



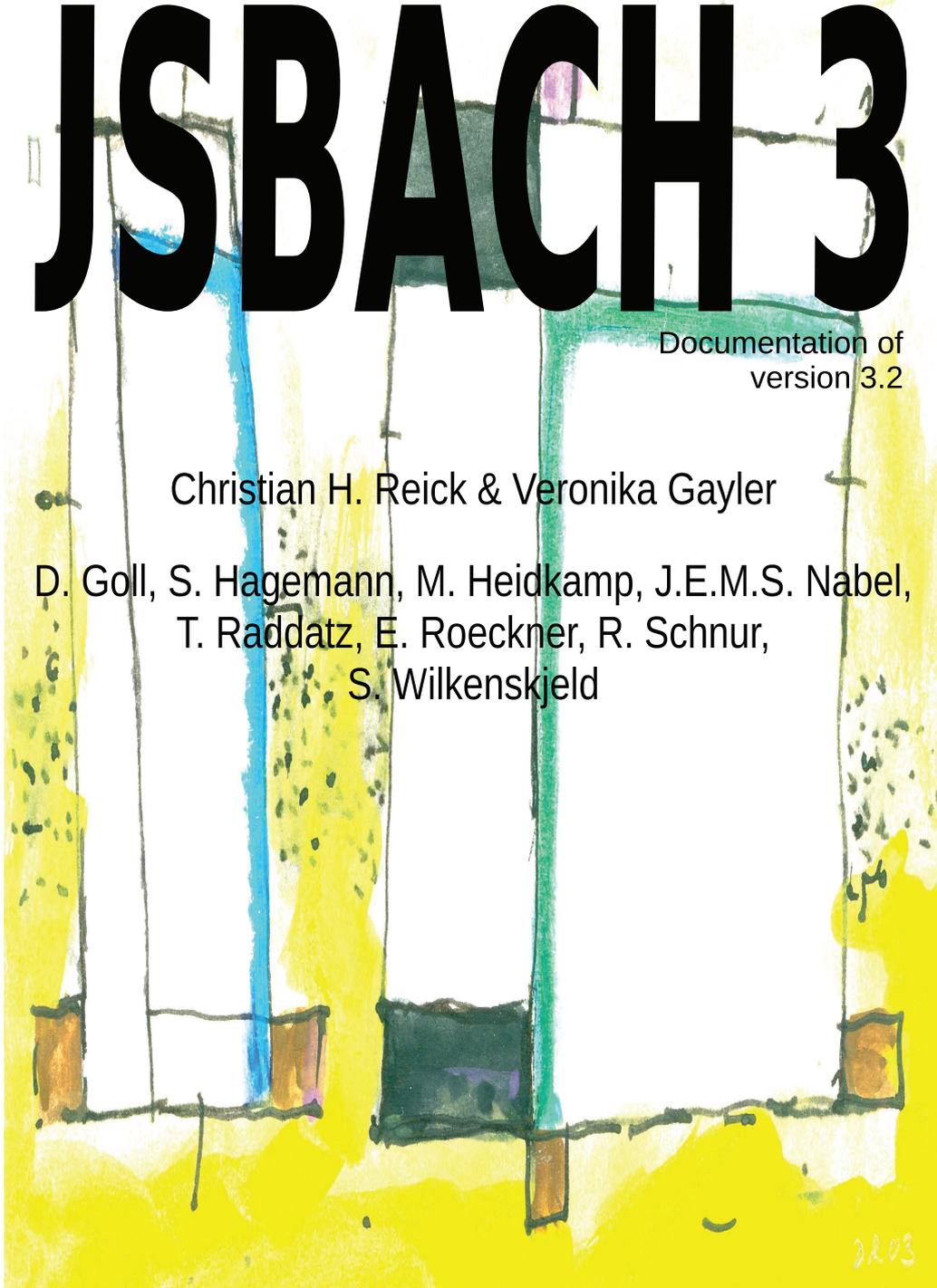
The land component of the MPI Earth System Model

# JSBACH 3

Documentation of  
version 3.2

Christian H. Reick & Veronika Gayler

D. Goll, S. Hagemann, M. Heidkamp, J.E.M.S. Nabel,  
T. Raddatz, E. Roeckner, R. Schnur,  
S. Wilkenskjeld



## Hinweis

Die Berichte zur Erdsystemforschung werden vom Max-Planck-Institut für Meteorologie in Hamburg in unregelmäßiger Abfolge herausgegeben.

Sie enthalten wissenschaftliche und technische Beiträge, inklusive Dissertationen.

Die Beiträge geben nicht notwendigerweise die Auffassung des Instituts wieder.

Die "Berichte zur Erdsystemforschung" führen die vorherigen Reihen "Reports" und "Examensarbeiten" weiter.

