The RTM package

a Learning Environment for Reaction Transport Modelling

Lubos Polerecky, Dries Bonte and Karline Soetaert

July 2021

Abstract

Since many years we have been teaching courses on environmental and reaction-transport modelling to master students at Ghent and Utrecht University. Our aim is to teach students how to create mechanistic mathematical models from scratch and apply them to advance their understanding of natural systems, including those perturbed by anthropogenic activities. Typically, students attending our classes have different backgrounds and highly variable levels of skills in quantitative and computational methods, which makes it difficult to find a teaching style that suits everyone. To overcome this issue, we revised our didactic approach and now teach in the form of a "flipped class", where students prepare themselves ahead of the class using material available on-line, and then, during the class, put the learned concepts into practice by solving problems and exercises with a direct feedback from us. To facilitate this teaching approach, we created educational videos, tutorials, and exercises that guide the students in learning and exploring the topics covered in the courses. Here, we present the R-package RTM, which merges this teaching material together with our modelling and teaching experiences under the common platform of the programming language R.

Introduction

The courses "Environmental Modelling" and "Reactive Transport Modelling in the Hydrosphere" have been taught at Ghent and Utrecht University, respectively, since many years. Students attending these courses learn how to make mechanistic models that start from first principles of physics, chemistry, biology, and ecology, are mathematically formulated as differential equations, and are solved numerically by a computer. The models aim to advance their understanding of natural systems by allowing prediction of the systems' behaviour under various scenarios (e.g., response of the global carbon cycle to fossil fuel burning).

The aim of the courses is to teach students how to make models from scratch. This differs from many modelling courses where students merely learn how to apply existing tools. More specifically, students learn to

- create conceptual models that describe environmental scenarios or problems,
- translate conceptual models into mechanistic models and, ultimately, mathematical equations that describe the variation of the state variables characterising the system in time and space,
- implement these mathematical equations in a computer program and solve them,
- interpret the results in terms of the original environmental scenario or problem, and
- apply the models to realistic scenarios.

We use R (R core team 2020) as the programming language for implementing and solving these models. This is mainly because there exists an "ecosystem" of R-packages that provide methods for solving the type of differential equations that we deal with in the courses, including *rootSolve*, *deSolve*, *ReacTran* (Soetaert, 2009; Soetaert et al., 2010; Soetaert and Meysman, 2012). These packages have been implemented so that they hide much of the mathematical details behind the methods, allowing students to focus on the original problem rather than being concerned with finding a numerical approach to solve it. Additionally, they provide auxiliary functions that facilitate tabulation and visualisation of model results, fitting of models to data, etc.

Our students typically follow a master program in Earth or Life Sciences, and are interested in disciplines such as oceanography, biology, biogeochemistry, hydrology, or physics. Some have a good understanding of the processes occurring in the natural environment, but have never written a computer program before; others have ample experience in programming, mathematics or computer science, but little knowledge about the processes in nature. Due to this variation in background, knowledge and skills, a big challenge is to find a teaching style that can stimulate efficient and effective learning for all students.

Our way of teaching: “flipped class”

In the “traditional” learning environment, the instructor explains the context, theory and examples during lectures, and encourages students to practice the taught material by themselves at home (e.g., by solving problems and exercises given as homework assignments). In our “flipped class” approach, this process is reversed. Students obtain the theoretical lectures in advance and prepare themselves before the class. Then, during the class, they practice the learned concepts by solving problems and exercises with a direct feedback from the instructor.

One of the advantages of the flipped class is that students can work more focused and obtain a more thorough understanding of the material. Typically, the level of understanding of students in the modelling class is highly diverse. Thus, in the traditional way of teaching, the brighter students often get bored as they feel that the material is being taught too slowly, while less bright students become demotivated or frustrated as they cannot follow. In the flipped class approach, students can take the time they need to prepare themselves for the lessons individually. Subsequently, while in the classroom, they can directly interact with tutors and therefore explore the topics to the depth determined by their own ability, motivation and ambition. This provides more room for tutoring “a la tete du client”.

The modelling course taught by Karline Soetaert and Dries Bonte at Ghent University has been in the form of a flipped class since the academic year 2019–2020, while the course taught by Lubos Polerecky and Karline Soetaert at Utrecht University first implemented this approach in the academic year 2020–2021. This way of teaching proved to be a positive experience: while in previous years students would complain that they do not get enough practice in class, there is now ample time for practical exercises. Also the class is much more dynamic, with dramatically improved student participation and involvement.

