Geosci. Model Dev. Discuss., 5, 993–1039, 2012
www.geosci-model-dev-discuss.net/5/993/2012/
doi:10.5194/gmdd-5-993-2012

Geoscientific Model Development Discussions

Models of soil organic matter decomposition: the SoilR package, version 1.0

C. A. Sierra, M. Müller, and S. E. Trumbore

Max Planck Institute for Biogeochemistry, Hans-Knöll-Str. 10, 07745 Jena, Germany

Received: 29 March 2012 – Accepted: 5 April 2012 – Published: 2 May 2012

Correspondence to: C. A. Sierra (csierra@bgc-jena.mpg.de)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

Organic matter decomposition is a very important process within the Earth System because it controls the rates of mineralization of carbon and other biogeochemical elements, determining their flux to the atmosphere and the hydrosphere. SoilR is a modeling framework that contains a library of functions and tools for modeling soil organic matter decomposition under the R environment for computing. It implements a variety of model structures and tools to represent carbon storage and release from soil organic matter. In SoilR, organic matter decomposition is represented as a linear system of ordinary differential equations that generalizes the structure of most compartment-based decomposition models. A variety of functions is also available to represent environmental effects on decomposition rates. This document presents the conceptual basis for the functions implemented in the package. It is complementary to the help pages released with the software.

1 Introduction

Organic matter decomposition is a fundamental process within the Earth System (Swift et al., 1979; Schlesinger, 1997; Jacobson, 2000). Through this process, carbon and other biogeochemical elements fixed by plants in the process of photosynthesis are transferred to the atmosphere and the hydrosphere in mineral form. This release of biogeochemical elements is fundamental for other processes in the Earth System such as the global energy balance, with important consequences for climate. Organic matter decomposition is also a basic process for the availability of biogeochemical elements necessary for plant growth, therefore it has important consequences for agriculture and humanity.

Given the importance of soil organic matter decomposition, many models have been developed describing its dynamics (Manzoni and Porporato, 2009), but only few attempts have been made to synthesize them (e.g. Paustian et al., 1997; Wu and

McGechan, 1998; Manzoni and Porporato, 2009). Although more than 250 different models of soil organic matter decomposition have been proposed since the 1930s, most of these models share common mathematical structures (Manzoni and Porporato, 2009). This suggests that it is possible to develop models that can generalize most of the models already proposed. In fact, Ågren and Bosatta (1998) have made important contributions to a general theory of organic matter decomposition with the development of the continuous quality theory.

In most models, organic matter is usually characterized by compartments with homogeneous decomposition rates, which in the continuous quality theory are approximated by a continuous function between a rank variable denoted as quality and the decomposition rate. The continuous quality approach introduces a high level of generality, but it also introduces limitations in terms of finding analytical solutions for complex representations of organic matter heterogeneity (Sierra et al., 2011). For this reason, the continuous quality theory has only been implemented to describe the dynamics of the first moment of the distribution of quality. Furthermore, the description of microbial dynamics in the continuous quality theory lacks the generality needed to encompass different mathematical representations of microbial-substrate interactions.

A general theory of soil organic matter decomposition can benefit greatly from a synthesis of the different modeling approaches already proposed. It would help to identify model structures that have been used frequently with certain degree of success to represent observed data.

Recently, Held (2005) has called to attention a growing gap between high-end simulations and theoretical understanding in climate modeling, which we believe also applies to Earth system modeling in general. Current models are highly complex and use sophisticated algorithms to represent different processes within the Earth system. However, it is difficult to obtain a basic understanding of system behavior from these models due to their complexity. Furthermore, these models include only one single set of functions to represent a specific process, which is equivalent to proposing one single hypothesis to explain system structure and function. Therefore, Held (2005) proposed

GMDD
5, 993–1039, 2012

the development of hierarchical models in which detailed models can be reduced in hierarchies that help to better understand system dynamics. In one end, the general model has a high level of abstraction and helps to elucidate basic properties of the system. At the other end, detailed models are specific realizations of the general models that can help to predict system behavior under specific conditions such as climate change or emission scenarios.

In this document we introduce SoilR, a modeling framework to represent the process of soil organic matter decomposition in terrestrial ecosystems. It was developed under the idea of hierarchical models that synthesize different approaches to represent the decomposition process. The current version is built under the mathematical formalism of linear dynamical systems to represent, in a very general form, soil organic matter as a state variable with time dependent inputs, outputs, and internal transformations.

A dynamical system, in a broad sense, is a system that evolves in time through the iterated application of an underlying dynamical rule (Jost, 2005). To describe the evolution of a dynamical system over time, it is necessary to represent the actual state of the system and a mathematical rule that dictates the change of state. There are many different ways to represent both the state of the soil system and its transition rules. SoilR provides the basic framework to accommodate different representations of state or system structure and its dynamics. This is accomplished by a library of different numerical functions that can represent many different possibilities of soil organic matter dynamics.

This document presents the main structural characteristics of SoilR and the quantitative tools that can be used to represent different soil biogeochemical processes. The first version of this tool is focused on organic matter decomposition, and other versions of the package will include nutrient dynamics and isotopic composition.

1.1 General information about SoilR and R

The modeling framework we describe in this document is implemented in the R environment for computing (R Development Core Team, 2011). However, numerical ecosystem models are frequently developed in low-level programming languages such as C or Fortran. There are many advantages of using these low-level languages, specially in terms of computational efficiency; however, they are difficult to learn for scientists not formally trained in programing. At a different side of the spectrum of programming languages is the R environment; a high-level language in which simplicity may compromise efficiency. For many applications in ecosystem modeling though, the nature and size of the problems are usually not large enough for this being an important issue. Simplicity however, has been a major constraint for a wide adoption of models in ecosystem science.

Another important issue for model development is accessibility. Models coded in licensed software impose limitations in accessibility and future developments of new tools and models. Open Source software is ideal for guaranteeing that code can be freely distributed, used, and modified by everyone.

SoilR was developed in the R environment for computing to provide simple access of soil organic matter decomposition models in an Open Source platform where the code is freely accessible. To see source code and examples of the functions implemented in SoilR the user only needs to type the name of the function in the R command shell. To obtain more detailed documentation, the user can simply type a question mark (?) followed by a function name.

R allows the integration of concepts from functional and object oriented programing and can interface with many other low-level programing languages (Chambers, 2008). R also allows the development of software using concepts of *literate programing* (Knuth, 1984), allowing the production of code and documentation within the same environment.

As an open source tool, SoilR is also open for contributions by users interested in improving the existing code, make corrections on bugs in the code, add new functions, improve efficiency and functionality, etc.

GMDD
5, 993–1039, 2012

1.2 Development philosophy

In the development of SoilR, we were guided by four basic ideas:

- *Open source and reproducibility*: users should have access to all functions and be able to make changes as desired. Similarly, all functions and analyses implemented in SoilR can potentially be scrutinized or reimplemented in other languages. Results obtained using SoilR are therefore reproducible.
- *High-level programing*: users should be less concerned with model implementation and more focused on scientific exploration. Functions implemented in SoilR must be easy to use and require little input from the user, except for the input data and parameter values required for specific simulations.
- *Flexibility*: users should be able to have alternatives for representing the same system in different ways. This is accomplished in SoilR with the development of libraries. For example, the user is free to choose the temperature dependence of decomposition rates from a long list of options. Users can also define their own functions and incorporate them in the system.
- *Interoperability*: users should have the option of using tools from other developers. In R, a large numbers of functions and tools are available for data analysis, programing, and visualization. These tools can be easily incorporated within SoilR to provide a large number of possibilities for data analysis, statistics, and simulation.

2 Theoretical framework

2.1 Brief history of SOM modeling

Although early models of soil organic matter decomposition employed geometric series or difference equations (e.g. Nikiforoff, 1936; Jenny et al., 1949), the predominant

GMDD
5, 993–1039, 2012

mathematical formalism since the 1940s is that of ordinary differential equations (Manzoni and Porporato, 2009). Representing decomposition of chemical substances by differential equations was introduced much earlier than that (Van't Hoff, 1884). However, within the ecological disciplines, Olson (1963) presented the first comprehensive treatment of mathematical models of organic matter decomposition, popularizing the model

$$\frac{dX}{dt} = L - kX, \tag{1}$$

where X is either oven-dry weight, organic carbon, or energy in organic matter; L is the income of organic matter; and k a decay constant.

Equation (1) treats soil organic matter as one single compartment with an overall decomposition rate representative of all substances within the soil matrix. It has been commonly noted that soil organic matter is heterogeneous, and the single exponential model of decomposition fails to account for this heterogeneity (Minderman, 1968; Swift et al., 1979). Earlier, Henin et al. (1959) proposed a model to account for the different rates of decomposition of labile and stable material, also considering the process of humification, i.e. the transfer of material from the labile to the stable pool. This model can be expressed as

$$\begin{aligned} \frac{dX_1}{dt} &= L - k_1 X_1 \\ \frac{dX_2}{dt} &= \alpha k_1 X_1 - k_2 X_2, \end{aligned} \tag{2}$$

where X_1 represents the labile pool and X_2 the stable pool. The parameter α represents the humification or transfer rate. A variant of this model have been widely used for studies of litter decomposition, in which the system of equations takes the form

(Minderman, 1968; Means et al., 1985)

$$\frac{dX_1}{dt} = \gamma L - k_1 X_1$$
$$\frac{dX_2}{dt} = (1-\gamma)L - k_2 X_2. \qquad (3)$$

In this case, the two pools decompose independently from one another and the amount of litter inputs L is partitioned between the pools according to the parameter γ.

Different variations of these models can be found in the literature, with different number of pools and transfer among compartments.

Two numerical compartment models have become standard in representing organic matter decomposition, these are the RothC (Jenkinson and Rayner, 1977; Jenkinson et al., 1990) and the Century (Parton et al., 1987) models. These two models have been used successfully to represent soil carbon dynamics at different spatial and temporal scales (Paul and Clark, 1996; Paustian et al., 1997). Although these models were developed on the grounds of pragmatism rather than based on strict mathematical formalisms (Bolker et al., 1998), they can be easily translated into systems of differential equations with the general model (Bolker et al., 1998; Paustian et al., 1997)

$$\begin{aligned}\frac{dX_1}{dt} &= f(\theta_1 k_1 X_1, \cdots, \theta_m k_m X_m)\\ &\vdots\\ \frac{dX_m}{dt} &= f(\theta_1 k_1 X_1, \cdots, \theta_m k_m X_m),\end{aligned} \qquad (4)$$

where θ is a parameter set modifying the decomposition rate k, and m the total number of compartments representing the system.

