Modeling Pit Lake Water Quality – How Do We Solve Multi-Disciplinary Problems?

Mike Müller

hydrocomputing GmbH & Co. KG, Zur Schule 20, 04158 Leipzig, Germany

Abstract Pit lakes are complex systems and their modeling requires an interdisciplinary approach. Coupled models for pit lake water quality modeling can often give convincing results. Further insight can be achieved by adapting the model to site-specific conditions. However, the communication in carrying out such a modeling project may prove less effective than should be. The paper provides suggestions on how to improve this communication process through information technology protocols. Finally, unifying the technical process of modeling and model development with the human interactions is proposed.

Keywords pit lakes, water quality modeling, multidisciplinary, collaboration

Introduction

Pit lakes belong to some of the most complex hydrological systems in terms of modeling, because important processes often involve several disciplines, such as hydrodynamics in the lake and the groundwater, hydro-geochemistry, and limnology. In addition, these systems are artificial and may be considerably different from natural conditions, and engineering measures often manipulate the lake to regulate the lake water level or to improve water quality.

This multidisciplinarity needs to be addressed in modeling approaches. While there is a considerable body of knowledge in modeling of single domains such as natural lakes, groundwater, erosion, and hydrogeochemisity, the interactions between these domains are much less understood. This may be attributed to two main reasons: (1) the technical difficulties to appropriately connect computational models of different domains, and (2) the different approach required when experts from different domains need to communicate and collaborate to solve a common problem. The technical difficulties, although not trivial, can be overcome by coupling established models from different domains.

There are several programs for modeling hydrodynamic processes in lakes such as DYRESM (Imerito 2013) for one-dimensional, CE-QUAL-W2 (Cole and Buchak 1995) for two-dimensional and ELCOM (Hodges, Dallimore 2013) for three-dimensional representation of lakes. Codes for modeling limnological water quality include AQUASIM (Reichert 1994), CAEDYM (Hipsey *et al.* 2007), and CE-QUAL-W2 (coupled hydrodymanics and water quality). Common models for hydro-chemical calculations are EQ3/6 (Wolery and Daveler 1992), PHREEQC (Parkhurst and Appelo 1999), and CORE2D V4 (Samper *et al.* 2012).

Salmon *et al.* (2008) describe a coupling of DYRESM and CAEDYM. Hipsey *et al.* (2006) used a coupling of ELCOM and CAEDYM for lake modeling. Another coupling is that of DYCD-CORE (Moreira 2010; Moreira *et al.* 2011). This paper uses PITLAKQ (Müller 2013) as an example. PITLAKQ is open source and freely available for further development. It couples CE-QUAL-W2 and PHREEQC as well as a few other models and additional algorithms to provide a comprehensive representation of important processes and their interactions. Modern software development techniques and powerful hardware facilitate the development and application of this type of model.

The second problem, the collaboration of experts from different, sometimes rather remote, fields is harder to solve. Setting-up, running and especially interpreting the result of such a model require in-depth knowledge in different domains. While experts know a lot about their fields, they may be novices in the field of their collaborators. This is compounded by ambiguous terminology definitions. The same technical term may describe something different in different technical disciplines, leading to potential miscommunication.

Important processes in pit lakes

There are many processes in a pit lake that can influence the water quality. Depending on the location and size of the lake, the sinks and sources of water and their water quality, and the intended use of the lake, emphasis on specific processes may vary. Processes thought to have a strong impact on the system may turn out not to be as significant as other processes that were not considered at all in the beginning. While some generalizations are possible, it seems that each lake is slightly or sometimes totally different and requires a modification in the modeling approach. The problem is that the need for these modifications often becomes clear only after some modeling has been done that shows problems in representing the system. Which process needs to be modeled in detail, which process can be simplified, and which process can be ignored can often only be determined by running sensitivity analyses with different processes included or excluded.

The challenge

Science and engineering are subdivided into subjects. Typically, people specialize in one subject and spend most of their professional life deepening their knowledge in it. While it is important to really know one subject and keep up with newest development in this field, it can become problematic when it comes to pit lake modeling. As can be seen from the above listed important processes, these processes cross over different disciplines such as hydrology, geochemistry, limnology, soil mechanics, groundwater hydrology, computational fluid dynamics and software engineering. These specialities may have overlaps but may come from different schools like natural science and engineering.

From the author's experience, frequently knowledge from people with diverse backgrounds, as indicated above, is needed to adequately tackle pit lake modeling problems. People from different disciplines tend to use different approaches to solve problems and often put emphasis on different tasks such as field sampling, laboratory experiments, data analyses or model building. While this is useful to cover aspects necessary to capture the nature of a pit lake, it can also cause new problems for solving the task at hand.