Implementation of the flipped classroom

To prepare for each lesson, students are offered the choice to either read one or more chapters from a modelling textbook (Soetaert and Herman, 2009), or watch a short video that explains the theory. Then they are encouraged to test their understanding of the theory by answering a number of short (often multiple-choice) questions. After this “priming”, students go to the classroom, where they submerge themselves, under the supervision of tutors, in solving more substantial exercises.

Videos that elaborate on the theory

The theory behind the material taught in the modelling courses is explained in a series of educational videos. The videos, which typically take around 15 minutes or less, were created as follows:

1. Didactically sound slide presentations were created.
2. The narrative for the material on the slides was written down.
3. The narrative was recorded and combined with the slides.

Steps 1 and 2 were created by the tutors, while step 3 was partially outsourced and partially edited by the tutors.

Creation of the educational videos offers several longer-term advantages. For example, they were made available to the public as a YouTube playlist (https://www.youtube.com/playlist?list=PLx8PHcDdmF-uD1Pr07TU9SzlhlGpfrxqn), and so can serve as an online course. Also, if students wish to refresh their knowledge of the subject, they can re-watch the videos at any time and in any order. Importantly, they can watch the videos at their own pace (e.g., if some students find the videos too slow, multi-tasking or the fast-forward knob is an option).

Quiz-like questions

Each video is accompanied by a set of quiz-like questions and small exercises. These were implemented in the form of *tutorials* provided by the R-package *learnr* (Schloerke et al., 2020). They mostly consist of multiple-choice questions, sprinkled here and there with a more substantial exercise that needs an R-code for it to be solved.

The quizzes were designed to allow the students to test their basic understanding of the subject explained in the video. Before we introduced these quizzes, we often noted that some students overestimate their grasp of the material and are overly confident, while others are baffled by the complexity of the material and may loose confidence that they will eventually manage to understand it. The quizzes allow students to make this assessment by, and for, themselves. They are not graded, but students are encouraged to use them for their own benefit, as they facilitate better preparation for the more substantial exercises in the classroom and ultimately for the exam. The quizzes also improve students' quantitative and problem-solving skills by giving them hints and guidance as to how to tackle specific questions.

Classroom exercises

While in the classroom, students work in small groups (2–4 students per group) on practical modelling *exercises* that address current environmental issues. It is required that students design, develop and implement their own models from scratch (e.g., they start from a conceptual diagram and progress until they develop an R-code that generates quantitative predictions).

At the time of writing this vignette, the exercises address, among others, the following environmental issues:

- *The CO_2 problem.* Atmospheric CO_2 concentrations are described by an Earth-system box model that comprises carbon fluxes among the major Earth's carbon reservoirs. The model is used to estimate the impact of anthropogenic activities (e.g., burning of fossil fuels, deforestation) on the atmospheric CO_2 concentrations and to explore the effects of certain mitigation strategies (e.g., ocean fertilisation). This is an example of a linear model.
- *The ozone problem.* Ozone is a green-house gas and, at high concentrations, harmful to humans and animals. It also plays an important protective role by absorbing harmful UV radiation. The ozone model is used to test how anthropogenic emissions of NO (nitric oxide) due to combustion or burning of fuels alter the natural ozone dynamics in the lower atmosphere. It is a good example of a "typical" chemistry model.
- *Ocean acidification.* This is a very complex problem as it deals with equilibrium chemistry. The topic is introduced by three exercises of increasing complexity. A more extensive model dealing with ocean acidification is not part of the mandatory course material, but is provided as auxiliary material that students may consult if they are interested in the topic (see below).
- *The COVID pandemic.* An epidemiological model (so-called SIR model) is developed to describe the spread of the SARS-CoV2 (corona) virus within a population. In addition to infection and recovery, the model includes hospitalisations, mortality, vaccination, and virus mutations. The students practice how to use information from epidemiologists (who are not modelers) to develop and apply such models.
- *Agricultural eutrophication.* This model describes how fertilization with nutrients affects the competition between crops and weeds, and explores the impact of different fertilization strategies on agricultural run-off. In addition to ecological interactions, the model also contains a (simplistic) economic component that lets students explore the monetary aspects of farming.