In general, the number of compartments in this type of models is less than 10 (Manzoni and Porporato, 2009), and the decomposition rate constant may be a function of temperature, moisture, and/or other edaphic conditions.

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

We make use of this mathematical abstraction (Eq. 4), to propose a general model of soil organic matter decomposition.

2.2 A general model of soil organic matter decomposition

Models of soil organic matter decomposition are, in their large majority, specific cases of linear dynamical systems (Bolker et al., 1998; Manzoni and Porporato, 2009; Luo and Weng, 2011). Making use of this property, we propose a model that generalizes the majority of all previously proposed compartment models. This general model is given by

$$\frac{d\mathbf{C}(t)}{dt} = \mathbf{I}(t) + \mathbf{A}(t)\mathbf{C}(t), \tag{5}$$

where $\mathbf{C}(t)$ is a $m \times 1$ vector of carbon stores in m pools at a given time t; $\mathbf{A}(t)$ is a $m \times m$ square matrix containing time-dependent decomposition rates for each pool and transfer coefficients between pools; and $\mathbf{I}(t)$ is a time-dependent column vector describing the amount of inputs to each pool m.

The matrix $\mathbf{A}(t)$ is particularly important because it defines both the model structure and the extrinsic effects on decomposition and transfer rates. For this reason we rewrite Eq. (5) as

$$\frac{d\mathbf{C}(t)}{dt} = \mathbf{I}(t) + \xi(t)\mathbf{A}\,\mathbf{C}(t), \tag{6}$$

where $\xi(t)$ is a time-dependent scalar containing the extrinsic or environmental effects on decomposition rates. Notice that the matrix $\mathbf{A}$ contains now constant coefficients defining model structure.

From this general Eq. (6), it is possible to derive a large variety of structures for compartment models.

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

2.3 The matrix A and model structure

Organic matter decomposition can be represented with a large variety of model structures and levels of connectivity among compartments (Swift et al., 1979; Bruun et al., 2008; Manzoni and Porporato, 2009). Different model structures are determined by the matrix **A** in linear dynamical systems (Bolker et al., 1998; Manzoni et al., 2009). For instance, the parallel or pure decay structure (Fig. 1) in compartment models is defined by a diagonal matrix of the form

$$\mathbf{A} = \begin{pmatrix} -k_1 & 0 & \cdots & 0 \\ 0 & -k_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -k_m \end{pmatrix},$$

where the entries in the diagonal represent the decomposition rate k_j for each compartment j. A required condition is that all $k_j \geq 0$.

Compartments connected in series (Fig. 1) can be represented with a matrix of the form

$$\mathbf{A} = \begin{pmatrix} -k_1 & 0 & 0 & \cdots & 0 \\ a_{2,1} & -k_2 & 0 & \cdots & 0 \\ 0 & a_{3,2} & -k_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -k_m \end{pmatrix},$$

where the entries $a_{i,j}$ are the transfer coefficients of material from pool j to pool i. A required condition is that all $a_{i,j} \geq 0$

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

Similarly, feedback between adjacent compartments (Fig. 1) is defined by a matrix of the form

$$\mathbf{A} = \begin{pmatrix} -k_1 & a_{1,2} & 0 & 0 & \cdots & 0 \\ a_{2,1} & -k_2 & a_{2,3} & 0 & \cdots & 0 \\ 0 & a_{3,2} & -k_3 & a_{3,4} & \cdots & 0 \\ 0 & 0 & a_{4,3} & -k_4 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -k_m \end{pmatrix}. \tag{7}$$

More complex model structures are created by replacing zero entries in the matrix **A**, representing transfers between different compartments i and j.

An important characteristic of the entries $a_{i,j}$ is that they are proportional to the decomposition rate, i.e. $a_{i,j} = \alpha_{i,j} k_i$, where $\alpha_{i,j}$ represents the proportion of the decomposition rate that is transferred to pool i from pool j. Furthermore, $0 \leq \alpha_{i,j} \leq 1$, and the column sum $\sum_i \alpha_{i,j} \leq 1$, with

$$r_j = 1 - \sum_i \alpha_{i,j} \tag{8}$$

representing the proportion of the decomposed material that gets released from the system from pool j.

2.4 The environmental term $\xi(t)$

The majority of organic matter decomposition models include functions $f(x)$ that modify decomposition rates according to a set of time-varying environmental conditions $\{x_1(t) \ldots x_n(t)\}$ such as temperature, moisture, evapotranspiration, etc. (Burke et al., 2003; Adair et al., 2008). In our model (Eq. 6), the representation of these environmental effects is done with the term $\xi(t)$, which is the result of the evaluation of the function

GMDD
5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

or set of functions $f(x_i(t))$, yielding a scalar value that can be directly multiplied to the matrix **A**. In this case,

$$\xi(t) = f(x_1(t), \ldots, x_n(t)). \tag{9}$$

The values of $f(x_i(t))$ are determined by different functions that depend on temperature, precipitation, and other environmental variables. Time-dependence is therefore introduced with times series of these environmental variables as input in the model. SoilR contains a library of functions that calculate environmental effects on decomposition rates based on functions reported for different models (Table 1).

More complex functions are also introduced in SoilR but are not included in Table (1) due to space limitations. The documentation and help files of SoilR contain a more detailed description of all functions.

2.5 Initial conditions

The linear dynamical system represented by Eq. (6), has many different solutions, but we are only interested in the solution that satisfies

$$\mathbf{C}(t = 0) = \mathbf{C_0}, \tag{10}$$

where $\mathbf{C_0}$ is a $m \times 1$ vector with the value of carbon content in the different compartments i. $\mathbf{C_0}$ must be specified in SoilR to run any possible model structure.

GMDD
5, 993–1039, 2012

2.6 The vector of inputs

Inputs to the system from above and belowground components are represented by the vector $\mathbf{I}(t)$, which for clarity can also be expressed as

$$\mathbf{I}(t) = I(t) \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_i \\ \vdots \\ \gamma_m \end{pmatrix} \tag{11}$$

where $I(t)$ is a time-dependent scalar representing the total amount of inputs and the coefficients $0 \leq \gamma_i \leq 1$ represent the partitioning among the different pools.

2.7 Carbon release

A variable of interest in modeling soil organic matter decomposition is the amount of carbon leaving the system over time either in the form of CO_2 gas or as dissolved organic carbon. We represent this flux with the general term $\mathbf{r}$, which is given by

$$\mathbf{r} = \mathbf{R}\,\mathbf{C}(t), \tag{12}$$

where $\mathbf{r}$ is a $m \times 1$ vector containing the instantaneous release of carbon for all pools, and $\mathbf{R}$ is a $m \times m$ diagonal matrix with the release coefficients r_j in its diagonal calculated from (8).

2.8 Analytical solution

Analytical solutions to Eq. (5) are implemented in SoilR only with the purpose of testing the performance of the numerical methods. However, we can only test cases under certain simplifications of the general model of Eq. (5). In particular, for a homogeneous

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

system with constant coefficients, which is analogous to the decomposition of a single cohort of organic matter (Ågren and Bosatta, 1998), Eq. (5) simplifies to

$$\frac{d\mathbf{C}(t)}{dt} = \mathbf{A}\mathbf{C}(t). \tag{13}$$

With initial conditions as in (10) the analytical solution to this problem is given by

$$\mathbf{C}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{C_0}. \tag{14}$$

If $\mathbf{I}(t)$ is not identically zero, then the solution of the linear system

$$\frac{d\mathbf{C}(t)}{dt} = \mathbf{I}(t) + \mathbf{A}\mathbf{C}(t),$$

with initial conditions as in (10), is given by

$$\mathbf{C}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{C_0} + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{I}(\tau)d\tau. \tag{15}$$

A detailed description of the calculation of the matrix exponential $e^{\mathbf{A}}$ is provided in the appendix.

3 Numerical implementation

The solution to the dynamical system described by Eq. (6) is discretized over time, with h denoting the time step and n the number of steps. The time step h may or may not be constant. Initial conditions are given at time $t_0 = 0$. The solution to the system is then given by

$$\mathbf{C}_{n+1} = \mathbf{C}_n + D_r[f'(\mathbf{C}_n), h], \tag{16}$$

where $D_r[f'(\mathbf{C}_n), h]$ is an r order finite difference approximation to the system of ODEs of Eq. (6) for each time step h (LeVeque, 2007); in other words, and ODE solver.

The choice for the ODE solver is flexible in SoilR. Currently, we provide the option to use a simple Euler forward method or an interface to the `deSolve` package of Soetaert et al. (2010).

The function `deSolve.lsoda.wrapper` in SoilR, is a wrapper to the function `lsoda` in package `deSolve`. `lsoda` uses variable-step, variable-order methods and switches between stiff and non-stiff methods during the simulation when the stiffness of the system changes (Soetaert et al., 2010).

3.1 The `Model` class and the `GeneralModel` function

The numerical implementation of the general model described by Eq. (6) is facilitated in SoilR by defining the class `Model` in R. A class in R is an attribute of an object that defines formally what information the object should contain and how those objects should behave when functions are applied to them (Chambers, 2008). This means that any model structure constructed with SoilR would have the same attributes and will provide consistent results when generic functions are applied to these objects.

The class `Model` is initialized in SoilR by calling the function `GeneralModel`, which only initialize the `Model` object but does not perform any simulation. This function can only accept five different arguments that are necessary to implement a specific case of the general model of Eq. (6). The first argument is a vector `t` which contains the time steps where the solution to the ODE system is sought. This vector can be of any length but must be of class `"numeric"`.

