On time discretizations for the simulation of the batch settling–compression process in one dimension

Raimund Bürger, Stefan Diehl and Camilo Mejías

ABSTRACT

The main purpose of the recently introduced Bürger–Diehl simulation model for secondary settling tanks was to resolve spatial discretization problems when both hindered settling and the phenomena of compression and dispersion are included. Straightforward time integration unfortunately means long computational times. The next step in the development is to introduce and investigate time-integration methods for more efficient simulations, but where other aspects such as implementation complexity and robustness are equally considered. This is done for batch settling simulations. The key findings are partly a new time-discretization method and partly its comparison with other specially tailored and standard methods. Several advantages and disadvantages for each method are given. One conclusion is that the new linearly implicit method is easier to implement than another one (semi-implicit method), but less efficient based on two types of batch sedimentation tests.

Key words | benchmark simulation, Bürger–Diehl simulation model, clarifier, compressive settling, linearly implicit time integration, numerical efficiency

Raimund Bürger (corresponding author)
Camilo Mejías
CI^{2}MA and Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas,
Universidad de Concepción,
Casilla 160-C,
Concepción,
Chile
E-mail: *rburger@ing-mat.udec.cl*

Stefan Diehl
Centre for Mathematical Sciences,
Lund University,
PO Box 118,
S-221 00 Lund,
Sweden

INTRODUCTION

Benchmark simulations of entire wastewater treatment plants (WWTPs) are today performed with one-dimensional simulation models of the secondary settling tank (Gernaey *et al.* 2014; Li & Stenstrom 2014). In the model by Bürger *et al.* (2011, 2013), sometimes referred to as 'Bürger–Diehl model', the physical phenomena of hindered settling, volumetric bulk flows, compression of the sludge at high concentrations and dispersion of the suspension near the feed inlet can be included in a flexible way. Each phenomenon is associated with a separate constitutive function with its model parameters, and can be activated or deactivated at the user's discretion. The possibility to include sludge compression is particularly important, since this improves the predictive power considerably (De Clercq *et al.* 2008; Ramin *et al.* 2014; Guyonvarch *et al.* 2015; Torfs *et al.* 2015; Van Loosdrecht *et al.* 2015). However, the inclusion of physical phenomena that result in second-order derivative terms in the model partial differential equation (PDE), e.g., compression, means that straightforward time discretization by easy-to-implement, explicit methods such as the explicit Euler method leads to long computational times.

Bürger *et al.* (2013) presented implementation details of a numerical algorithm, which gives reliable simulations with respect to the underlying physical principles and can be obtained with a user-defined accuracy. For long-time simulations of entire WWTPs, it is important to keep discretization errors small. This calls for a fine resolution in both space (many layers in the settler) and time (short time steps), which implies long computational times. Conversely, fast computations obtained with a low resolution in space and time come at the cost of poor accuracy.

In scientific computing, the numerical error is a measure of how close a numerical solution is to the exact solution or reference solution (obtained by a very fine discretization) of the model, i.e., the governing differential equation. The efficiency of a numerical method is assessed by relating the numerical errors to the computational (central processing unit; CPU) times necessary to obtain the numerical solutions for different discretizations. It is then said that one numerical method is more efficient than another if it allows a numerical solution with a determined numerical error to be obtained in less CPU time, or equivalently, a given budget of CPU time allows one to obtain a

doi: 10.2166/wst.2015.572

more accurate numerical solution by the first method than by the second.

The simulation model by Bürger *et al.* (2013) is based on a method-of-lines formulation of the underlying nonlinear PDE. This means a system of time-dependent ordinary differential equations (ODEs), one for each layer of the settler. Simulations of PDEs are stable and reliable if a so-called CFL (Courant–Friedrichs–Lewy) condition is satisfied. This gives a maximal time step Δt for each given layer thickness Δz. If only hindered settling and bulk flows are included, then the CFL condition means that Δt can be chosen proportional to Δz, i.e., Δt–Δz, which results in fast simulations. When compression or dispersion is included and standard ODE solvers used, the CFL condition states that Δt–$(\Delta z)^2$, which means very small Δt when the error should be reduced (small Δz is chosen).