## Anschrift / Address

Max-Planck-Institut für Meteorologie  
Bundesstrasse 53  
20146 Hamburg  
Deutschland

Tel./Phone: +49 (0)40 4 11 73 - 0  
Fax: +49 (0)40 4 11 73 - 298

name.surname@mpimet.mpg.de  
www.mpimet.mpg.de

## Notice

The Reports on Earth System Science are published by the Max Planck Institute for Meteorology in Hamburg. They appear in irregular intervals.

They contain scientific and technical contributions, including Ph. D. theses.

The Reports do not necessarily reflect the opinion of the Institute.

The "Reports on Earth System Science" continue the former "Reports" and "Examensarbeiten" of the Max Planck Institute.

## Layout

Bettina Diallo and Norbert P. Noreiks  
Communication

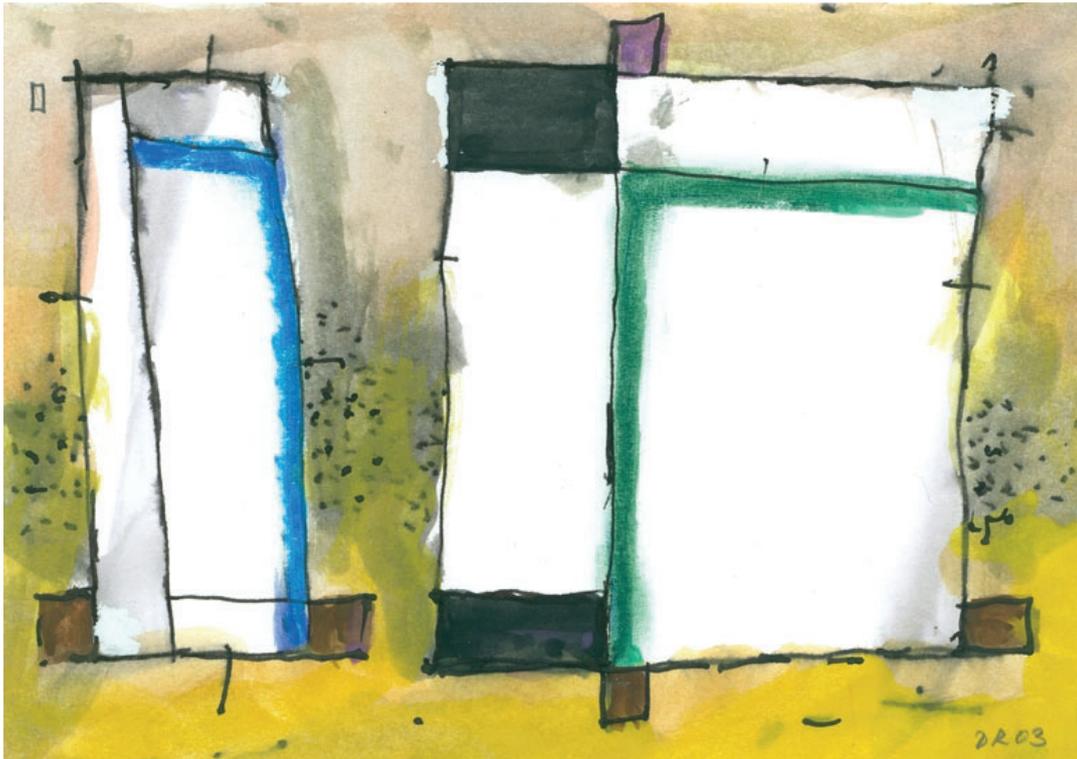
## Copyright

Photos below: ©MPI-M  
Photos on the back from left to right:  
Christian Klepp, Jochem Marotzke,  
Christian Klepp, Clotilde Dubois,  
Christian Klepp, Katsumasa Tanaka



# JSBACH 3

The land component of the MPI Earth System Model



Documentation of version 3.2

Christian H. Reick & Veronika Gayler

D. Goll, S. Hagemann, M. Heidkamp,  
J.E.M.S. Nabel, T. Raddatz, E. Roeckner,  
R. Schnur, S. Wilkenskjeld

Hamburg 2021

Christian H. Reick  
Veronika Gayler

D. Goll  
S. Hagemann  
M. Heidkamp  
J.E.M.S. Nabel  
T. Raddatz  
E. Roeckner  
R. Schnur  
S. Wilkenskjeld

Max-Planck-Institut für Meteorologie  
Bundesstrasse 53  
20146 Hamburg

*Title ports on Ecover by C.H. Reick and main title by B. Diallo, both employing  
a painting by Dieter Reick (2003).  
Page 271: Linocut by Dieter Reick (~1950).*

*© Max Planck Institute for Meteorology and the authors  
D-20146 Hamburg, Germany, February 23, 2021*

DOI: 10.17617/2.3279802

Berichte zur Erdsystemforschung / Max-Planck-Institut für Meteorologie  
*Reports on Earth System Science / Max Planck Institute for Meteorology*

