# Advantages of concurrent use of multiple software frameworks in water quality modelling using a database approach

Luuk P.A. van Gerven[1,2 #], Robert J. Brederveld[3], Jeroen J.M. de Klein[2], Don L. DeAngelis[4], Andrea S. Downing[5], Michiel Faber[3], Daan J. Gerla[6], Jochem 't Hoen[2], Jan H. Janse[1,7], Annette B.G. Janssen[1,2], Michel Jeuken[8], Bob W. Kooi[9], Jan J. Kuiper[1,2], Betty Lischke[10], Sien Liu[8], Thomas Petzoldt[11], Sebastiaan A. Schep[3], Sven Teurlincx[1], Christophe Thiange[8], Dennis Trolle[12], Egbert H. van Nes[2], Wolf M. Mooij[1,2]

*[1] Department of Aquatic Ecology, Netherlands Institute of Ecology (NIOO-KNAW), P.O. Box 50 NL-6700 AB, Wageningen, The Netherlands*
*[2] Aquatic Ecology and Water Quality Management Group, Department of Environmental Sciences, Wageningen University, P.O. Box 47, NL-6700 AA Wageningen, The Netherlands*
*[3] Witteveen+Bos, P.O. Box 233, 7400 AV Deventer, The Netherlands*
*[4] USGS/Biological Resources Division and Department of Biology, University of Miami, PO Box 249118, Coral Gables, FL 33124, USA*
*[5] Department Systems Ecology, Stockholm University, SE-10691, Stockholm, Sweden*
*[6] Royal Netherlands Institute for Sea Research (NIOZ), Department of Ecosystem Studies, P.O. Box 140, 4400 AC Yerseke The Netherlands*
*[7] PBL, Netherlands Environmental Assessment Agency, P.O. Box 303, 3720 AH Bilthoven, The Netherlands*
*[8] Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands*
*[9] Faculty of Earth and Life Sciences, Department of Theoretical Biology, Vrije Universiteit, De Boelelaan 1085, 1081 HV Amsterdam, The Netherlands*
*[10] Institute of Biochemistry and Biology, Department of Ecology and Ecosystem Modelling, University of Potsdam, Am Neuen Palais 10, 14469 Potsdam, Germany*
*[11] Faculty of Environmental Sciences Institute of Hydrobiology, Technische Universität Dresden, 01062 Dresden, Germany*
*[12] Department of Bioscience, Aarhus University, Vejlsøvej 25, 8600 Silkeborg, Denmark*

[#]email corresponding author: l.vangerven@nioo.knaw.nl

Manuscript for publication in the Fundamental and Applied Limnology Thematic Issue on "Modelling in Freshwater Ecosystems".

**Abstract**

Water quality modelling deals with multidisciplinary questions ranging from fundamental to applied. Addressing this broad range of questions requires multiple analysis techniques and therefore multiple frameworks. Through the recently developed database approach to modelling (DATM), it has become possible to run a model in multiple software frameworks without much overhead. Here we apply DATM to the ecosystem model for ditches PCDitch and its twin model for shallow lakes PCLake. Using DATM, we run these models in six frameworks (ACSL, DELWAQ, DUFLOW, GRIND for MATLAB, OSIRIS and R), and report on the possible model analyses with tools provided by each framework. We conclude that the dynamic link between frameworks and models resulting from DATM has the following main advantages: it allows one to use the framework one is familiar with for most model analyses and eases switching between frameworks for complementary model analyses, including the switch between a 0-D and 1-D to 3-D setting. Moreover, the strength of each framework – including runtime performance – can now be easily exploited. We envision that a community-based further development of the concept can contribute to the future development of water quality modelling, not only by addressing multidisciplinary questions but also by facilitating the exchange of models and process formulations within the community of water quality modellers.

Keywords: Database Approach To Modelling, DATM, PCLake, PCDitch, OSIRIS, ACSL, R, GRIND, DUFLOW, DELWAQ, Modelling Framework, Model Implementation, Model Analysis, Differential Equations, Community-based Modelling

## Introduction

Water quality modelling often deals with multidisciplinary issues ranging from fundamental questions aiming at a more thorough understanding of theoretical principles to applied questions like the scenario-wise evaluation of potential measures for ecosystem management. This diversity in questions requires a multitude of model analysis techniques and therefore a multitude of software frameworks, as there is no single framework that captures all these techniques. Ideally, one would like to easily implement a model in a framework of choice and easily switch between existing frameworks to exploit the myriad of available analysis techniques.

The number of software frameworks that is available to implement water quality models is large and still increasing (Argent 2004). This makes it nearly impossible to have an overview of existing frameworks and their capabilities. As a result, experienced users stick to the framework they have invested in, instead of exploiting the rich array of choices that exists. At the same time, new users choose the framework they have easiest access to, and for which they can get support from experienced users in their direct vicinity.

Switching between frameworks currently takes considerable effort as models are often locked in a single framework, in that they are written in framework-specific code and can only be accessed through framework-specific user interfaces (David et al. 2013). This phenomenon is also referred to as framework invasiveness (Lloyd et al. 2011). This harbours the risk that framework familiarity tends to define which model to use, instead of the ecological question that needs to be answered (Argent 2004). The observed multitude of frameworks and their locked-in models leads us to conclude that the landscape of water quality modelling is highly fragmented. This fragmentation often leads to a 'reinvention of the wheel' and 'tunnel-vision' in water quality modelling, as there is no healthy cross-fertilization of ideas between models and frameworks (Mooij et al. 2010).

A database approach to modelling (DATM) was recently proposed to address this challenge (Mooij et al. 2014). In this approach, the knowledge incorporated in a model is stored in a database, independently of program language and framework. In order to run the model in a certain framework, the information in the database is translated and augmented with language and framework specifics. This process is automated so that the model can easily be re-implemented in the framework after it has been modified in the database. Thus, with DATM it becomes easy to switch between multiple frameworks and exploit their joint multitude of model analysis techniques to address the multidisciplinary questions such as encountered in water quality modelling.

Here we apply DATM to analyse the ecosystem model for ditches PCDitch (Janse 1998) and its twin model for shallow lakes PCLake (Janse 1997) in six different software frameworks, including non-spatial and 1-D to 3-D implementations of the models. After determining the runtime of the models in the different frameworks, we analysed both models and report on the used framework tools for sensitivity analysis, calibration, validation, uncertainty analysis, bifurcation analysis and scenario analysis. We discuss the benefits and potential pitfalls of using DATM in water quality modelling with respect to exploiting the complementarity and redundancy among frameworks. Additionally, we discuss the possibilities it creates for model and framework review. Because DATM relies on mathematics, we conclude that it can effectively be used for a much wider range of models and frameworks than studied here, and may contribute to the future development of water quality modelling.

## Methods

### *Framework-implementation of the models*

We used DATM to implement PCDitch and PCLake in six different frameworks. In this process, the ordinary differential equations (ODE's) of the models are translated into framework-specific code (Mooij et al. 2014). The complete implementation process – from building the translators to performing test runs – is described in detail in Appendix 1.

Using DATM, ODE-based models – describing the change of a state variable in time – can also be implemented in a spatial setting. For example, we implemented PCDitch and PCLake in the frameworks DUFLOW and DELWAQ, which are suited for 1-D to 3-D water quality modelling (see next paragraph). Then the ODE's are embedded in partial differential equations (PDE's) describing the change of a state variable in time and space. Also lattice differential equations (LDE's) can be used for spatial modelling, and are supported as well by DATM. PDE's have a continuous spatial structure, whereas LDE's are discrete in space. This discretization causes additional dynamics compared to PDE's (Chow et al. 1996). To avoid these effects of discretization, mostly PDE's are used in spatial water quality modelling.

To illustrate how a model – consisting of processes described by ODE's – can be embedded in LDE's and PDE's, consider an ODE describing how the change of substance *C* with time depends on an inflow (with concentration $C_{in}$), an outflow and the model term $f_R(C,t)$ describing the model processes for substance *C*:

$$\frac{dC}{dt} = a(C_{in} - C) + f_R(C,t) \, , \tag{1}$$

where *a* is the dilution rate, which is the inverse of the water retention time. A spatial dimension can be added by regarding spatially connected compartments, which can be described by LDE's. For a chain of compartments where compartment *i* receives water from the upstream compartment *i-1* this results in

$$\frac{dC_i}{dt} = a(C_{i-1} - C_i) + f_R(C_i,t) + d(C_{i-1} - 2C_i + C_{i+1}) \, , \tag{2}$$

where dispersion of substances in space – with dispersion rate *d* – comes into play. Note that each compartment is a perfectly-mixed discrete spatial unit. For a continuous spatial structure, the change of substance *C* with time at a certain location can be described by a PDE:

$$\frac{\partial C}{\partial t} = -\nabla \cdot (\vec{u}C) + f_R(C,t) + D\nabla^2 C \, , \tag{3}$$

where the first term describes the transport of *C* by flow (advection) related to flow velocity $\vec{u} = (u_x, u_y, u_z)$ and the third term describes the transport by dispersion, where *D* is the dispersion coefficient.

In each case, DATM provides a framework independent description of the process terms $f_R(C,t)$ and merges these with the framework specific hydrodynamic terms in a format prescribed by the framework.

***Frameworks***

We used the frameworks ACSL, GRIND for MATLAB, OSIRIS and R (here all used for 0-D modelling), and DELWAQ and DUFLOW (used for 1-D to 3-D modelling). These frameworks were chosen for their capabilities but also for practical reasons, such as the availability of the framework and the experience of one or more of the authors with a given framework. We summarise technical details in Table 1, such as the programming language – which is important for the runtime performance –, the user-interface and the licensing policy.