The purpose of this contribution is to investigate different time-integration methods, of which one is new, with respect to efficiency and other aspects, such as implementation complexity. There is a qualitative difference in the numerical treatment depending on whether only hindered settling and bulk flows are considered or, in addition, compression or dispersion are included. For clarity of presentation, we limit ourselves here to batch sedimentation in a vessel with a constant cross-sectional area, and for which the depth z is measured from the suspension surface downwards to the bottom at $z = B$. The model PDE is

$$\frac{\partial C}{\partial t} = -\frac{\partial f(C)}{\partial z} + \frac{\partial}{\partial z}\left(d(C)\frac{\partial C}{\partial z}\right), \quad 0 < z < B, \quad t > 0, \tag{1}$$

where the flux function $f(C) = Cv_{\mathrm{hs}}(C)$ contains the hindered settling velocity function $v_{\mathrm{hs}}(C)$ and the compression function is $d(C) = Kv_{\mathrm{hs}}(C)\sigma_{\mathrm{e}}{}'(C)$, where K is a constant containing the solid and fluid mass densities and the acceleration of gravity, and σ_{e} is the effective solid stress function, which is zero below a critical concentration and increasing above.

The investigated methods are used for the simulation of two different batch settling tests in a vessel with $B = 1$ m. While the present study is limited to batch settling in a column, the method proposed herein should be appropriate to handle secondary settling tanks as well (under suitable modifications that would mainly affect the convective flux, but not the compression term for whose discretization the present method has been tailored). It is therefore essential to demonstrate that the method can robustly handle several scenarios, in particular a scenario that exhibits some spatial variation of the solution. For this reason we consider a conventional Kynch test (KT; Kynch 1952), that is, the settling of an initially homogeneous suspension, and a Diehl test (DT; Diehl 2007), where a body of concentrated suspension is initially located above clear liquid, separated by, for example, a membrane. The DT shows a stronger variation of the solution than the KT. Both tests provide complementary information that can be used to identify large portions of the flux function f (Betancourt *et al.* 2014).

METHODS

As hindered settling velocity function we choose for simplicity $v_{\mathrm{hs}}(C) = v_0 \mathrm{e}^{-rC}$ with $v_0 = 10$ m/h and $r = 0.45$ m^3/kg, and the effective solids stress chosen is

$$\sigma_{\mathrm{e}}(C) = \begin{cases} 0 & \text{for} \quad 0 \le C \le C_{\mathrm{c}}, \\ \alpha(C - C_{\mathrm{c}}) & \text{for} \quad C > C_{\mathrm{c}}, \end{cases}$$

where $\alpha = 0.5$ m^2/s^2 and $C_{\mathrm{c}} = 6$ kg/m^3. To obtain a working numerical method an important preparation is to compute the primitive

$$D(C) := \int_0^C d(s)\,\mathrm{d}s$$

of the compression function $d(C)$. In the present case, with the functions $v_{\mathrm{hs}}(C)$ and $\sigma_{\mathrm{e}}(C)$ chosen herein, we can obtain $D(C)$ in closed algebraic form; if this is not possible (for other choices of these functions), the function $D(C)$ can be obtained numerically (Bürger *et al.* 2013). Now (1) can be written as

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial z}\left(-f(C) + \frac{\partial D(C)}{\partial z}\right).$$

For batch sedimentation, this PDE should be complemented with initial data $C(z, 0) = C_0(z)$ and zero-flux boundary conditions

$$-f(C) + \left.\frac{\partial D(C)}{\partial z}\right|_{z=0} = 0 = -f(C) + \left.\frac{\partial D(C)}{\partial z}\right|_{z=B}. \tag{2}$$

Suitable numerical schemes for the approximate solution of (1) are based on subdividing the depth interval $[0, B]$ into a number N of layers of equal thickness $\Delta z = B/N$. A discretization in space leads to the following method-of-lines formulation (Bürger *et al.* 2013), which is a system of N ODEs:

$$\frac{\mathrm{d}C_1}{\mathrm{d}t} = -\frac{G_{3/2}}{\Delta z} + \frac{J_{3/2}}{\Delta z}, \tag{3}$$

$$\frac{\mathrm{d}C_j}{\mathrm{d}t} = -\frac{G_{j+1/2} - G_{j-1/2}}{\Delta z} + \frac{J_{j+1/2} - J_{j-1/2}}{\Delta z}, \quad j = 2, \ldots, N-1, \tag{4}$$

$$\frac{\mathrm{d}C_N}{\mathrm{d}t} = \frac{G_{N-1/2} - J_{N-1/2}}{\Delta z}, \tag{5}$$

where $C_j(t)$ is the concentration of layer j, $G_{j+1/2}$ denotes the numerical convective flux due to hindered settling between layer j and layer $j+1$, which can be chosen, for instance, as the Godunov flux, and $J_{j+1/2}$ is the compressive numerical flux chosen as

$$J_{j+1/2} = \frac{D(C_{j+1}) - D(C_j)}{\Delta z}.$$

Note that (2) implies that the boundary fluxes $G_{1/2} = J_{1/2} = G_{N+1/2} = J_{N+1/2} = 0$. Implementation details on how to compute these fluxes are provided by Bürger *et al.* (2013). The last term in (4) now becomes the usual second-order difference approximation of $\partial^2 D(C)/\partial z^2$, i.e.,

$$\frac{J_{j+1/2} - J_{j-1/2}}{\Delta z} = \frac{D(C_{j+1}) - 2D(C_j) + D(C_{j-1})}{\Delta z^2}. \tag{6}$$