240  
2021

ISSN 1614-1199

# JSBACH 3

The land component of the MPI Earth System Model



# Foreword

Why should one name a land surface model after a composer? This funny idea was brought up around 1999 by Colin Prentice, a former director of the Max Planck Institute for Biogeochemistry in Jena, when he and his coworker Wolfgang Knorr realized in 1999 that climate models should be “biologized”. After the achievements made in the 90s by coupling atmospheric general circulation models with ocean circulation models, and after the successful coupling with models for ocean biogeochemistry, it was the logical next step to include the so far missing variability from the interrelation between the land carbon cycle and the climate. By that time, Guy Brasseur had just taken one of the three director positions at the Max Planck Institute for Meteorology in Hamburg (MPI-M), determined to couple the ozone chemistry model MOZART originating from NCAR with the Hamburg atmospheric circulation model ECHAM. Clearly, from the Jena perspective, the Earth’s biology is at least equally important as atmospheric chemistry. Hence, in order not to stand back against the MOZART coupling initiative, the name of an at least equally famous composer was needed for the new land surface component. And since it was clear from the outset that this “biologization” could only be achieved by establishing an appropriate research group of scientists from Jena at the home of ECHAM in Hamburg, the German composer J.S. Bach made the run: most conveniently, his name contains the letters “J” and “H”, like Jena and Hamburg. So JSBACH stands now for Jena Scheme for Biosphere-Atmosphere Coupling in Hamburg.

But from a name to a working model takes a while. Fortunately, Wolfgang’s clever marketing of his Biosphere-Energy-Transfer-Hydrology model BETHY in applying for money at different funding agencies turned out to be very successful so that, around the end of 2001, the so-called “Jena group” became established at the MPI-M, financed from different national and European sources. Although well supported by Erich Roeckner and his ECHAM group, the “veggies” from Jena, talking about weird things like Rubisco production and sapwood diameter, remained a bit of “outsiders” at the MPI-M – best pictured by the fact that they remained located in the “Pavilion” (an euphemism for a kind of provisional building, better termed “barrack”), when the whole institute moved into the brand new MPI building. But steadily, the Hamburg institute started relying on “having” a vegetation group – first asking only for presentation material but finally showing pride in hosting such a group. No wonder: carbon cycle studies started to be a must for a first-class climate research institute.

Of course, this recognition depended on the realization of JSBACH, which was tackled by four postdocs: Karl-Georg Schnitzler, Thomas Raddatz, and Christian Reick from the “Jena group”, and Reiner Schnur from Erich Roeckner’s group at MPI-M . The first and most difficult task was to re-code ECHAM5 in order to collect all land surface processes in one place and to sort them into a modular structure. This allowed to test Erich’s requirement not to change the simulated climate by running the revised model in a kind of twin set-up – calculating all land surface processes twice, with the old ECHAM5 code and concurrently with the new JSBACH code.

Thereafter, a first version of JSBACH was finalized by transferring the photosynthesis

code of BETHY, integrating a new phenology model, and adding five land carbon pools. This setup was used to participate in the Coupled Climate Carbon Cycle Model Intercomparison Project (C<sup>4</sup>MIP) in 2005, which led to the first mentioning in the scientific literature [45], a paper that was well recognized in the fourth Assessment Report of the IPCC and being the first real success of the JSBACH project.

Nevertheless, the JSBACH group shrank as Colin Prentice moved to Bristol to take the lead in the QUEST project. Full integration into the MPI-M was accomplished not before the arrival of Martin Claussen in 2005 who, as a new director, established a whole department at MPI-M devoted to land processes in the Earth system.

With this long-term perspective for JSBACH at MPI-M, with the development of further vegetation components (land use change (2006), dynamic vegetation (2008)), and with a growing number of publications, the wish for a JSBACH documentation developed. But, as always, advancing to new horizons is much more exciting than consolidating the ground behind. Writing documentation was therefore rather slow and, in view of the ever ongoing code changes, the rudimentary writings were often outdated before they could be finalized. Hence, for years the documentation existed only as a draft, always trying to catch up with the JSBACH development. Nevertheless, this 'living document' helped generations of PhD students to get acquainted with the innards of JSBACH, keeping them fully aware that the final instance to answer questions on model details is always the code. Therefore, in retrospect, this permanent draft state might have been fully appropriate for documenting a model undergoing permanent change.

This situation has now changed because the ECHAM-based development line of JSBACH has come to an end. The next-generation JSBACH (JSBACH 4) will be a complete re-write in the context of ICON<sup>1</sup>. Therefore, without the threat of ever getting outdated, now the time has come to publish a final documentation of the latest ECHAM-based JSBACH version 3.2.