*ACSL*

The Advanced Continuous Simulation Language (ACSL) is among the first modelling frameworks used to simulate continuous systems of time-dependent nonlinear differential equations (Mitchell & Gauthier 1976). It is an equation-oriented language developed to represent mathematical models in an easily readable way. ACSL includes a MACRO capability to duplicate (sets of) states. In ecosystem modelling, this characteristic can be used to implement species within functional groups (for example see Janse 2005).

*GRIND for MATLAB*

GRIND for MATLAB (hereafter referred to as GRIND and based on the C program Great Integrator Differential equations (De Boer & Pagie 1983)) is a modelling framework used to analyse time-dependent differential equations and difference equations as well as matrix and vector models (http://www.sparcs-center.org/grind). GRIND is developed for theoretical ecology and features phase-plane and bifurcation analyses. It is mainly used to analyse simple models with a few equations, and for teaching purposes.

*OSIRIS*

The Object-oriented Simulation Framework for Individual-based Simulations (OSIRIS) is a modelling framework that was originally developed for the implementation of event-driven spatially-explicit individual-based models (Mooij & Boersma 1996). It was extended, however, to implement models of differential equations. A particular feature of OSIRIS is that the input and output files are structured in a database. DATM can be seen as an extension of this design.

*R*

R is a programming language and environment developed for statistical computing and graphics (R Core Team 2013). It attracted the attention of the scientific community and has gained much popularity in recent years, as a result of its open licensing under the GNU General Public License and well documented package system (Fox 2009). This has promoted the community-based development of more than five thousand add-on packages. One of these packages, deSolve, allows for a simple implementation of dynamic models based on differential equations (Soetaert et al. 2010).

*DELWAQ*

DELWAQ (Delft Water Quality) is a water quality module embedded in the hydrodynamic framework Delft-3D or SOBEK (Delft Hydraulics 1995, Deltares 2013). It is used for 1-D, 2-D and 3-D water quality modelling in seas, estuaries, streams, ditches and lakes. Given externally calculated hydrodynamics, DELWAQ simulates the transport of substances and sediment in a user-defined spatial configuration. Built-in water quality processes can be switched on or off by the user. The user can install additional processes, thus allowing to link DELWAQ with models such as PCLake.

*DUFLOW*

The DUFLOW (Dutch Flow model) water quality modelling framework was originally developed for and used by Dutch Water Boards for simulating 1-D unsteady flow in streams and ditches (Clemmens et al. 1993, Spaans et al. 1989). Later, it was extended to simulate water quality in 1-D. Water quality models can be implemented in Duflow as a set of differential equations. The process equations are evaluated for each hydrological unit when simulating the transport of substances in a user-defined spatial configuration.

***Models***

*PCDitch*

PCDitch describes ditch ecosystems (Janse 1998). It covers six functional groups of macrophytes – essential for the ecological functioning of ditches (Portielje & Roijackers 1995) – and one group of algae (Fig. 1). These groups compete for nutrients and light, each with a different competition strategy as defined by their growth form. PCDitch describes the cycling of dry weight and nutrients (N and P) for all model compartments, in both the upper sediment layer and the water column. PCDitch is mainly used to study the critical nutrient loading at which ditches become dominated by free-floating plants instead of submerged plants (van Liere et al. 2007). This critical loading is relevant for management because dense mats of free-floating plants form a threat to biodiversity (Scheffer et al. 2003).

*PCLake*

PCLake describes shallow lake ecosystems (Janse 1997). It is a food web model that covers the interaction between different trophic levels (fish, zooplankton and primary producers) within an ecosystem context. Similarly to PCDitch, it includes the water column and the upper sediment layer and describes the cycling of dry weight and nutrients (N and P) over the different model components (Fig. 1). PCLake is primarily used to define critical nutrient loadings at which a vegetation-dominated clear-state turns into a phytoplankton-dominated turbid-state or vice versa. Critical transitions to and from a turbid state often occur at different nutrient loadings, implying alternative stable states. PCLake has proved successful in answering both fundamental and applied questions and therefore bridges the gap between these two (Mooij et al. 2010).

## Results

### ***Runtime performance***

Before analysing the models in the different frameworks, we first determined the runtime of the models per framework-implementation. Not surprisingly, the runtime differed considerably between frameworks (Table 2). The runtime of a 50-year PCDitch and PCLake run (default settings) was of the order of *seconds* using OSIRIS and ACSL, of the order of *minutes* using DUFLOW and DELWAQ and of the order of *hours* using GRIND and R. The latter frameworks use interpreted languages, which explains the long runtimes. However, in both GRIND and R, the models could be compiled using C++ and then called from the framework. This reduced the runtime considerably − up to 5000 times − leading to runtimes similar to OSIRIS and ACSL. The longer runtimes of the spatially explicit frameworks DUFLOW and DELWAQ were – expectedly – due to the fact that they solved PDE’s instead of ODE’s and used integrators with a fixed time step, whereas the other frameworks used integrators with a variable time step (see Table A1 in Appendix 1).

### ***Analysis***

The multi-framework implementation of PCLake and PCDitch allowed us to have a great amount of choice and versatility in the methodologies for analysis, fully validating the purpose of DATM. We carried out sensitivity analyses, calibration/validation/uncertainty analyses and bifurcation analyses using built-in tools (GRIND and OSIRIS) and by writing our own scripts (ACSL and R). Furthermore, we used DELWAQ and DUFLOW for spatial scenario analyses. Table 3 gives an overview of the published and unpublished analyses on PCLake and PCDitch that we are aware of, classified per analysis and framework. Note that the analyses that date back to before 2012 were not carried out with DATM-implementations of the models, but resulted from non-automated framework-to-framework translation of the models. We report on these pre-DATM analyses, as they illustrate the tools provided by the different frameworks for model analysis. They also illustrate the need to run a model in multiple frameworks to address multidisciplinary questions. This aspect, together with the realization of how much effort it took to translate a model from one framework to another, led, in fact, to the development of DATM. In addition to reporting on the published and unpublished analyses (Table 3) – where the figures show only analyses resulting from DATM-implementations – we point to powerful framework tools that come within reach through using DATM.

*Sensitivity analysis*

A sensitivity analysis quantifies how changes in the model input (i.e., parameters, initial states or external inputs) affect the model outcome (Klepper 1997). We used sensitivity analyses to identify the most sensitive parameters. This gives insight into model behaviour and can be used to select parameters for calibration.

With PCLake, ACSL with SIMLAB (EC-JRC-ISIS 2002) was used to carry out a stepwise sensitivity analysis (Janse et al. 2010) by (1) screening the parameters to select a subset of most sensitive ones using the Morris method (Morris 1991) and (2) performing a global sensitivity analysis on this subset using the FAST (‘Fourier Amplitude Sensitivity Test’) method (Saltelli et al. 2008). With PCDitch, we used GRIND to calculate sensitivity indices for the parameters by Monte Carlo sampling followed by a regression. This information was used to cluster parameters with a similar or opposite effect on the model outcome (Fig. 2), showing that the parameters of a certain functional group of water plants in PCDitch are closely linked. ACSL, GRIND and OSIRIS offer basic One-At-a-Time (OAT) sensitivity analysis tools (Saltelli et al. 2008). These tools were not used, because they are less suited for non-linear models like PCDitch and PCLake than the tools described above.

R offers a variety of sensitivity tools. For example the Flexible Modelling Environment (FME) package (Soetaert & Petzoldt 2010) contains functions for global and local sensitivity analyses. FME can also evaluate the identifiability of parameter sets, which is useful for over-parameterized models like PCLake and PCDitch (Mieleitner & Reichert 2006). Non-identifiability occurs commonly in PCLake and PCDitch, as a change in one parameter can often be compensated by changing other parameters (Janse et al. 2010).

*Calibration*

Calibration aims at improving the fit between a model and measured data. Various optimization techniques exist that randomly or actively search parameter space for the best fit. The fit is usually measured through the root mean squared error (RMSE) or mean relative error (MRE) (e.g. Trolle et al. 2014). Calibration can also be regarded as a way to find reasonable values for poorly defined or unmeasurable parameters (inverse modelling), a situation that is common for most environmental models (van Oevelen et al. 2010).

Using ACSL with SIMLAB, PCDitch was calibrated against experimental ditches by Simulated Annealing (van Laarhoven & Aarts 1987) and PCLake was calibrated – using a Bayesian procedure − on 43 mainly Dutch shallow lakes, aiming at a compromise fit rather than calibration on a specific lake (Janse et al. 2010), where the state of the lake (turbid or clear) was predicted well for 91% of the lakes. The outcome of the multi-lake calibration can be used as a starting point for the optimization of parameters for a specific lake. For example, the OSIRIS implementation was used to calibrate PCLake against two Danish shallow lakes: Lake Arreskov (Nielsen 2013) and Lake Engelsholm (Fig. 3) for which a reasonable fit was obtained between measured and simulated algae biomass from 1999 until 2001. In the latter study an ensemble of parameters combinations and ranges was used that would allow the simulated output time series to encompass all or most of the observations. In another study, PCLake was calibrated against the Dutch large shallow lake Markermeer (Elzinga 2013) by using GRIND for local optimization (simplex method: Press et al. 2009) and global optimization (shuffled complex evolution (SCE-UA) method).

R also offers a variety of calibration tools (see http://cran.r-project.org/web/views/Optimization.html).