The second argument is a matrix of class `"TimeMap"` which implements the time dependence of the matrix **A** of Eq. (5). The class `TimeMap` is native to SoilR and was implemented to prevent extrapolations beyond the range of input data. Due to the time dependence of the matrix **A**, it needs to be implemented as a function as required by the majority of ODE solvers (Soetaert et al., 2010) specifying the starting and ending times of the simulation.

The third argument of the function `GeneralModel` is a vector of class `"numeric"` containing the initial values $\mathbf{C}_0$ of the ODE system. The fourth argument is again an object of class `"TimeMap"` containing the inputs $\mathbf{I}(t)$ to the systems in a vector form. The length of this vector must be equal to the dimension of the matrix $\mathbf{A}$, and initial and ending times must be specified for this vector. The fifth argument is another object of class `"function"` specifying the ODE solver (see previous section).

Once a new object of class `Model` is initialized with a function call to `GeneralModel`, this new object can be queried to apply specific methods to it. For example, to obtain the amount of carbon over time solving the system of ODEs, any object of class `Model` can be queried by applying the function `getC`. The call to this function returns a $n \times m$ matrix with the amount of carbon for each pool m at each time step n. Similarly, to obtain the amount of carbon release over time, any object of class `Model` can be queried with the function `getReleaseFlux` to obtain a $n \times m$ matrix with the amount of released carbon for each pool m at each time step n.

This implementation of specific models as a class with generic methods will allow the integration of new functions without major modifications to our current implementation. For example, once nutrient cycling and isotope dynamics are incorporated into SoilR, new methods will be developed independently without major modifications to the current implementation of carbon stocks and release.

Specific model structures such as those presented in Fig. (1) are implemented as separate functions that simply call the general function `GeneralModel`. However, within the implementation of these specific model structures we introduced a series of tests that would avoid the specification of model constructs without biological meaning. In particular, these functions check for the correspondence between the dimensions of the matrix $\mathbf{A}$ and the vectors $\mathbf{I}$ and $\mathbf{C}_0$, and guarantee that the elements in the main diagonal of the matrix $\mathbf{A}$, i.e. the decomposition rates, are negative. In the vignette provided with SoilR we present some examples on how these specific model structures are implemented, which also serves as a template to implement new model structures

GMDD
5, 993–1039, 2012

as desired by the user. Examples on how to use these model structures as a function are presented in Sect. 4.

3.2 Version control system, unit testing, and automatic documentation

The development of SoilR is aided by a significant amount of existing open source software. To solve the ordinary differential equations produced by our framework, we rely on the well tested and documented `deSolve` package developed by Soetaert et al. (2010). We also use the open source symbolic python library SymPy (SymPy Development Team, 2012) to compute analytical solutions for the models for which this is possible. The analytical solutions obtained from SymPy are used to automatically create unit tests for SoilR. To constantly run these tests, we use another open source software, the `Runit` package (Burger et al., 2009). The tests are distributed with the release version of SoilR and thus add to the transparency of its development. In addition, we use `Sweave` (Leisch, 2002, 2003) and the `inlinedocs` package (Hocking et al., 2012) to produce documentation and encourage a literate programming style (Knuth, 1984). As version control system, we use Mercurial (O'Sullivan, 2009) and the Trac (Edgewall Software, 2011) online project management tool, which includes ticket system, wiki, and online access to our source code.

3.3 Documentation

There are different types of documentation for SoilR. The first source of information is this document, which introduces the science and some general technical details. A second source of information is the documentation to each function provided within the package itself. To view this documentation, the user only needs to open R and type `help.start()`. This will open a help window on a web-browser. There the user only needs to go to Packages/SoilR to view a list of all the functions implemented. Clicking on each function will show details about the arguments of each function and examples on how to use them. For specific functions, the user can also just type the name of

GMDD
5, 993–1039, 2012

the function preceded by the question mark on the R command shell. For example, typing `?TwopParallelModel` in the R command shell will open a help window with the description of the function. To view the source code of the function, the user only needs to type the name of the function (without the question mark) on the R command shell.

A third source of information are the so called Package Vignettes. These are short documents illustrating the use of the package for specific purposes. Currently, we provide one vignette with version 1.0 of SoilR. This vignette illustrates the implementation of any model structure within SoilR. For future versions, we will provide vigenettes about fitting specific model structures to data and how to use SoilR for modeling radiocarbon.

3.4 Installing and loading SoilR

SoilR can be obtained from the Comprehensive R Archive Network (CRAN), the official repository for R packages with mirrors in places all over the world. Packages stored in CRAN can be downloaded directly from an R session. It can also be obtained from R-Forge, a repository for package developers. To install SoilR from CRAN the user simply needs to type in the R command shell `install.packages("SoilR")`. To install from R-Forge, the statement is `install.packages("SoilR", repos="http://R-Forge.R-project.org")`. After installing the package, simply type `library(SoilR)` and the package is loaded into your R session.

4 Examples

In this section we present examples on how to run some of the functions implemented in SoilR based on the theoretical framework presented previously. Additional details about the implementations of each function and instructions on how to implement new model structures are presented in the vignette 'Implementing Compartment Models in

GMDD
5, 993–1039, 2012

SoilR: the `GeneralModel` Function' provided with the package. To view this vignette, simply type `vignette("GeneralModel", package="SoilR")` in the R command shell.

4.1 Implementation of a two pool model with connection in series: the ICBM model

One of the first models ever proposed to represent soil organic matter dynamics was a two-pool model with connection in series (c.f. Eq. 2, Henin et al., 1959). More than 50 years later, Andren and Katterer (1997) proposed the ICBM model, which is practically the same model proposed earlier by Henin et al. (1959), but including a term for temperature and moisture dependence of decomposition rates. The set of differential equations of the ICBM model are given by

$$\begin{aligned} \frac{dC_1}{dt} &= I - k_1 \xi C_1 \\ \frac{dC_2}{dt} &= \alpha k_1 \xi C_1 - k_2 \xi C_2, \end{aligned} \tag{17}$$

where α is a humification or transfer coefficient and ξ a parameter representing external effects on decomposition rates. In the ICBM model, C_1 represents a "young" pool and C_2 an "old" pool. This set of equations can be rewritten using our model formulation, which gives

$$\frac{d\mathbf{C}}{dt} = I \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \xi \begin{pmatrix} -k_1 & 0 \\ \alpha k_1 & -k_2 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}, \tag{18}$$

with initial conditions $(C_{1,0}, C_{2,0})^{\mathbf{T}}$, and where

$$\mathbf{A} = \begin{pmatrix} -k_1 & 0 \\ a_{2,1} & -k_2 \end{pmatrix},$$

GMDD
5, 993–1039, 2012

with $a_{2,1} = \alpha k_1$, $\gamma = 1$, and $\xi(t) = \xi$.

The function `ICBMModel` in SoilR implements this model structure requiring as its arguments: (1) a vector of any length with the points in time when we are interested in finding a solution, (2) a column vector of decomposition rates $(k_1, k_2)^{\mathbf{T}}$, (3) the value of α, (4) the value of ξ, (5) a column vector with the initial amount of carbon at the beginning of simulation $(C_{1,0}, C_{2,0})^{\mathbf{T}}$, and (6) the mean annual carbon input to the soil I. Andren and Katterer (1997) provided values for these arguments from a 35-year field experiment manipulating carbon and nitrogen inputs to the soil in Sweeden. For the case of a treatment in which the soil was left as bare fallow without N or C inputs, the ICBM model can be parameterized as

$$\frac{d\mathbf{C}}{dt} = 0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 1.32 \begin{pmatrix} -0.8 & 0 \\ 0.13 * 0.8 & -0.00605 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}, \tag{19}$$

with initial conditions $(0.3, 3.96)^{\mathbf{T}}$.

Assuming SoilR is already installed, it is only necessary to write the following lines of code to run the ICBM model

```
library(SoilR)
times=seq(0,20,by=0.1)
Bare=ICBMModel(t=times, ks=c(k1=0.8
        k2=0.00605), h=0.13, r=1.32,
         c0=c(C10=0.3,C20=3.96), In=0)
```

The first line simply loads the SoilR library within the R session. The second line simply defines the time vector for which we want to obtain a solution. In this case, it is a series of numbers from 0 to 20 by steps of 0.1 years. The call to `ICBMModel` simply initialize the model and checks for consistency on its arguments. In this example, the decomposition rates are given in units of year^{-1}, and the initial amounts of carbon in units of kg C m^{-2}.

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

To obtain the amount of carbon over time it is necessary to invoke the function `getC` storing the output into an object. For example, to store the amount of carbon from the object `Bare` the user can type

```
CtBare=getC(Bare)
```

This new object, `CtBare`, is a matrix with 2 columns (2 pools) and 201 rows (201 points in time, from 0 to 20 in increments of 0.1). To obtain the total amount of carbon, i.e. the sum of the pools, the R function `rowSums` can be used. For example, plotting the total amount of carbon over time as well as the carbon on each pool only requires these lines of code

```
plot(times, rowSums(CtBare),type="l",
    ylim=c(0,5), ylab="Topsoil carbon
    mass (kg m-2)",xlab="Time (years)",
    lwd=2)
lines(times, CtBare[,1],lty=2)
lines(times,CtBare[,2],lty=3,col=2,lwd=2)
legend("topright",c("Total carbon",
  "Carbon in pool 1", "Carbon in pool 2"),
  lty=c(1,2,3),col=c(1,1,2),lwd=c(2,1,2),
  bty="n")
```

If the total amount of carbon is needed for further calculations, the output of `rowSums()` can be stored in an object with any name.

We implemented the different N and C treatments reported in Andren and Katterer (1997) from the set of parameters reported by those authors. The code necessary to reproduce Fig. 2 in Andren and Katterer (1997) is provided as an example with the function `ICBMModel` and can be accessed by typing `attr(ICBMModel,"ex")`, or

GMDD
5, 993–1039, 2012

from the html help in R. This code produces Fig. 2, which is identical to Fig. 2 in Andren and Katterer (1997).

4.2 Alternative two-pool models

The ICBM model described in the previous section, although useful, does not offer too much flexibility in terms of the input arguments. For example, the litter inputs to the system could vary over time as well as the temperature and moisture effects on decomposition rates. In addition, there are other possibilities to implement a two pool model depending on the type of connection between pools (Fig. 1).

