ENGINEERING MATHEMATICS
AND COMPUTING LAB

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Dynamic Simulation of Land Management Effects on Soil N_2O Emissions using a coupled Hydrology-Ecosystem Model on the Landscape Scale

Martin Wlotzka, Vincent Heuveline, Edwin Haas, Steffen Klatt, David Kraus, Klaus Butterbach-Bahl, Philipp Kraft, Lutz Breuer

Preprint No. 2013-03

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)
ISSN 2191–0693
Preprint No. 2013-03

Affiliation of the Authors

Martin Wlotzka[a,1], Vincent Heuveline[a], Edwin Haas[b], Steffen Klatt[b], David Kraus[b], Klaus-Butterbach-Bahl[b], Philipp Kraft[c], Lutz Breuer[c]

[a] *Engineering Mathematics and Computing Lab (EMCL),*
Karlsruhe Institute of Technology (KIT), Germany
[b] *Institute for Meteorology and Climate Research (IMK),*
Karlsruhe Institute of Technology (KIT), Germany
[c] *Institute of Landscape Ecology and Resources Management,*
Justus-Liebig-University of Giessen, Germany
[1] *Corresponding Author: Martin Wlotzka, martin.wlotzka@uni-heidelberg.de*

Impressum

Heidelberg University
Interdisciplinary Center for Scientific Computing (IWR)
Engineering Mathematics and Computing Lab (EMCL)

Speyerer Str. 6,
69115 Heidelberg
Germany

Dynamic Simulation of Land Management Effects on Soil N_2O Emissions using a coupled Hydrology-Ecosystem Model on the Landscape Scale

Martin Wlotzka, Vincent Heuveline, Edwin Haas, Steffen Klatt, David Kraus,
Klaus Butterbach-Bahl, Philipp Kraft, Lutz Breuer

December 01, 2013

Agricultural soils are the primary anthropogenic source of atmospheric N_2O. Greenhouse gas (GHG) emissions from soils are mainly the result of microbial processes such as nitrification / denitrification. These processes have a strong dependency on environmental factors like temperature, moisture, soil and vegetation properties or the land management. Therefore emissions occur with a high spatial and temporal variability giving rise to hot spots and hot moments. Quantifying sources and sinks of GHG like CO_2, N_2O and CH_4 for natural, agricultural and forest ecosystems is crucial for our understanding of impacts of land management on the biosphere-atmosphere exchange of GHG and for the development of mitigation options. GHG exchange from soils is driven by complex microbial and plant nutrient turnover processes and it is the net result of all physicochemical and biological processes involved in production, consumption and transport. Process oriented biogeochemical models are useful tools for integrating our knowledge of the key processes and drivers to estimate carbon and nitrogen (C and N) trace gas emissions from soils. In this study we have coupled the LandscapeDNDC ecosystem model to the CMF (Catchment Modelling Framework) hydrology model generating a modelling system capable to asses the C and N cycling and their feedbacks to crop growth and microbial processes on the landscape scale. The deployed coupling approach by the use of the parallel MPI based OpenPALM coupler enables the simulation of lateral exchange of nutrients (nitrate) with the soil water fluxes and therefore to assess the C and N cycling on the landscape scale. In this study we describe the coupling approach and present simulation results of crop growth, nutrient cycling and resulting nitrous oxide emissions on a virtual landscape.

1 Models

Process oriented biogeochemical models are useful tools for integrating our knowledge of the key processes and drivers to estimate carbon and nitrogen (C and N) trace gas emissions from soils. GHG exchange from soils is driven by complex microbial and plant nutrient turnover processes and it is the net result of all physicochemical and biological processes involved in production, consumption and transport.

1.1 Biogeochemistry

In the following we will outline the most important processes, which directly influence nitrous oxide (N_2O) emissions namely microbiological nitrification and denitrification and diffusive gas transport.

Nitrification is an aerobic process and thus bound to oxygen contained within the soil matrix. Denitrification, in contrast, is an anaerobic process, which is strongly inhibited in the presence of even low concentrations of oxygen [7]. Both processes depend on the so-called anaerobic volume fraction f_{av}, which is calculated in dependency on oxygen concentration.

Let $\Omega \subset \mathbb{R}$ be a one dimensional vertical soil column, then the anaerobic volume fraction is defined as

$$
\begin{aligned}
f_{\mathrm{av}} &: \Omega \to [0,1] \ , \\
f_{\mathrm{av}}(x) &:= e^{-\sqrt{\alpha c_{O_2}}} \ ,
\end{aligned}
$$

with $\alpha > 0$ being an appropriate scaling parameter. This definition is used for nitrification as well as denitrification rate calculations.

Nitrification encompasses two steps, firstly turning ammonium (NH_4^+) to nitrite (NO_2^-) and secondly turning nitrite to nitrate (NO_3^-). Nitrous oxide is produced as byproduct of the first step depending on temperature (T), water content (θ), pH and nitrification rate:

$$\partial_t c_{\mathrm{N_2O}} = K_{\mathrm{N_2O}} \;\; f_{\mathrm{tm,N_2O}}(T,\theta) \;\; f_{\mathrm{pH,N_2O}} \;\; \partial_t c_{\mathrm{NO_2}}.$$

The nitrification rate is given by

$$\partial_t c_{\mathrm{NO_2}} = K_{\mathrm{NO_2}} \;\; \frac{2\, c_{\mathrm{am}}(T,\theta)}{\frac{1}{f_{\mathrm{pH,NO_2}}} + \frac{1}{f_{\mathrm{NH_4}}}} \;\; f_{\mathrm{av}}.$$

$K_{\mathrm{N_2O}}$, $K_{\mathrm{NO_2}}$ and $K_{\mathrm{NH_4}}$ are process specific constants, which can be partly derived from experimental measurements. Following the concept of microbial activity after [2] the term $c_{\mathrm{am}}(T,\theta)$ represents the current active part of total microbial biomass (c_{m}):

$$c_{\mathrm{am}}(T,\theta) := c_{\mathrm{m}} \;\; f_{\mathrm{tm,C}}(T,\theta).$$

The microbial activity $f_{\mathrm{tm,C}}(T,\theta)$ again depends on temperature and water content. The functions $f_{\mathrm{tm},\cdot} : \mathbb{R} \times \mathbb{R}_0^+ \to [0,1]$ is given by the harmonic mean of factors accounting for temperature and water content, respectively. The water content dependency is modeled by a Weibull distribution. The terms $f_{\mathrm{pH}} : \mathbb{R}_0^+ \to [0,1]$ and $f_{\mathrm{NH_4}} : \mathbb{R}_0^+ \to [0,1]$ are reduction factors accounting for unfavourable pH and ammonium concentration.

In contrast to nitrification as nitrate production process, denitrification describes the stepwise reduction of nitrate to molecular nitrogen (N_2) via nitrite, nitric oxide (NO) and nitrous oxide. Description of denitrification was proposed in [14].

