Implementation and Simulation of Ecological Models in R: an Interactive Approach

Thomas Petzoldt[*]

February 9, 2004

In recent years, the R system has evolved to a mature environment for statistical data analysis, the development of new statistical techniques, and, together with an increasing number of books, papers and online documents, an impressing basis for learning, teaching and understanding statistical techniques. Moreover, due to the advantages of the S language and the OpenSource availability of the R software and its libraries, this system is also suitable to implement simulation models, which are common in ecology.

However, there remains one problem. When implementing models of different type, e.g. differential equations or individual-based models, the result may be a lot of different simulation programs and the interactive ease of a command driven R-system gets lost. The need to edit parameter files or the sourcecode directly is a serious disadvantage, particularly when sharing such models.

As a first step towards an intercheangable but easy to use format I propose a standardized list-structure to allow users to run, modify, implement, discuss and share ecological models. Each simulation model can be implemented as a list (simecol simulation model object) with an intentionally simple structure. The list has very few mandatory components, namely `equations`, which contains the model equations, rules or arbitrary program code, `params` with the constant model parameters and `init` which is used to define the initial state as vector, matrix, list or via a special initialization function.

For dynamic models the vector `times` is used to define simulation interval and time steps and `inputs` for variable data (e.g. time dependend environmental forcings). The list component `solver` defines the solver to be used to simulate the model. It can be either `iteration` for discrete event models, an ODE-solver (e.g. `lsoda`) for ODE models or a user-provided algorithm. Simulation results are stored within the simecol object for later use, e.g. plotting or statistical analysis.

In this way, a simecol object contains the entire dataset, that is needed to run a basic simulation simply by entering the model object via `source()` or `data()` from the harddisk or the internet and then to run and plot the model via `plot(simulate(simecol.object))`. As an additional advantage R-helpfiles and package vignettes can be used as usable and standardized methods to overcome the documentation dilemma.

Interface functions are provided to get or set model parameters, time steps and initial values, but it is also possible to modify the components of simecol objects directly, e.g. the model equations. Compared to a strictly object oriented approach, this might seem dangerous and may lead to an inconsistent state, but on the other hand the list-based approach makes implementation and share of new models extremely simple. Using this, several applications with different types of ecological models will be demonstrated to show the interactive potential of this approach.

[*] Institute of Hydrobiology, Dresden University of Technology, 01062 Dresden, petzoldt@rcs.urz.tu-dresden.de