The parallel pool model structure can be implemented with the function `TwopParallelModel`, while a more general version of a series model structure can be implemented with the function `TwopSeriesModel`. Similarly, the feedback model structure can be implemented with the function `TwopFeedbackModel`. The inputs of litter and the modification of decomposition rates by external factors can be either constant or a function of time. Furthermore, the functions presented in Table 1, and their combination, can be used as arguments into the different model structures providing a large variety of options to model decomposition with just two pools. In fact, the same flexibility can be obtained with any number of pools with the application of the more general function `GeneralModel`.

As an example, we show the differences obtained by running three different versions of a two-pool model with the same amount of carbon at the beginning of the simulation, similar rates of litter inputs, and equal decomposition rates (Fig. 3). As an illustration, we also ran the simulations with different options for the time dependence of the litter inputs and the decomposition rates. In the first simulation, we ran a model with a structure of parallel compartments. In this simulation, the litter inputs and decomposition rates were constant, but the decomposition rates were modified by average values of temperature and moisture according to the functions proposed in the Daycent model (Table 1). For the second simulation, we ran a two-pool model with connection in series introducing temporal variability in the amount of inputs using a sine function that

GMDD
5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

artificially represents an annual cycle. In the third simulation, we ran a two-pool model with connection in series among compartments. The amount of litter inputs over time were calculated using random numbers over time. In this simulation, we also produced random numbers of temperature and moisture and applied the functions to modify decomposition rates according to the Century and the Demeter models (Fig. 3). The code to reproduce Fig. 3 is provided in the example of the function `TwopFeedbackModel`.

These simulations, without being necessarily realistic, simply show that small differences in model structure can produce very different predictions, even when the main parameters of the model remain unchanged. The simulations also serve to illustrate different possibilities in the use of the basic functions of SoilR to represent the process of organic matter decomposition over time.

To implement more sophisticated models with a higher degree of complexity, it is possible to specify a larger amount of pools with complex functions representing the dependence of litter inputs, decomposition rates, and transfer between pools with other external variables such as temperature, moisture, soil texture, nutrient status, among many other.

4.3 Implementation of the RothC model

RothC is a popular and widely used model for predicting organic matter dynamics over time. Although earlier versions of the model included five active pools and one inert pool (Jenkinson and Rayner, 1977), more recent versions only include four active pools plus the inert pool (Jenkinson et al., 1990). RothC is implemented within SoilR and we provide details here about this implementation to illustrate the use and potential implementation of any other model. RothC can be described by the following set of differential equations

Discussion Paper | Discussion Paper | Discussion Paper | Discussion Paper |

GMDD

5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

$$
\begin{aligned}
\frac{dC_1}{dt} &= \gamma I - k_1 C_1 \\
\frac{dC_2}{dt} &= (1-\gamma) I - k_2 C_2 \\
\frac{dC_3}{dt} &= \alpha_{1,3} k_1 C_1 + \alpha_{2,3} k_2 C_2 - k_3 C_3 + \alpha_{3,3} k_3 C_3 + \alpha_{4,3} k_4 C_4 \\
\frac{dC_4}{dt} &= \alpha_{1,4} k_1 C_1 + \alpha_{2,4} k_2 C_2 + \alpha_{3,4} k_3 C_3 - k_4 C_4 + \alpha_{4,4} k_4 C_4 \\
\frac{dC_5}{dt} &= 0
\end{aligned}
\tag{20}
$$

where C_1 represents the decomposable plant material (DPM) pool, C_2 the resistant plant material (RPM) pool, C_3 the microbial biomass (BIO) pool, C_4 the humified organic matter (HUM) pool, and C_5 the inert organic matter pool (IOM). This set of equations can be rewritten in the form

$$
\frac{d\mathbf{C}}{dt} = I \begin{pmatrix} \gamma \\ 1-\gamma \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -k_1 & 0 & 0 & 0 & 0 \\ 0 & -k_2 & 0 & 0 & 0 \\ \alpha_{1,3} k_1 & \alpha_{2,3} k_2 & -k_3(1-\alpha_{3,3}) & \alpha_{4,3} k_4 & 0 \\ \alpha_{1,4} k_1 & \alpha_{2,4} k_2 & \alpha_{3,4} & -k_4(1-\alpha_{4,4}) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{pmatrix}
$$

or as

$$
\frac{d\mathbf{C}}{dt} = I \begin{pmatrix} \gamma \\ 1-\gamma \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -k_1 & 0 & 0 & 0 & 0 \\ 0 & -k_2 & 0 & 0 & 0 \\ a_{1,3} & a_{2,3} & -k_3 + a_{3,3} & a_{4,3} & 0 \\ a_{1,4} & a_{2,4} & a_{3,4} & -k_4 + a_{4,4} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{pmatrix}
\tag{21}
$$

GMDD
5, 993–1039, 2012

Title Page
Abstract | Introduction
Conclusions | References
Tables | Figures
◀| |▶
◀ ▶
Back | Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

The values of the decomposition rates are constant and given by: $k_1 = 10$, $k_2 = 0.3$, $k_3 = 0.66$, and $k_4 = 0.02$ (Jenkinson et al., 1990; Coleman and Jenkinson, 1999). The value of the transfer coefficients is determined by a function of soil texture. For the microbial biomass pool, transfer coefficients are calculated as

$$a_{3,j} = k_{3,j} \frac{0.46}{x+1} \tag{22}$$

where x is a value that determines the proportion of decomposed material that is respired as CO_2 and is given by

$$x = 1.67(1.85 + 1.60 \exp(-0.0786\, p\text{Clay})) \tag{23}$$

where pClay is percent clay in mineral soil (Jenkinson et al., 1990). Similarly, the transfer coefficients for the humified pool are given by

$$a_{4,j} = k_{4,j} \frac{0.54}{x+1}. \tag{24}$$

The partitioning of incoming plant material is determined by the ratio DPM/RPM, which in RothC is set as 1.44. Therefore, $\gamma = 0.59$. Now, the basic structure of the RothC model can be written as

$$\frac{d\mathbf{C}}{dt} = I \begin{pmatrix} 0.59 \\ 0.41 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -10 & 0 & 0 & 0 & 0 \\ 0 & -0.3 & 0 & 0 & 0 \\ 1.02 & 0.03 & -0.59 & 0.01 & 0 \\ 1.19 & 0.04 & 0.08 & -0.02 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{pmatrix} \tag{25}$$

The annual amount of inputs is set in the RothC model as $I = 1.7$ MgC ha^{-1} yr^{-1}.

With this parameterization, it is possible to run the RothC model without varying environmental effects on decomposition rates and observe how the system approaches steady-state for the different pools (Fig. 4). Parameter values, initial conditions, and litter inputs can be changed easily within SoilR to compare different predictions from this model.

GMDD
5, 993–1039, 2012

4.4 Applications for large-scale modeling

R can handle a large variety of data structures, which offers interesting possibilities for the application of the functions implemented in SoilR. One important type of data structure is spatial data with global coverage. There is a large number of R packages to import and manipulate spatial data, which facilitates the technical aspects for running SoilR functions at the global scale.

As an example, we show here a simple calculation of climate decomposition indexes (CDIs) (Adair et al., 2008) using a global dataset of temperature, precipitation, and potential evapotranspiration available at 0.5 degree resolution on NetCDF files from the WATCH dataset (Weedon et al., 2011). The input dataset contained monthly average temperature, precipitation and evapotranspiration for the period 1958 to 2001. We calculated the CDIs as

$$\mathrm{CDI} = \xi(t) = f(T)f(W) \tag{26}$$

with $f(T)$ implemented by the function `fT.Century1`, and $f(W)$ by the function `fW.Century` as described in Table 1. Results can be easily plotted on a map (Fig. 5) and exported again to NetCDF files. Different combinations of the functions described in Table 1 can be used to calculate CDIs (Adair et al., 2008) and evaluate different hypotheses about the response of decomposition rates to moisture, temperature, and other variables.

5 Discussion

Many models of soil organic matter decomposition have been proposed previously, and there even exist some open source tools to implement some of these models (e.g. Easter et al., 2007). We have developed a tool for implementing and running a large variety of these models with the idea of facilitating comparison among multiple models

GMDD
5, 993–1039, 2012

in an easy to use interface. In this section, we discuss some of the advantages and disadvantages of our approach.

5.1 Parameter space and structural domain

Models of organic matter decomposition are basically hypothetical abstractions about the structure and dynamics of soil organic matter. The multitude of models previously developed suggests that there exists a large number of hypotheses about the structure and functioning of soil organic matter, but there is basically little consensus on whether one model structure (hypothesis) would have more support on observations than other model structures (Manzoni and Porporato, 2009).

The majority of modeling studies have focused on finding the set of parameter values of a particular model that best fit some observed data. This approach is useful and has provided much insight on understanding the rates of soil organic matter decomposition. However, from the perspective of assessing different hypotheses about the structure and dynamics of soil organic matter, parameter estimation can only give a narrow view of the more complex spectrum resulting from the combination of structure domain and the parameter space of models.

SoilR provides the possibility of assessing both model structure and parameter values broadening the spectrum of ideas that can be assessed within one single analytical framework. As a tool, it goes beyond than simply providing the best set of parameters from a particular model that best fit some observed data. SoilR allows for the exploration and testing of different hypotheses about processes within soils.

We believe this approach complements well, and could be even more powerful, than previous approaches to assess performance of model structure with inter-comparisons among different modeling groups (e.g., Melillo et al., 1995; Wu and McGechan, 1998; Cramer et al., 2001; Randerson et al., 2009). Multi-model inter-comparison projects do not necessarily cover the whole domain of model structures, and may be subject to important issues such as independence of code, bias of the whole model ensemble, inappropriate metrics to define model performance, etc (Knutti et al., 2009; Knutti,

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

2010). SoilR can partially help to overcome some of these problems for assessing the performance of model structure through the option of using alternative functions to represent the same process. Philosophically, this approach is also similar to testing multiple working hypotheses.