*Validation*

There are many different definitions for model validation. Here, we refer to it as 'testing whether a model is acceptable for its intended use' (Refsgaard & Henriksen 2004). This is often done by confronting a model with measured data of systems that were not used for model calibration. In general, validation involves computing the goodness of fit between simulated and measured data and analysing whether the residuals are random or systematic.

Using ACSL, PCLake was validated against data for 34 Dutch shallow lakes (van Puijenbroek et al. 2004) and on 9 lakes that were not used for the Bayesian calibration procedure referred to above (Janse et al. 2010), using MRE as the fit criterion.

GRIND and OSIRIS provide basic $R^2$ validation tools and R offers more specific validation tools, such as in the packages FME and qualV (Jachner et al. 2007).

*Uncertainty analysis*

Uncertainty analysis measures the reliability of model output given uncertainties in model input, initial values and model structure (O'Neill & Gardner 1979).

The Bayesian procedure already applied for calibration (Janse et al. 2010) served to quantify uncertainties in the critical nutrient loading computed by PCLake in ACSL. Uncertainty ranges were computed from posterior parameter distributions; i.e. prior parameter distributions that were narrowed down by validating modelled output against observation data. The OSIRIS implementation was used to evaluate uncertainty of PCLake output, based on 900 simulations with randomly sampled values from a uniform distribution for the most sensitive parameters, sampled within a range of -20% to +20% of their default value (Nielsen et al. accepted for publication). The same OSIRIS implementation was used for a simple structural uncertainty analysis, as OSIRIS can be directed from a database environment which allows for comparing different versions of the model code, facilitating structural uncertainty analysis. OSIRIS was used to change the structure of PCLake by adding organic matter in three different ways (Lischke et al. 2014). This addition affected the hysteresis curve – which indicates at which nutrient loadings the lake switches from a clear to turbid state and vice versa – such that organic matter input increases the chance for a lake to become or stay turbid (Fig. 4).

GRIND and R also include tools for uncertainty analysis. In GRIND the Monte Carlo sampling method used for sensitivity analysis can also be used for a classical uncertainty analysis where the effects of prior distributions of parameters on the model outcomes can be evaluated (van Nes & Scheffer 2003). R provides a variety of tools related to model input uncertainty (see http://cran.r-project.org/web/views/Bayesian.html) and some packages facilitate structural uncertainty analysis (for example simecol (Petzoldt & Rinke 2007)).

*Bifurcation analysis*

Bifurcation analysis is used to reveal qualitative changes in long-term (asymptotic) model behaviour due to changes in parameters (e.g., mortality rates) or external forcings (e.g., nutrient loading). It can be used to determine the shape of the ecological stability landscape (Scheffer et al. 2001). The potential of this technique to analyse complex simulation models is easily overlooked.

We carried out bifurcation analyses in R to find critical nutrient loadings for PCLake (not shown here), leading to a hysteresis curve like in Fig. 4. For PCDitch, ACSL was used for this purpose (van Liere et al. 2007), where no hysteresis was found because the critical nutrient loading towards and from duckweed dominance was the same. These bifurcation analyses can be combined with scenario evaluation. For example, Janse et al. (2008) studied the importance of basic system characteristics (e.g. depth, fetch, sediment type) in PCLake using ACSL. Others focussed on the effects of global warming (Mooij et al. 2009, Mooij et al. 2007) using OSIRIS.

OSIRIS provides a simple bifurcation tool that calculates the effect of a stepwise varied parameter on a response variable. This tool is extensively used in teaching. We wrote scripts with a similar procedure for ACSL and R. GRIND features the automated ‘paranal’ routine for this approach. For more powerful bifurcation analyses, a switch to specialized frameworks, such as AUTO (Doedel & Oldeman 2009) or MATCONT (Dhooge et al. 2003), would be preferred. These frameworks are mainly used to analyse minimal dynamic models, consisting of only a few equations models (e.g. Kooi 2003), but could also be used for more complex models.

*Scenario analysis*

With a calibrated and validated model, scenario analyses can be carried out to evaluate potential future scenarios and the effectiveness of measures for ecosystem management.

The effect of nutrient loading on transient dynamics of floating and submerged plants were explored with PCDitch in ACSL (Janse & van Puijenbroek 1998). Using the same implementation, the effect of sediment type, flow rate and water depth on the critical nutrient loading was studied (van Liere et al. 2007). With PCLake in ACSL, the impact on the critical nutrient loading of herbivory by birds and fish (Janse et al. 1998), of global warming (Mooij et al. 2007) and of the size of surrounding marsh zone (Janse et al. 2001). The nutrient removal capacity of the marsh zone, leading to lower in-lake nutrient concentrations, was assessed with PCLake in OSIRIS (Sollie et al. 2008). We performed 1-D scenario tests with PCDitch in DUFLOW by looking at the effect of global warming on duckweed abundance in a spatial network of ditches (Fig. 5), showing that duckweed benefits from higher temperatures at the cost of submerged water plants. With PCLake in DELWAQ we performed a 3-D scenario analysis in the large shallow Chinese lake Taihu (Fig. 6) showing that the occurrence of summer algal blooms depends on the history of the lake, such as whether the lake was initially clear or turbid.

GRIND and R (using package deSolve) offer the possibility to define events, by making sudden changes in the values of state variables. This can be used to mimic discrete events in ecosystem management such as the removal of fish.

**Discussion**

Water quality modelling deals with multidisciplinary questions ranging from fundamental to applied. This diversity in questions requires a multitude of analysis techniques and therefore a multitude of frameworks to run a model in. Here we applied a database approach to modelling (DATM) for this purpose, which facilitates the implementation of a model in a framework and makes it easy to switch between frameworks to make use of the myriad of analysis tools offered by the frameworks. We would like to stress that we advocate the idea behind DATM rather than the specific implementation that we created. This idea is to put the process formulations of water quality models (terms $f_R(C,t)$ in Eq. 1-3) in *a framework independent* database and write translators to merge these process formulations with framework specific features such as spatial discretization and hydrodynamical process formulations. We are aware that many modelling packages contain *a framework dependent* library of process routines (e.g. DELWAQ, DUFLOW, etc.). From here it is only a small step to DATM. We see the technical simplicity of DATM as a strength and would welcome alternative implementations of the idea.

In this study we implemented PCDitch and PCLake in six different frameworks which revealed two clear benefits of DATM: 1) the possibility to use the framework one is familiar with for many analyses, and 2) the possibility to switch easily to other frameworks to exploit additional tools. This includes the switch between a 0-D and 1-D to 3-D implementation of the model. The first benefit arises from the redundancy in analysis tools amongst frameworks, while the second benefit stems from their complementarity. A surprising side effect of our efforts was that it ignited a healthy competition among the developers of some of the frameworks (in particular, GRIND and OSIRIS) to include missing tools in their framework after being convinced of their usefulness in other frameworks. The field of water quality modelling can undoubtedly benefit from such cross-fertilization of ideas.

The comparison of model runtimes in the different frameworks revealed huge differences. It showed the amount by which a model runs faster – up to 5000 times – in a framework that uses a compiled programming language instead of an interpreted language. This gain in runtime will especially pay off for large models when performing multiple model runs for example for a sensitivity or bifurcation analysis. Fortunately, some frameworks like R and GRIND which are based on interpreted languages offer the possibility to compile the model code to increase runtime performance. Furthermore, there is an obvious computational cost to implement a model in a spatial setting, increasing the runtime up to 100 times or more (dependent on the spatial configuration) compared to a 0-D setting. When applying models spatially, it is therefore recommendable to first explore the model behaviour in a 0-D setting.

Increasing redundancy among frameworks may lead to a point where one can ask what the added value is of using an approach like DATM over using encompassing frameworks such as the Delta Shell (Donchyts & Jagers 2010) or FABM (Trolle et al. 2012), which are both currently under construction. Indeed, an implicit motivation of many framework developers seems to be to make other frameworks superfluous. We note, however, that many of the frameworks from the early days of ecological simulation in the 1970s are still maintained even though many new frameworks have become available since then (Argent 2004). Based on this observation, we can reasonably expect that fragmentation of the field of water quality modelling, when it comes to framework use, is there to stay and can only be overcome by a DATM-like approach that operates at a different level.

While implementing PCDitch and PCLake in the six frameworks and discussing our results with experts in the field, we were pointed to a number of other frameworks for which DATM translators will be useful, either because of their additional tools, or simply because they are extensively used. Among these frameworks are AUTO (Doedel & Oldeman 2009), FABM (Trolle et al. 2012), FST (van Kraalingen et al. 2003), MATCONT (Dhooge et al. 2003), Mathematica (Wolfram 1999), Python (van Rossum & Drake 2001), Simecol (Petzoldt & Rinke 2007), SIMILE (Muetzelfeldt & Massheder 2003), SMART (Grant & Lai 1998) and Stella (Richmond & Peterson 1985).

In addition to the findings of this study, a database approach like DATM may facilitate the inspection and review of the assumptions underlying model code (Mooij et al. 2014). It offers a transparent way to inspect and review a model through standard database queries that can select the groups of states, parameters or equations to be studied in detail. This is especially rewarding for large models like PCDitch and PCLake, as demonstrated by our collaborative projects and student courses. Framework review, which, like model review, is often lacking in the scientific review process, can also be greatly enhanced with a database approach, by checking that a model gives the same results when implemented in another framework (Joppa et al. 2013). Furthermore, an approach like DATM provides a framework-independent way to store a given model version and a common ground for multiple users to work on a given version, even if these users prefer different frameworks for model analysis. This helps to maintain coherence in model development and promotes community-based model development.