Let $\Gamma = \{\mathrm{NO}_3^-, \mathrm{NO}_2^-, \mathrm{NO}, \mathrm{N_2O}\}$ be the set of reducible N compounds γ. The terms $\partial_t c_{\gamma^-}$ denote consumption rates of the next higher oxidized N-compound. The turnover rate of each N-compound $\gamma \in \Gamma$ is given by

$$\partial_t c_\gamma = -c_{\mathrm{am}}(T,\theta) \left(\frac{\mu_\gamma}{Y_\gamma} + m_\gamma \;\; \frac{\gamma}{\sum\limits_{\gamma\in\Gamma} \gamma} \right) \;\; \frac{\sum\limits_{\gamma\in\Gamma} \gamma}{K_\gamma + \sum\limits_{\gamma\in\Gamma} \gamma} \;\; f_{\mathrm{C}} \;\; f_{\mathrm{O_2}} \;\; f_{\mathrm{pH},\gamma} \; + \; \partial_t c_{\gamma^-},$$

where μ_γ, Y_γ and m_γ are constants representing microbial growth rate, yield and maintenance respiration for each N-compound γ, respectively. As for nitrification there is for each step of denitrification a specific pH-reduction given by $f_{\mathrm{pH,N}_\gamma} : \mathbb{R}_0^+ \to [0,1]$. Moreover, denitrification is limited by carbon availability $f_{\mathrm{C}} : \mathbb{R}_0^+ \to [0,1]$. Oxygen consumption occurs due to heterotrophic and autotrophic respiration while Oyxgen supply is determined by soil diffusion, which in turn strongly depends on soil water content. The diffusion for volatile compounds $\psi \in \Psi$ is calculated using the Fickian ansatz:

$$\partial_t c_\psi = \partial_\psi [D_{\psi,\mathrm{e}}(\theta) \partial_\psi c_\psi]$$

with $D_{\psi,\mathrm{e}}(\theta)$ being the species typical diffusion coefficient D_ψ reduced by the effective diffusion coefficient $D_{\mathrm{e}}(\theta)$ depending on water content, porosity n and temperature T:

$$D_{\psi,\mathrm{e}}(\theta) := D_\psi \;\; D_{\mathrm{e}}(\theta) = D_\psi \;\; (1-\theta)^{\nu_1} \;\; n^{\nu_2} \;\; \left(\frac{T}{T_0}\right)^{\nu_3}$$

The processes described above can be written in a closed form as

$$\partial_t c = f_{\mathrm{bgc}}(\theta, c, t). \tag{1}$$

1.1.1 LandscapeDNDC

LandscapeDNDC [8] is a newly developed ecosystem model including granular functionalities for simulating biogeochemical carbon and nitrogen (C & N) cycling, plant growth and the water cycle on site and regional scale. It belongs to the process-based biogeochemical models simulating ecosystem functioning on the basis of the underlying plant physiological, soil microbial and physico-chemical processes. The model enables the combined simulation of different ecosystems of different temporal and spatial scales. All calculations are structured in a modular form representing different ecosystem components/functionalities, like plant growth, water dynamics, microclimate, soil biogeochemistry and microbiology. Land use management for agricultural and forest ecosystems such as fertilization, tillage, harvest, thinning and others can also be simulated.

For the simulation of forest ecosystems, the PNET-N-DNDC [23] – an advanced DNDC version for forest ecosystems – has been implemented and integrated into LandscapeDNDC. This includes the PNET-N-DNDCtropica for tropical forest systems from [12] and [24]. For modelling arable and grassland ecosystems, the DNDC functionalities regarding crop growth processes and agricultural management activities are included in LandscapeDNDC. In contrast to the different versions of the DNDC model (agricultural DNDC, Forest-DNDC, Wetland-DNDC) LandscapeDNDC is built upon one generalized soil bio-geochemical process description to be applied to the different ecosystem (arable, grassland, forest). The model is using meteorological data (e.g., max. and min. air temperatures, precipitation, radiation) as well as management data (e.g., seeding/harvesting, tillage, fertilizer application) with a time resolution of at least a day as driving input. Furthermore, information about soil and vegetation properties (e.g., texture, pH, crop types) serve as initialization parameters to calculate daily rates of plant N uptake, litter production, mineralisation, nitrification, denitrification and others.

1.1.2 Discretisation

LandscapeDNDC enables the simulation of ecosystem processes on the regional scale incorporating many site scale simulations (*kernels*) each representing any of the supported ecosystems. However, no spatial relationships are associated with kernels, i.e. a LandscapeDNDC simulation region is a set of independent single sites. Each site may therefore take the shape of an arbitrary polygon within an unstructured grid.

In addition to the longitude-latitude grid soil horizons are discretised according to available soil strata properties (e.g. bulk density, porosity) for various depths d_l, $(l = 1, \dots, L)$. Requiring $d_{l-1} < d_l$ and choosing $d_0 = 0$, yields strata heights $H_l = d_l - d_{l-1}$. Finally, soil layer heights h_k are given by

$$h_{\kappa(l-1)+1}, \dots, h_{\kappa(l)} = \frac{H_l}{\lambda_l} \quad \text{with} \quad \kappa(l) = \sum_{j=1}^{l} \lambda_j,$$

where $\lambda_l \in \mathbb{N}$ is an optional discretisation parameter defaulting to 1. The total number of soil layers Λ is hence equivalent to $\kappa(L)$. Note that soil layers in the same strata have initially identical properties. For the proposed modelling setup all kernels use the same soil layer discretisation.

1.1.3 Communication

Communication and neighbourhood relations amongst kernels, e.g. to address lateral transport of nutrients between adjacent kernels, is not handled by LandscapeDNDC but superimposed by external transport models by means of code coupling. An example using the hydrological model framework CMF is subject of this study.

1.1.4 Implementation

Due to the design concept of treating kernels as independent subtasks, LandscapeDNDC can store them in a simple array type data structure without any additional information. Simulations are then carried out by iterating over the list of kernels and running each one for a single time step. It is worth pointing out that having synchronized kernels is essential for the proposed coupling scheme to work.

For small numbers of kernels a nearly linear speed up can be achieved by using multi-core processing units. LandscapeDNDC uses OpenMP to execute kernels in parallel on a single multi-core CPU. For large simulations input data and kernel states require substantial amounts of memory and input data refresh time. Data refreshes occur periodically after an input buffer's content has been completely consumed. At this point input data is read from disk into each kernels input data buffers. A prominent example of *streamed* input data are driving forces like climate. Both of these issues can be elleviated by parallelizing LandscapeDNDC and deploying on multiprocessor systems or clusters. LandscapeDNDC support MPI parallelization and scales perfectly for all numbers of nodes up to the total number of kernels. Every node processes a partition of the set of kernels. Such a domain decomposition is imposed by the selection strategy for kernels used by all LandscapeDNDC instances. For example, a simple Round Robin scheme would select for node r from the full domain all kernels K where

$$K \mod P \equiv r \qquad (P \in \mathbb{N} \text{ is the number of processing units}),$$

is satisfied. Here, kernels are gaplessly numbered starting at 0.

1.2 Hydrology

Water plays an important role on the emission of trace gases. Soil moisture can form a barrier for gas exchange between soil air and atmospheric air. It promotes the formation of temporal anaerobic zones in the soil. Furthermore, water is the most important and effective transport medium for reactive nitrogen in the soil, both vertically and laterally. The formation of trace gas emission hot spots and hot moments often depends on the water quantity and quality [7].

Complex, process based biogeochemical models often include submodels for water transport in order to simulate the water filled pore space dynamically in time and the transport of nutrients [8,15,18]. However, water flow is limited to a one dimensional, vertical domain to reduce complexity in these models. Regional coverage is maintained by sequentially calculating a large number of soil columns without any interaction. But in reality, subsurface water flow and solute transport is a non-linear three-dimensional process.

1.2.1 Water Flow in Porous Media and Transport of a Dissolved Substance

Soil is a porous medium from the mathematical point of view. Porous media can be defined as a portion of space with the following properties:

- The space is occupied by a number of phases. A phase is defined as that portion of the space which is occupied by a material with uniform properties, and which is separated from other materials by a well-defined interface.
- At least one of the phases is solid.
- The phases are distributed throughout the whole space.

The solid phase of the porous media of interest in this work is the soil. The voids or open spaces between the particles of the soil are referred to as pores. The pore space is occupied by water and air, which are the two other phases.