5.2 Model hierarchies and functional programing

The gap between simulations and understanding described by Held (2005) is currently exacerbated by the continuous increase in detail and complexity of simulation models. Held (2005) suggests that a way forward to close the gap between simulations and understanding is by the development of model hierarchies in which large-scale complex models are particular cases of general models that are more amenable for understanding of system structure and behavior.

SoilR can also be viewed as a system for hierarchical modeling of soil processes. Consider for example, the environmental or external effects on decomposition rates, which here are denoted by the term $\xi(t)$. In its more simple and general case, the external effects can be simply a constant ($\xi(t) = c$) that allows the understanding of model behavior without changes in environmental conditions. Simulations can then be run with a changing environment, for example with variable soil moisture ($\xi(t) = f(W(t))$). Soil moisture could depend on other variables such as precipitation and potential evapotranspiration ($W(t) = f(P(t), PET(t))$), which in turn can be dependent on other functions. In this form, a hierarchy of models is build with the dependence of different functions on other functions.

In terms of programing, this concept of model hierarchies can be easily implemented in a functional programing style. A function that performs certain task can have as its arguments other functions that perform other tasks. These functions can be independent among each other so many different functions that perform the same task can be available within the same modeling environment. This is one of our goals with SoilR, to provide a modeling environment that serves as a repository of different functions that can perform the same task so their performance can be easily compared. It also allows

GMDD
5, 993–1039, 2012

building soil organic matter decomposition models in a hierarchical framework because functions can have a large number of dependences on other functions creating a bridge between simple general models and detailed modeling constructions under the same basic principles.

5.3 Limitations

Obviously, our approach to organic matter decomposition modeling with SoilR has limitations. The first most obvious limitation, is the use of high-level programing language that can have important problems in computational efficiency. This could be an important problem if specific models structures with a large number of computations per time step are applied to a large number of points such as a grid of global points. Other programing languages such as C or Fortran may be more suitable for these tasks. SoilR is more suitable for the exploration of different model structures and hypotheses about soil processes rather than for large computational tasks, however, some paralellization tools within R could be used for this purpose. We recommend however, that once a specific model structure is identified as useful for a large computation, the entire model object is translated to other language. We are exploring this possibility to include in future releases of SoilR.

Another potential limitation is the incompatibility, in the current version of SoilR, for representing the decomposition process as a non-linear dynamical system. Non-linear dynamics are important for representing microbial processes such as priming, and are very relevant for simulations at short time scales (Wutzler et al., 2008; Manzoni and Porporato, 2007, 2009). A potential solution for this limitation in SoilR, is to provide a framework to linearize non-linear systems and compute solutions as presented here. This functionality can be possibly included in future releases.

There might be other limitations that we are not currently aware of. Probably over time, and with the input from other users, we will become aware of those limitations.

GMDD
5, 993–1039, 2012

6 Conclusions

We have developed a modeling environment for representing the process of soil organic matter decomposition. This tool, SoilR, is an open source package for the implementation and testing of different representations of the process of soil organic matter decomposition. The main characteristic of SoilR is its hierarchical structure in which we describe a general model that can accommodate any possible model structure of a multi-pool model of decomposition. More detailed models can be implemented to simulate specific controls on the decomposition process. This allows for testing of multiple working hypotheses about the structure and functioning of soils and their behavior over time. SoilR not only allows for exploring dynamics on the parameter space of a model but also on the structural domain.

This first version of SoilR only allows simulations of organic matter decomposition. Future versions will include representations of other biogeochemical elements such as nitrogen and phosphorus as well as their isotopic composition. A module for parameter fitting will also be included in future releases.

Appendix A

Computation of the matrix exponential

The matrix exponential $e^{\mathbf{A}}$ is defined similar to the exponential of a real number as

$$e^{\mathbf{A}} = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k. \tag{A1}$$

However, this definition can not be applied to actually compute $e^{\mathbf{A}}$ for a given matrix. To do so, we use the following theory from matrix algebra:

Discussion Paper | Discussion Paper | Discussion Paper | Discussion Paper |

GMDD

5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

1. For a diagonal matrix

$$\mathbf{d} = \begin{pmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_m \end{pmatrix},$$

The matrix exponential is given by

$$e^{\mathbf{d}} = \begin{pmatrix} e^{k_1} & 0 & \cdots & 0 \\ 0 & e^{k_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{k_m} \end{pmatrix}, \tag{A2}$$

2. If $\mathbf{D}$ is related to $\mathbf{d}$ by a similarity transformation $\mathbf{D} = \mathbf{PdP}^{-1}$ we can compute the exponential by

$$e^{\mathbf{D}} = \mathbf{P}e^{\mathbf{d}}\mathbf{P}^{-1} \tag{A3}$$

due to the fact that

$$\mathbf{PP}^{-1} = 1$$

3. For a nilpotent matrix $\mathbf{N}$ with $\mathbf{N}^q = 0$, the infinite sum in (A1) is reduced to polynomial because all powers $\mathbf{N}^k$ vanish for $k > q$.

4. If the minimal polynomial of $\mathbf{A}$ can be factored into a product of first degree polynomials, $\mathbf{A}$ can be decomposed into a diagonalizable $\mathbf{D}$ and a nilpotent matrix $\mathbf{N}$, where $\mathbf{D}$ and $\mathbf{N}$ commute (Jordan-Chevalley decomposition)

$$\mathbf{A} = \mathbf{D} + \mathbf{N}. \tag{A4}$$

$$\mathbf{DN} = \mathbf{ND} \tag{A5}$$

GMDD
5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

5. The Jordan canonical form (J.c.f) is a representative of the equivalence class of matrices similar (in the sense of A3) to the tridiagonal matrices as, for example in (7).

6. The J.c.f $\mathbf{J}$ of $\mathbf{A}$ consists of blocks that fulfill (A4) and (A5) where $\mathbf{D}$ is not only diagonalizable but even diagonal. Additionally, the exponential of $e^{\mathbf{J}}$ can be composed from the exponentials of the Jordan blocks, in the same way as $\mathbf{J}$ is composed of the blocks itself. Let $J_{\lambda_i k}$ denote the Jordan block related to the eigenvalue λ_i with block-size k then $\mathbf{J}$ as well as $e^{\mathbf{J}}$ can be expressed as block diagonals:

$$\begin{aligned} \mathbf{J} &= \mathbf{J}_{\lambda_{1}1} \oplus \cdots \oplus \mathbf{J}_{\lambda_1 m_1} \oplus \cdots \oplus \mathbf{J}_{\lambda_n 1} \oplus \cdots \oplus \mathbf{J}_{\lambda_n m_n} \\ e^{\mathbf{J}} &= e_1^{\mathbf{J}_{\lambda_1}} \oplus \cdots \oplus e^{\mathbf{J}_{\lambda_1 m_1}} \oplus \cdots \oplus e^{\mathbf{J}_{\lambda_n 1}} \oplus \cdots \oplus e^{\mathbf{J}_{\lambda_n m_n}} \end{aligned} \tag{A6}$$

In conclusion, for all cases where the J.n.f of $\mathbf{A}$ exists we can compute the matrix exponential combining (A3) and (A6), which results in an analytical solution of the form

$$e^{\mathbf{A}} = \mathbf{P} e^{\mathbf{J}} \mathbf{P}^{-1}.$$

We give a small example to illustrate this process for finding analytical solutions. Consider the following matrix

$$\mathbf{A}t = \begin{pmatrix} -t & 0 & 0 & 0 & 0 & 0 \\ 0 & -2t & 0 & 0 & 0 & 0 \\ 0 & 0 & -2t & 0 & 0 & 0 \\ 0 & 0 & 0.5t & -2t & 0 & 0 \\ 0 & 0 & 0 & 0.5t & -2t & 0 \\ 0 & 0 & 0 & 0 & 0.5t & -2t \end{pmatrix}$$

Discussion Paper | Discussion Paper | Discussion Paper | Discussion Paper |

GMDD

5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

The jordan normal form is given by

$$\mathbf{J} = \begin{pmatrix} -t & 0 & 0 & 0 & 0 & 0 \\ 0 & -2t & t & 0 & 0 & 0 \\ 0 & 0 & -2t & t & 0 & 0 \\ 0 & 0 & 0 & -2t & t & 0 \\ 0 & 0 & 0 & 0 & -2t & 0 \\ 0 & 0 & 0 & 0 & 0 & -2t \end{pmatrix}$$

The transformation matrix with $\mathbf{A} = \mathbf{PJP}^{-1}$ is:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0.125 & 0 & 0 & 0 & 0 \end{pmatrix}$$

There are three jordan blocks, the first related to the eigenvalue −1 of size 1, the second related to the eigenvalue −2 of size 4, and the third also related to the eigenvalue −2 but of size 1.

The matrix exponentials of the first and the last blocks are very easy to compute because the blocks are just 1×1 matrices.

$$e^{\mathbf{b}_1} = \frac{t}{e}$$

$$e^{\mathbf{b}_3} = \frac{t}{e^2}$$

Now consider the second Jordan block

$$\mathbf{b}_2 = \begin{pmatrix} -2t & t & 0 & 0 \\ 0 & -2t & t & 0 \\ 0 & 0 & -2t & t \\ 0 & 0 & 0 & -2t \end{pmatrix}$$

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

We decompose it into a diagonal and a nilpotent part

$$\mathbf{b_2} = \mathbf{d} + \mathbf{N} = \begin{pmatrix} -2t & 0 & 0 & 0 \\ 0 & -2t & 0 & 0 \\ 0 & 0 & -2t & 0 \\ 0 & 0 & 0 & -2t \end{pmatrix} + \begin{pmatrix} 0 & t & 0 & 0 \\ 0 & 0 & t & 0 \\ 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and compute the exponentials of the two parts separately starting with the diagonal part.

$$e^{\mathbf{d}} = \begin{pmatrix} e^{-2t} & 0 & 0 & 0 \\ 0 & e^{-2t} & 0 & 0 \\ 0 & 0 & e^{-2t} & 0 \\ 0 & 0 & 0 & e^{-2t} \end{pmatrix}$$

To compute the exponential of the nilpotent part, we look first at the powers of **N**

$$\mathbf{N}^2 = \begin{pmatrix} 0 & 0 & t^2 & 0 \\ 0 & 0 & 0 & t^2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{N}^3 = \begin{pmatrix} 0 & 0 & 0 & t^3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{N}^4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

GMDD
5, 993–1039, 2012
SoilR v. 1.0
C. A. Sierra et al.