The method-of-lines formulation (3)–(5) is converted into a fully discrete scheme by time discretization. Although a variety of methods could be used for (3)–(5), there is nothing to gain by applying a standard ODE solver of accuracy higher than first order, e.g., a Runge–Kutta method. On the contrary, this will only cause longer computational times producing the same error as explicit Euler time stepping (Diehl *et al.* 2015).

For the discussion of the various time-stepping methods, which will take the numerical solution from $t = t_n$ to $t_{n+1} = t_n + \Delta t$, it will be important to carefully distinguish between numerical fluxes and concentrations that are evaluated at the old time step t_n and those evaluated at t_{n+1}, which we mark by the respective upper index n and $n+1$. The time-integration methods compared herein are the following, where k_1 and k_2 are constants that depend on the choice of the functions f and d.

The explicit Euler method

The explicit Euler method was used for all simulations in Bürger *et al.* (2011, 2013). The CFL condition of the fully discrete scheme can be captured by $\Delta t \le k_2 \Delta z^2$ when Δz is small. The method is easy to implement since all terms in the right-hand side of (3)–(5) are evaluated at $t = t_n$, and therefore each concentration C_j^{n+1} is an explicit function of the known ones at the previous time t_n. Thus, the fully discrete version of (4) is

$$\frac{C_j^{n+1} - C_j^n}{\Delta t} = -\frac{G_{j+1/2}^n - G_{j-1/2}^n}{\Delta z} + \frac{D(C_{j+1}^n) - 2D(C_j^n) + D(C_{j-1}^n)}{\Delta z^2}, \quad j = 2, \ldots, N-1,$$

with analogous formulas replacing the boundary updates (3) and (5).

The semi-implicit method

The semi-implicit (SI) method (Bürger *et al.* 2006) is described in detail by Diehl *et al.* (2015). The CFL condition is $\Delta t \le k_1 \Delta z$. To advance the solution from t_n to t_{n+1} we must solve the following nonlinear system of algebraic equations, where C_j^{n+1}, $j = 1, \ldots, N$, are the unknowns:

$$\frac{C_j^{n+1} - C_j^n}{\Delta t} = -\frac{G_{j+1/2}^n - G_{j-1/2}^n}{\Delta z} + \frac{D(C_{j+1}^{n+1}) - 2D(C_j^{n+1}) + D(C_{j-1}^{n+1})}{\Delta z^2}, \quad j = 2, \ldots, N-1, \tag{7}$$

supplemented with analogous formulas replacing the boundary updates (3) and (5). These equations are solved iteratively, for example, by the Newton–Raphson method.

The linearly implicit method

The idea of the linearly implicit (LI) method goes back to Berger *et al.* (1979) and is based on first considering the contribution from the compressive flux term (6) from time t_n to t_{n+1}. The purpose of the LI method is to avoid the numerical solution of the nonlinear system of Equation (7). The nonlinearities in (7) are found in the evaluations of the function D. The idea is to replace $D(C_j^n)$ in (7) by ξq_j^n (where ξ is a parameter connected to the convergence of the method) so that

$$\frac{C_j^{n+1}-C_j^n}{\Delta t}=-\frac{G_{j+1/2}^n-G_{j-1/2}^n}{\Delta z}+\xi\frac{q_{j-1}^{n+1}-2q_j^{n+1}+q_{j+1}^{n+1}}{\Delta z^2}, \tag{8}$$

and to find a simple update formula for q_j^n, $j=1,\ldots,N$, to be executed first in each time step. A stable implicit Euler time step implies the formula

$$\frac{q_j^{n+1}-q_j^n}{\Delta t}=\xi\frac{q_{j-1}^{n+1}-2q_j^{n+1}+q_{j+1}^{n+1}}{\Delta z^2}.$$

This means that a linear system of equations should be solved for q_j^{n+1} at the next time step. Then (8) can be updated explicitly to obtain C_j^{n+1}.