Let the soil occupy a domain $\Omega \subset \mathbb{R}^3$. An important property of such a water-bearing formation is the porosity

$$\begin{aligned} n &: \Omega \to [0,1]\,, \\ n(x) &:= \lim_{vol(V)\to 0} \frac{\text{volume of the pores in } V}{vol(V)}\,, \end{aligned}$$

where $V \subset \Omega$ contains the point $x \in \Omega$ and $vol(V)$ is the volume of V. This definition is subject to the continuum paradox: on the one hand, the limit is necessary to allow the description of phenomena at a point by means of infinitesimal calculus; on the other hand, the volume V should contain a meaningful ensemble average over pores of many different sizes.

The volumetric water content of a soil can be defined in a similar way, namely

$$\begin{aligned} \theta &: \Omega \times (0,T) \to [0,1]\,, \\ \theta(x,t) &:= \lim_{vol(V)\to 0} \frac{\text{volume of water in } V \text{ at time } t}{vol(V)}\,, \end{aligned}$$

where $(0,T)$ with some $T > 0$ is the time interval under consideration. Clearly, $\theta \leq n$ since the maximum water content is achieved when the pores are entirely filled. This state is called full saturation and then $\theta = n$ holds by definition. In general, when water enters the soil, a certain amount of entrapped air might reside in dead end pores, which cannot be displaced by water. Then the maximum water content is denoted θ_s, which is called the state of satiation. However, the two terms are sometimes used interchangeably in the literature [4]. The residual water content θ_r consists of the moisture in dead end pores or otherwise so strongly held that it is unavailable for flow. It is convenient to define the effective saturation or wetness as

$$\theta_e := \frac{\theta - \theta_r}{\theta_s - \theta_r} \, .$$

1.2.2 Conservation of Mass

We give a brief idea how the conservation of mass leads to the basic equations which govern the water flow in porous media and the transport of a dissolved substance.
The mass m_w of the amount of water, which resides in a control volume $V \subset \Omega$ of the porous media, is given by

$$m_w = \int_V \rho\theta \, dv \, ,$$

where ρ denotes the density of the water. Similarly, the mass m_s of the solute which is dissolved in the water contained in V is given by

$$m_s = \int_V c\theta \, dv \, ,$$

where $c : \Omega \times (0,T) \to \mathbb{R}$ denotes the concentration of the solute. The physical principle of conservation of mass gives

$$\begin{aligned} \frac{dm_w}{dt} &= \frac{d}{dt}\int_V \rho\theta \, dv = \int_V \partial_t(\rho\theta) + \nabla \cdot (\rho\theta\boldsymbol{u}) \, dv = 0 \, , \\ \frac{dm_s}{dt} &= \frac{d}{dt}\int_V c\theta \, dv = \int_V \partial_t(c\theta) + \nabla \cdot (c\theta\boldsymbol{u}) \, dv = \int_V \phi \, dv \, , \end{aligned}$$

where $\boldsymbol{u} : \Omega \times (0,T) \to \mathbb{R}^3$ denotes the velocity field of the water and $\phi : \Omega \times (0,T) \to \mathbb{R}$ represents any sources or sinks of the substance. The equalities follow from the Reynolds transport theorem [1]. We denote by $\boldsymbol{q} = \theta\boldsymbol{u}$ the volumetric water flux, i.e. the volume of water crossing unit area perpendicular to the direction of flow in unit time. Since the equations above hold for any control volume $V \subset \Omega$ at any time $t \in (0,T)$, the integrands must vanish. This leads to the continuity equations

$$\partial_t\theta + \nabla \cdot \boldsymbol{q} = 0 \qquad \text{(water flow)}\, , \tag{2a}$$

$$\partial_t(c\theta) + \nabla \cdot (c\boldsymbol{q}) = \phi \qquad \text{(transport)}\, . \tag{2b}$$

The water density ρ has dropped out of Equation (2a) since it is convenient to assume it beeing constant in hydrology.

1.2.3 Darcy's Law for Saturated Flow in Porous Media

Let $p : \Omega \times (0,T) \to \mathbb{R}$ denote the water pressure in the soil. The pressure p can be expressed with an equivalent height ψ of a water column as a hydrostatic pressure $p = \rho g \psi$, where g denotes the gravitational acceleration. ψ is also called pressure head. This leads to the definition of the hydraulic head

$$\begin{aligned} &h : \Omega \times (0,T) \to \mathbb{R} \\ &h := \psi + z \, , \end{aligned}$$

where z is the coordinate on the vertical axis relative to some reference height.

In an 1856 report on the public fountains and water supply for the city of Dijon (France), Henry Darcy presented the results of his experiments on the seepage of water through a pipe filled with sand. He found that the rate of flow through a sand layer was directly proportional to the cross sectional area of the sand column and to the difference of the hydraulic head across the layer, and inversely proportional to the length of the sand column [4,22].

More precisely, Darcy's law for the saturated flow of water in porous media can be stated as

$$\boldsymbol{q} = -K\nabla h\,, \tag{3}$$

where K denotes the proportionality factor known as hydraulic conductivity and $\boldsymbol{q}$ is the volumetric flux.

1.2.4 Richards Equation for Unsaturated Flow in Porous Media

Buckingham postulated in 1907 that Darcy's law is also valid for a soil which is only partly saturated [17]. In this case the hydraulic conductivity is a function of the water content, $K = K(\theta)$ [4]. For a medium in fully saturated state $\theta = \theta_s$, the saturated conductivity is denoted $K_s := K(\theta_s)$. The relative conductivity is defined as

$$K_r := \frac{K}{K_s}\,.$$

The functional relation between the conductivity and the water content is a characteristic property of the medium. Two well-established modelizations of K_r as a function of θ_e are given as follows:

1. Brooks and Corey [3] developed the parameterization

$$K_r^{BC}(\theta_e) := \theta_e^{\epsilon}\,, \tag{4}$$

   which is based on the material-specific constant ϵ. K_r^{BC} is called Brooks-Corey retention curve.

2. Based on Mualem's model of soil water retention, Van Genuchten [6,16] developed the parameterization

$$K_r^{GM}(\theta_e) := \sqrt{\theta_e}\Big[1 - (1 - \theta_e^{\frac{1}{m}})^m\Big]^2\,, \tag{5}$$

   where the material-specific parameter $m \in (0,1)$ must be derived from experiments. K_r^{GM} is called Van-Genuchten-Mualem retention curve.

Substituting $\boldsymbol{q}$ according to Darcy's law (3) in the continuity equation (2a) yields

$$\partial_t\theta - \nabla\cdot\big(k(\theta)\nabla h\big) = 0\,, \tag{6}$$

which is known as the Richards equation [4,21].

1.2.5 Boundary Conditions

Let $\Gamma_{\mathrm{s}} \subset \partial\Omega$ denote the upward part of the boundary representing the soil surface. The boundary conditions on Γ_s reflect meteorological and environmental factors like precipitation and evaporation. Precipitation rates can be taken from recorded measurement data or from weather forecasts. Evaporation rates are calculated on the basis of recorded or forecasted soil and air temperature and sun radiation, and the wetness of the near-surface soil.

The surface boundary condition for Equation (2a) reads

$$\boldsymbol{q} = \boldsymbol{q}_{\mathrm{in}} - \boldsymbol{q}_{\mathrm{out}}(\theta) \quad \text{on } \Gamma_{\mathrm{s}}\,,$$

where $\boldsymbol{q}_{\mathrm{in}}$ represents precipitation inflow and $\boldsymbol{q}_{\mathrm{out}}$ denotes the outflow caused by evaporation.

The surface boundary condition for Equation (2b) reads

$$c\boldsymbol{q} = c_{\mathrm{in}}\boldsymbol{q}_{\mathrm{in}} \quad \text{on } \Gamma_{\mathrm{s}}\,.$$

Here, c_{in} denotes the concentration of the substance in the precipitated water.

The neighborhood of the lowest point on the boundary, where water flows out of the area of study, is denoted by Γ_{out}. We impose a Dirichlet boundary condition on the hydraulic head

$$h = h_{\mathrm{out}} \quad \text{on } \Gamma_{\mathrm{out}}\,.$$

The value of h_{out} is taken such that it causes an outflow due to a pressure difference.