In this study, we reported on our findings with applying DATM in the field of water quality modelling with fairly complex models. Our results are directly applicable, however, to other process models that are based on ODE's, whether simple or complex. These models can then be implemented by DATM in a 0-D setting or a spatial setting (PDE's). There are nonetheless some limitations. For instance, physiologically structured models of animal populations – that are defined in terms of PDE's with age, size, energy reserves and/or ontogenetic development as one of the integration variables (de Roos & Persson 2012) – cannot yet be implemented in the current definition of DATM, just like discrete time models, and the structured versions thereof, such as population matrix models (Caswell 1989). We see it as a future challenge to implement these types of models in DATM. The potential of DATM to easily combine models and frameworks does not imply that tools that were developed for the analysis of the simpler models will always work for more complex models. For instance, the sophisticated bifurcation tools that continue (un)stable equilibria along one or two parameter axes do not apply to seasonally forced models like PCDitch and PCLake, but only work for models with a constant forcing leading to stable steady states or periodic solutions like limit cycles.

We conclude that a database approach to modelling can be useful to address multidisciplinary questions in water quality modelling, as it makes the multitude of analysis techniques provided by different frameworks easily accessible. Thereby, it allows one to fully exploit the strength of each framework. We envision that a community-based further development of the concept can contribute to the future development of water quality modelling by facilitating the exchange of models and process formulations within the community of water quality modellers.

**Acknowledgements**

We thank Peter Leffelaar, Huub Scholten, Anne van Dam and Martin Droog for their helpful discussions and comments during the development of this paper. This work is financed by the Netherlands Foundation for Applied Water Research (STOWA) project no. 443237 and the China-Netherlands Joint Scientific Thematic Research Programme (JSTP) of the Netherlands Organisation for Scientific Research (NWO) project no. 842.00.009.

**References**

Aldenberg, T., Janse, J. H. and Kramer, P. R. M., 1995: Fitting the dynamic model PCLake to a multi-lake survey through Bayesian Statistics. Ecological Modelling **78**: 83-99.

Argent, R. M., 2004: An overview of model integration for environmental applications—components, frameworks and semantics. Environmental Modelling & Software **19**: 219-234.

Broers, M., 2012: Effect of climate change on the ecology of a ditch. BSc. Thesis. Wageningen University and Research Centre, Wageningen, The Netherlands.

Caswell, H., 1989: Matrix population models. -- Wiley Online Library.

Chow, S. N., Mallet-Paret, J. and Van Vleck, E. S., 1996: Dynamics of lattice differential equations. International Journal of Bifurcation and Chaos **6**: 1605-1621.

Clemmens, A., Holly, F. and Schuurmans, W., 1993: Description and Evaluation of Program: Duflow. Journal of Irrigation and Drainage Engineering **119**: 724-734.

de Boer, R. and Pagie, L., 1983: GRIND: Great integrator differential equations. Bioinformatics Group, University of Utrecht, The Netherlands.

David, O., J. C. Ascough, I., Lloyd, W., Green, T. R., Rojas, K. W., Leavesley, G. H. and Ahuja, L. R., 2013: A software engineering perspective on environmental modeling framework design: The Object Modeling System. Environ. Model. Softw. **39**: 201-213.

de Roos, A. M. and Persson, L., 2012: Population and community ecology of ontogenetic development. -- Princeton University Press.

Delft Hydraulics, 1995: DELWAQ version 4.0 Technical Reference Manual. Delft, The Netherlands.

Deltares, 2013: D-Water Quality, Water quality and aquatic ecology modelling suite. User Manual - Water Quality and Aquatic Ecology, and Technical Reference Manual - Processes Library Description. Delft, The Netherlands: Deltares.

Dhooge, A., Govaerts, W. and Kuznetsov, Y. A., 2003: MATCONT: a MATLAB package for numerical bifurcation analysis of ODEs. ACM Transactions on Mathematical Software (TOMS) **29**: 141-164.

Doedel, E. J. and Oldeman, B., 2009: Auto 07p: Continuation and bifurcation software for ordinary differential equations. Technical report, Concordia University, Montreal, Canada.

Donchyts, G. and Jagers, B., 2010: DeltaShell - an open modelling environment. In: International Environmental Modelling and Software Society (iEMSs) 2010 International Congress on Environmental Modelling and Software.

EC-JRC-ISIS, 2002: SimLab v. 2.2, User manual. European Commission, Joint Research Centre, Inst. for Systems, Informatics and Safety, Ispra, Italy.

Elzinga, J., 2013: Projected Social and Ecological Impacts of Marker Wetlands Projects on the Lake Markermeer Region. MSc. thesis. Wageningen University and Research Centre, Wageningen, The Netherlands.

Forrester, J. W., 1961: Industrial dynamics 2. -- MIT press Cambridge, MA.

Fox, J., 2009: Aspects of the Social Organization and Trajectory of the R Project. The R Journal **1**: 5-13.

Grant, H. and Lai, C. K., 1998: Simulation modeling with artificial reality technology (SMART): an integration of virtual reality and simulation modeling. In: Simulation Conference Proceedings, 1998. Winter 1. -- IEEE, pp. 437-441.

Jachner, S., van den Boogaart, G. and Petzoldt, T., 2007: Statistical methods for the qualitative assessment of dynamic models with time delay (R Package qualV). Journal of Statistical Software **22**: 1-31.

Janse, J. H., Ligtvoet, W., Van Tol, S. and Bresser, A. H. M., 2001: A model study on the role of wetland zones in lake eutrophication and restoration. The Scientific World Journal **1**: 605-614.

Janse, J. H., 1997: A model of nutrient dynamics in shallow lakes in relation to multiple stable states. Hydrobiologia **342**: 1-8.

Janse, J. H., 1998: A model of ditch vegetation in relation to eutrophication. Water Science and Technology **37**: 139-149.

Janse, J. H., 2005: Model studies on the eutrophication of shallow lakes and ditches. PhD Thesis, Wageningen University, Wageningen.

Janse, J. H., De Senerpont Domis, L. N., Scheffer, M., Lijklema, L., Van Liere, L., Klinge, M. and Mooij, W. M., 2008: Critical phosphorus loading of different types of shallow lakes and the consequences for management estimated with the ecosystem model PCLake. Limnologica-Ecology and Management of Inland Waters **38**: 203-219.

Janse, J. H., Scheffer, M., Lijklema, L., Van Liere, L., Sloot, J. S. and Mooij, W. M., 2010: Estimating the critical phosphorus loading of shallow lakes with the ecosystem model PCLake: Sensitivity, calibration and uncertainty. Ecological Modelling **221**: 654-665.

Janse, J. H., Van Donk, E. and Aldenberg, T., 1998: A model study on the stability of the macrophyte-dominated state as affected by biological factors. Water Research **32**: 2696-2706.

Janse, J. H. and van Puijenbroek, P. J. T. M., 1998: Effects of eutrophication in drainage ditches. Environmental Pollution **102**: 547-552.

Joppa, L. N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., Gavaghan, D. and Emmott, S., 2013: Troubling Trends in Scientific Software Use. Science **340**: 814-815.

Klepper, O., 1989: A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary (S.W. Netherlands). PhD Dissertation, LUW Wageningen The Netherlands, 270 pp.

Klepper, O., 1997: Multivariate aspects of model uncertainty analysis: tools for sensitivity analysis and calibration. Ecological Modelling **101**: 1-13.

Kooi, B., 2003: Numerical bifurcation analysis of ecosystems in a spatially homogeneous environment. Acta biotheoretica **51**: 189-222.

Lischke, B., Hilt, S., Janse, J. H., Kuiper, J. J., Mehner, T., Mooij, W. M. and Gaedke, U., 2014: Enhanced Input of Terrestrial Particulate Organic Matter Reduces the Resilience of the Clear-Water State of Shallow Lakes: A Model Study. Ecosystems **17**: 616-626.

Lloyd, W., David, O., Ascough II, J., Rojas, K., Carlson, J., Leavesley, G., Krause, P., Green, T. and Ahuja, L., 2011: Environmental modeling framework invasiveness: Analysis and implications. Environmental Modelling & Software **26**: 1240-1250.

Mieleitner, J. and Reichert, P., 2006: Analysis of the transferability of a biogeochemical lake model to lakes of different trophic state. Ecological Modelling **194**: 49-61.

Mitchell & Gauthier Associates, 1995: Advanced continuous simulation language (ACSL): reference manual, Version 11. -- MGA Software.

Mitchell, E. E. and Gauthier, J. S., 1976: Advanced continuous simulation language (ACSL). Simulation **26**: 72-78.

Mooij, W. M., De Senerpont Domis, L. N. and Janse, J. H., 2009: Linking species-and ecosystem-level impacts of climate change in lakes with a complex and a minimal model. Ecological Modelling **220**: 3011-3020.

Mooij, W. M. and Boersma, M., 1996: An object-oriented simulation framework for individual-based simulations (OSIRIS): *Daphnia* population dynamics as an example. Ecological Modelling **93**: 139-153.

Mooij, W. M., Brederveld, R. J., de Klein, J. J. M., DeAngelis, D. L., Downing, A. S., Faber, M., Gerla, D. J., Hipsey, M. R., t Hoen, J., Janse, J. H., Janssen, A. B. G., Jeuken, M., Kooi, B. W., Lischke, B., Petzoldt, T., Postma, L., Schep, S. A., Scholten, H., Teurlincx, S., Thiange, C., Trolle, D., van Dam, A. A., van Gerven, L. P. A., van Nes, E. H. and Kuiper, J. J., 2014: Serving many at once: How a database approach can create unity in dynamical ecosystem modelling. Environmental Modelling & Software.

