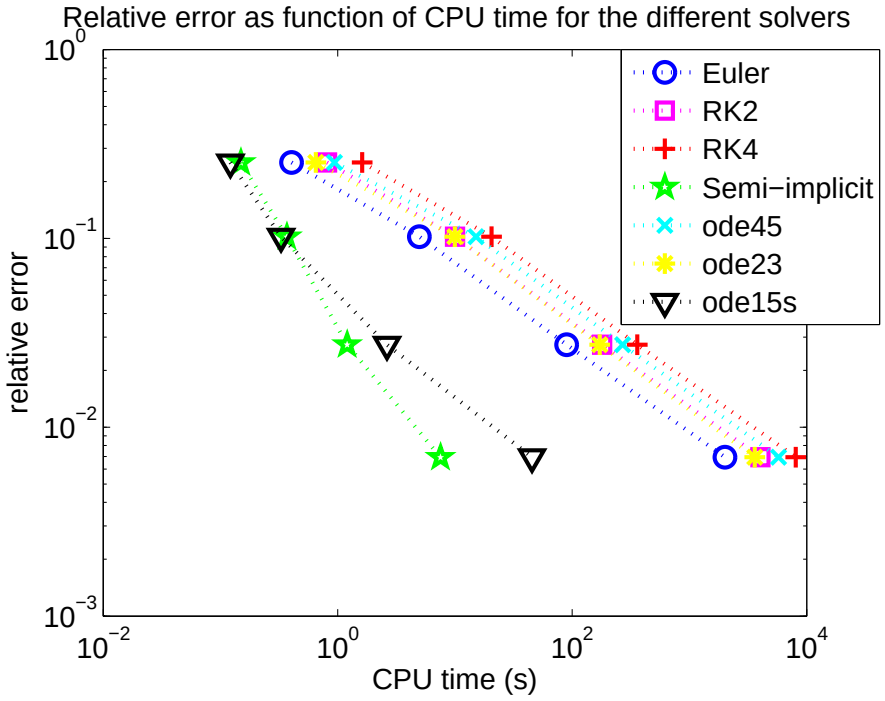

Figure 14: The relative errors for the different methods as function of the CPU time for Scenario 3 with the Thomas algorithm used in the semi-implicit method, tolerance in Newton step 10^{-4}.

Method	N	e_X	CPU time [s]
Euler	10	0.2523	0.4045
	30	0.1021	4.9697
	90	0.0274	89.5946
	270	0.0069	2.01E+3
RK2	10	0.2523	0.8119
	30	0.1020	10.0253
	90	0.0274	179.9050
	270	0.0069	4.03E+3
RK4	10	0.2523	1.6178
	30	0.1020	20.5211
	90	0.0274	359.3240
	270	0.0069	8.07E+3
ode45	10	0.2523	0.9370
	30	0.1020	15.1285
	90	0.0274	268.9525
	270	0.0069	5.75E+3
ode23	10	0.2523	0.6522
	30	0.1020	9.9725
	90	0.0274	171.5899
	270	0.0069	3.60E+3
ode15s	10	0.2522	0.1215
	30	0.1020	0.3284
	90	0.0274	2.6345
	270	0.0069	45.5111
Semi-implicit (Thomas)	10	0.2520	0.1497
	30	0.1019	0.3689
	90	0.0274	1.2032
	270	0.0069	7.5478

Table 10: Errors and CPU times in Scenario 3 for the different methods.