The other parts of the boundary are treated as solid walls through

$$\boldsymbol{q} = 0 \quad \text{on } \partial\Omega\backslash\big(\Gamma_{\mathrm{s}} \cup \Gamma_{\mathrm{out}}\big)\,.$$

1.2.6 Finite Volume Discretization

We give a short description of a finite volume discretization of the flow and transport problem. Equations (2a) and (2b) are integrated over a control volume $V \subset \Omega$ and the divergence theorem [9] is applied, yielding

$$\partial_t \int_V \theta \, dv + \int_{\partial V} \boldsymbol{q} \cdot \boldsymbol{n} \, ds = 0 \,, \tag{7a}$$

$$\partial_t \int_V c\theta \, dv + \int_{\partial V} c\boldsymbol{q} \cdot \boldsymbol{n} \, ds = \int_V \phi \, dv \,, \tag{7b}$$

where ∂V denotes the boundary of V and $\boldsymbol{n}$ denotes the outer unit normal field on ∂V. Let the computational domain Ω be covered by a mesh $\Omega_h = \bigcup_{i=1}^{N} C_i$ of polyhedral, non-overlapping cells. The water content in cell C_i is defined as

$$w_i := \int_{C_i} \theta \, dv \; (i = 1, ..., N) \,,$$

and the solute content is

$$s_i := \int_{C_i} c\theta \, dv \; (i = 1, ..., N) \,.$$

Clearly, the average concentration of the solute in cell C_i is $c_i = \frac{s_i}{w_i}$.

Taking the control volume as any of the mesh cells, the boundary integrals in Equations (7a) and (7b) turn into sums of fluxes to adjacent cells:

$$\int_{\partial C_i} \boldsymbol{q} \cdot \boldsymbol{n} \, ds = \sum_{j \in N_i} \boldsymbol{q}_{ij} A_{ij} \; (i = 1, ..., N) \,,$$

$$\int_{\partial V} c\boldsymbol{q} \cdot \boldsymbol{n} \, ds = \sum_{j \in N_i} c_{ij} \boldsymbol{q}_{ij} A_{ij} \; (i = 1, ..., N) \,,$$

where N_i is the index set of the adjacent cells C_j sharing an interface of area A_{ij} with cell C_i. Thus, a system of ordinary differential equations for the water and solute content of the cells results:

$$\dot{w}_i + \sum_{j \in N_i} \boldsymbol{q}_{ij} A_{ij} = 0 \qquad (i = 1, ..., N) \,, \tag{8a}$$

$$\dot{s}_i + \sum_{j \in N_i} c_{ij} \boldsymbol{q}_{ij} A_{ij} = \bar{\phi}_i \qquad (i = 1, ..., N) \,, \tag{8b}$$

where $\bar{\phi}_i = \int_{C_i} \phi \, dv$. The boundary conditions can easily be incorporated in this notation.

In general, the flux $\boldsymbol{q}_{ij} = \boldsymbol{q}(w_i, w_j)$ from cell C_i to C_j is a function of the water content. For Richards equation (6), the fluxes are approximated as

$$\boldsymbol{q}_{ij} = -\sqrt{K(w_i)K(w_j)} \frac{h_i - h_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2} \,, \tag{9}$$

where $K(w_i)$ is the conductivity and h_i is the hydraulic head of cell C_i, and $\boldsymbol{x}_i$ is the coordinate vector of some reference point of cell C_i. Here, the gradient of the hydraulic head is approximated by a finite difference, and the conductivity is taken as the geometric mean of the two cells.

In Equation (8b), the value of c_{ij} is taken as

$$c_{ij} = \begin{cases} c_i & \text{if } \boldsymbol{q}_{ij} > 0 \,, \\ c_j & \text{if } \boldsymbol{q}_{ij} < 0 \,, \\ 0 & \text{else} \,, \end{cases} \tag{10}$$

which is the average concentration of the substance in the cell where the flux originates from.

For the sake of simplicity, we subsume the system of ordinary differential equations (8) in the notation

$$\partial_t(w, s) = f_{\text{hyd}}(w, s, t) \,. \tag{11}$$

1.2.7 Catchment Modelling Framework

The Catchment Modelling Framework (CMF) [13] is a C++ library for creating hydrological simulation models. It offers a variety of classes and functions which represent the ingredients of a finite volume discretization as proposed by Qu and Duffy [20]. Hydrological models are set up as a network using node and connection objects. The nodes of the network are the cells of Ω_h. They have the hydrological meaning of a water storage keeping track of the water content. They are equipped with material specific constants, boundary conditions and a position in the geometry. The connection objects represent the edges of the network between adjacent cells. They contain the definition of the flux approximation. We use the Richards approximation (9) between adjacent cells, and evaporation and rainfall on the soil surface. CMF also provides conceptual equations for water transport for large scale applications as well as flux connections for surface water, which are not shown in this paper.
For modeling the transport of dissolved substances, CMF attaches solute storage objects to each water storage. The solute storages keep track of the solute content in the cells. Both storage types provide methods to calculate the time derivative of their state. For each cell $C_i \in \Omega_h$, the corresponding water storage computes $\dot{w}_i$, and the attached solute storage computes $\dot{s}_i$ by iterating over the flux connections and evaluating the contributions using (9) and (10). Thus, the CMF model calculates the right hand side function f_{hyd} of Equation (11). This function evaluation is parallelized with OpenMP for shared memory machines. For the time integration, CMF provides a number of well known ODE solvers, ranging from a naive explicit Euler, a classical Runge-Kutta-Fehlberg method to the implicit multistep and error controlled CVODE solver by Hindmarsh et al. [10], which we use for this study.

2 Model Coupling

The space-discretized models for the biogeochemical processes resulting from (1), and the water flow and transport of dissolved substances (11) form the coupled system of ordinary differential equations

$$\partial_t(w,s) = f_{\text{hyd}}(w,s,t) \text{ in } [0,T]\,, \qquad \text{(water flow and solutes transport)} \tag{12a}$$

$$\partial_t s = f_{\text{bgc}}(w,s,t) \text{ in } [0,T]\,, \qquad \text{(biogeochemical processes)} \tag{12b}$$

$$w(0) = w_0\,, \tag{12c}$$

$$s(0) = s_0\,. \tag{12d}$$

We propose two time stepping schemes for solving the coupled system. Both schemes employ an operator splitting such that the global time steps of the coupled system are composed of local time steps of the individual models. The time interval $[0,T]$ is divided into a sequence of discrete time steps $0 = t_0 < t_1 < ... < t_{n_{max}} = T$.

2.1 Consecutive Operator Splitting

A simple global time stepping scheme uses the consecutive operator splitting described in Algorithm 1. In each cycle of the time loop, the hydrological model is solved for the current time interval (line 3). This gives the water state w_{n+1} for the next time step, and an intermediate result $s^{(\text{h})}_{n+1}$ for the solute state. These states are taken as input for the biogeochemical model. It is subsequently solved for the same time interval (line 4), yielding the solute state s_{n+1} for the next time step.