Miscommunications among specialists may arise because:

- They have different assumptions about the problem.
- They use slightly different terminology for the same thing.
- They use the same terminology for different things.
- They see the task from their domain and give it the most weight.
- They don't always fully understand what the other team members do.

We need diversity in backgrounds and approaches but at the same time a communication between team members that allows for working together towards one goal rather than working side-by-side focusing on individual specialties.

The software engineering approach – Example PITLAKQ

One attempt to model processes controlling pit lake water quality comprehensively is PITLAKQ. This modeling software is a new implementation of the work by Müller (2004) and continuous modifications and applications to different sites (Müller *et al.* 2008, 2011; Werner *et al.* 2008; Müller and Eulitz 2010). The idea here is not to implement a fully new model but rather to reuse existing modeling tools from different domains and to combine them to a new, integrated tool.

This approach has the advantage that many person years of programming, evaluating and improving of these tools can be uti-

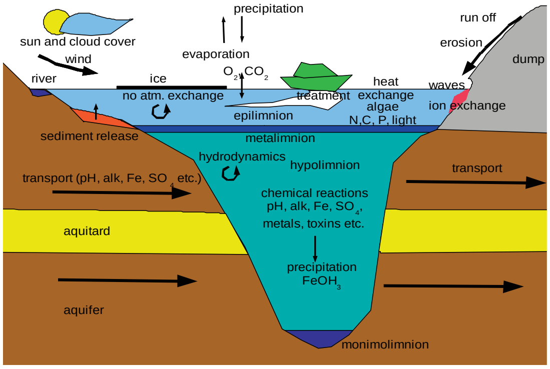

***Fig. 1** Schematic of processes in PITLAKQ.*

lized. Combining CE-QUAL-W2 and PHREEQC allows building upon the power of these public domain and well established tools.

One disadvantage of combining existing tools is that each comes with its own assumptions. Often this is not a substantial problem as along as the programmer and user of PITLAKQ takes this into account. But there are surprising cases that only become obvious after modeling results show unexpected effects. While some of these effects can be accommodated by catching potential errors with sanity checks, in the end the user needs to be sufficiently familiar with how both of these basic models work to successfully apply PITLAKQ.

Fig. 1 shows the processes implemented in PITLAKQ schematically. About 22 major process groups can be distinguished. This is sufficient for various pit lakes, focusing on acidic lakes. Often, a new site requires some modification of processes by adding a configuration option or adding a new process or variation of a process. While CE-QUAL-W2 is written in Fortran and PHREEQC in C, PITLAKQ is implemented in Python (Python Developers Team 2013). Very important is the interfacing of codes with Python to access the data structures from each individual model that allows for a fast implementation and testing process. Furthermore, all pre- and post-processing tools are written in Python. This allows working efficiently with large datasets which would be difficult or laborious to do with traditional tools such as spreadsheets. For many tasks the user needs only very basic Python knowledge to successfully modify pre- or post-processing scripts. This provides great flexibility to adapt to new types of problems.

Applications of PITLAKQ show that due to the dynamic nature of coupling, the coupled model often yields significantly different results to those obtained from uncoupled models that use pre-set boundary condition to represent the coupling partner. The models often show surprising results and trigger deeper investigations. These investigations often give a good indication of which data would need to be measured in more detail and which are not as important.

The human communication approach – interdisciplinary cooperation

While the implementation and the use of PITLAKQ are not trivial, it is a useful tool that can be applied and adapted to a variety of pit lake modeling problems. However, it does not do away with the need for expert knowledge from different domains outlined above.

As simple as it sounds, communication is paramount. Two types of communication should be included:

1. There should be regular physical meetings of all team members.
2. There should be a common document stating the objective of the modeling.

The common document should be continuously updated during the project and it should answer these questions:

- Is the objective still valid?
- Does the objective need modifications?
- What are the likely important processes for the objective?
- What measured values are needed and how are they turned into parameters?
- How sensitive are the parameters?

The document should contain a glossary explaining the used terminology. Even those seemingly simple terms like model, parameter, measured value or system should be defined. The definition should preferably be formulated using wording that can be understood by all team members while avoiding too much jargon.

The unified approach – effective communication supported at the technical level

Principles

Technical communication needs to be combined with the software engineering approach. One way of doing this would be through programming. Programming, as it is used here, can be defined as the expression of ideas in a formalized way that is restricted or guided by the programming languages. A program should primarily be useful for human communication and only secondly be executable by a computer. Programming can be done at various levels. For example, there is the distinction between systems programmers, library programmers and application programmers. Each type of programmer works with a different level of abstraction and with different objectives targeting different users of the software.