Hamburg, December 2020

---

<sup>1</sup>ICON is the new Earth System and Numerical Weather Prediction Model developed jointly by MPI-M and the German Weather Service ; see e.g. <https://code.mpimet.mpg.de/projects/iconpublic>

## Acknowledgement

The success of JSBACH would not have been possible without the ideas and commitment of many national and international colleagues. Due to the long JSBACH development history it is hardly possible to provide a complete list of all contributors. Nevertheless, we would like to name Christian Beer, Tanja Blome, Victor Brovkin, Andreas Chlond, Martin Claussen, Altug Ekici, Suman Halder, Thomas Kleinen, Silvia Kloster, Wolfgang Knorr, Daniela Kracher, Gitta Lasslop, Jari Liski, Sylvia Nyawira, Bikash Parida, Julia Pongratz, Petri Räisänen, Tom Riddick, Karl-Georg Schnitzler, Gregor Schürmann, Sanna Sevanto, Tobias Stacke, Tea Thum, Freja Vamborg, Philipp de Vrese, Karl-Hermann Wieners, and Sönke Zaehle, who substantially contributed to the model version documented here. In addition, JSBACH could not have been what it is today without the continuous everyday support by many colleagues throughout MPI-M. And last not least, we are grateful for the friendly and competent support by the members of the German Climate Computing Center (DKRZ).



# Contents

<b>Foreword</b>	<b>v</b>
<b>Contents</b>	<b>ix</b>
<b>1 Overview on JSBACH</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Land processes in JSBACH and the sequence of calculations . . . . .	2
1.3 Basic concepts . . . . .	3
1.4 Technical aspects . . . . .	8
<b>2 Land physics</b>	<b>11</b>
2.1 Introduction . . . . .	12
2.2 Surface energy balance and soil heat budget . . . . .	12
2.3 Water budget . . . . .	14
2.4 Surface exchange Fluxes . . . . .	22
2.5 Coupling to the atmosphere . . . . .	25
2.6 Lakes . . . . .	35
2.7 Glaciers . . . . .	36
2.8 River runoff: The Hydrological Discharge Model . . . . .	36
2.9 Diagnostics . . . . .	38
2.10 Implementation details . . . . .	38
<b>3 Phenology</b>	<b>43</b>
3.1 Introduction . . . . .	43
3.2 The phenology model LoGro-P . . . . .	43
3.3 Diagnostics . . . . .	50
3.4 Implementation details . . . . .	50
<b>4 Biophysical land surface properties</b>	<b>55</b>
4.1 Surface albedo . . . . .	55
4.2 Surface roughness . . . . .	60
4.3 Diagnostics . . . . .	61
4.4 Implementation details . . . . .	62
<b>5 Plant productivity and stomatal control</b>	<b>65</b>
5.1 Introduction . . . . .	65
5.2 Radiation in the canopy . . . . .	66
5.3 Photosynthesis . . . . .	72
5.4 Diagnostics . . . . .	83

<b>6</b>	<b>Natural carbon cycle</b>	<b>85</b>
6.1	Introduction . . . . .	85
6.2	From GPP to NPP . . . . .	86
6.3	Dynamics of vegetation carbon . . . . .	89
6.4	Dynamics of litter and soil carbon in Yasso . . . . .	93
6.5	Dynamics of litter and soil carbon in CBALANCE . . . . .	98
6.6	Diagnostics . . . . .	100
6.7	Implementation details . . . . .	100
<b>7</b>	<b>Natural nitrogen cycle</b>	<b>105</b>
7.1	Introduction . . . . .	105
7.2	Structure of the JSBACH nitrogen allocation model . . . . .	106
7.3	Representation of nitrogen limitation . . . . .	112
7.4	Mineral input and output fluxes . . . . .	116
7.5	Diagnostics . . . . .	121
7.6	Implementation details . . . . .	122
<b>8</b>	<b>Natural disturbances of vegetation</b>	<b>125</b>
8.1	Introduction . . . . .	125
8.2	Wind-throw . . . . .	126
8.3	Vegetation fires . . . . .	126
8.4	Carbon relocation from disturbances . . . . .	128
8.5	Nitrogen relocation from disturbances . . . . .	133
8.6	Diagnostics . . . . .	134
8.7	Implementation details . . . . .	134
<b>9</b>	<b>Natural land cover change</b>	<b>137</b>
9.1	Introduction . . . . .	137
9.2	Leading principles . . . . .	138
9.3	Dynamics of natural vegetation shifts . . . . .	139
9.4	Disturbances . . . . .	140
9.5	Bioclimatic limits . . . . .	141
9.6	Dynamics of inhospitable land (deserts) . . . . .	142
9.7	Potential and actual natural vegetation cover . . . . .	143
9.8	Carbon and nitrogen relocation following land cover change . . . . .	144
9.9	Diagnostics . . . . .	147
9.10	Implementation details . . . . .	148
<b>10</b>	<b>Anthropogenic land cover change</b>	<b>149</b>
10.1	Introduction . . . . .	150
10.2	Change in vegetation distribution . . . . .	151
10.3	Carbon and nitrogen relocation . . . . .	167
10.4	Diagnostics . . . . .	180
10.5	Implementation details . . . . .	180