Algorithm 1 Consecutive Operator Splitting

```
1: Set initial values $w_0$, $s_0$, $n = 0$.
2: while $n < n_{max}$ do
3:     Solve $\partial_t(w,s) = f_{\text{hyd}}(w,s,t)$ in $[t_n, t_{n+1}]$, $w(t_n) = w_n$, $s(t_n) = s_n$ to obtain $w_{n+1}$, $s^{(\text{h})}_{n+1}$.
4:     Solve $\partial_t s = f_{\text{bgc}}(w_{n+1},s,t)$ in $[t_n, t_{n+1}]$, $s(t_n) = s^{(\text{h})}_{n+1}$ to obtain $s_{n+1}$.
5:     $n \leftarrow n + 1$
6: end while
```

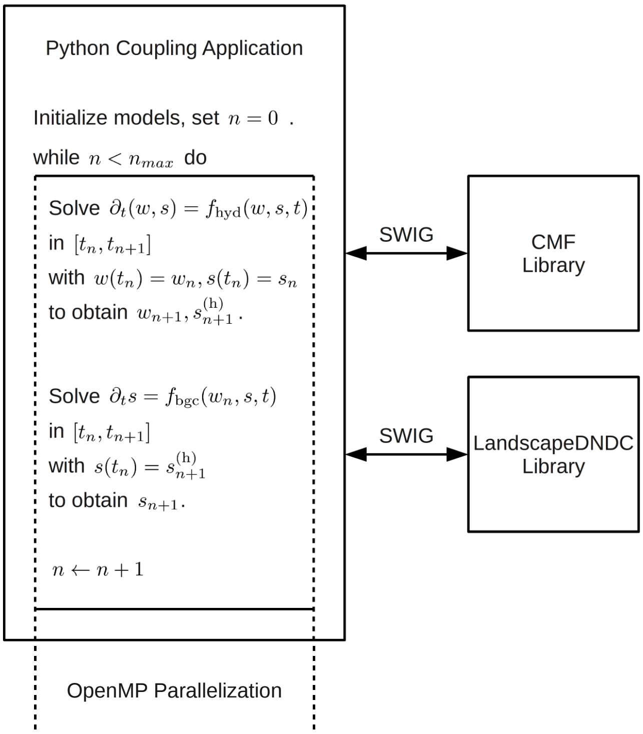
Python Coupling Application
Initialize models, set $n = 0$.
while $n < n_{max}$ do
Solve $\partial_t(w, s) = f_{\text{hyd}}(w, s, t)$
in $[t_n, t_{n+1}]$
with $w(t_n) = w_n, s(t_n) = s_n$
to obtain $w_{n+1}, s^{(\text{h})}_{n+1}$.
Solve $\partial_t s = f_{\text{bgc}}(w_n, s, t)$
in $[t_n, t_{n+1}]$
with $s(t_n) = s^{(\text{h})}_{n+1}$
to obtain s_{n+1} .
$n \leftarrow n + 1$
OpenMP Parallelization
SWIG
CMF
Library
SWIG
LandscapeDNDC
Library

Figure 1: Coupling scheme for simulations with Python.

2.2 Simulation with Python

A straightforward way to simulate the coupled system is to create an application which contains both models. We decided to implement this coupling application in Python (www.python.org). The Python programming language is freely available for a wide range of machines like laptops, workstations and clusters. We compiled the models into Python modules using SWIG (www.swig.org). The SWIG tool generates interface code for accessing the C++ models from Python. It allows to use the model objects and functions in Python applications. This makes it easy to set up simulations on various platforms, and it enables for a fast prototyping.

We implemented the consecutive operator splitting scheme in a Python coupling application. The models can exchange data by accessing global variables. The internal OpenMP parallelization of the models is not affected by the SWIG wrapping. Thus, the simulation application can be run in parallel on shared memory platforms. The python coupling scheme is depicted in Figure 1.

2.3 Concurrent Operator Splitting

A more sophisticated global time stepping scheme uses the concurrent operator splitting described in Algorithm 2. In each cycle of the time loop, the models are solved for one time interval using only input data which is available from the last time step (lines 3 and 4). This allows for concurrent computations in both models. The hydrological model produces the water state w_{n+1} for the next time step, and the intermediate result $s^{(\text{h})}_{n+1}$ for the transported solute. The biogeochemical model computes the intermediate solute state $s^{(\text{b})}_{n+1}$. The difference $s^{(\text{b})}_{n+1} - s_n$ represents the production or loss of NO_3 due to the soil chemical processes. The solute state s_{n+1} for the next time step is taken as the sum of the transported intermediate result and the production or loss term (line 5).

Algorithm 2 Concurrent Operator Splitting

```
Set initial values $w_0$, $s_0$, $n = 0$.
while $n < n_{max}$ do
    Solve $\partial_t(w, s) = f_{\text{hyd}}(w, s, t)$ in $[t_n, t_{n+1}]$, $w(t_n) = w_n$, $s(t_n) = s_n$ to obtain $w_{n+1}$, $s^{(\text{h})}_{n+1}$.
    Solve $\partial_t s = f_{\text{bgc}}(w_n, s, t)$ in $[t_n, t_{n+1}]$, $s(t_n) = s_n$ to obtain $s^{(\text{b})}_{n+1}$.
    Set $s_{n+1} = s^{(\text{h})}_{n+1} + s^{(\text{b})}_{n+1} - s_n$.
    $n \leftarrow n + 1$
end while
```

As in the consecutive operator splitting case, each of the models can be implemented in parallel. The concurrent operator splitting introduces a second level of parallelism. The models can compute the results concurrently in each iteration of the time loop, since their input depends only on data from the last time step. This allows to employ individual parallelization concepts for the models, and to run them on individual sets of processors.

2.4 Simulation with OpenPALM

In order to exploit the opportunities for parallelism offered by the concurrent operator splitting, we use the OpenPALM software coupler tool [5, 19] for the simulation. The fundamental idea of OpenPALM is to consider complex simulations as a coupled application. The OpenPALM approach is to divide the computations into a number of tasks and to define a coupling algorithm which controls the execution and interaction of these tasks. Each task can be implemented individually, offering the possibility to reuse existing codes with minimal modifications for building a computational component that can be coupled to other components.

OpenPALM features both levels of parallelism. On the one hand, independent tasks can run concurrently on separate sets of processors. OpenPALM handles the concurrent execution of such tasks, establishes an intercommunication context between them and grants a synchronization mechanism. On the other hand, OpenPALM is able to couple components which are internally parallelized, using OpenMP as well as MPI. For such components, OpenPALM establishes a private intracommunication context and manages data exchange between different sets of processes.

In order to grant modularity of the components and full flexibility for defining a coupling algorithm, OpenPALM implements the end-point communication scheme. It provides two basic communication routines for sending and receiving data objects, `PALM_Put` and `PALM_Get`, respectively. Components can use these routines to exchange data without the need to have a reference on the origin and destination of the communication. Instead, `PALM_Put` and `PALM_Get` announce a request for sending and receiving data. OpenPALM acts as a broker on these requests, i.e. it derives the actual communication pattern from the current state of the coupling algorithm and organizes a rendez-vous of the components which are involved in the communication. In order to avoid deadlocks between senders and receivers, OpenPALM provides buffer storage for pending communications.

In the OpenPALM terminology, a computational component which can be executed in a coupling algorithm is called a unit. The task represented by a unit may vary from simple algebraic operations over linear or nonlinear solvers up to complex physical models. The granularity of the tasks can be freely chosen by the user. Units are defined by so-called identity cards which can be recognized by OpenPALM. The identity card of a unit basically provides information about its data objects which may be exchanged with other units and, in case of a parallel unit, the data distribution. Units can be implemented in Fortran, C and C++. Existing codes can easily be turned into a unit by creating an appropriate identity card and by changing the original program entry point (`program` statement in Fortran or `main` function in C/C++) into a normal subroutine or function. OpenPALM spawns separate executables for calling the unit functions. Inside the unit source code, the `PALM_Put` and `PALM_Get` primitives may be used for communication with other units. According to the end-point communication scheme, units do not need to know their communication partners, but OpenPALM arranges the connection. This modularity allows both to individually develop units for a coupled application in collaborative projects and to reuse existing units in any coupling algorithm.