Why not add the "special domain programmer", a specialist in a domain who has sufficient programming knowledge to write his own, simple, programs and use libraries supplied by other programmers with deeper software engineering knowledge. While this does not sound like a viable way using languages such as C or Fortran, the advent of programming languages that allow a much higher level of abstraction changes the situation. Python is a general purpose programming language but has turned out to be useful for people who spend a minor proportion of their time programming, such as many scientists and engineers. They can learn enough programming in a rather limited time and can apply their knowledge to solve problems that otherwise would be very tedious or impossible to solve.

What would be the advantages of such an approach?

- Everybody in the team would use the same language, the programming language, to express their ideas.
- All ideas can be verified by running the resulting program.
- Outputs can be evaluated and automatically compared to expected values.
- The distinction between modeler and non-modeler in the team would become much weaker.
- Potentially all participating team members would get a much better understanding of the whole project.
- New ideas can be implemented in an unprecedented way.
- Of course there are disadvantages:
- This would be very new to many people and might not be accepted.
- Everybody who wants to participate has to learn the programming language.

Partial models

A partial model represents only a small part of a model focusing on one process or a few interrelated processes. Instead of interacting with the rest of the complex model, it receives predefined values the user has specified. This approach allows testing the behavior of individual parts of the model without all the complexity of the coupled model.

For example, the release of substances from the lake bottom sediment may depend on the water quality of the lake water right at the sediment-water interface. PITLAKQ can represent this process but needs parameters that have to be obtained through laboratory experiments. Naturally, the experimenter doing the laboratory study has the most expertise in measuring parameters. Now, instead of handing the parameters over to a modeler, the experimenter can use them in the partial model and see first hand what effect different parameters have on the release from the sediment.

Those partial models typically have very short run times in the order of seconds or a few minutes. Therefore, several model runs can be done as a parameter study. Using a few simple programming constructs such as loops and if-statements, many of these studies can be semi-automated. When the application programmer provides a good and simple programming interface, experimenters, who might not program otherwise, can perform parameter studies and adapt them flexibly to their needs.

There are several outcomes of this approach. (1) The experimenter better understands the impact of the parameter they measure. (2) The experimenter can better estimate the impact of the inherent variability in the measurement on the modeling results. (3) The coupled model will receive better parameters and their effects on the whole system can be better understood.

Of course, the extraction of partial models from a large model has one big disadvantage: it removes the feedback otherwise received from the other parts of the model. Therefore, the results from a partial model must be interpreted with caution. Despite this drawback, partial models can be useful and can help to better integrate the project team.

Interface definition between models

Another way where some basic programming techniques can help improve the communication are interface definitions between processes. Let us assume that there are biological processes in the lake that impact the hydro-geo-chemistry and vice versa. Two experts, one from the field of limnology and one from hydro-geo-chemistry describe what happens from their individual point of view. Naturally they will focus on their own domain, stressing either the limnological or chemical processes. Furthermore, both describe environmental factors that account for the "outside world" of the process from their point of view.

In addition to writing down their assumptions about the environment, both experts specify an interface in a programming language. That is essentially a list of environmental parameters with allowed ranges. When formalized in a programming language, the application programmer can write a small control program that checks for potential inconsistencies. This can be the basis for further discussions and adjustments of these interfaces. This is different from actually running the model because only the data flow between units representing different processes is checked and no process is modeled. The effort and therefore the turnaround times are much smaller than running a process model.

Conclusions

Pit lake modeling is a complex task that needs an interdisciplinary solution. Furthermore, sites and objectives are often so different that they require an adapted modeling solution. The coupling of existing models in combination with adding new model processes has proven to give new insight into pit lake behavior. The technical solution alone is not enough and a clear communication between all participants needs to be established, making objectives and meanings clear, as well as eliminating ambiguities in the terminology. This paper suggests that both the technical and the human communication part can be combined using programming. While this seems to be a highly technical solution, the use of a modern programming language such as Python, which is relatively simple yet powerful and widely used by occasional programmers, can actually facilitate communication among scientists. The restrictiveness of a programming language can help to achieve clearer problem definitions, resulting in better mutual understanding.

Acknowledgement

The author thanks Thomas Cole for CE-QUAL-W2 and giving valuable advice on details of its source code. Many thanks go to David Parkhurst for providing PHREEQC and long discussions on programming details. Furthermore, the author thanks Katja Eulitz for her valuable feedback on how to improve the handling and the process representation in PITLAKQ.

References

Castendyk and Eary (eds) (2009) Mine pit lakes: Characteristics, predictive modelling, and sustainability. Society for Mining, Metallurgy and Exploration, Littleton.

Cole, TM, Buchak, EM (1995) CE-QUAL-W2: A two-dimensional, laterally averaged, hydrodynamic and water quality model 2.0, US Army Corps of Engineers, Waterways Experiment Station, USA.