<b>APPENDICES</b>	<b>183</b>
<b>A Compiling and Running</b>	<b>184</b>
A.1 Overview . . . . .	184
A.2 JSBACH Compilation . . . . .	185
A.3 Running JSBACH . . . . .	186
A.4 Running CBALONE . . . . .	187
A.5 Running HD standalone . . . . .	188
<b>B Preprocessor Directives</b>	<b>189</b>
<b>C Input Files</b>	<b>190</b>
C.1 Overview . . . . .	191
C.2 The JSBACH namelists file <code>namelist.jsbach</code> . . . . .	193
C.3 The land cover type library File <code>lctlb.def</code> . . . . .	209
C.4 Initial data file <code>jsbach.nc</code> . . . . .	213
C.5 Boundary and Initial Data for the Hydrological Discharge (HD) model . . . . .	219
C.6 Initial data for the Carbon Pools: <code>Cpools.nc</code> . . . . .	220
C.7 Initial data for the Nitrogen Pools: <code>Npools.nc</code> . . . . .	222
C.8 Initial data for foliage projective cover (FPC): <code>fpc.nc</code> . . . . .	223
C.9 Initial data for climate buffer variables: <code>climbuf.nc</code> . . . . .	224
C.10 Forcing data for JSBACH standalone runs: <code>climate.nc</code> . . . . .	225
C.11 Forcing data for CBALONE: <code>exp_date.jsbach_yDay_Mean.nc</code> . . . . .	226
<b>D Output files</b>	<b>228</b>
D.1 Overview . . . . .	228
D.2 The jsbach stream variables . . . . .	230
D.3 The land stream variables . . . . .	230
D.4 The nitrogen ( <code>nitro</code> ) stream variables . . . . .	230
D.5 The surface ( <code>surf</code> ) stream variables . . . . .	231
D.6 The vegetation ( <code>veg</code> ) stream variables . . . . .	231
D.7 The Yasso stream variables . . . . .	233
D.8 The area weighted ( <code>accw</code> ) stream variables . . . . .	234
D.9 The carbon dioxide ( <code>co2</code> ) stream variables . . . . .	234
D.10 The tracer stream variables . . . . .	234
D.11 CBALONE output . . . . .	234
<b>E Post-processing</b>	<b>243</b>
E.1 Monthly means . . . . .	243
E.2 Post-processing and visualization with <code>visualize_all.ksh</code> . . . . .	243
E.3 Monitoring . . . . .	251
<b>F Generation of maps for leaf and soil albedo</b>	<b>252</b>
<b>G Equilibration of C, N, and vegetation distribution</b>	<b>255</b>
<b>H Regional and site-level Simulations</b>	<b>258</b>
<b>I Some Abbreviations</b>	<b>260</b>
<b>Bibliography</b>	<b>261</b>



# Chapter 1

## Overview on JSBACH

### Contents

---

1.1	Introduction . . . . .	1
1.2	Land processes in JSBACH and the sequence of calculations . . . . .	2
1.3	Basic concepts . . . . .	3
1.3.1	Representation of vegetation by PFTs . . . . .	3
1.3.2	Tiling of grid boxes . . . . .	3
1.4	Technical aspects . . . . .	8
1.4.1	The JSBACH main routine . . . . .	8
1.4.2	Flux- versus parameter-aggregation . . . . .	8
1.4.3	Basic data structures . . . . .	10
1.4.4	Infrastructure . . . . .	10

---

### 1.1 Introduction

JSBACH is the land component of the atmospheric component ECHAM [55], and thereby part of the Max Planck Institute Earth System Model (MPI-ESM) [89]. Historically, JSBACH grew out of ECHAM5 [116] by collecting all land processes into a separate land component – then called JSBACH – accessed from ECHAM each time step via a single subroutine call. Accordingly, JSBACH inherits all the land processes originally present in ECHAM5, in particular the way the surface energy balance is solved, and how the land processes are coupled to the atmosphere. While some of the JSBACH source codes are still the original ECHAM5 (or even ECHAM3) codes (e.g. the model for soil heat transport described in section 2.2), others have been modernized: the ECHAM5 bucket hydrology has been replaced by a diffusive transport model for soil water (section 2.3.2), and leaf area index (chapter 3) and albedo of dry surfaces (section 4.1) are no more prescribed but replaced by state dependent models. But the main motivation for the development of JSBACH was the wish to study the coupled climate-carbon dynamics. Accordingly, very soon submodels for the absorption of solar radiation in vegetation canopies (section 5.2), leaf photosynthesis (section 5.3), and allocation and turnover of carbon in vegetation and soils (chapter 6) had been added and underwent many changes since then. Later on, to study natural and anthropogenic climate change, components

for natural shifts in biogeography (chapter 9) and land use change (section 10) were added. And most recently, the model was complemented by including nitrogen cycling (chapter 7).

Design target upon all these changes and additions was to conserve energy, water, carbon, nitrogen, and – upon land cover changes – also area. For water, carbon, nitrogen and area this aim is met to numerical accuracy while, for energy, tiny leaks still exist that could be closed only by a considerable enhancement of the land physics, e.g. by accounting for the temperature of rain water or the heat produced by heterotrophic respiration. Another design target concerned modularity: Subcomponents describing particular processes should be independent in the sense that they provide their own data structures, state initialization, and definition of input and output fields.