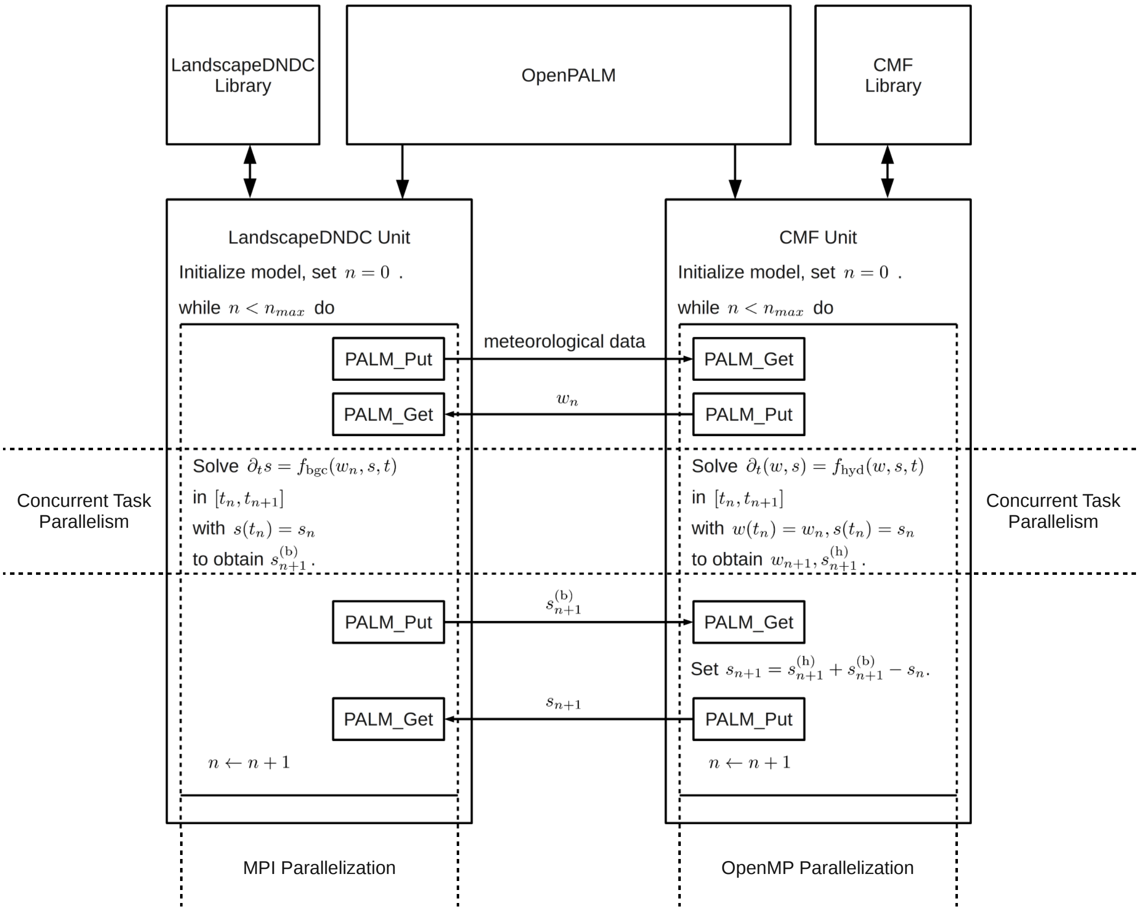
LandscapeDNDC Library
OpenPALM
CMF Library
LandscapeDNDC Unit
Initialize model, set $n = 0$.
while $n < n_{max}$ do
CMF Unit
Initialize model, set $n = 0$.
while $n < n_{max}$ do
PALM_Put
meteorological data
PALM_Get
PALM_Get
w_n
PALM_Put
Concurrent Task Parallelism
Solve $\partial_t s = f_{\text{bgc}}(w_n, s, t)$
in $[t_n, t_{n+1}]$
with $s(t_n) = s_n$
to obtain $s^{(\text{b})}_{n+1}$.
Solve $\partial_t(w, s) = f_{\text{hyd}}(w, s, t)$
in $[t_n, t_{n+1}]$
with $w(t_n) = w_n, s(t_n) = s_n$
to obtain $w_{n+1}, s^{(\text{h})}_{n+1}$.
Concurrent Task Parallelism
PALM_Put
$s^{(\text{b})}_{n+1}$
PALM_Get
Set $s_{n+1} = s^{(\text{h})}_{n+1} + s^{(\text{b})}_{n+1} - s_n$.
PALM_Get
s_{n+1}
PALM_Put
$n \leftarrow n + 1$
$n \leftarrow n + 1$
MPI Parallelization
OpenMP Parallelization

Figure 2: Coupling scheme for simulations with OpenPALM.

We built two units, one for the hydrological model using CMF, and one for the biogeochemical model using LandscapeDNDC. The CMF unit uses the OpenMP parallelization as described in Section 22.2. The MPI parallelization of the LandscapeDNDC unit is based on a domain decomposition. The biogeochemical model kernels are distributed among a number of computational nodes. Each node can advance its model kernels for one time step independent of each other. Both units use the `PALM_Put` and `PALM_Get` routines to exchange the water and solute states, and to transfer meteorological data. The coupling scheme is depicted in Figure 2.

3 Application

We designed a numerical experiment to show the feasibility of the described approach for quantifying the ecosystem carbon and nitrogen cycling on the landscape scale. The computational domain Ω_h with $N = 13120$ cells was a virtual landscape forming a valley. The horizontal discretization yielded 40 x 41 cells with a size of 10 by 10 meters each. The vertical discretization consisted of 8 soil layers of different heights (0.5, 0.5, 2, 3, 3, 3, 3 and 5 meters). We placed a stream outlet Γ_{out} with $h_{\text{out}} = 0.1$ m below ground near the lowest point of the geometry, see Figure 3. Precipitation and evaporation fluxes occured at the surface boundary Γ_{s}. All other parts of the boundary were equipped with the no-flow condition $\boldsymbol{q} = 0$. The hydraulic head was set to 0.5 meter below surface as initial condition. The initial NO_3 concentration was set to $c = 10^{-3}e^{-4d}$ kg per m^3, where d denotes the soil depth. Temporal resolution of the data exchange of the coupled system was one day.

The landscape represented an intensive agricultural system of maize rotations. Fertilization with

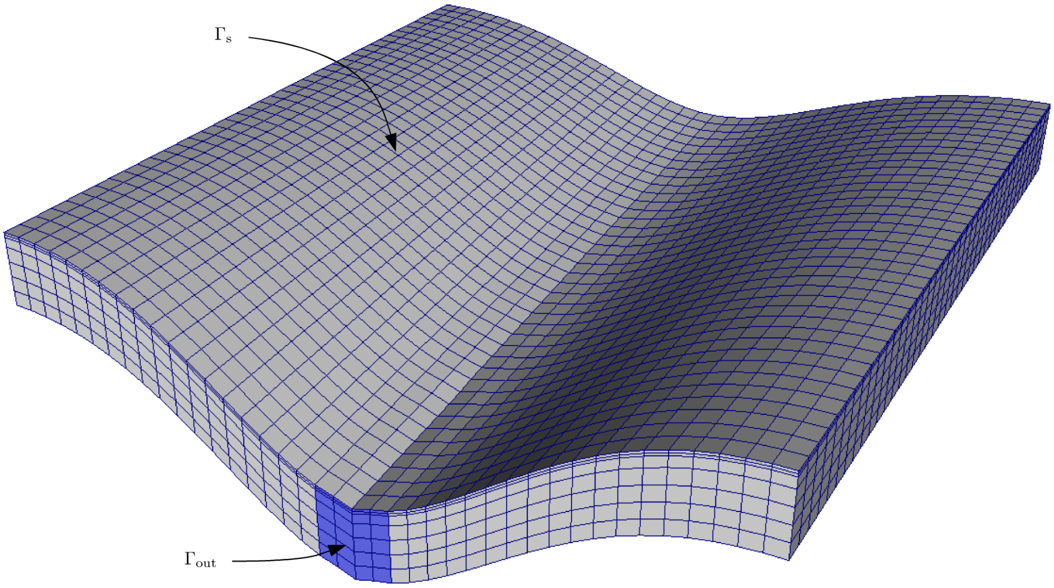

Figure 3: Computational mesh Ω_h with $N = 13120$ cells.

organic and inorganic N was more than 300 kg-N/ha/a. It was split into several applications in the growing season in spring and after harvest in autumn uniformly on the surface. The domain represented an average pre-alpine climate with approx. 800 mm average precipitation and an average temperature of 9.6 °C throughout the surface.