To completely describe the LI method, we define

$$\xi:=\gamma\max_{0\le C\le C_{\max}}d(C),$$

where $C_{\max}$ is a (nominal) maximum concentration and $\gamma>1$ is a parameter. This parameter can be chosen at the user's discretion. To provide some guidance we mention that based on theoretical considerations (not detailed here) and numerical evidence (presented below), the closer $\gamma>1$ is chosen to the value 1, the more accurate is the scheme. However, the admissible time step Δt behaves as $\Delta t\to 0$ when $\gamma\to 1$. Consequently, the choice of γ is subject to the competing goals of accuracy (small errors) and speed (short CPU times) of the simulation. A detailed discussion of the optimal choice of γ under these restrictions is provided below.

To advance the solution from t_n to t_{n+1}, one proceeds as follows:

1. For $j=1,\ldots,N$, set $q_j^n=D(C_j^n)/\xi$.
2. Solve the following linear system for $q_1^{n+1},\ldots,q_N^{n+1}$:

$$\begin{aligned}\frac{q_1^{n+1}-q_1^n}{\Delta t}&=-\xi\frac{q_1^{n+1}-q_2^{n+1}}{\Delta z},\\ \frac{q_j^{n+1}-q_j^n}{\Delta t}&=\xi\frac{q_{j-1}^{n+1}-2q_j^{n+1}+q_{j+1}^{n+1}}{\Delta z^2},\quad j=2,\ldots,N-1,\\ \frac{q_N^{n+1}-q_N^n}{\Delta t}&=\xi\frac{q_{N-1}^{n+1}-q_N^{n+1}}{\Delta z}.\end{aligned} \tag{9}$$

3. Calculate $C_1^{n+1},\ldots,C_N^{n+1}$ from

$$\frac{C_1^{n+1}-C_1^n}{\Delta t}=-\frac{G_{3/2}^n}{\Delta z}+\frac{q_1^{n+1}-q_1^n}{\Delta t},$$

$$\frac{C_j^{n+1}-C_j^n}{\Delta t}=-\frac{G_{j+1/2}^n-G_{j-1/2}^n}{\Delta z}+\frac{q_j^{n+1}-q_j^n}{\Delta t},$$
$$j=2,\ldots,N-1,$$

$$\frac{C_N^{n+1}-C_N^n}{\Delta t}=\frac{G_{N-1/2}^n}{\Delta z}+\frac{q_N^{n+1}-q_N^n}{\Delta t}.$$

Note that the linear system (9) can be written as follows, where $\mu:=\Delta t/\Delta z^2$:

$$\begin{bmatrix}1+\xi\mu & -\xi\mu & 0 & \cdots & 0\\ -\xi\mu & 1+2\xi\mu & -\xi\mu & \ddots & \vdots\\ 0 & \ddots & \ddots & \ddots & 0\\ \vdots & \ddots & -\xi\mu & 1+2\xi\mu & -\xi\mu\\ 0 & \cdots & 0 & -\xi\mu & 1+\xi\mu\end{bmatrix}\begin{pmatrix}q_1^{n+1}\\ \vdots\\ q_N^{n+1}\end{pmatrix}=\begin{pmatrix}q_1^n\\ \vdots\\ q_N^n\end{pmatrix},$$

This tridiagonal linear system of equations can easily be solved, for example by the Thomas algorithm (cf., e.g., Diehl *et al.* 2015). A preliminary analysis on the stability of the scheme (not presented here) implies that the CFL condition of the LI scheme is

$$\Delta t\le\frac{\Delta z}{2\max\limits_{0\le C\le C_{\max}}f'(C)}\left(1-\frac{1}{\gamma}\right). \tag{10}$$

Other time-stepping methods

Any ODE solver could be used for the method-of-lines system (3)–(5). Possibly competitive methods are adaptive step-size methods and we choose here the ODE solver ode15s in Matlab (2014), which is an implicit multi-step method of variable order with step-size control. Of several standard ODE solvers investigated by Diehl *et al.* (2015), this was the second most efficient one (after the SI method) in the investigations with both stand-alone settler simulations and benchmark simulations for the entire activated sludge process.