Mooij, W. M., Janse, J. H., De Senerpont Domis, L. N., Hülsmann, S. and Ibelings, B. W., 2007: Predicting the effect of climate change on temperate shallow lakes with the ecosystem model PCLake. In: Shallow Lakes in a Changing World. -- Springer, pp. 443-454.

Mooij, W.M., Trolle, D., Jeppesen, E., Arhonditsis, G., Belolipetsky, P.V., Chitamwebwa, D.B.R., Degermendzhy, A.G., DeAngelis, D.L., De Senerpont Domis, L.N., Downing, A.S., Elliott, J.A., Fragoso Jr, C.R., Gaedke, U., Genova, S.N., Gulati, R.D., Håkanson, L., Hamilton, D.P., Hipsey, M.R., t Hoen, J., Hülsmann, S., Los, F.H., Makler-Pick, V., Petzoldt, T., Prokopkin, I.G., Rinke, K., Schep, S.A., Tominaga, K., van Dam, A.A., van Nes, E.H., Wells, S.A., Janse, J.H., 2010. Challenges and opportunities for integrating lake ecosystem modelling approaches. Aquatic Ecology **44**: 633-667.

Morris, M. D., 1991: Factorial sampling plans for preliminary computational experiments. Technometrics **33**: 161-174.

Muetzelfeldt, R. and Massheder, J., 2003: The Simile visual modelling environment. European Journal of Agronomy **18**: 345-358.

Nielsen, A., Trolle, D., Bjerring, R., Sondergaard, M., Olsen, J. E., Jeppesen, E., Janse, J. H. and Mooij, W. M., accepted for publication: Effects of climate and nutrient load on the water quality of shallow lakes assessed through ensemble runs by PCLake.

O'Neill, R. V. and Gardner, R. H., 1979: Sources of uncertainty in ecological models. IN: B. P. Zeigler, M. S. Elzas, G. J. Kliv, T. I. Oren (eds.), Methodology in Systems Modelling and Simulation. p. 447-463. North-Holland Publishing Co.

Petzoldt, T. and Rinke, K., 2007: Simecol: an object-oriented framework for ecological modeling in R. Journal of Statistical Software **22**: 1-31.

Portielje, R. and Roijackers, R. M. M., 1995: Primary succession of aquatic macrophytes in experimental ditches in relation to nutrient input. Aquatic Botany **50**: 127-140.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., 2009: Numerical recipes in C+: the art of scientific computing 994. -- Cambridge University Press Cambridge.

R Core Team, 2013: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: http://www.R-project.org/.

Refsgaard, J. C. and Henriksen, H. J., 2004: Modelling guidelines—terminology and guiding principles. Advances in Water Resources **27**: 71-82.

Richmond, B. and Peterson, S., 1985: Stella. Lyme, NH: High Performance Systems.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M. and Tarantola, S., 2008: Global sensitivity analysis: the primer. -- Wiley. com.

Scheffer, M., Carpenter, S., Foley, J. A., Folke, C. and Walker, B., 2001: Catastrophic shifts in ecosystems. Nature **413**: 591-596.

Scheffer, M., Szabó, S., Gragnani, A., van Nes, E. H., Rinaldi, S., Kautsky, N., Norberg, J., Roijackers, R. M. M. and Franken, R. J. M., 2003: Floating plant dominance as a stable state. Proceedings of the National Academy of Sciences **100**: 4040-4045.

Soetaert, K. and Petzoldt, T., 2010: Inverse modelling, sensitivity and monte carlo analysis in R using package FME. Journal of Statistical Software **33**: 1-28.

Soetaert, K., Petzoldt, T. and Setzer, R. W., 2010: Solving differential equations in R: Package deSolve. Journal of Statistical Software **33**: 1-25.

Sollie, S., Janse, J. H., Mooij, W. M., Coops, H. and Verhoeven, J. T., 2008: The contribution of marsh zones to water quality in Dutch shallow lakes: A modeling study. Environmental management **42**: 1002-1016.

Spaans, W., Booij, N., Praagman, N., Norman, R. and Lander, J., 1989: DUFLOW: A Micro-Computer Package for the Simulation of One-Dimensional Unsteady Flow in Open Channel Systems. Bureau SAMWAT, The Hague, Netherlands.

Stowa & MX.Systems, 2004: Duflow Modelling Studio: Duflow Manual version 3.7. -- STOWA.

Trolle, D., Elliot, J. A., Mooij, W. M., Janse, J. H., Bolding, K., Hamilton, D. P. and Jeppesen, E., 2014: Advancing projections of phytoplankton responses to climate change through ensemble modelling. Environmental Modelling and Software, in press.

Trolle, D., Hamilton, D. P., Hipsey, M. R., Bolding, K., Bruggeman, J., Mooij, W. M., Janse, J. H., Nielsen, A., Jeppesen, E. and Elliott, J. A., 2012: A community-based framework for aquatic ecosystem models. Hydrobiologia **683**: 25-34.

van Kraalingen, D. W. G., Rappoldt, C. and van Laar, H. H., 2003: The Fortran simulation translator, a simulation language. European Journal of Agronomy **18**: 359-361.

van Laarhoven, P. J. M. and Aarts, E. H. L., 1987: Simulated annealing. -- Springer.

van Liere, L., Janse, J. H. and Arts, G., 2007: Setting critical nutrient values for ditches using the eutrophication model PCDitch. Aquatic Ecology **41**: 443-449.

van Liere, L., Janse, J. H., Jeuken, M., van Puijenbroek, P. J. T. M., Schoumans, O., Hendriks, R., Roelsma, J. and Jonkers, D., 2002: Effect of nutrient loading on surface waters in polder Bergambacht, The Netherlands. In: Agricultural Effects on Ground and Surface Waters: Research at the edge of Science and Society 273. -- IAHS, Wageningen.

van Nes, E. H., Lammens, E. H. and Scheffer, M., 2002: PISCATOR, an individual-based model to analyze the dynamics of lake fish communities. Ecological modelling **152**: 261-278.

van Nes, E. H. and Scheffer, M., 2003: Alternative attractors may boost uncertainty and sensitivity in ecological models. Ecological Modelling **159**: 117-124.

van Oevelen, D., Van den Meersche, K., Meysman, F. J., Soetaert, K., Middelburg, J. J. and Vézina, A. F., 2010: Quantifying food web flows using linear inverse models. Ecosystems **13**: 32-45.

van Puijenbroek, P. J. T. M., Janse, J. H. and Knoop, J. M., 2004: Integrated modelling for nutrient loading and ecology of lakes in The Netherlands. Ecological Modelling **174**: 127-141.

van Rossum, G. and Drake, F. L., 2001: Python Reference Manual, PythonLabs, Virginia, USA.

Witteveen + Bos, 2008a: Watersysteemanalyse Loosdrechtse Plassen. Report nr. LOOS20-1-1 (In Dutch).

Witteveen + Bos, 2008b: Nadere uitwerking waterkwaliteit en ecologische kwaliteit Wieringerrandmeer. Report nr. WRW5-32 (In Dutch).

Witteveen + Bos, 2009: Effecten peilbeheer en waterberging Oldambtmeer. Report nr. VDM53-1-1 (In Dutch).

Witteveen + Bos, 2010a: Ecologische modellering Loenderveense Plas en Terra Nova. Report nr. LN4-1-1 (In Dutch).

Witteveen + Bos, 2010b: Waterkwaliteit herinrichting Nieuwe Driemanspolder. Report nr. ZTM99-1-1 (In Dutch).

Witteveen + Bos, 2013a: Toetsing, verbetering en ontsluiting ecologische modellen PCLake en PCDitch aan de hand van praktijktoepassingen: case Zuidlaardermeer. Report nr. STO170-1-10 (In Dutch).

Witteveen + Bos, 2013b: Toetsing, verbetering en ontsluiting ecologische modellen PCLake en PCDitch aan de hand van praktijktoepassingen: case Bergse Plassen. Report nr. STO170-1-13 (In Dutch).

Witteveen + Bos, 2013c: Toetsing, verbetering en ontsluiting ecologische modellen PCLake en PCDitch aan de hand van praktijktoepassingen: case Stad van de Zon. Report nr. STO170-1-14 (In Dutch).

Wolfram, S., 1999: The MATHEMATICA® Book, Version 4. -- Cambridge university press.

Table 1. Technical details of the frameworks used for water quality modelling with PCLake and PCDitch.

| Framework | programming language | language type[a] | user-interface[b] | automatic equation sorting[c] | availability |
|---|---|---|---|---|---|
| ACSL | ACSL | c | cl / GUI | yes | paid license |
| GRIND | MATLAB | i[1] | cl / GUI[2] | yes | free[3] |
| OSIRIS | C++ | c | cl / GUI[4] | no | free |
| R | R | i[1] | cl / GUI | no | free |
| DELWAQ | FORTRAN | c | cl / GUI | no | free |
| DUFLOW | DUPROL[5] | c | GUI | no | free |

[a] c = compiled, i = interpreted, [b] cl = command-line, GUI = Graphical User Interface, [c] automatic sorting guarantees that variables are not used until they are assigned a value, [1] models can be compiled in C++ (MATLAB) or also in FORTRAN (R) and linked to MATLAB / R to increase runtime performance, [2] to enter the equations as Forrester diagrams (Forrester 1961), [3] GRIND runs in MATLAB which is not free of charge, [4] Microsoft Excel or Microsoft Access, [5] DUFLOW Program Language.