Hipsey, MR., Alexander, RA, Dallimore, CJ (2006) Simulation model of Lake Coeur d'Alene: Volume 2 –

Water quality modelling using ELCOM-CAEDYM, CWR Report.

Hipsey, MR, Romero, JR, Antenucci, JP, Imberger, J (2007) The Computational Aquatic Ecosystem Dynamics Model (CAEDYM): a versatile water quality model for coupling with hydrodynamic drivers, Pro. of the 7th Int. Conference on Hydroinformatics, Nice, France, 03–08 Sept. 2006, Vol 1., 526–533.

Hodges B, Dallimore C (2013) Estuary, Lake and Coastal Ocean Model: ELCOM, v2.2 Science Manual, Centre for Water Research University of Western Australia.

Imerito, A (2013) Dynamic Reservoir Simulation Model DYRESM, v4.0 Science Manual, Centre for Water Research University of Western Australia.

Müller, M (2004) Modellierung von Stofftransport und Reaktionen mit einem neuentwickelten, gekoppelten Grund und Oberflächenwassermodell am Beispiel eines Tagebaurestsees, Ph.D. Thesis, Dresden.

Müller M, Eulitz K (2010) Characterizing Water Quality of Pit Lake through Modeling, in Proceedings Mine Water & Innovatiev Thinking, C. Wolkersdorfer and A. Freund (eds), International Mine Water Symposium, 5–9 September 2010, Sydney, Nova Scotia, Canada, Cape Breton University Press, Sydney, Nova Scotia, Canada, pp. 375–378.

Müller M, Eulitz K, McCullough CD, Lund MA (2011) Model-based Investigations of acidity sources and sinks of a pit Lake in Western Australia, in Proceedings Mine Water – Managing Challenges, T.R. Rüde, A. Freund and C. Wolkersdorfer (eds), International Mine Water Congress, 4–11 September 2011, Aachen, Germany, RWTH Aachen, Germany, pp. 41–45.

Müller M, Werner F, Eulitz K, Graupner B (2008) Water Quality Modeling of Pit Lakes: Development of a Multiply-Coupled Groundwater Lake Circulation and Chemical Model, in Proceedings 10th International Mine Water Association Congress – Mine Water and the Environment, N. Rapantova, H. Zbynek and J. Zeman (eds), 10th International Mine Water Association Congress, 2–5 June 2008, Karlovy Vary, Czech Republic, Ostrava (VSB – Technical University of Ostrava), pp. 547–549.

Parkhurst DL, Appelo CAJ (1999) User’s guide to PHREEQC (Version 2) – A computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations, Water-Resources Investigations Report 99–4259, U.S. Geological Survey, 310 p.

McCullough CD (ed) (2011) Mine pit lakes: Closure and Management. Australian Centre for Geomecanics, Perth

Moreira S (2010) Coupled hydrodynamic and geochemical models for open pit lakes: application to the as pontes pit lake. Ph.D: thesis. Universidad a Coruña, Spain.

Moreira *et al.* (2011) Coupled hydrodynamic and geochemical model of the meromictic Waldsee in East Germany. Aquatic Geochemistry 17, 265–280.

Müller M (2013) PITLAKQ Documentation, viewed April 10, 2013. www..pitlakq.com.

Python Developers Team (2013) Python viewed April 10, 2013. www.python.org.

Reichert P (1994) AQUASIM – A tool for simulation and data analysis of aquatic systems, Water Sci. Tech., 30(2), 21–30.

Salmon SU, Oldham CE, Ivey GN (2008) Assessing internal and external controls on lake water quality: Limitations on organic carbon-driven alkalinity generation in acidic pit lakes. Water Resources Research 44, W10414, doi:10.1029/2007WR005959

Samper J, Yang C, Zheng L, Montenegro L, Xu T., Dai Z, Zhang G, Lu C, Moreira S (2012) CORE2D V4: A Code for Water Flow, Heat and Solute Transport, Geochemical Reactions, and Microbial Processes. In: Zhang F, Yeh GT, Parker J (2012) Groundwater Reactive Transport Models, 160–185.

Werner F, Eulitz K, Graupner B, Müller M (2008) Pit Lake Baerwalde Revisited: Comparing Predictions to Reality, in Proceedings 10th International Mine Water Association Congress – Mine Water and the Environment, N. Rapantova, H. Zbynek and J. Zeman (eds), 10th International Mine Water Association Congress, 2–5 June 2008, Karlovy Vary, Czech Republic, Ostrava, pp. 623–626.

Wolery TJ, Daveler SA (1992) EQ6, A computer program for reaction pat modeling of aqueous geochemical systems: theoretical manual, user’s guide, and related documentation (version 7.0). Lawrence Livermore National Laboratory, UCRL-MA-110662.