5.1.5 Smearing of shock-waves

As is seen in Figures 10, 11, 13 and 14 the RK methods are both slower and sometimes slightly more inaccurate than the simpler Euler method. A trouble with the RK methods is that they tend to smoothen out shock waves, i.e. they add more numerical viscosity. Consider a situation with a settler tank with no bulk flows, dispersion or compression, i.e. similar to Scenario 1 but with the Kynch batch flux density function (3.11). Let the initial concentration be 0 for layers $j = -1, 0 \ldots, p-1$, and 1 kg/m^3 for layers $j = p, p+1, \ldots, N+2$. Note that $f_{\text{bk}} > 0$, so there will be a shock wave moving downwards. Hence the concentration in layer p should tend to zero as fast as possible. For the the layer p, (4.18) reduces to

$$\frac{\mathrm{d}Z_p}{\mathrm{d}t}(t_n) = \frac{G^n_{p-1} - G^n_p}{\Delta x} =: F_1.$$

Now take an Euler step in this layer,

$$Z^1_p = \underbrace{Z^0_p}_{=1} + \frac{\Delta t}{\Delta x}(\underbrace{G^0_{p-1}}_{=0} - \underbrace{G^0_p}_{=f_{\text{bk}}(1)}) = 1 - \frac{\Delta t}{\Delta x} f_{\text{bk}}(1).$$

Now using the midpoint method (RK2), one notes that $F_2 = f(t_n + \Delta t/2, Z^n_j + \Delta t F_1/2)$. Here the function f is given by the right hand side in (4.18). Now

$$Z^1_p = \underbrace{Z^0_p}_{=1} + \Delta t f(\underbrace{t_0}_{=0} + \Delta t/2, \underbrace{Z^0_p}_{=1} + \underbrace{\Delta t F_1/2}_{=-\frac{\Delta t f_{\text{bk}}(1)}{2\Delta x}}) = 1 + \Delta t f\left(\Delta t/2, 1 - \frac{\Delta t f_{\text{bk}}(1)}{2\Delta x}\right) =$$

$$= 1 - \frac{\Delta t}{\Delta x} \underbrace{f_{\text{bk}}\left(\underbrace{1 - \frac{\Delta t f_{\text{bk}}(1)}{2\Delta x}}_{<1}\right)}_{<f_{\text{bk}}(1)} > 1 - \frac{\Delta t}{\Delta x} f_{\text{bk}}(1).$$

Thus it is clear that a step with the midpoint method yields a larger concentration than if one uses the Euler method. The concentration will go slower to zero using the midpoint method than the Euler method and thus the shock wave will be smeared out more since the solution should go quickly to zero. The situation is similar with the RK4 method. The reason for trusting Euler is that, as mentioned before, simulations have indicated that it does converge to the exact solution. Figure 15 shows a simulation of a scenario with no bulk flows, no dispersion and no compression. The Kynch batch flux density function is the one given by Equation (3.11). The solution is obtained with the Euler method using 270 internal layers. Figures 16 and 17 show how the shock wave looks for the three different solvers at two fixed time points. The two RK methods are very similar to each other and smoothen out the shock wave more than the Euler method.

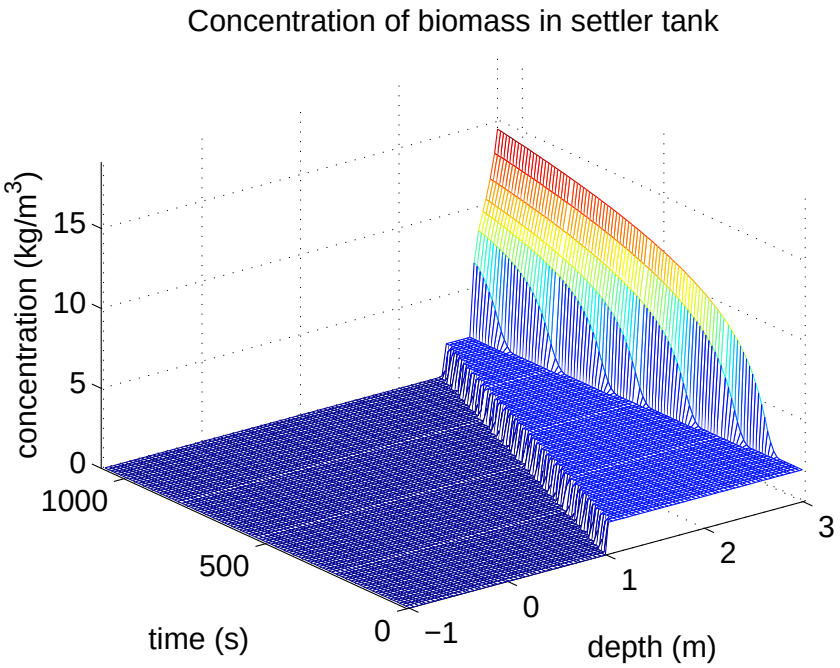

Figure 15: Traveling shock wave using an Euler solver with $N = 270$ plotted for 100×100 points.