Table 2. Runtime (min:sec) of a 50-year run of PCDitch and PCLake in different frameworks while producing daily output for all state variables. We used the preferred integrators (Table A1) with default settings on absolute and relative tolerance. The integrators had a variable time step, except those of DELWAQ and DUFLOW whose fixed time step was set to 10 minutes. The calculations were performed on a standard desktop PC with Intel Core i5-2500 @ 3.30 GHz processor.

| | **ACSL** | **GRIND** | **OSIRIS** | **R** | **DELWAQ** | **DUFLOW** |
|---|---|---|---|---|---|---|
| PCLake | 0:03 | 0:02[1] / 30:32 | 0:06 | 0:03[1] / 284:25 | 9:04[2] / 10:29[3] | 16:16 |
| PCDitch | 0:19 | 0:16[1] / 186:36 | 0:37 | 0:02[1] / 179:25 | 7:37[2] | 6:23 |

[1] model code was compiled in C++ and then called from the framework

[2] model was run in 1-D environment with hydrodynamics calculated by SOBEK

[3] model was run in 3-D environment with hydrodynamics calculated by Delft-3D

Table 3. Overview of the published and unpublished analyses on PCDitch (in bold) and PCLake, classified per analysis and framework. X=already performed, where studies that used the DATM-approach are underlined, x=potentially to be performed.

| **analysis** | **ACSL** | **GRIND** | **OSIRIS** | **R** | **DELWAQ** | **DUFLOW** |
|---|---|---|---|---|---|---|
| Sensitivity analysis | X[1] | X[2] | X[3] | x | | |
| Calibration | X[1,**4**,5] | X[6] | X[3,7] | x | | |
| Validation | X[8] | x | X[7] | x | | |
| Uncertainty analysis | X[1,5] | x | X[3,9] | x | | |
| Bifurcation analysis | X[10-13,**14**] | x | X[9] | X[15] | | |
| Scenario analysis | | | | | | |
| 0-D | X[13,**14**,16-21,**22,23**] | x | X[24-27] | X[28] | | |
| 1-D to 3-D | | | | | x[29] | X[30,31] |

[1] Janse et al. (2010), [2] see Fig. 2, [3] Nielsen (2013), [4] Janse (1998), [5] Aldenberg et al. (1995), [6] Elzinga (2013), [7] Trolle et al. (2014) (see Fig. 3), [8] van Puijenbroek et al. (2004), [9] Lischke et al. (2014) (see Fig. 4), [10] Janse et al. (2008), [11] Mooij et al. (2009), [12] Janse (1997), [13] Janse et al. (1998), [14] van Liere et al. (2007), [15] unpublished results, [16] Mooij et al. (2007), [17]Janse et al. (2001), [18] Witteveen + Bos (2008a), [19] Witteveen + Bos (2008b), [20] Witteveen + Bos (2009), [21] Witteveen + Bos (2010a), [22] Witteveen + Bos (2010b), [23] Janse & van Puijenbroek (1998), [24] Sollie et al. (2008), [25] Witteveen + Bos (2013a), [26] Witteveen + Bos (2013b), [27] Witteveen + Bos (2013c) , [28] Broers (2012), [29] see Fig. 6 [30] van Liere et al. (2002), [31] see Fig. 5.

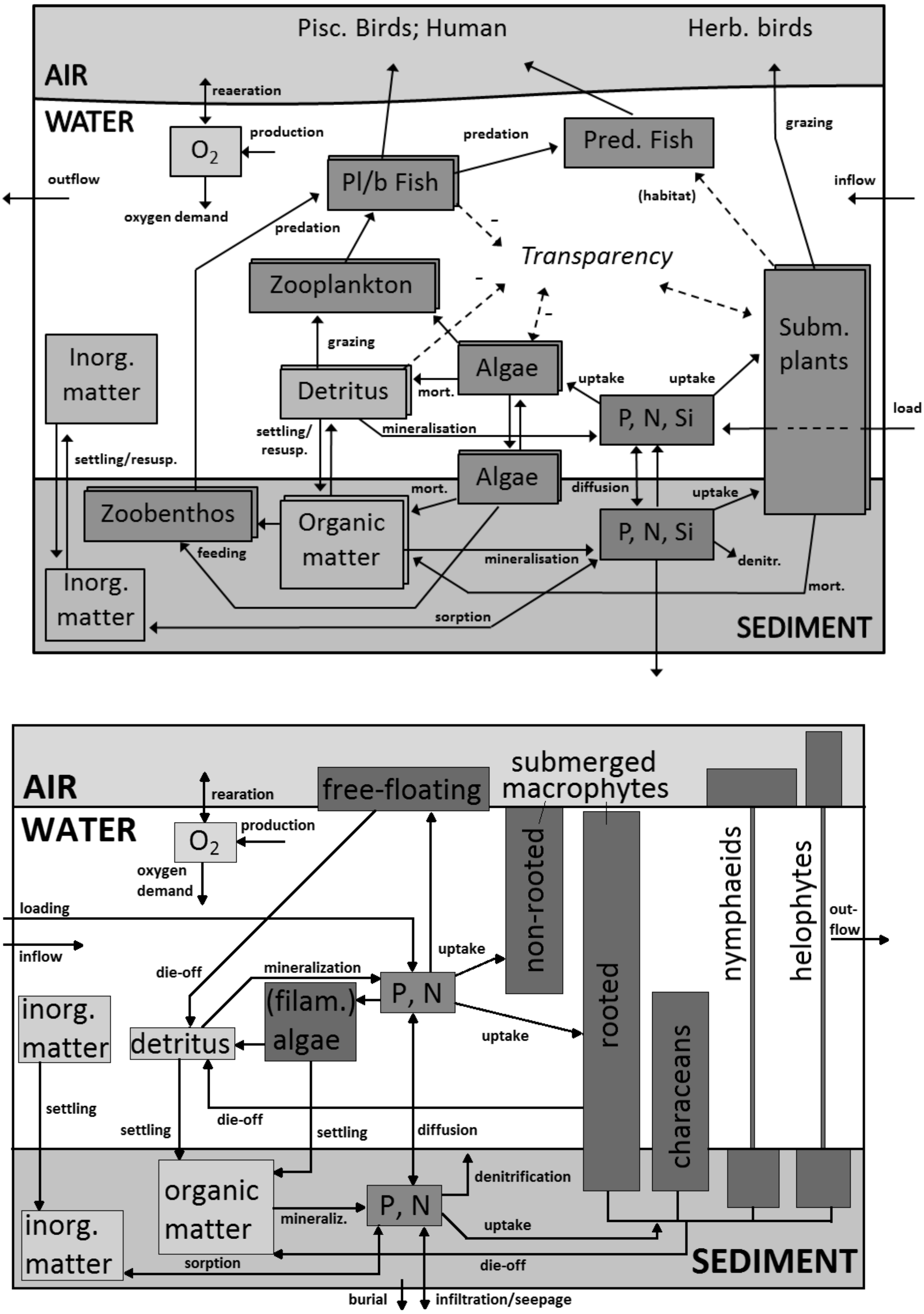


Figure 1: Model structure of PCLake (upper figure) and PCDitch (lower figure), modified from Janse (1997) and Janse (1998).