Notice that $n^k = 0$ for $k > 3$. Therefore, we only need the first 4 terms of (A1). Accordingly, we have

$$e^{\mathbf{N}} = \begin{pmatrix} 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 \\ 0 & 1 & t & \frac{1}{2}t^2 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Combining the two results we get:

$$\begin{aligned} e^{\mathbf{b_2}} &= e^{\mathbf{d}+\mathbf{N}} \\ &= e^{\mathbf{d}} e^{\mathbf{N}} \\ &= \begin{pmatrix} e^{-2t} & te^{-2t} & \frac{1}{2}t^2e^{-2t} & \frac{1}{6}t^3e^{-2t} \\ 0 & e^{-2t} & te^{-2t} & \frac{1}{2}t^2e^{-2t} \\ 0 & 0 & e^{-2t} & te^{-2t} \\ 0 & 0 & 0 & e^{-2t} \end{pmatrix} \end{aligned}$$

Now we assemble e^J from the exponentials of the blocks. Next to the big one there are two others which have only size 1. Ordering from big to small we get

$$e^{\mathbf{J}t} = \begin{pmatrix} e^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{-2t} & te^{-2t} & \frac{1}{2}t^2e^{-2t} & \frac{1}{6}t^3e^{-2t} & 0 \\ 0 & 0 & e^{-2t} & te^{-2t} & \frac{1}{2}t^2e^{-2t} & 0 \\ 0 & 0 & 0 & e^{-2t} & te^{-2t} & 0 \\ 0 & 0 & 0 & 0 & e^{-2t} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-2} \end{pmatrix}$$

GMDD

5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

To arrive at the final result we have to transform back to the original base and eventually get:

$$e^{\mathbf{A}t} = \mathbf{P}e^{\mathbf{J}t}\mathbf{P}^{-1}$$

$$= \begin{pmatrix} e^{-1} & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{-2} & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{-2t} & 0 & 0 & 0 \\ 0 & 0 & 0.5te^{-2t} & e^{-2t} & 0 & 0 \\ 0 & 0 & 0.12t^2e^{-2t} & 0.5te^{-2t} & e^{-2t} & 0 \\ 0 & 0 & 0.02t^3e^{-2t} & 0.12t^2e^{-2t} & 0.5te^{-2t} & e^{-2t} \end{pmatrix},$$

which is the analytical solution.

Supplementary material related to this article is available online at: http://www.geosci-model-dev-discuss.net/5/993/2012/gmdd-5-993-2012-supplement.zip.

Acknowledgements. Financial support for the development of this project was provided by the Max Planck Society. We would like to thank Ulrich Weber and Martin Jung for providing global climate data used in the example. Maarten Braakhekke provided helpful comments on a previous version of this manuscript.

The service charges for this open access publication have been covered by the Max Planck Society.

References

Adair, E., Parton, W., Del Grosso, S., Silver, W., Harmon, M., Hall, S., Burke, I., and Hart, S.: Simple three-pool model accurately describes patterns of long-term litter

Discussion Paper | Discussion Paper | Discussion Paper | Discussion Paper |

GMDD

5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

Title Page

Abstract | Introduction

Conclusions | References

Tables | Figures

◀| |▶

◀ ▶

Back | Close

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

decomposition in diverse climates, Glob. Change Biol., 14, 2636–2660, doi:10.1111/j.1365-2486.2008.01674.x, 2008. 1003, 1018, 1034
Ågren, G. and Bosatta, E.: Theoretical ecosystem ecology: understanding element cycles, Cambridge University Press, 234 pp., 1998. 995, 1006
Andren, O. and Katterer, T.: ICBM: The Introductory Carbon Balance Model for Exploration of Soil Carbon Balances, Ecol. Appl., 7, 1226–1236, http://www.jstor.org/stable/2641210, 1997. 1011, 1012, 1013, 1014, 1036
Bolker, B. M., Pacala, S. W., and Parton, W. J.: Linear analysis of soil decomposition: insights from the CENTURY model, Ecol. Appl., 8, 425–439, doi:10.1890/1051-0761(1998)008[0425:LAOSDI]2.0.CO;2, 1998. 1000, 1001, 1002
Bruun, S., Six, J., Jensen, L., and Paustian, K.: Estimating turnover of soil organic carbon fractions based on radiocarbon measurements, Radiocarbon, 47, 99–113, 2008. 1002
Burger, M., Juenemann, K., and Koenig, T.: RUnit: R Unit test framework, R package version 0.4, 25, 2009. 1009
Burke, I., Kaye, J., Bird, S., Hall, S., McCulley, R., and Sommerville, G.: Evaluating and testing models of terrestrial biogeochemistry: the role of temperature in controlling decomposition, Models in ecosystem science. Princeton University Press, Princeton, New Jersey, USA, 225–253, 2003. 1003, 1034
Chambers, J.: Software for data analysis: Programming with R, Springer Verlag, 498 pp., 2008. 997, 1007
Coleman, K. and Jenkinson, D.: ROTHC-26.3 A model for the turnover of carbon in soils, Rothamsted Research, 47 pp., 1999. 1017
Cramer, W., Bondeau, A., Woodward, F. I., Prentice, I. C., Betts, R. A., Brovkin, V., Cox, P. M., Fisher, V., Foley, J. A., Friend, A. D., Kucharik, C., Lomas, M. R., Ramankutty, N., Sitch, S., Smith, B., White, A., and Young-Molling, C.: Global response of terrestrial ecosystem structure and function to CO_2 and climate change: results from six dynamic global vegetation models, Glob. Change Biol., 7, 357–373, doi:10.1046/j.1365-2486.2001.00383.x, 2001. 1019
Easter, M., Paustian, K., Killian, K., Williams, S., Feng, T., Al-Adamat, R., Batjes, N., Bernoux, M., Bhattacharyya, T., Cerri, C., Cerri, C., Coleman, K., Falloon, P., Feller, C., Gicheru, P., Kamoni, P., Milne, E., Pal, D., Powlson, D., Rawajfih, Z., Sessay, M., and Wokabi, S.: The GEFSOC soil carbon modelling system: A tool for conducting regional-scale soil carbon inventories and assessing the impacts of land use change on soil carbon, Agriculture, Ecosys-

GMDD
5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

tems & Environment, 122, 13–25, doi:10.1016/j.agee.2007.01.004, http://www.sciencedirect.com/science/article/pii/S01678809070%00333, 2007. 1018

Edgewall Software: The Trac User and Administration Guide 0.12, http://trac.edgewall.org/wiki/TracGuide, 2011. 1009

Foley, J.: An equilibrium model of the terrestrial carbon budget, Tellus B, 47, http://www.tellusb.net/index.php/tellusb/article/view/16050, 2011. 1034

Grosso, S. D., Parton, W., Mosier, A., Holland, E., Pendall, E., Schimel, D., and Ojima, D.: Modeling soil CO_2 emissions from ecosystems, Biogeochem., 73, 71–91, doi:10.1007/s10533-004-0898-z, 2005. 1034

Harmon, M. and Domingo, J.: A user's guide to STANDCARB version 2.0: a model to simulate the carbon stores in forest stands, Department of Forest Science, Oregon State University, Corvallis, Oregon, 2001. 1034

Held, I. M.: The Gap between Simulation and Understanding in Climate Modeling, B. Am. Meteorol. Soc., 86, 1609–1614, doi:10.1175/BAMS-86-11-1609, 2005. 995, 1020

Henin, S., Monnier, G., and Turc, L.: Un aspect de la dynamique des matieres organiques du sol, Comptes rendus hebdomadaires des séances de l'Académie des sciences, 248, 138–141, 1959. 999, 1011

Hocking, T. D., Wutzler, T., Ponting, K., and Grosjean, P.: Sustainable, Extensible Documentation Generation using inlinedocs, J. Statist. Softw., in review, 2012. 1009

Jacobson, M.: Earth system science: from biogeochemical cycles to global change, Academic Press, 527 pp., 2000. 994

Jenkinson, D. S. and Rayner, J. H.: The Turnover of Soil Organic Matter in Some of the Rothamsted Classical Experiments, Soil Sci., 123, 298–305, http://journals.lww.com/soilsci/Fulltext/1977/05000/The_Turno%ver_of_Soil_Organic_Matter_in_Some_of_the.5.aspx, 1977. 1000, 1015

Jenkinson, D. S., Andrew, S. P. S., Lynch, J. M., Goss, M. J., and Tinker, P. B.: The Turnover of Organic Carbon and Nitrogen in Soil, Philos. T. Roy. Soc. B., 329, 361–368, http://www.jstor.org/stable/76840, 1990. 1000, 1015, 1017, 1034

Jenny, H., Gessel, S. P., and Bingham, F. T.: Comparative Study of Decomposition Rates of Organic Matter in Temperate and Tropical Regions, Soil Sci., 68, 419–432, http://journals.lww.com/soilsci/Fulltext/1949/12000/Comparative_Study_of_Decomposition_Rates_of.1.aspx, 1949. 998

GMDD

5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

Jost, J.: Dynamical systems: examples of complex behaviour, Springer Verlag, Berlin, 189 pp., 2005. 996

Kelly, R. H., Parton, W. J., Hartman, M. D., Stretch, L. K., Ojima, D. S., and Schimel, D. S.: Intra-annual and interannual variability of ecosystem processes in shortgrass steppe, J. Geophys. Res., 105, 20093–20100, doi:10.1029/2000JD900259, http://dx.doi.org/10.1029/2000JD900259, 2000. 1034

Kirschbaum, M. U.: The temperature dependence of soil organic matter decomposition, and the effect of global warming on soil organic C storage, Soil Biology and Biochemistry, 27, 753–760, doi:10.1016/0038-0717(94)00242-S, http://www.sciencedirect.com/science/article/pii/003807179400%242S, 1995. 1034

Knuth, D. E.: Literate Programming, The Computer Journal, 27, 97–111, http://comjnl.oxfordjournals.org/content/27/2/97.abstract, 1984. 997, 1009