JSBACH uses the I/O, memory handling, parallelization, time stepping, and calendar infra-structure of ECHAM. Details are found in the technical documentation of ECHAM [109]. As part of ECHAM, JSBACH runs on the same grid and time step as ECHAM. Within JSBACH there is no communication between grid boxes. In JSBACH, each grid box is equipped with a sub-structure (“tiling”) to represent diversity of land cover types within a grid box, in particular the diversity of plant functional types (PFTs) representing various vegetation types. Both concepts – tiling and PFTs – are basic to JSBACH and described in detail in section 1.3 below.

Despite being integral part of ECHAM, JSBACH can be run independently forced by meteorological data (observed or simulated). This configuration is called ‘JSBACH standalone’ (see Appendix A.3). Another configuration covers only the longterm dynamics (decadal/centennial) of carbon and nitrogen turnover and vegetation biogeography. This configuration – called CBALONE – is typically used for speeding up the generation of equilibrium states for land carbon and vegetation distribution: The resulting equilibrium states can be used as re-start condition for JSBACH standalone and ECHAM/JSBACH (see Appendix G). Finally, also the model for hydrological discharge (HD model) can be run separately (see Appendix A.5).

All configurations have their own driver calling JSBACH each time step (ECHAM/JSBACH, JSBACH standalone), or once a day (CBALONE). The following documentation concentrates on the first two configurations. In these two configurations the driver calls the JSBACH main routine `jsbach_inter_1d`. This subroutine sequentially calls the different subprocesses implemented in JSBACH, possibly only part of the processes, depending on the runtime configuration (see appendix C.2). Section 1.4.1 below shortly describes this JSBACH main routine and the information exchanged with the driver.

## 1.2 Land processes in JSBACH and the sequence of calculations

When running JSBACH as part of ECHAM, or when running it in the standalone configuration, the JSBACH main routine is called once each time step. In this main routine, the land processes, implemented as subroutines, are called sequentially. This is depicted in Fig. 1.1. Whether all processes are called or only some of them is controlled by namelist switches (see appendix C.2), for some processes allowing for alternative implementations to be used (e.g for anthropogenic land cover change). As sketched in the figure, the sequence of calculations starts by calculating natural landcover changes, i.e. biogeographic shifts in the distribution of PFTs (chapter 9). This includes wild fires that lead to CO<sub>2</sub> emissions. Next, modifications from anthropogenic landcover changes are computed according to externally prescribed maps (chapter 10). These anthropogenic landcover changes involve transfer of

carbon and nitrogen between plants and soils,  $\text{CO}_2$  and  $\text{NO}_x$  emissions from deforestation fires, and a re-shuffling of carbon and nitrogen between different PFTs. Having performed the changes in landcover distribution, plant carbon assimilation is simulated next by first determining the fraction of photosynthetically active solar radiation (PAR) absorbed in the vegetation canopy ( $f_{\text{apar}}$ ) (section 5.2). Then, using the leaf area index from the previous time step, in a first call of the photosynthesis routine (section 5.3), the stomatal conductance in absence of water stress is determined. When calling the land physics routines, this “unstressed” stomatal conductance is scaled with the plant available soil water to yield the “stressed” stomatal conductance from which the plant productivity can then be determined in a second call of the photosynthesis routine. The land physics routines update the soil water budget including calculation of runoff and drainage (section 2.3), recalculate the temperature profile in the soil (section 2.2), and compute heat and moisture fluxes across the land-atmosphere interface. The stressed stomatal conductance enters in particular when updating the soil hydrological status. Heat and moisture fluxes are computed by using the thermal and hydrological information provided from the driver (e.g. ECHAM). The second call of the photosynthesis routine computes gross primary productivity (GPP) and dark respiration per leaf area. Accounting for maintenance and growth respiration, scaling from leaf level to the whole plant results in net primary productivity (NPP) (section 6.2). When calling the carbon and nitrogen allocation routines (chapters 6 and 7), NPP is distributed to the different plant organs. The allocation routines account also for the transfer of living biomass to the litter carbon pools and estimate heterotrophic respiration. The latter is done by the Yasso soil carbon model (chapter 6).  $\text{CO}_2$ -uptake from plants and respiration flux from the soils both contribute to the  $\text{CO}_2$  exchange with the atmosphere. Close to the end of the main routine, the phenology routine is called for updating the leaf area index (section 3) before then albedo (section 4.1) and surface roughness (section 4.2) are recomputed.

## 1.3 Basic concepts

### 1.3.1 Representation of vegetation by PFTs

The diversity of vegetation is represented in JSBACH by so-called “Plant Functional Types” (PFTs). Each PFT is globally endowed with particular properties with respect to the various processes JSBACH is accounting for. Examples for such properties are photosynthetic pathway type (C3, C4) and other associated parameters of the photosynthesis models, type of phenology (e.g. grass, raingreen, etc.) and other phenology parameters (maximum Leaf Area Index etc.). Each submodel (process) in JSBACH can associate its own parameter set with each PFT. These specifications are found in the so-called “land cover type library file” (see C.3) containing all defining parameters for each PFT. Thereby, the number of PFTs and their properties can be freely varied in JSBACH, although new parameter sets or PFTs may need careful recalibration of the model.