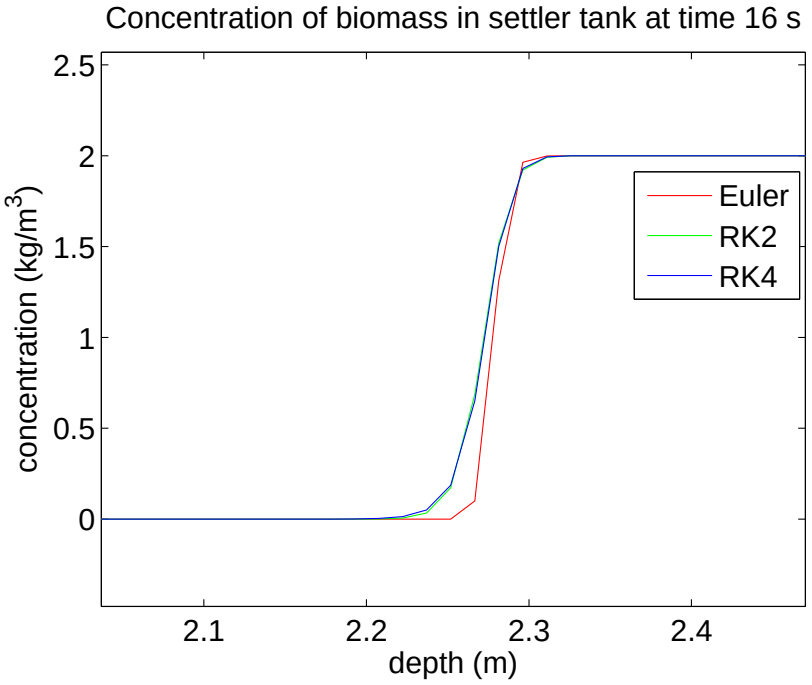

Figure 16: The shock wave is smeared out more with the RK2- and RK4-methods than with Euler.

State variable	Influent concentration
S_{I}	0.030 g/l
$X_{\mathrm{B,A}}$	0 g/l
S_{O}	0 g/l
X_{P}	0 g/l
S_{NO}	0 g/l
S_{I}	7 mol/m^3

Table 11: Additional storm weather influent data.