RESULTS AND DISCUSSION

For both the KT and the DT we measure the performance of the numerical methods in terms of numerical error and CPU time. The error of a total simulated solution $C_{\mathrm{tot},N}$ with N layers up to a time T is calculated by comparing with a reference solution C_{ref} obtained by Euler's method and $N = 2430$ layers ($\Delta z = B/N = 1\ \mathrm{m}/2430 \approx 0.41\ \mathrm{mm}$). The relative L^1 error is calculated as

$$E_N = \int_0^T \int_0^B |C_{\mathrm{tot},N}(z,t) - C_{\mathrm{ref}}(z,t)|\,\mathrm{d}z\,\mathrm{d}t \Big/ \int_0^T \int_0^B C_{\mathrm{ref}}(z,t)\,\mathrm{d}z\,\mathrm{d}t.$$

For the KT we choose an initial concentration of $C_0 = 5\,\mathrm{kg/m^3}$ and $T = 1.5\,\mathrm{h}$, which means that almost steady state has been reached. This test is also used to study the dependence of the numerical solutions produced by the LI method on the choice of the parameter γ. For each run, the time step Δt is chosen according to the CFL condition (10) (with equality). Note that the right-hand side of (10) is a decreasing function of γ and it tends to infinity as γ approaches 1 from above. Figure 1 shows numerical solutions obtained for three different values of γ. This figure and Figure 2 indicate that for a given number of layers $N > 30$, the values of γ closest to 1 produce the solutions with the smallest errors, but the highest CPU times. We note that, for instance, for a given value of N the time step for $\gamma = 3$ can be chosen twice larger than that for $\gamma = 1.5$, so the total CPU time for $\gamma = 3$ should be about half of that for $\gamma = 1.5$. To assess which value of γ is optimal, an efficiency plot (of relative L^1 error versus CPU time) is helpful, see Figure 2. The curves are roughly L-shaped, indicating that γ should be chosen close to the point corresponding to the point of 'bend', since smaller or larger values would lead to larger numerical errors (at only slightly smaller CPU times) or to smaller numerical errors at much increased CPU times. Based on these considerations, the choice $\gamma = 3$ seems a good compromise for further comparison with other methods.

For the DT we choose the initial data

$$C(z,0) = \begin{cases} 10\ \mathrm{kg/m^3} & \text{for} \quad 0 < z < 0.4\ \mathrm{m}, \\ 0 & \text{for} \quad 0.4\ \mathrm{m} < z < B = 1\ \mathrm{m}. \end{cases}$$

Figure 3 shows the numerical solution by the LI method with $\gamma = 3$ and a contour plot of the reference solution. The right plot illustrates that in those regions where $C < C_{\mathrm{c}}$, therefore $d = 0$, and (1) reduces to a first-order hyperbolic PDE, iso-concentration curves are straight lines, in complete agreement with the corresponding theory.

Figure 4 shows the efficiency curves for all methods investigated plus those produced by employing Matlab's ode15s solver. It turns out that the latter method is the least efficient, and that for both tests the implicit methods are most efficient, as expected from the corresponding CFL conditions. Moreover, for a given number of layers N, the LI method is the fastest, although not necessarily the one with the smallest error. Note that the efficiency curve for the SI method for the DT (the right plot of Figure 3) is composed of four symbols only. In fact, no information is available for the run with $N = 810$ since the Newton–Raphson iterations did not converge for that case. While this situation could be easily overcome by the ad–hoc remedy of further reducing Δt (say, to 80% of its maximal value determined by the CFL condition), it alerts us to a more fundamental problem observed with the SI method, namely that the convergence of (iterative) solvers for the nonlinear equations is not ensured a priori.

For accurate simulation results, Δz should be chosen small, whereby the CFL condition implies that the explicit

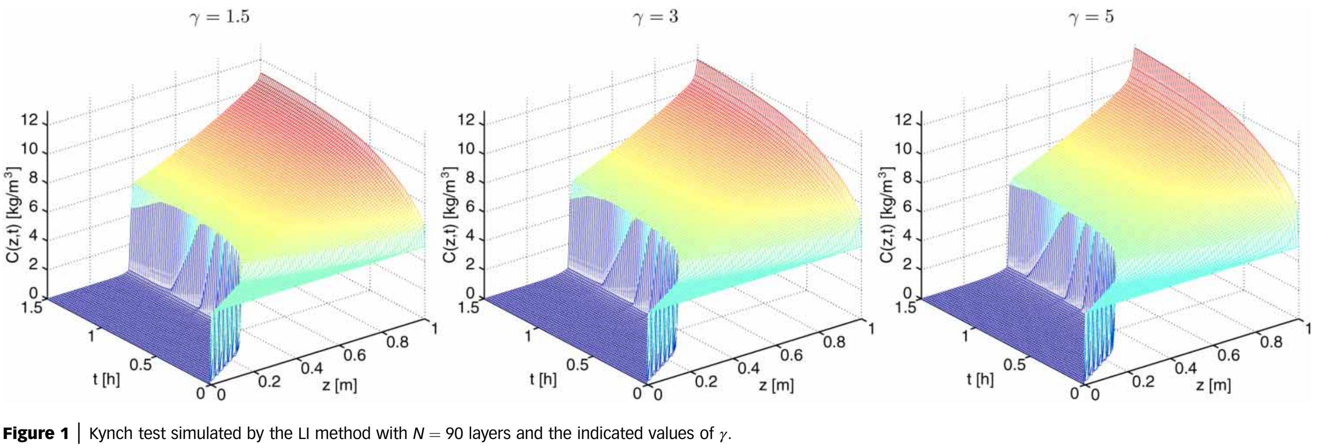

Figure 1 | Kynch test simulated by the LI method with $N = 90$ layers and the indicated values of γ.