### 1.3.2 Tiling of grid boxes

The basic land surface unit in JSBACH is a grid box having a well-defined geographic location. Each grid box is divided into “tiles” to allow for the representation of sub-grid scale heterogeneity. These tiles are not specified by their geographic location in a grid box, but only by the fraction of the grid box they cover. The number of tiles per grid box is a globally fixed number but is configurable via namelist (see appendix C.2.11). Each tile of each grid box is associated with one of several land cover types, which can be either a PFT or “glacier”.

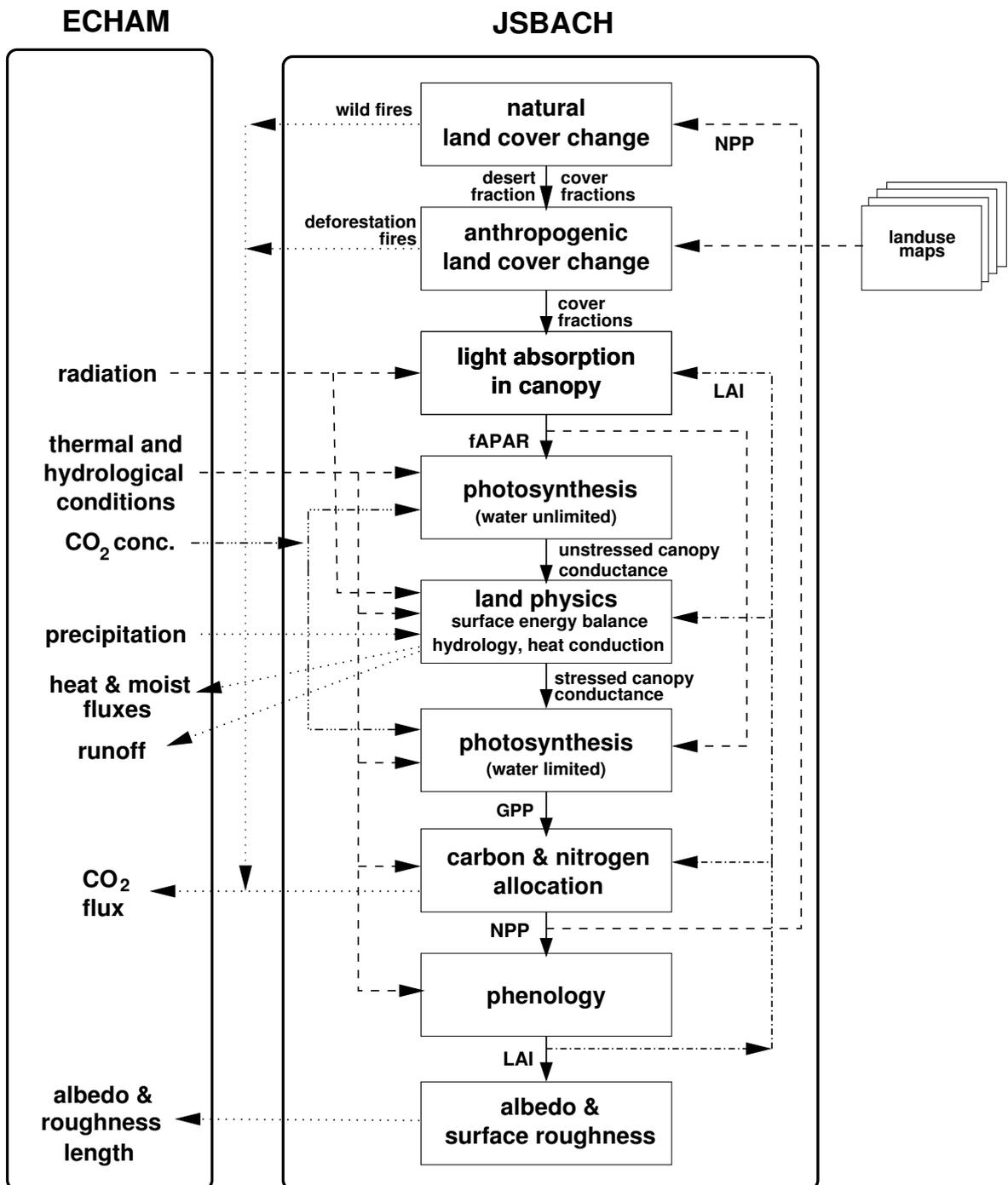


Figure 1.1: Sketch of JSBACH and its coupling to ECHAM

The mapping between tiles and landcover types is prescribed from one global map for each tile that is read in during the initialization of JSBACH (see appendix C.4.6) and kept fixed throughout a simulation. Conversely, considering a particular grid box, not every land cover type must be linked with a tile of that grid box: for example, to save computational time, it may be useful to link tropical PFTs only with grid boxes in the tropics, but not in the boreal zone. This tiling concept is quite flexible and could also be applied to additional non-PFT cover types like wetlands or urban areas. Usually the number of tiles is chosen to be smaller than the available number of PFTs to save computation time and memory.