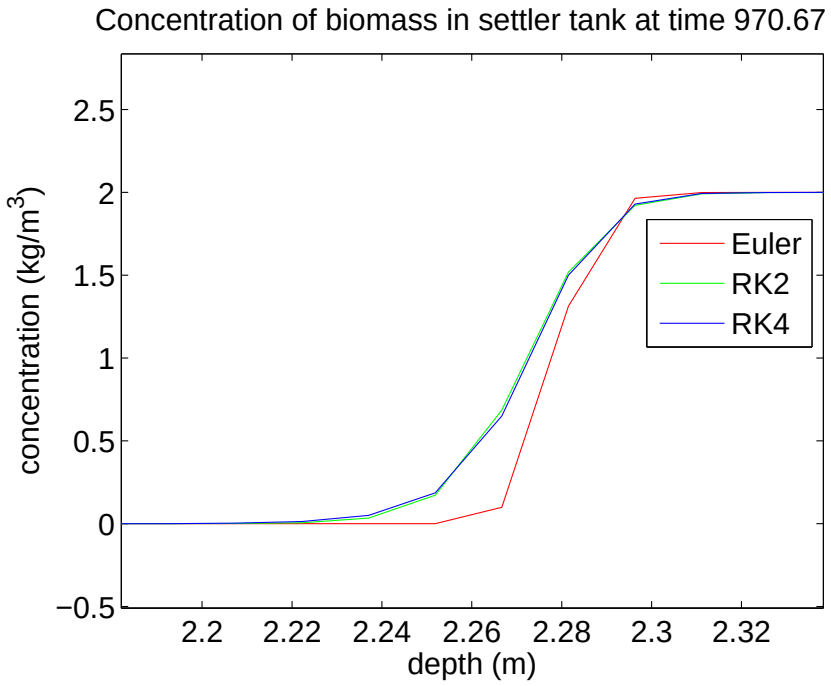

Figure 17: The shock wave is smeared out more with the RK2- and RK4-methods than with Euler.

5.2 Extended model

5.2.1 Storm weather influent data

Figures 18, 19 and 20 show storm weather influent data that is provided by Alex et al. (2008). Table 11 contains additional influent data. The influent data are given in intervals of fifteen minutes during two weeks. To get data at arbitrary time points linear interpolation has been used.

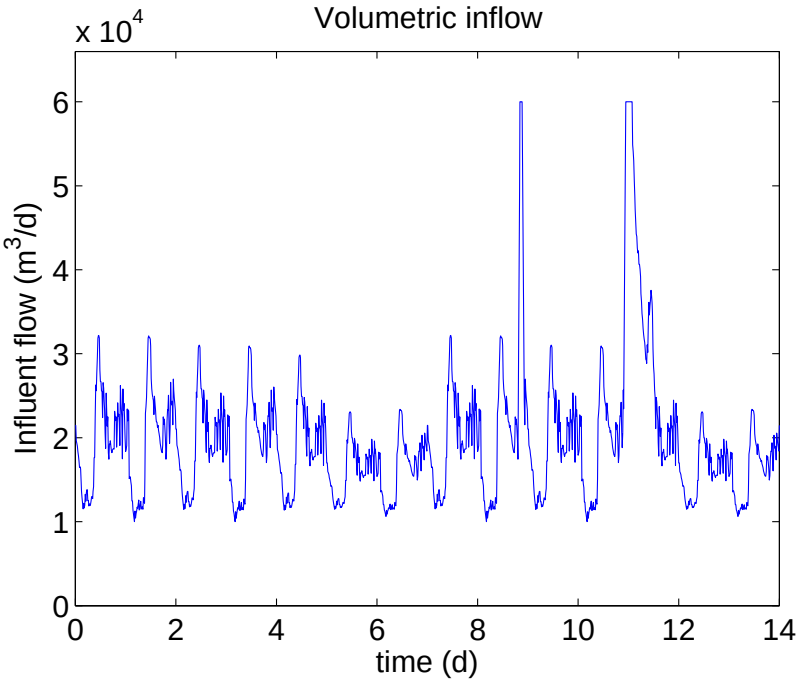

Figure 18: Storm weather influent data.