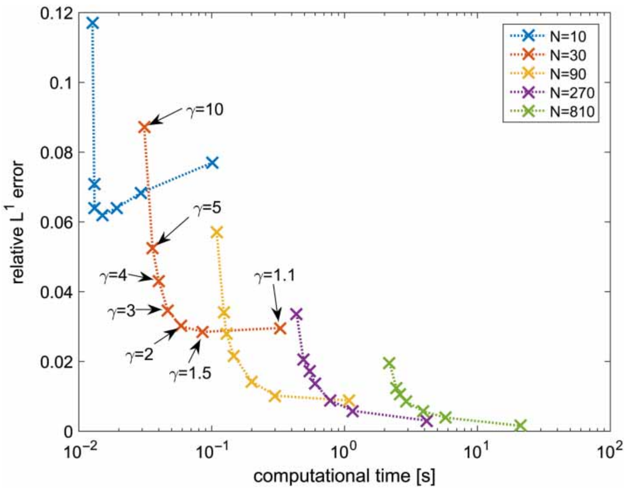

Figure 2 | LI method applied to KT: efficiency plots (relative L^1 error versus CPU time) for the indicated values of N, determined for $\gamma = 10, 5, 4, 3, 2, 1.5, 1.1$ (from top to bottom, the curve for $N = 30$ is labeled for illustration).

Euler method requires much smaller time steps than the other methods. On the other hand, the SI method needs more computations at every time step and requires the evaluation of the Jacobian matrix of the nonlinear algebraic system of equations. However, even when this system is solved by the Newton–Raphson method, the SI method has turned out to be far more efficient than explicit Euler (Diehl *et al.* 2015). To reduce the implementation complexity and to remove the necessity to solve numerically a system of nonlinear algebraic equations at every iteration in the SI method, whose convergence depends on the Newton–Raphson iterations, the LI method can be used instead. The advantage of solving a linear system only in each iteration, which also means an easier implementation, is paid fort by the price of a larger numerical error.

CONCLUSIONS

With the aim of finding an efficient time-integration method for the simulation of sedimentation with compression, we have in this work confined ourselves to two batch sedimentation tests for the comparison of some methods. The ideas of the new LI method and its implementation have been provided. When evaluated on the batch settling tests, the following advantages (+) and disadvantages (−) can be concluded.

- Explicit Euler method:
  + Implementation easy.
  + Convergence proof exists also for continuous sedimentation (numerical approximate solutions converge to the PDE solution as $\Delta z \to 0$) by Bürger *et al.* (2005). Robust method.
  − Not efficient.
- SI method:
  + Most efficient of the investigated methods. Under the assumption that the Newton–Raphson iterations find a solution each time step, convergence of the numerical method was proved for batch sedimentation by Bürger *et al.* (2006).
  − Implementation more complex than Euler and LI. At each time step, a nonlinear system of algebraic equations is solved, e.g., by Newton–Raphson