Knutti, R.: The end of model democracy?, Climatic Change, 102, 395–404, doi:10.1007/s10584-010-9800-2, 2010. 1019

Knutti, R., Furrer, R., Tebaldi, C., Cermak, J., and Meehl, G. A.: Challenges in Combining Projections from Multiple Climate Models, J. Climate, 23, 2739–2758, doi:10.1175/2009JCLI3361.1, 2009. 1019

Leisch, F.: Sweave, part I: Mixing R and LaTeX, R News, 2, 28–31, 2002. 1009

Leisch, F.: Sweave, part II: Package vignettes, R News, 3, 21–24, 2003. 1009

LeVeque, R.: Finite difference methods for ordinary and partial differential equations, Society for Industrial and Applied Mathematics, Philadelphia, 2007. 1007

Lloyd, J. and Taylor, J. A.: On the Temperature Dependence of Soil Respiration, Funct. Ecol., 8, 315–323, http://www.jstor.org/stable/2389824, 1994. 1034

Luo, Y. and Weng, E.: Dynamic disequilibrium of the terrestrial carbon cycle under global change, Trends in Ecol. Evol., 26, 96–104, doi:10.1016/j.tree.2010.11.003, http://www.sciencedirect.com/science/article/pii/S01695347100%02661, 2011. 1001

Manzoni, S. and Porporato, A.: A theoretical analysis of nonlinearities and feedbacks in soil carbon and nitrogen cycles, Soil Biol. Biochem., 39, 1542–1556, 2007. 1021

Manzoni, S. and Porporato, A.: Soil carbon and nitrogen mineralization: Theory and models across scales, Soil Biol. Biochem., 41, 1355–1379, doi:10.1016/j.soilbio.2009.02.031, http://www.sciencedirect.com/science/article/pii/S00380717090%00765, 2009. 994, 995, 999, 1000, 1001, 1002, 1019, 1021

Manzoni, S., Katul, G. G., and Porporato, A.: Analysis of soil carbon transit times and age distributions using network theories, J. Geophys. Res., 114, doi:10.1029/2009JG001070, http://dx.doi.org/10.1029/2009JG001070, 2009. 1002

Means, J. E., Cromack Jr, K., and MacMillan, P. C.: Comparison of decomposition models using wood density of Douglas-fir logs, Canadian J. Forest Res., 15, 1092–1098, doi:10.1139/x85-178, http://www.nrcresearchpress.com/doi/abs/10.1139/x85-178, 1985. 1000

Melillo, J., Borchers, J., and Chaney, J.: Vegetation/ecosystem modeling and analysis project: Comparing biogeography and geochemistry models in a continental-scale study of terrestrial ecosystem responses to climate change and CO_2 doubling, Glob. Biogeochem. Cy., 9, 407–438, 1995. 1019

Minderman, G.: Addition, Decomposition and Accumulation of Organic Matter in Forests, J. Ecol., 56, 355–362, http://www.jstor.org/stable/2258238, 1968. 999, 1000

Nikiforoff, C.: Some general aspects of the chernozem formation, Soil Sci. Soc. Amer. J., 1, 333–342, 1936. 998

Olson, J. S.: Energy Storage and the Balance of Producers and Decomposers in Ecological Systems, Ecology, 44,322–331, http://www.jstor.org/stable/1932179, 1963. 999

O'Sullivan, B.: Mercurial: the definitive guide, Oreilly & Associates Inc, http://mercurial.selenic.com, 2009. 1009

Parton, W., Schimel, D., Cole, C., and Ojima, D.: Analysis of factors controlling soil organic matter levels in Great Plains grasslands, Soil Sci. Soc. Am. J., 51, 1173–1179, 1987. 1000

Parton, W. J., Morgan, J. A., Kelly, R. H., and Ojima, D. S.: Modeling soil C responses to environmental change in grassland systems, in: The potential of US grazing lands to sequester carbon and mitigate the greenhouse effect, edited by: Follett, R., Kimble, J., and Lal, R., 371–398, CRC, 2001. 1034

Paul, E. and Clark, F.: Soil microbiology and biochemistry, Academic Press, San Diego, USA, 340 pp., 1996. 1000

Paustian, K., Ågren, G., and Bosatta, E.: Modelling litter quality effects on decomposition and soil organic matter dynamics, in: Driven by nature: plant litter quality and decomposition, edited by: Cadisch, G. and Giller, K. E., 313–335, CABI Publishing, UK, 1997. 994, 1000

R Development Core Team: R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org/, ISBN 3-900051-07-0, 2011. 996

GMDD
5, 993–1039, 2012

Randerson, J. T., Hoffman, F. M., Thornton, P. E., Mahowald, N. M., Lindsay, K., Lee, Y.-H., Nevison, C. D., Doney, S. C., Bonan, G., Stöckli, R., Covey, C., Running, S. W., and Fung, I. Y.: Systematic assessment of terrestrial biogeochemistry in coupled climate–carbon models, Global Change Biology, 15, 2462–2484, doi:10.1111/j.1365-2486.2009.01912.x, 2009. 1019

Schlesinger, W.: Biogeochemistry: An Analysis of Global Change, Academic Press, 588 pp., 1997. 994

Sierra, C. A., Harmon, M. E., and Perakis, S. S.: Decomposition of heterogeneous organic matter and its long-term stabilization in soils, Ecol. Monogr., 81, 619–634, doi:10.1890/11-0811.1, 2011. 995

Soetaert, K., Petzoldt, T., and Setzer, R. W.: Solving Differential Equations in R: Package deSolve, J. Stat. Softw., 33, 1–25, http://www.jstatsoft.org/v33/i09, 2010. 1007, 1009

Swift, M., Heal, O., and Anderson, J.: Decomposition in terrestrial ecosystems, University of California Press, http://books.google.de/books?id=hSWMkhdSfPAC, 1979. 994, 999, 1002

SymPy Development Team: SymPy: Python library for symbolic mathematics http://www.sympy.org., 2012. 1009

Van't Hoff, J.: Etudes de dynamique chimique, Frederick Muller & Co., Amsterdam, 214 pp., 1884. 999

Weedon, G. P., Gomes, S., Viterbo, P., Shuttleworth, W. J., Blyth, E., Österle, H., Adam, J. C., Bellouin, N., Boucher, O., and Best, M.: Creation of the WATCH Forcing Data and Its Use to Assess Global and Regional Reference Crop Evaporation over Land during the Twentieth Century, J. Hydrometeorol., 12, 823–848, doi:10.1175/2011JHM1369.1, 2011. 1018, 1039

Wu, L. and McGechan, M.: A Review of Carbon and Nitrogen Processes in Four Soil Nitrogen Dynamics Models, J. Agr. Eng. Res., 69, 279–305, doi:10.1006/jaer.1997.0250, http://www.sciencedirect.com/science/article/pii/S00218634979%0250X, 1998. 994, 1019

Wutzler, T. and Reichstein, M.: Colimitation of decomposition by substrate and decomposers – a comparison of model formulations, Biogeosciences, 5, 749–759, doi:10.5194/bg-5-749-2008, 2008. 1021

GMDD
5, 993–1039, 2012

SoilR v. 1.0

C. A. Sierra et al.

Table 1. Functions implemented in SoilR to represent the effects of temperature T, and moisture W on decomposition rates.

$f(x)$	Terms	Function name	Source
$f(T) =$			
$Q_{10}^{(T-10)/10}$	T: mean temperature	`fT.Q10`	
$\frac{47.9}{1+\exp\left(\frac{106}{T+18.3}\right)}$	T: monthly temperature (°C)	`fT.RothC`	(Jenkinson et al., 1990)
$\left(\frac{T_{max}-T}{T_{max}-T_{opt}}\right)^{0.2} \exp\left(\frac{0.2}{2.63}\left(1-\left(\frac{T_{max}-T}{T_{max}-T_{opt}}\right)^{2.63}\right)\right)$	T, T_{max} T_{opt}: monthly average, maximum, and optimal temperature	`fT.Century1`	(Burke et al., 2003)
$3.439\exp\left(\frac{0.2}{2.63}\left(1-\left(\frac{T_{max}-T}{T_{max}-T_{opt}}\right)^{2.63}\right)\left(\frac{T_{max}-T}{T_{max}-T_{opt}}\right)^{0.2}\right)$	T, T_{max} T_{opt}: monthly average, maximum, and optimal temperature	`fT.Century2`	(Adair et al., 2008)
$0.8\exp(0.095T_s)$	T_s: Soil temperature	`fT.Daycent1`	(Kelly et al., 2000)
$0.56+(1.46\arctan(\pi 0.0309(T_s-15.7)))/\pi$	T_s: Soil temperature	`fT.Daycent2`	(Parton et al., 2001; Grosso et al., 2005)
$0.198+0.036T$	T: monthly temperature	`fT.linear`	(Adair et al., 2008)
$\exp\left(308.56\left(\frac{1}{56.02}-\frac{1}{(T+273)-227.13}\right)\right)$	T: monthly temperature	`fT.LandT`	(Lloyd and Taylor, 1994)
$\exp(-3.764+0.204T(1-0.5T/36.9))$	T: mean temperature	`fT.KB`	(Kirschbaum, 1995)
$\exp((\ln(Q_{10})/10)(T-20))$	T: mean temperature. Q_{10}: temperature coefficient	`fT.Demeter`	(Foley, 2011)
$\exp(-(T/(T_{opt}+T_{lag}))^{T_{shape}})Q_{10}^{(T-10)/10}$	T, T_{max} T_{opt}: monthly average, maximum, and optimal temperature	`fT.Standcarb`	(Harmon and Domingo, 2001)
$f(W) =$			
$\frac{1}{1+30\exp(-8.5W)}$	$W=P/\mathrm{PET}$, P: monthly precipitation, PET: monthly potential evapotranspiration	`fW.Century`	(Parton et al., 2001; Adair et al., 2008)
$\left(\frac{W-b}{a-b}\right)^{d((b-a)/(a-c))}\left(\frac{W-c}{a-c}\right)^{d}$	W: water filled pore space. a, b, c, d: empirical coefficients	`fW.Daycent1`	(Kelly et al., 2000)
$5(0.287)+(\arctan(\pi 0.009(RWC-17.47))/\pi)$	W: volumetric water content	`fW.Daycent2`	(Grosso et al., 2005)
$0.25+0.75(M/M_{sat})$	M: soil moisture. M_{sat}: saturated soil mositure	`fW.Demeter`	(Foley, 2011)
$(1-\exp(-(3/M_{min})(M+a)))^{b}\exp(-(M/(M_{max}+c))^{d})$	M, M_{min}, M_{max}: average, minimum and maximum moisture content in litter pool. a,b,c,d: empirical coefficients	`fW.Standcarb`	(Harmon and Domingo, 2001)