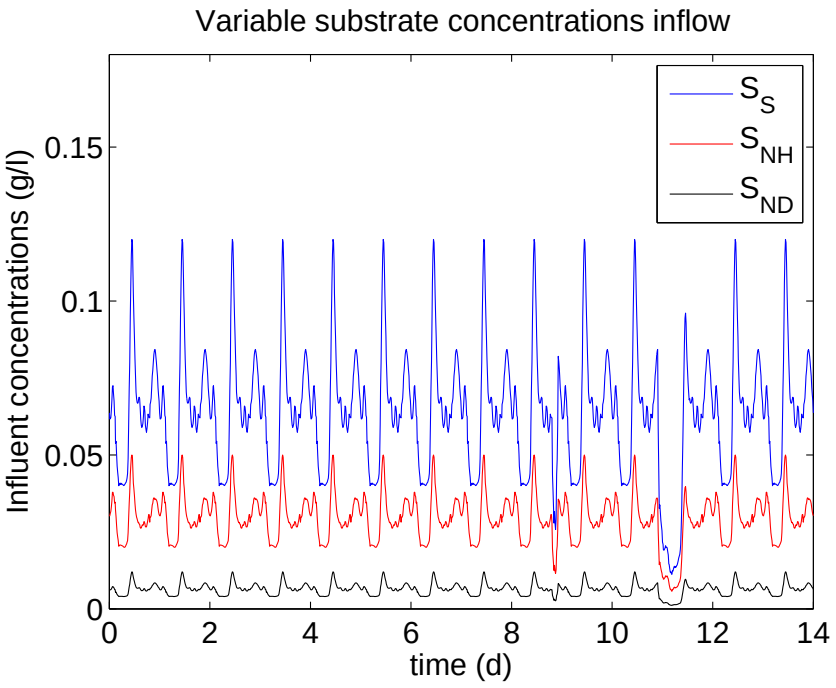

Figure 19: Storm weather influent data.

Parameter	Value
a	0.5 $\mathrm{m}^2/\mathrm{s}^2$
X_c	6 kg/m^3
r	0.35
w	0.0155
α_1	0.0023 m^{-1}
α_2	0.8533 s/m^2
r_V	0.45 m^3/kg

Table 12: List of additional parameter values used in the storm weather scenario.

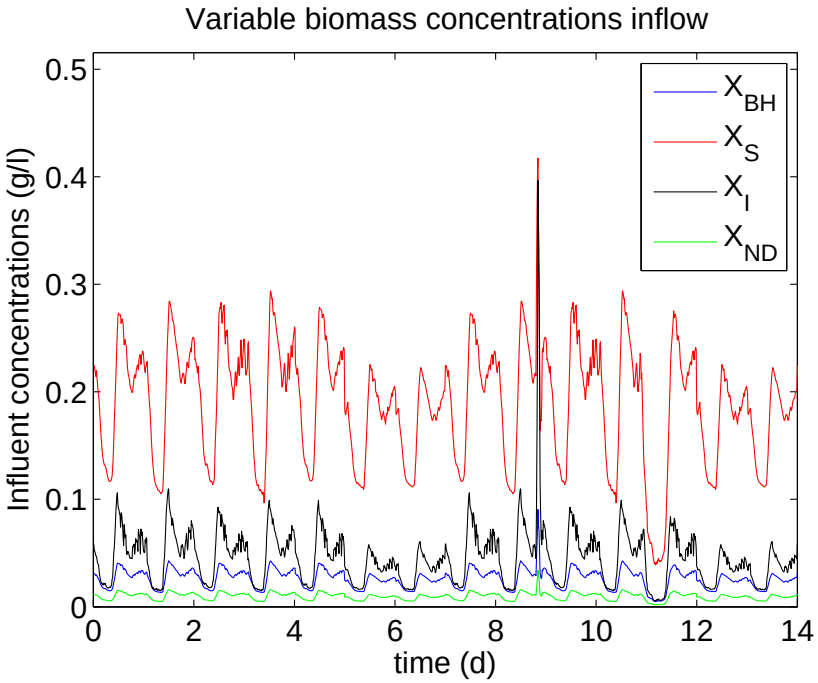

Figure 20: Storm weather influent data.

5.2.2 Storm weather scenario

The storm weather influent data will be used in the simulation. Relevant parameter values are given in Tables 3 and 12. The simulation will take the entire system, with all reactor tanks and all substrate and biomass components, into account.

The system is initialized in a steady state that has been obtained through a simulation with constant influent. Figure 21 shows the simulation of the scenario using the Euler method on a space grid with 810 internal layers. The storm weather event occurs around a simulation time of some 10 days which can clearly be seen as the

settler becomes more overloaded during the event.