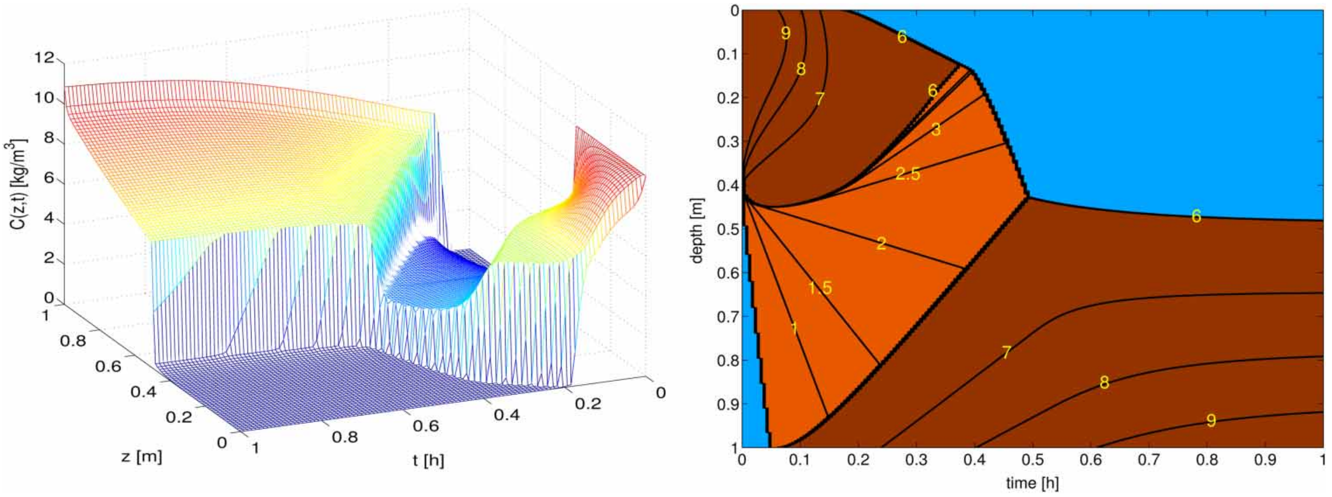

Figure 3 | Numerical solution of the DT. Left: solution by the LI method with $\gamma = 3$ and $N = 90$. Right: contour plot of the reference solution, showing areas of $C = 0$ (light grey; blue in online version), $0 < C < C_c$ (medium grey; light brown) and $C > C_c$ (dark grey; dark brown) and iso-concentration lines corresponding to the annotated values of C [kg/m^3]. The full colour version of this figure is available online.

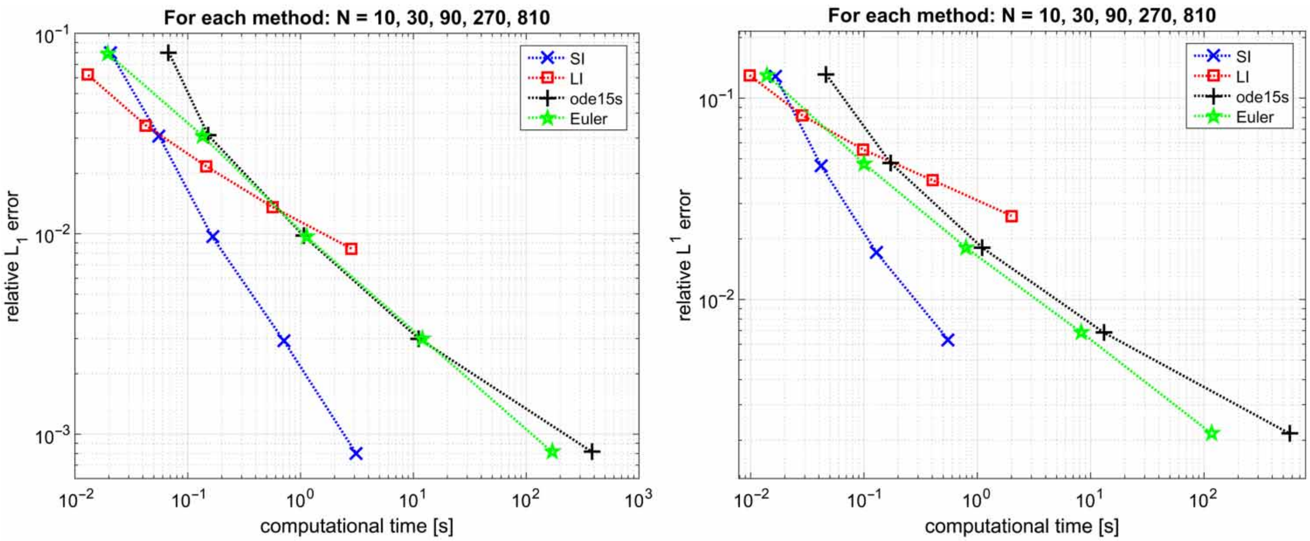

Figure 4 | Efficiency plots for the KT (left) and the DT (right) and the Euler, SI and LI methods ($\gamma = 3$) and Matlab's ode15s solver. In the right plot the cross corresponding to $N = 810$ for the SI method is not shown since the Newton–Raphson iterations did not converge, i.e., the method failed.

iterations, which require tolerance parameter to be set. There is no guarantee that these iterations converge.

- LI method:
  + Implementation easy. The ingredient in addition to the Euler method is basically that a linear system of equations is solved at each time step. Robust method.
  ± Second most efficient for $N \approx 100$ for batch sedimentation. The efficiency can be adjusted to some extent by a parameter. Fastest method for a given $N \geq 30$, but least accurate. Convergence proof in preparation.
- Matlab's ode15s:
  + Ready-to-use standard time solver.
  – Implementation most complex of the investigated methods. Well-established robust ODE solver for stiff problems, but not developed for solutions containing discontinuities.
  – Least efficient of the investigated methods.