- *Dead zones in a river.* This one-dimensional reaction-transport model describes the distribution of dissolved oxygen concentrations in a river subjected to excessive nutrient input (ammonia). The model is used to estimate the maximal nutrient inputs that can be allowed to keep the minimal oxygen concentration above a certain critical level.

Additional exercise topics include

- introduction to R for modellers,
- conceptual diagrams and formulating of mass balance equations,
- dissolution of silica particles (e.g., diatom frustules),
- bacteria-mediated mineralisation of detritus,
- the P cycle in marine sediments (a simplified early diagenetic model),
- the coupled cycles of C, N, O, and S in aquatic sediments (a complex early diagenetic model),
- food web modelling of a lake or river ecosystem,
- competition between plants in grasslands,
- competition between floating plants and algae in shallow lakes,
- virus dynamics in the ocean,
- bio-economic modelling of a Scallop aquaculture.

New material can be added in the future, thus creating a repository from which modules can be selected based on students' interests that vary from year to year.

Auxiliary material

In addition to tutorials and exercises, the RTM package provides a number of smaller "readers" that explain various topics in somewhat greater detail. Their aim is to satisfy the curiosity of students with the greatest motivation to learn modelling in R. The topics covered in these readers are not compulsory for the course, so the students get to see them after the mid-term exam, but before they start to work on their final projects. They include

- implementation of *events* in dynamic models developed in R,
- use of *forcing functions* based on data in models developed in R,
- display of *observed data* alongside model results in R,
- *fitting* of a 1D reaction-transport model to data in R,
- *visualisation* of multi-dimensional outputs from a 1D reaction-transport model in R,
- modelling of *pH dynamics* in a 1D reaction-transport model in R,
- mathematical analysis of the response of a system to a *perturbation from an equilibrium*,
- development of *interactive applications* in R,
- the basics about the *numerical methods* used for reaction-transport modelling in R,
- the basics about collaborative model development via *git*.

Model templates

Although students develop their models from scratch, they do not necessarily start from an empty sheet when implementing them in R. Instead, they start from a number of R Markdown templates (Allaire et al., 2021) that are provided by the RTM package. Thus, rather then becoming easily confused or frustrated by the need to generate a syntactically correct R-code, students can focus on the implementation of the modelled processes and generation and visualisation of outputs using R-code chunks provided by the templates. The templates included in the package facilitate the development of

- dynamic models without the spatial component (*RTM_0D*),
- dynamic one-dimensional reaction-transport models (*RTM_1D*),
- dynamic one-dimensional reaction-transport models in porous media (*RTM_porous1D*).

Additionally, they include templates used in some of the exercises, including models that deal with

- local equilibrium chemistry (*RTM_equilibrium*),
- a basic food-web and element cycling in an aquatic ecosystem (*RTM_npzd*).

Final projects

After completing the tutorials, exercises, and the written mid-term exam, students proceed with a work on a larger project of their choice. This involves the development of a model about a specific environmental topic and is done in small groups of 2–4 people. In contrast to the exercises made in the classroom, they do this with less guidance from the tutors. They also need to document their work in a written report, where they outline the scientific rationale, describe their model (including assumptions and implementation), and present and interpret the results, much like they would do when writing a scientific paper. Note that the topics of these final projects are *not* part of the RTM package.

References

JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang and Richard Iannone (2021). rmarkdown: Dynamic Documents for R. R package version 2.9. https://rmarkdown.rstudio.com

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

B. Schloerke, J.J. Allaire and B. Borges (2020). learnr: Interactive Tutorials for R. R package version 0.10.1. https://CRAN.R-project.org/package=learnr

K. Soetaert (2009). rootSolve: Nonlinear root finding, equilibrium and steady-state analysis of ordinary differential equations. R-package version 1.6

K. Soetaert, T. Petzoldt, R. W. Setzer (2010). Solving Differential Equations in R: Package deSolve. Journal of Statistical Software, 33(9), 1–25. http://www.jstatsoft.org/v33/i09/DOI10.18637/jss.v033.i09

K. Soetaert and F. Meysman (2012). Reactive transport in aquatic ecosystems: Rapid model prototyping in the open source software R. Environmental Modelling & Software, 32, 49–60.