The geometry enforced interflow transport of nutrients from upland into the riparian zone, entering the stream which was formed at the bottom of the slopes.

Nitrate and ammonium play a significant role in plant nutrition. Due to the low mobility of ammonium, we only accounted for the nitrate transport. It was formed by microbial processes converting NH_4 via NO_2 into NO_3 at aerobic conditions as described in section 2. It was also substrate for the denitrification, the microbial transformation of NO_3 into N_2, which occured at anaerobic conditions.

In Figure 4 the soil water redistribution after a simulation time of 420 days is illustrated. Rainfall events and corresponding water fluxes along the slopes resulted in saturated soil water conditions along the riparian zone.

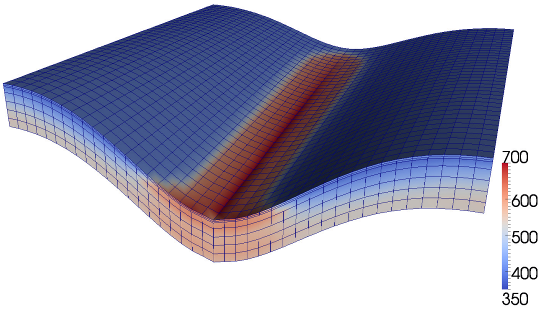

Figure 4: Soil water content [mm/m^3] after 420 days.

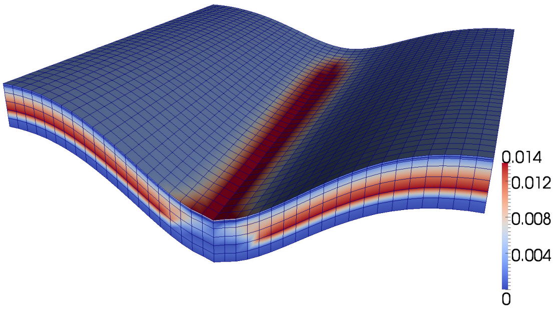

Figure 5: Nitrate (NO_3) concentration in the soil water after 420 days.

Soil nitrate concentrations following water fluxes are shown in Figure 5. Nitrate was formed by mineralization and consecutive nitrification of plant litter, and it was percolated into deeper soil layers. It was transported towards the riparian zone where most of the excess nitrate was discharged towards the surface water due to soil water saturation.

The vegetation development feedback due to the nutrient availability in the landscape resulted in a biomass production gradient along the slopes. Figure 6 illustrates the strong variations of the gross primary productivity of the plant growth. Downslope regions with large nutrient and water availability showed a strong increase in plant growth as there was no limitation by nitrogen, whereas upslope regions were less productive. The fertilization regime supplied enough excess nitrogen to be redistributed within the landscape.

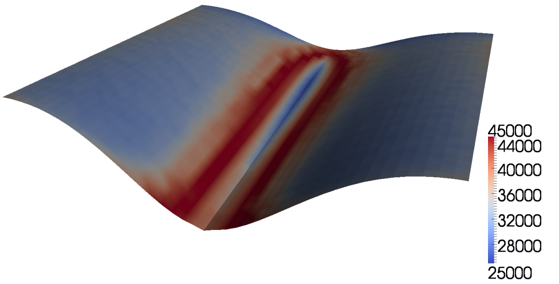

Figure 6: The spatial distribution of the plant growth / gross primary production as a result of differences in nutrient availability.

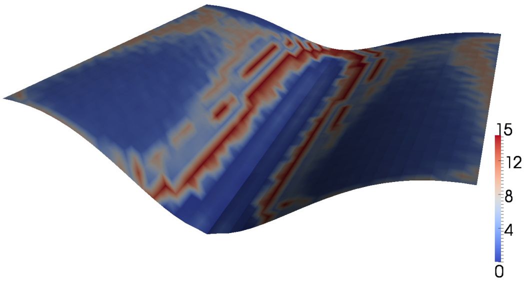

Figure 7: The spatial distribution of the nitrous oxide (N_2O) emissions as a result of the nitrification / denitrification process of nutrient availability.

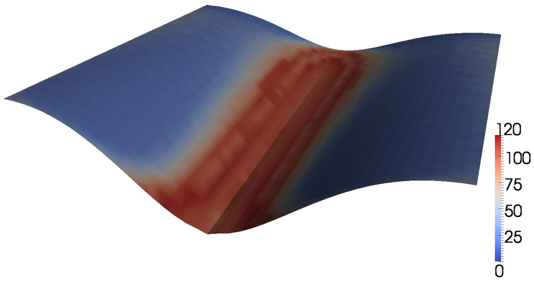

Figure 8: The spatial distribution of the dinitrogen emissions. Dinitrogen is the endproduct of the denitrification process transforming nitrate via nitrite into dinitrogen by denitrifying microorganisms at anaerobic soil conditions.

Figure 8 reflects hot spots of denitrification, which is illustrated by the yearly accumulated N_2 emissions. Fully saturated conditions in the riparian zone lead to anaerobic conditions, which in turn lead to the complete reduction of nitrate to dinitrogen. Under such conditions almost all N_2O which was produced during denitrification, was further reduced to N_2. In contrast, the transition from complete anaerobic to aerobic zones in higher altitudes yielded increased N_2O emissions because denitrification was not completed. As a consequence of this incomplete process, N_2O diffused to the surface and was emitted to the atmosphere. The transition zone between aerobic and anaerobic regions as well as the temporal availability of nitrogen varied. These variations caused emission patterns that form hot spot regions around the riparian zone as illustrated in figures 7 and 8.

4 Conclusion

The results of the simulation with the coupled model system show that considering lateral transport of water and nutrients exposes more realistic regional emission patterns due to spatial gradients in nutrient availability and their feedbacks to crop growth and microbial activity.

Despite the lack of validation of the modelling approach, the presented simulation results highlight the effect of generating indirect N_2O emissions caused by lateral nutrient redistribution and near-surface water eutrophication.

These effects are well known and have been observed before. IPCC [11] acknowledges them in its methodologies to asses the source strengths of the associated GHG emissions by the use of specific emission factors. These factors are constant and do not reflect any local conditions, e.g. climate, management, hydrology and soil properties. The coupled simulations do reflect many effects associated with the nutrient transport and have therefore proved to be a powerful tool to asses the effects of nutrient cycling within landscapes. This is a fundamental requirement for the identification of hot spots and hot moments of GHG emissions and for their mitigation.

References

[1] R. Aris. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics.* Dover, 1989.

[2] S.A. Blagodatsky and O. Richter. Microbial growth in soil and nitrogen turnover: a theoretical model considering the activity state of microorganisms. *Soil Biology and Biochemistry*, 30(13):1743 – 1755, 1998.

[3] R. H. Brooks and A. T. Corey. Hydraulic Properties of Porous Media. *Hydrology Papers No. 3, Colorado State University*, 1964.

[4] W. Brutsaert. *Hydrology.* Cambridge University Press, 2005.

[5] S. Buis, A. Piacentini, D. Declat, and the PALM Group. Palm: A computational framework for assembling high performance computing applications. *Concurrency and Computation: Practice and Experience*, 18:231–245, 2006.