ACKNOWLEDGEMENTS

RB is supported by Fondecyt project 1130154; Conicyt project Anillo ACT1118 (ANANUM); Red Doctoral REDOC.CTA, MINEDUC project UCO1202; BASAL project CMM, Universidad de Chile and CI^{2}MA, Universidad de Concepción; and Fondef ID15I10291. RB and CM are also supported by CRHIAM, Proyecto Conicyt Fondap 15130015. SD acknowledges support from The ÅForsk Foundation, Sweden.

REFERENCES

Berger, A. E., Brézis, H. & Rogers, J. C. W. 1979 A numerical method for solving the problem $u_t - \Delta f(u) = 0$. *RAIRO Anal. Numér.* **13**, 297–312.

Betancourt, F., Bürger, R., Diehl, S. & Mejas, C. 2014 Advanced methods of flux identification for clarifier-thickener simulation models. *Minerals Eng.* **63**, 2–15.

Bürger, R., Karlsen, K. H. & Towers, J. D. 2005 A model of continuous sedimentation of flocculated suspensions in clarifier-thickener units. *SIAM J. Appl. Math.* **65**, 882–940.

Bürger, R., Coronel, A. & Sepúlveda, M. 2006 A semi-implicit monotone difference scheme for an initial-boundary value problem of a strongly degenerate parabolic equation modeling sedimentation-consolidation processes. *Math. Comp.* **75**, 91–112.

Bürger, R., Diehl, S. & Nopens, I. 2011 A consistent modeling methodology for secondary settling tanks in wastewater treatment. *Water Res.* **45**, 2247–2260.

Bürger, R., Diehl, S., Farås, S., Nopens, I. & Torfs, E. 2013 A consistent modelling methodology for secondary settling tanks: a reliable numerical method. *Water Sci. Technol.* **68**, 192–208.

De Clercq, J., Nopens, I., Defrancq, J. & Vanrolleghem, P. A. 2008 Extending and calibrating a mechanistic hindered and compression settling model for activated sludge using indepth batch experiments. *Water Res.* **42**, 781–791.

Diehl, S. 2007 Estimation of the batch-settling flux function for an ideal suspension from only two experiments. *Chem. Eng. Sci.* **62**, 4589–4601.

Diehl, S., Farås, S. & Mauritsson, G. 2015 Fast reliable simulations of secondary settling tanks in wastewater treatment with semi-implicit time discretization. *Comput. Math. Applic.* **70**, 459–477.

Gernaey, K. V., Jeppsson, U., Vanrolleghem, P. A. & Copp, J. B. 2014 *Benchmarking of Control Strategies for Wastewater Treatment Plants*. IWA Scientific and Technical Report No. 23. IWA Publishing, London, UK.

Guyonvarch, E., Ramin, E., Kulahci, M. & Plósz, B. G. 2015 iCFD: interpreted computational fluid dynamics–Degeneration of CFD to one-dimensional advection-dispersion models using statistical experimental design–The secondary clarifier. *Water Res.* **83**, 396–411.

Kynch, G. J. 1952 A theory of sedimentation. *Trans. Farad. Soc.* **48**, 166–176.

Li, B. & Stenstrom, M. K. 2014 Research advances and challenges in one-dimensional modeling of secondary settling tanks–a critical review. *Water Res.* **65**, 40–63.

MATLAB 2014 Release 2014b. The MathWorks, Inc., Natick, MA, USA.

Ramin, E., Wágner, D. S., Yde, L., Binning, P. J., Rasmussen, M. R., Mikkelsen, P. S. & Plósz, B. G. 2014 A new settling velocity model to describe secondary sedimentation. *Water Res.* **66**, 447–458.

Torfs, E., Maere, T., Bürger, R., Diehl, S. & Nopens, I. 2015 Impact on sludge inventory and control strategies using the benchmark simulation model no. 1 with the Bürger-Diehl settler model. *Water Sci. Technol.* **71**, 1524–1535.

Van Loosdrecht, M. C. M., Lopez-Vazquez, C. M., Meijer, S. C. F., Hooijmans, C. M. & Brdjanovic, D. 2015 Twenty-five years of ASM1: past, present and future of wastewater treatment modelling. *J. Hydroinformatics* **17**, 697–718.

First received 24 June 2015; accepted in revised form 27 October 2015. Available online 7 November 2015