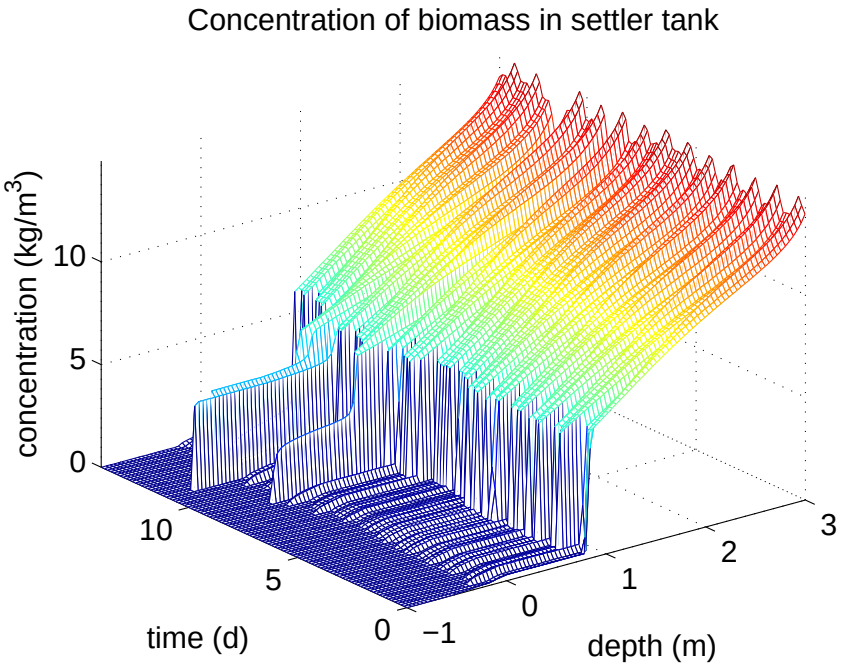

Figure 21: Storm weather scenario. Reference solution obtained using the Euler method with 810 internal layers plotted for 100×100 points.

5.2.3 Efficiency of different solvers

To measure the efficiency of the different solvers using the extended model another error measure is also introduced. This relative error will measure the deviations of the seven substrate component concentrations in the effluent layer, that is layer $j = 0$. This relative error is defined as

$$e_S := \frac{\sum_{k=1}^{7} \int_0^T |S_{0,k}^N(t) - S_{0,k}^{\text{ref}}(t)| \, \mathrm{d}t}{\sum_{k=1}^{7} \int_0^T S_{0,k}^{\text{ref}}(t) \, \mathrm{d}t}. \tag{5.2}$$

The efficiency plots using the two different error measures are shown in Figures 22 and 23 respectively, values are found in Table 13. The results clearly shows that the semi-implicit method performs best with the RK4 method being the worst.

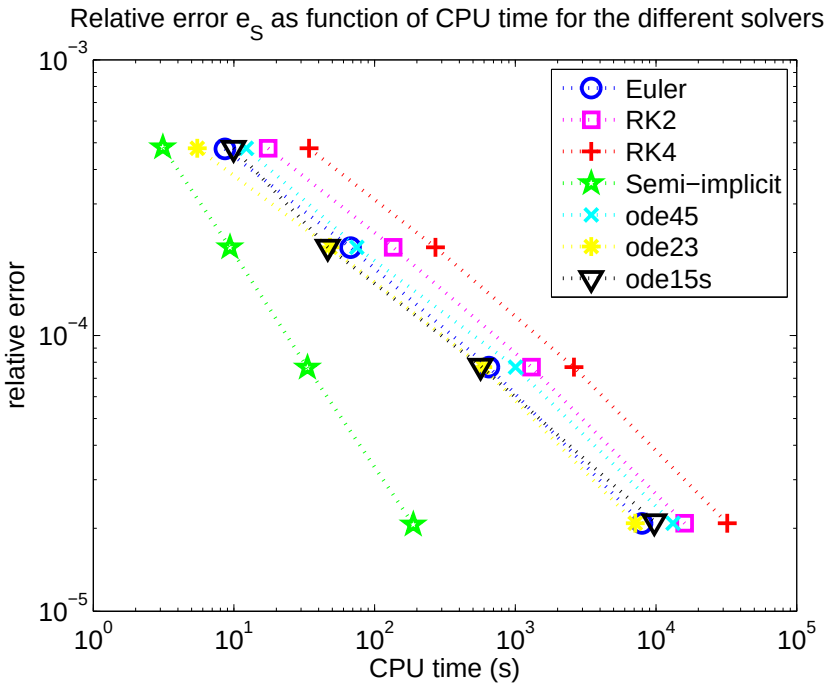

Figure 22: The relative errors, for the different solvers for the substrate effluent concentrations, e_S in the sedimentation tank.