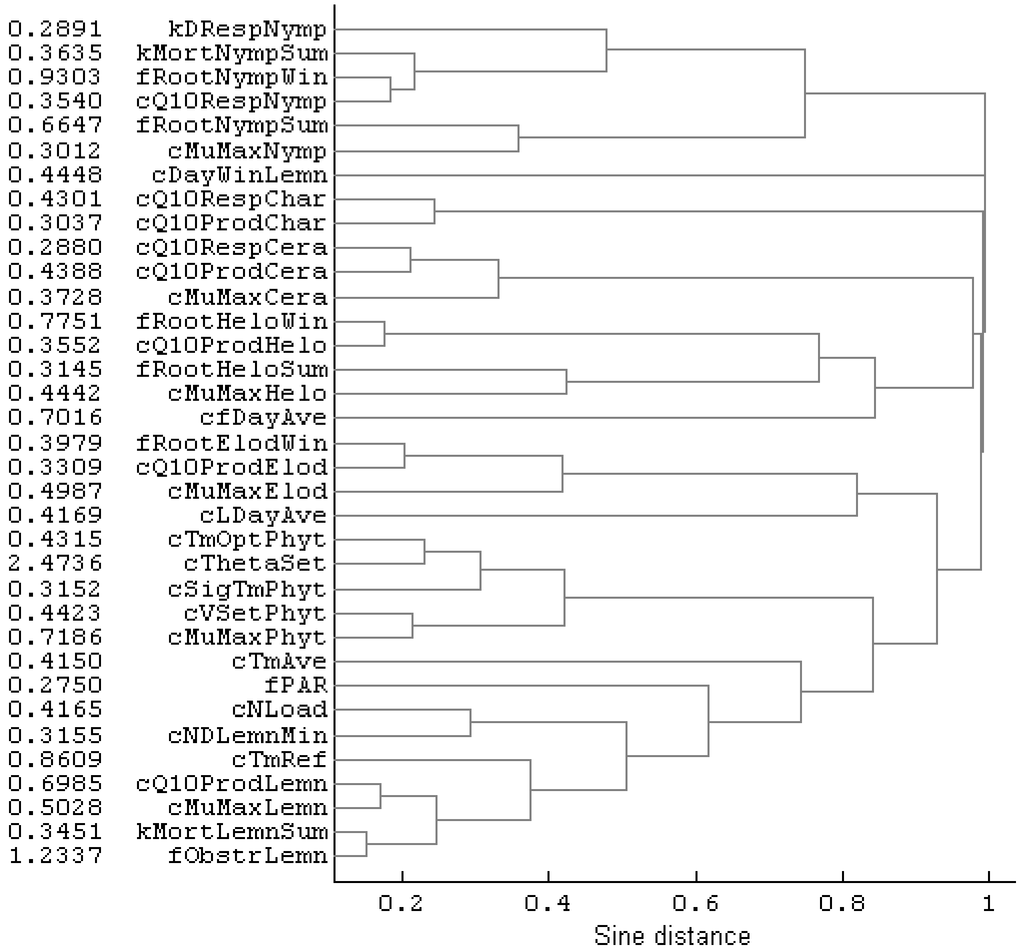


Figure 2: Example of a sensitivity analysis for PCDitch performed with GRIND (based on Klepper 1989). This dendrogram shows clusters of the most sensitive model parameters with a similar or opposite effect on the model results (biomass of all water plants at several times during the run). The 'sine distance' is used as a similarity measure of the parameters. The value of the sensitivity index before each parameter is the length of the vector of the sensitivity coefficients which is a measure of the total strength of the effect of the parameter. For more details see Klepper (1989) and van Nes et al. (2002).

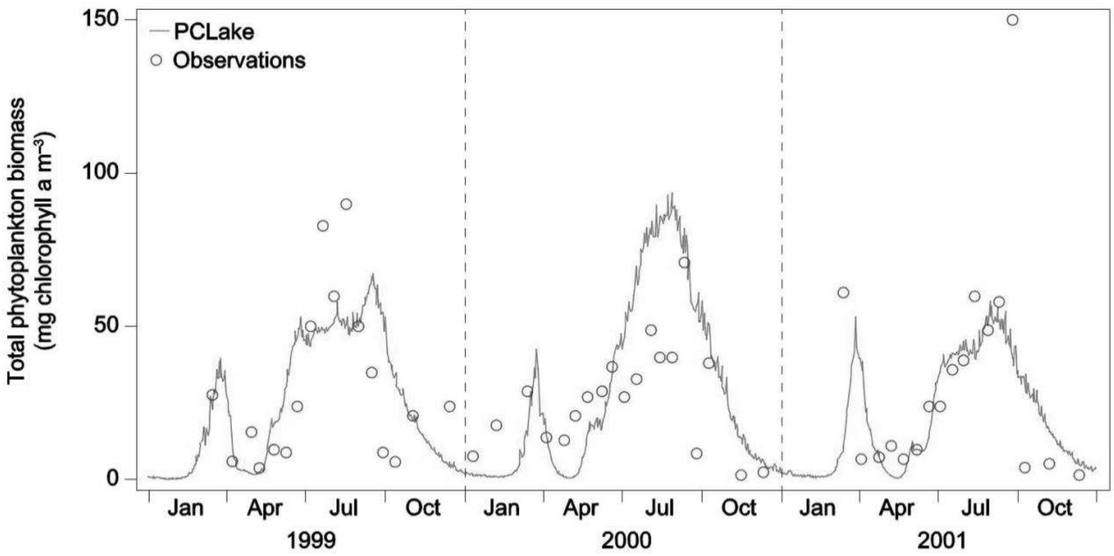


Figure 3: Example of calibration (years 1999-2000) and validation (year 2001) of PCLake performed with OSIRIS for Lake Engelsholm, a shallow eutrophic lake in Denmark (figure was modified from Trolle et al. 2014).

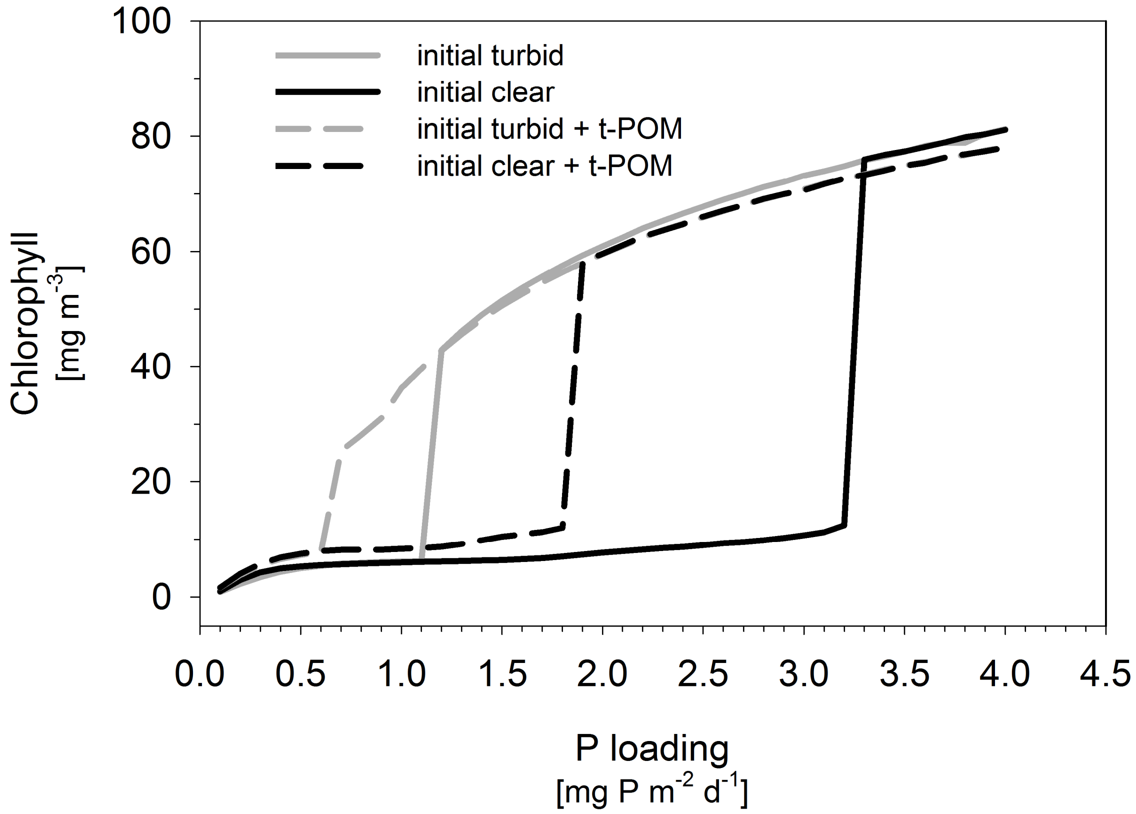


Figure 4: Example of a bifurcation analysis combined with a structural uncertainty analysis of PCLake in OSIRIS (figure copied from Lischke et al. 2014). It shows the bifurcation points of an average temperate shallow lake – the critical external phosphorus loadings at which the lake switches from a clear state to a turbid state and vice versa – for different model structures, so whether allochtonous terrestrial particulate organic matter (t-POM) is taken into account or not. The turbidity is represented by the average chorophyll-a concentration in the last year of a 30 year run and the t-POM input (only in autumn) equals 8 g dw $m^{-2}$ $day^{-1}$ to mimic leaf fall.

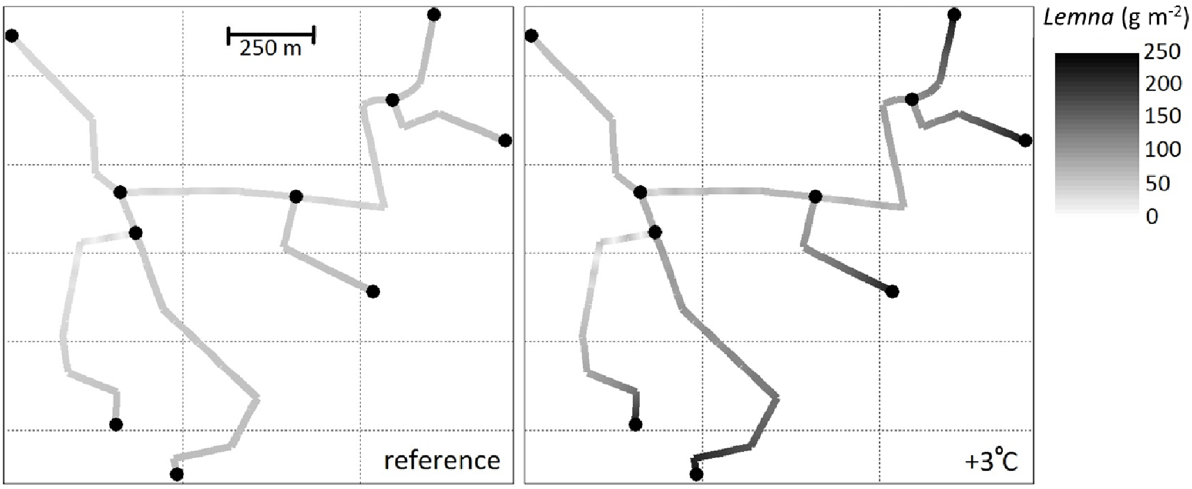


Figure 5: Example of a 1-D scenario analysis for PCDitch performed with DUFLOW. It shows the duckweed biomass on a summer day in a spatial network of ditches for a reference year (left) and a 3℃ warmer year (right).