[6] M. T. Van Genuchten. A Closed Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Science Society of America Journal*, pages 892–898, 1980.

[7] Peter M Groffman, Klaus Butterbach-Bahl, Robinson W Fulweiler, Arthur J Gold, Jennifer L Morse, Emilie K Stander, Christina Tague, Christina Tonitto, and Philippe Vidon. Challenges to incorporating spatially and temporally explicit phenomena (hotspots and hot moments) in denitrification models. *Biogeochemistry*, 93(1-2):49–77, 2009.

[8] E. Haas, S. Klatt, A. Froehlich, P. Kraft, C. Werner, R. Kiese, R. Grote, L. Breuer, and K. Butterbach-Bahl. LandscapeDNDC: a process model for simulation of biosphere-atmosphere-hydrosphere exchange processes at site and regional scale. *Landscape Ecology*, 2012.

[9] H. Heuser. *Lehrbuch der Analysis.* Teubner, 2002.

[10] Hindmarsh, A. C. and Brown, P. N. and Grant, K. E. and Lee, S. L. and Serban, R. and Shumaker, D. E. and Woodward, C. S. . "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers," ACM Transactions on Mathematical Software. Technical report, LLNL technical report UCRL-JP-200037, 2005.

[11] IPCC. Climate change 2007: The physical science basis - Contribution of working group to the fourth assessment report of the intergovernmental panel on climate change. Technical report, Intergovernmental Panel on Climate Change, Cambridge, United Kingdom and New York, NY, USA, 2007.

[12] Ralf Kiese, Changsheng Li, David W. Hilbert, Hans Papen, and Klaus Butterbach-Bahl. Regional application of pnet-n-dndc for estimating the n2o source strength of tropical rainforests in the wet tropics of australia. *Global Change Biology*, 11(1):128–144, 2005.

[13] P. Kraft, K. B. Vach, H.-G. Frede, and L. Breuer. A hydrological programming language extension for integrated catchment models. *Environmental Modelling and Software*, 2011.

[14] P A Leffelaar and W W Wessel. Denitrification in A Homogeneous, Closed System: Experiment and Simulation. *Soil Science*, 146(5), 1988.

[15] Changsheng Li, Steve Frolking, and Tod A Frolking. A model of nitrous-oxide evolution from soil driven by rainfall events. 1. Model structure and sensitivity. *Journal of Geophysical Research*, 97(D9):9759–9776, 1992.

[16] Y. Mualem. A New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Media. *Water Resources Research*, 12:513–522, 1976.

[17] T. N. Narasimhan. Buckingham, 1907: An Appreciation. *Vadose Zone Journal*, pages 434–441, 2005.

[18] Parton, W.J., D.S. Ojima, C.V. Cole, D.S. Schimel. A general model for soil organic matter dynamics: Sensitivity to litter chemistry, texture and management, Quantitative Modeling of Soil Forming Processes. Technical report, Soil Science Society of America, 1994.

[19] A. Piacentini and the PALM Group. Palm: A dynamic parallel coupler. *Lecture Notes in Computer Science*, 2565:479–492, 2003.

[20] Y Qu and C J Duffy. A semidiscrete finite volume formulation for multiprocess watershed simulation. *Water Resources Research*, 2007.

[21] L. A. Richards. Capillary Conduction of Liquids through Porous Mediums. *Physics*, 1:318–333, 1931.

[22] C.T. Simmons. Henry Darcy (1803-1858): Immortalised by his scientific legacy. *Hydrogeology Journal*, pages 1023–1038, 2008.

[23] F Stange, K Butterbach-Bahl, H Papen, S Zechmeister-Boltenstern, CS Li, and J Aber. A process-oriented model of N2O and NO emissions from forest soils 2. Sensitivity analysis and validation. *Journal Of Geophysical Research-Atmospheres*, 105:4385–4398, 2000.

[24] C. Werner, K. Butterbach-Bahl, E. Haas, T. Hickler, and R. Kiese. A global inventory of n2o emissions from tropical rainforest soils using a detailed biogeochemical model. *Global Biogeochemical Cycles*, 21(3), 2007.

Preprint Series of the Engineering Mathematics and Computing Lab

recent issues

No. 2013-02 Martin Baumann, Jochen Förstner, Jonas Kratzke, Sebastian Ritterbusch, Bernhard Vogel, Heike Vogel: Model-based Visualization of Instationary Geo-Data with Application to Volcano Ash Data

No. 2013-01 Martin Schindewolf, Björn Rocker, Wolfgang Karl, Vincent Heuveline: Evaluation of two Formulations of the Conjugate Gradients Method with Transactional Memory

No. 2012-07 Andreas Helfrich-Schkarbanenko, Vincent Heuveline, Roman Reiner, Sebastian Ritterbusch: Bandwidth-Efficient Parallel Visualization for Mobile Devices

No. 2012-06 Thomas Henn, Vincent Heuveline, Mathias J. Krause, Sebastian Ritterbusch: Aortic Coarctation simulation based on the Lattice Boltzmann method: benchmark results

No. 2012-05 Vincent Heuveline, Eva Ketelaer, Staffan Ronnas, Mareike Schmidtobreick, Martin Wlotzka: Scalability Study of HiFlow^3 based on a Fluid Flow Channel Benchmark

No. 2012-04 Hartwig Anzt, Armen Beglarian, Suren Chilingaryan, Andrew Ferrone, Vincent Heuveline, Andreas Kopmann: A unified Energy Footprint for Simulation Software

No. 2012-03 Vincent Heuveline, Chandramowli Subramanian: The Coffee-table Book of Pseudospectra

No. 2012-02 Dominik P.J. Barz, Hendryk Bockelmann, Vincent Heuveline: Electrokinetic optimization of a micromixer for lab-on-chip applications

No. 2012-01 Sven Janko, Björn Rocker, Martin Schindewolf, Vincent Heuveline, Wolfgang Karl: Software Transactional Memory, OpenMP and Pthread implementations of the Conjugate Gradients Method - a Preliminary Evaluation

No. 2011-17 Hartwig Anzt, Jack Dongarra, Vincent Heuveline, Piotr Luszczek: GPU-Accelerated Asynchronous Error Correction for Mixed Precision Iterative Refinement

No. 2011-16 Vincent Heuveline, Sebastian Ritterbusch, Staffan Ronnås: Augmented Reality for Urban Simulation Visualization

No. 2011-15 Hartwig Anzt, Jack Dongarra, Mark Gates, Stanimire Tomov: Block-asynchronous multigrid smoothers for GPU-accelerated systems

No. 2011-14 Hartwig Anzt, Jack Dongarra, Vincent Heuveline, Stanimire Tomov: A Block-Asynchronous Relaxation Method for Graphics Processing Units

No. 2011-13 Vincent Heuveline, Wolfgang Karl, Fabian Nowak, Mareike Schmidtobreick, Florian Wilhelm: Employing a High-Level Language for Porting Numerical Applications to Reconfigurable Hardware

No. 2011-12 Vincent Heuveline, Gudrun Thäter: Proceedings of the 4th EMCL-Workshop Numerical Simulation, Optimization and High Performance Computing

No. 2011-11 Thomas Gengenbach, Vincent Heuveline, Mathias J. Krause: Numerical Simulation of the Human Lung: A Two–scale Approach

The responsibility for the contents of the working papers rests with the authors, not the Institute. Since working papers are of a preliminary nature, it may be useful to contact the authors of a particular working paper about results or caveats before referring to, or quoting, a paper. Any comments on working papers should be sent directly to the authors.

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)

$\frac{\partial u}{\partial t} - \nu\Delta u$
$+ (u \cdot \nabla) u + \frac{1}{\rho} \nabla p$
$\frac{\partial u}{\partial t}$

www.emcl.iwr.uni-heidelberg.de