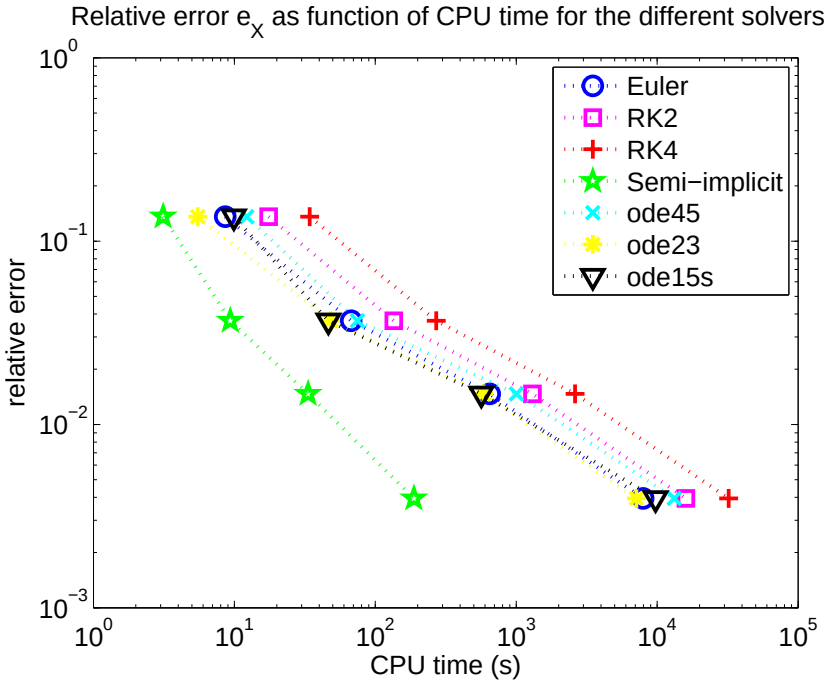

Figure 23: The relative errors, for the different solvers for the biomass concentration, e_X in the sedimentation tank.

Method	N	e_X	e_S	CPU time [s]
Euler	10	0.1360	0.4754E-3	8.6221
	30	0.0368	0.2083E-3	67.3952
	90	0.0147	0.0768E-3	648.7921
	270	0.0039	0.0208E-3	7.95E+3
RK2	10	0.1360	0.4775E-3	17.5132
	30	0.0368	0.2088E-3	135.4051
	90	0.0147	0.0768E-3	1.30E+3
	270	0.0039	0.0208E-3	1.60E+4
RK4	10	0.1360	0.4775E-3	34.2078
	30	0.0368	0.2088E-3	270.8111
	90	0.0147	0.0768E-3	2.61E+3
	270	0.0039	0.0208E-3	3.215E+4
ode45	10	0.1360	0.4775E-3	12.2417
	30	0.0368	0.2088E-3	74.8202
	90	0.0147	0.0768E-3	1.00E+3
	270	0.0039	0.0208E-3	1.32E+4
ode23	10	0.1360	0.4775E-3	5.4927
	30	0.0368	0.2088E-3	48.1658
	90	0.0147	0.0768E-3	586.8117
	270	0.0039	0.0208E-3	7.12E+3
ode15s	10	0.1359	0.4781E-3	9.9340
	30	0.0368	0.2095E-3	46.4258
	90	0.0147	0.0775E-3	565.9028
	270	0.0039	0.0212E-3	9.73E+3
Semi-implicit (Thomas)	10	0.1362	0.4816E-3	3.1209
	30	0.0368	0.2090E-3	9.3616
	90	0.0147	0.0765E-3	33.3418
	270	0.0039	0.0206E-3	188.2093

Table 13: Errors and CPU times in the storm weather scenario for the different methods.