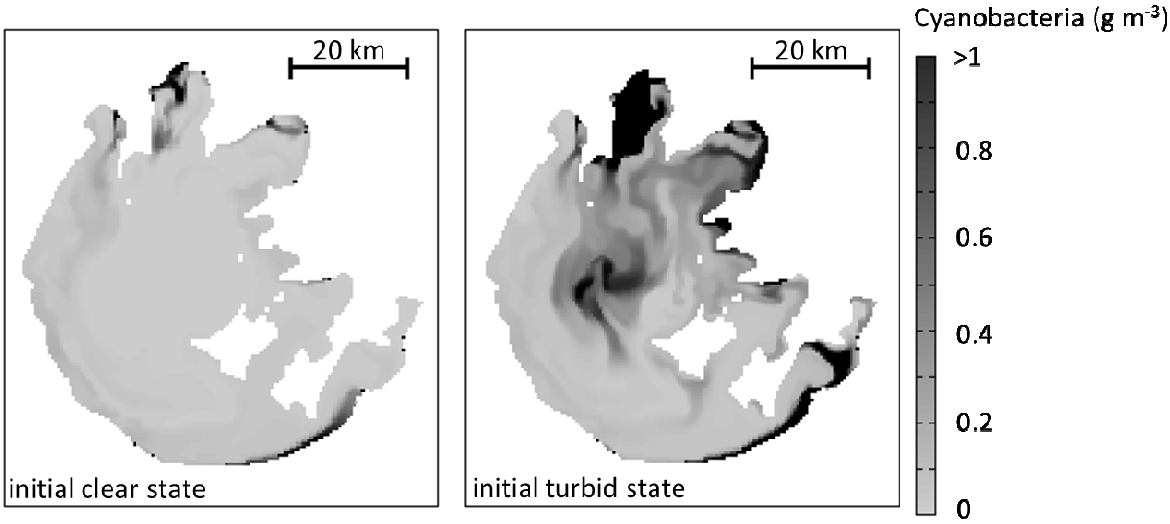


Figure 6: Example of a 3-D scenario analysis with PCLake in DELWAQ. It shows preliminary results of summer algal blooms in the large shallow Chinese lake Taihu, when starting from a clear water state (left) and a turbid state (right).

## Appendix 1: Implementing PCDitch and PCLake in the frameworks by using DATM

*Step 1: Building the DATM translator*

Framework-specific DATM translators turn essential model information – stored in the database − into an operational implementation of the model in the framework of choice. Essential model information includes the model equations and the information needed to run the model such as initial values for the state variables, parameter values, boundary conditions and runtime options (Mooij et al. 2014). We used Microsoft Excel to store, view and edit DATM information on PCDitch and PCLake and wrote the translators in Visual Basic for Applications (VBA).

The translators to GRIND and R were easiest to develop, typically taking a day, as they only dealt with translating the essential model information into the right syntax of the framework language. These two translators produced 'clean' and readable code with no overhead such as declaration statements and integration calls. These extra statements were needed by compiled language frameworks ACSL, OSIRIS, DUFLOW and DELWAQ. Translators for the spatially explicit frameworks DUFLOW and DELWAQ were most time-consuming to write, typically taking a week. First, those parts of the model that handle the built-in flow of water and substances (see first term in Eq. 1) needed to be excluded, as flow is managed by the frameworks themselves. Thereafter, the model had to be linked to the hydrodynamic variables covered by the framework. Finally, model state variables that are subjected to flow (e.g. free-floating plants) had to be declared as such in the frameworks. The translators gathered this information from extra fields in the model database.

While developing the translators we experienced difficulties in translating 'dynamic parameters', i.e. parameters that are modified by the model as the simulation proceeds. For example, PCDitch and PCLake store year-to-year variations in phenological parameters that indicate the start of the growing season as such dynamic parameters. For the frameworks ACSL, DUFLOW, OSIRIS and R storing dynamic information as parameters is no problem. In GRIND and DELWAQ, however, we had to use framework-specific constructs to implement dynamic parameters.

*Step 2: Debugging the generated code and checking it at t=0*

All frameworks easily picked up syntax errors with the debuggers incorporated in their compilers or interpreters. This held particularly for integrated development environments, such as the free C++ environment Code::Blocks (http://www.codeblocks.org/) which we used for compiling the OSIRIS code.

After syntax errors were resolved, we proceeded with checking for initial errors, i.e errors in the calculation of all identifiers in the model (parameters, initial states, auxiliary variables and derivatives) at t=0, before numerical integration has started. The initial errors that we encountered included missing pairs of parentheses that resulted in an incorrect evaluation of the equations. Debuggers cannot detect such errors in equation logic. We therefore checked the calculated values against known correct output, in our case of the ACSL implementation of the models. Checking for initial errors proved to be a powerful tool. Indeed, at t=0, errors in model equations are not yet propagated and the variables for which values do not match are direct clues to erroneous equations.

*Step 3: Choosing the integrator and setting up the simulation*

To run the model, the ordinary differential equations (ODE's) of PCLake and PCDitch had to be solved by numerical integration. The choice for an integrator and its step-size are important, as it influences the accuracy of results and model runtime performance. The various frameworks offer a list of different integrators to choose from (Table A1). Highlighted in Table A1 are the integrators that showed the best performances running PCDitch and PCLake in terms of accuracy and runtime in ACSL, GRIND, OSIRIS and R. In general, implicit integrators were more suited than explicit integrators, as PCLake and PCDitch contain stiff equations. Integrators that use a variable time step performed better in terms of runtime (up to an order of magnitude) than the ones that use a fixed time step while maintaining a good accuracy.

Model equations are solved differently in the spatially explicit frameworks (DUFLOW and DELWAQ). Here, the ODE's of PCLake and PCDitch are embedded as an extra term in the advection-dispersion equations (see the second term in Eq. 3). These are partial differential equations (PDE's) in time and space that describe the transport of substances, given previously calculated hydrodynamics (flow velocities). To solve these PDE's, DELWAQ has fourteen numerical integration methods to choose from, all with a fixed time step but varying implicitness. DUFLOW has one method of which the fixed time step and the implicitness of the time-integration can be set. For PCDitch, problems with negative state values were avoided by selecting a more implicit time-integration and a smaller time step, at the cost of a longer runtime. Note that in both DUFLOW and DELWAQ, the embedded PCLake or PCDitch term in the advection-dispersion equations (term 2 in Eq. 3) is always evaluated explicitly.

To set up the simulation, some frameworks required additional information next to the framework-specific model code and integrator. DUFLOW and DELWAQ required information on spatial configuration and associated boundary conditions to be defined in the user-interface. Furthermore, all frameworks required forcing functions to represent variable input (e.g. that of temperature and nutrient loading). For some frameworks (DUFLOW and DELWAQ), simulation options such as runtime and integration options, could only be defined in the user-interface and not passed to the framework by the DATM translators.

*Step 4: Dynamic test runs*

As is common in dynamic test runs, we experienced runtime errors varying from small deviations from known correct output, to negative values of state variables, or even an early termination of the model run due to a division by zero. However, a proper check against initial errors (see step 2) usually prevented most runtime errors. Model implementation in hydrodynamic frameworks (e.g. DELWAQ) produced an extra type of runtime error related to incorrect communication between model and framework on hydrodynamics and boundary conditions. We identified and corrected these errors by comparing the water and nutrient balance for a single-cell model implementation in DELWAQ with a 0-D control model. In our case this was the OSIRIS-implementation of the models.

Once runtime errors were solved, dynamic test runs allowed testing framework performance, in terms of accuracy and runtime. Besides the inevitable small deviations of numerical origin, all frameworks that we tested produced the same output.

Table A1. Available solvers per framework. The solvers of ACSL, OSIRIS, R and GRIND apply to ODE's, whereas the solvers of DELWAQ and DUFLOW deal with PDE's. In bold are the solvers that we preferred based on their performance in terms of numerical error and runtime.

| Framework | Solvers |
|---|---|
| ACSL[1] | Euler, rk2, rk4, rk2f, **rk5f**, Adams-Moulton, bdf (Gear's method) |
| GRIND | Euler[2], rk4[2], **ode45**[2], ode23, ode113, ode15s, ode23s, ode23t, ode23tb |
| OSIRIS | Euler, rk4, **rk45ck** |
| R[3] | Euler, rk2, rk4, rk23, rk23bs, rk34f, rk45f, rk45ck, rk45e, rk45dp6, rk45dp7, rk78dp, rk78f, ode23, ode45, lsoda, lsode, lsodes, lsodar, **vode**, daspk, radau, bdf, bdf_d, Adams, impAdams, impAdams_d, iteration |
| DELWAQ[4] | upwind scheme (US), second order Runge-Kutta, Lax Wendroff method, alternating direction implicit method, flux-corrected transport scheme (FCTS), implicit US with direct solver, implicit US with iterative solver, horizontal: US and vertical: implicit in time and central discretisation (ITCD), horizontal: FCTS and vertical: ITCD, horizontal: US and vertical: implicit in time and upwind discretisation (ITUD), horizontal: FCTS and vertical: ITUD, horizontal: implicit US and vertically: centrally discretised with iterative solver, ADI scheme for 3D models with central discretization in the vertical, ADI scheme for 3D models with upwind discretization in the vert., **local-theta FCTS** |
| DUFLOW[5] | **implicit scheme with direct solver** |

[1] Mitchell & Gauthier Associates (1995), [2] Also available when the model is compiled using C++, [3] Using R package 'deSolve' (Soetaert et al. 2010), [4] Deltares (2013), [5] Stowa & MX.Systems (2004)