Appendix - About the RTM package

The RTM package contains tutorials, exercises and extra material (readers) that students study and work through during the course *Reactive transport modeling in the Hydrosphere* (taught at Utrecht University, The Netherlands) or *Environmental modelling* (taught at Ghent University, Belgium). Anyone who wants to learn environmental or reaction-transport modelling can consult the tutorials and make the exercises. The answers are also provided, in a sub-directory under the *inst* package directory.

Tutorials

The function *RTMtutorial* runs a specific tutorial. To list all possible tutorials, use the command

```
RTMtutorial("?")
```

x	description
introduction	About the modelling course at Utrecht University
why	Why is modelling useful?
conceptual	Making conceptual models

x	description
massbalance	Creating mass balance equations
Rmodels	Dynamic modelling in the R language
largescale	Large-scale models (e.g. System-Earth's carbon cycle)
chemical	Elementary chemical reactions
equilibrium	Equilibrium (reversible) chemical reactions
enzymatic	Enzymatic reactions
partitioning	Partitioning between phases
ecology	Ecological interactions
transportProcesses	The general reaction-transport equation
transportFluxes	Transport fluxes due to advection and diffusion/dispersion
transportPorous	Reaction-transport in porous media
transportBoundaries	Boundary conditions in teaction-transport models
transportR	Implementation of reaction-tranport models in R

Exercises

The function *RTMexercise* opens the questions for each exercise. Note that to display the exercise in a PDF format, you need to have LaTeX installed on your computer. To list all possible exercises, use the command

```
RTMexercise("?")
```

	x	description	type
2	conceptual	Translating problems into a conceptual diagram	massbalance
3	massbalance_chemistry	Creating mass balance equations in chemistry	massbalance
4	massbalance_ecology	Creating mass balance equations in ecology	massbalance
5	carbonCycle	An Earth-system box model of the carbon cycle	linearmodels
6	ozone	Ozone dynamics in the troposphere	chemistry
7	dissolutionSi	Dissolution kinetics of Si particles	chemistry
8	equilibriumNH3	Equilibrium chemistry - ammonium/ammonia	chemistry
9	equilibriumHCO3	Equilibrium chemistry - the carbonate system	chemistry
10	equilibriumOMD	Equilibrium chemistry - impact of mineralisation on pH	chemistry
11	detritus	Bacterial decay of detritus	biogeochemistry
12	npzd	nutrients, phytoplankton, zooplankton, detritus	biogeochemistry
13	crops_weed	Crops and weed competition for nutrients	biogeochemistry
14	COVID	The COVID pandemic (population dynamics)	epidemiology
15	virus	Virus dynamics in the ocean	epidemiology
16	bears_salmon	Foodweb model comprising bears, salmon and scavengers	ecology
17	plant_coexistence	Competition and coexistence of plants in grasslands	ecology
18	hyacinth_algae	Competition between floating plants and algae	ecology
19	aquaculture	Model of scallop aquaculture, including economics	individualbased
20	riverAnoxia	Anoxia in an estuary (1-D model)	reaction_transport
21	Pdiagenesis	Simple phosphorus diagenesis in marine sediments (1D)	reaction_transport
22	diagenesis	Complex diagenesis in marine sediments (C,N,O2,S)	reaction_transport

Auxiliary material

The function *RTMreader* opens the auxiliary material. Note that to display these readers in a PDF format, you need to have LaTeX installed on your computer. To list all possible documents, use the command

```
RTMreader("?")
```

x	description
events	Events in dynamic models developed in R
forcings	Forcing functions based on data in models developed in R
observations	Showing observed data alongside model results in R
fitting	Fitting a 1D reaction-transport model to data in R
visualisation	Visualising outputs from a 1D reaction-transport model in R
pHprofiles	Estimating pH in a 1D reaction-transport model in R
perturbation_I	Response of systems to a perturbation from equilibrium - Part I
perturbation_II	Response of systems to a perturbation from equilibrium - Part II
interactive	Developing interactive applications in R
numericalR	Numerical methods used for reaction-transport modelling in R
git_sharing_code	Sharing code with the world via Git in RStudio

RTM templates

To open a model template in Rstudio, choose one of the files from the menu File → New File → R Markdown → From Template. Save the file under a different name after opening the template.