6 Conclusions and summary

The aim of the thesis was to implement two different models of a wastewater treatment plant and investigate which time discretization schemes are most efficient when running simulations. The speed of the simulations is limited by a CFL condition, which if violated may cause numerical instability. It has been investigated how effective different methods have performed during simulations of some different scenarios.

The results show that there is no point in using higher order methods with fixed time step size such as RK2 and RK4. These methods are slower since they can cannot take large enough time steps to make up for the extra computations needed at every time step. They are also slightly more inaccurate since they tend to smoothen out shock waves more than the Euler method. The efficiency of the built-in solvers in MATLAB, ode45, ode23 and ode15s, is scenario dependent. One solver may perform very well for a scenario but far worse for another one. The simplest method of them all, the explicit Euler method performs surprisingly well and is always a better choice than the RK2- and RK4-methods. In many simulations it was also better than all the MATLAB built-in solvers. The best method of them all seems to be the semi-implicit method. Since the CFL condition of this method does not contain any second-order derivative terms it is allowed to take much larger time steps. The cost is that on each time step one has to solve a nonlinear system of equations. This is done with the Newton-Rhapson method which converges very fast and thus this cost is rather small in comparison to the gain.

For the wastewater treatment sector these results may be of interest. The results indicate that simulations are preferably performed using the semi-implicit method described in this thesis. For simulations where dispersion and compression effects are absent and the semi-implicit method is of no use, the Euler method is probably the best choice. The RK2- and RK4-methods are always worse than these two and can be rejected while the adaptive methods in MATLAB may be a good choice depending on the scenario.

References

Alex, J., Benedetti, L., Copp, J., Gernaey, K. V., Jeppsson, U., Nopens, I., Pons, M.-N., Steyer, J.-P., and Vanrolleghem, P. (2008). Benchmark simulation model no. 1 (BSM1). Technical report, Dept. of Industrial Electrical Engineering and Automation Lund University.

Bürger, R., Coronel, A., and Sepulveda, M. (2005). A semi-implicit monotone difference scheme for an initial-boundary value problem of a strongly degenerate parabolic equation modeling sedimentation-consolidation processes. *Mathematics of Computation*, 75(253):91–112.

Bürger, R., Diehl, S., Farås, S., and Nopens, I. (2012a). On reliable and unreliable numerical methods for the simulation of secondary setlling tanks in wastewater treatment. *Computers Chem. Eng.*, 41.

Bürger, R., Diehl, S., Farås, S., Nopens, I., and Torfs, E. (2012b). A consistent modelling methodology for secondary setlling tanks: A reliable numerical method.

Conte, S. D. and de Boor, C. (1972). *Elementary Numerical Analysis: An Algorithmic Approach*. McGraw-Hill.

Copp, J. (2002). *The COST Simulation Benchmark: Description and Simulator Manual*. Directorate-General for Research.

Diehl, S. (1996). Introduction to the scalar non-linear conservation law. Technical report, Department of Mathematics Lund Institute of Technology.

Diehl, S. (2012). Shock-wave behaviour of sedimentation in wastewater treatment: A rich problem. In *Analysis for Science, Engineering and Beyond, Springer Proceedings in Mathematics 6*, chapter 7, pages 175–214. Springer-Verlag Berlin Heidelberg.

DuChateau, P. and Zachmann, D. W. (1989). *Applied Partial Differential Equations*. Harper & Row.

Edsberg, L. (2008). *Introduction to Computation and Modelling for Differential Equations*. John Wiley & Sons, Inc. Hoboken, New Jersey.

Thomas, L. H. (1949). Elliptic problems in linear differential equations over a network. Technical report, Watson Sci. Comput.

Vesilind, P. A. (1968). Theoretical considerations: Design of prototype thickeners from batch settling tests.

Master's Theses in Mathematical Sciences 2013:E62
ISSN 1404-6342

LUTFMA-3255-2013

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/