GMDD

5, 993–1039, 2012

Title Page
Abstract | Introduction
Conclusions | References
Tables | Figures
◂ | ▸
Back | Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

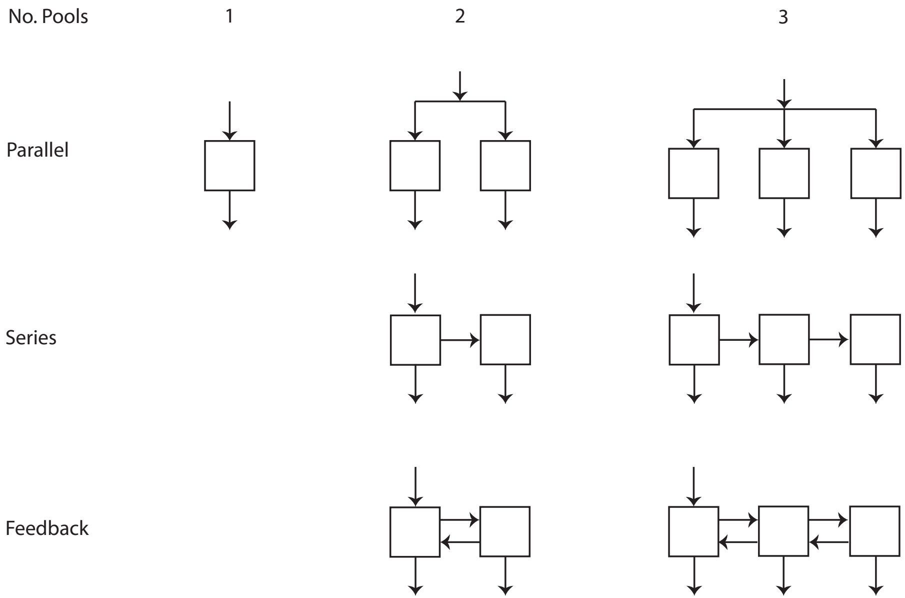

Fig. 1. Basic model structures implemented in SoilR. Squares represent the compartments, and arrows represent inputs and outputs to and from the compartments. These model structures are special cases of the matrix **A**.

GMDD

5, 993–1039, 2012

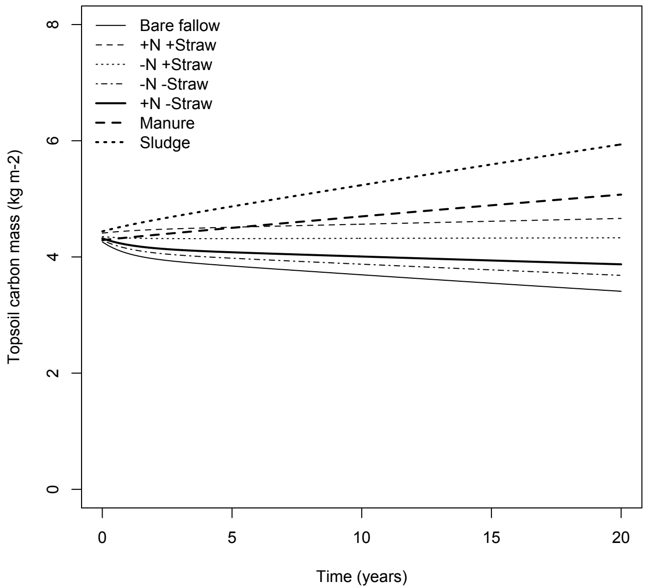

Fig. 2. Model predictions with the version of the ICBM model implemented in SoilR. This graph reproduces Fig. 2 in Andren and Katterer (1997). This figure can be reproduced typing `example(ICBMModel)` or `attr(ICBMModel,"ex")` in SoilR.

GMDD
5, 993–1039, 2012

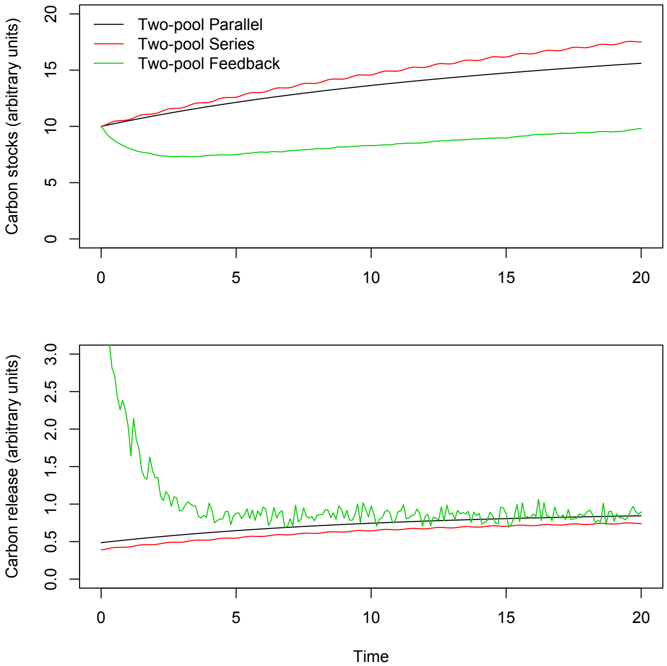

Fig. 3. Examples of three different representations of a two-pool model with different model structures and environmental effects on decomposition rates. The upper panel shows carbon stocks and the lower panel carbon release. Additional details about the implementation are given in the text.

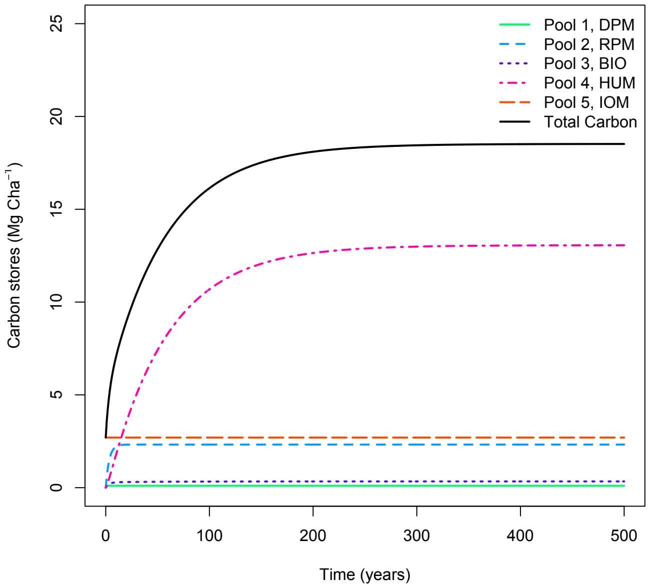

Fig. 4. Carbon accumulation for the different pools included in the RothC model. DPM: the decomposable plant material pool, RPM: resistant plant material, BIO: microbial biomass pool, HUM: humified organic matterpool, and IOM: inert organic matter pool.

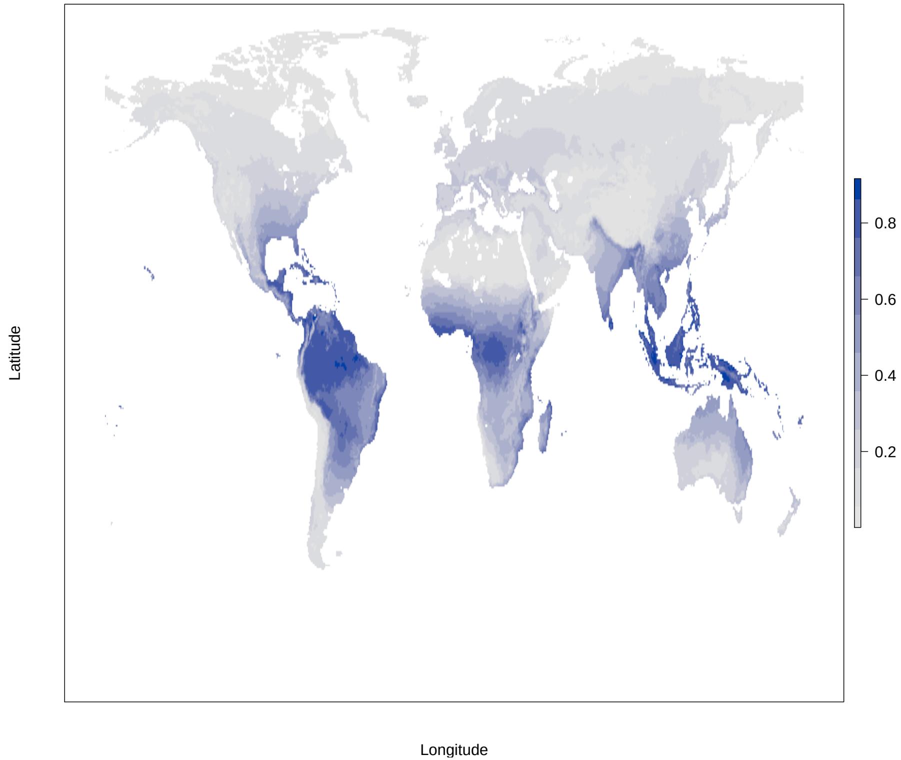

Fig. 5. Climate decomposition index (CDI) calculated as the product of a function of temperature (`fT.Century1`) and a function of precipitation and potential evapotranspiration (`fW.Century`) using monthly data from the WATCH dataset (Weedon et al., 2011).

GMDD
5, 993–1039, 2012