One peculiarity of the tiling concept in JSBACH is that one land cover type is not associated with a tile: non-ice-covered regions inhospitable to plants like rocky surfaces or deserts are instead determined by associating with each grid box a value  $veg_{max}$  indicating the fraction of a grid box hospitable to vegetation. Accordingly,  $f_{bare} := 1 - veg_{max}$  is the fraction of inhospitable land in a grid box (also called “bare land” in the following), and the area  $V_{veg}$  of a grid box accessible to vegetation is given by

$$(1.1) \quad V_{veg} = A \, veg_{max},$$

where  $A$  is the area of the considered grid box ( $m^2$ ).

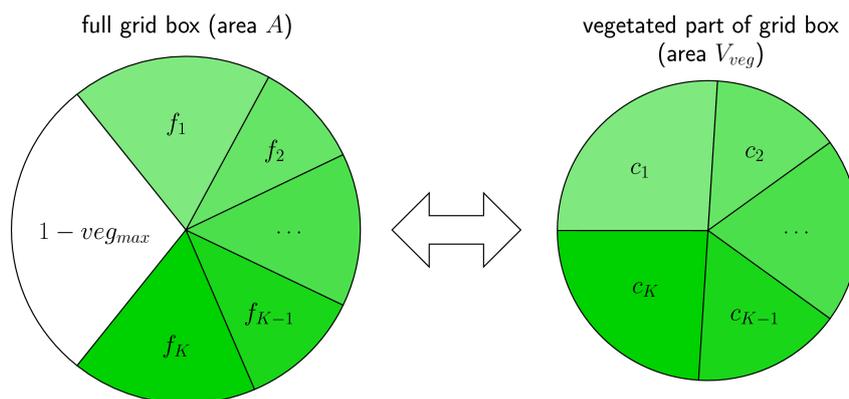


Figure 1.2: Tiling of a grid box and cover fractions: tile  $i$  has cover fraction  $f_i$  relative to the full grid box and cover fraction  $c_i$  relative to the vegetated part of the grid box.

Internally, JSBACH is not using *area* to characterize the extent of a particular vegetation type, but the *fraction* of the area covered by vegetation (compare Fig. 1.2). Denoting by  $v_i$  the area covered by the PFT associated with tile  $i$  in a grid box (units:  $m^2$ ), such fractions are most naturally introduced by

$$(1.2) \quad f_i = \frac{v_i}{A}, \quad i = 1, 2, \dots, K,$$

where  $K$  is the number of tiles in a grid box. Obviously

$$(1.3) \quad V_{veg} = \sum_{i=1}^K v_i$$

so that

$$(1.4) \quad \sum_{i=1}^K f_i + (1 - veg_{max}) = 1.$$

Because of the implicit handling of bare land it is more convenient to describe land cover only with respect to that part of the grid box where vegetation can grow. Associated cover fractions are introduced by

$$(1.5) \quad c_i = \frac{v_i}{V_{veg}} = \frac{f_i}{veg_{max}}, \quad i = 1, 2, \dots, K,$$

and they sum up to 1:

$$(1.6) \quad \sum_{i=1}^K c_i = 1.$$

Together with  $veg_{max}$  these  $c_i$  are the basic quantities characterizing extent of landcover in JSBACH. Sloppily, the quantities  $c_i$  will be referred to by the term “cover fraction”.  $veg_{max}$  and the cover fractions are either prescribed by external maps, or they are re-computed every year by the dynamic biogeography component of JSBACH (chapter 9) in relation to the prevailing climate, and the  $c_i$  are possibly further modified by one of the implemented land use change schemes (chapter 10).

The cover fractions are of general importance for JSBACH because most computations are based on densities instead of extensive quantities. E.g. the prognostic variables for the content of the carbon pools in JSBACH have the unit *moles carbon per square meter vegetated area*, instead of simply *moles of carbon*. The advantage of this approach is that however small the area covered by a vegetation type may be, the density values stay finite, in contrast to the associated extensive quantities that may fall below computational accuracy for small areal extent of the particular vegetation. This feature allows to follow the performance of vegetation in a grid box even if its extent is so minute that e.g. the total carbon content is numerically zero. Moreover, by reprocessing the output of a JSBACH simulation it is thereby possible to infer e.g. what would have happened with the carbon dynamics if other cover fractions would have been assumed in the simulations. In this way, expensive additional simulations can be circumvented.

In addition, JSBACH accounts for gaps in vegetation canopy. This is necessary because e.g. transpiration and carbon uptake happen only where leaves are present, i.e. within the canopies, but not in the canopy gaps. Conversely, canopy gaps contribute to bare land evaporation and modify albedo differently than a closed canopy. Therefore, the size of canopy gaps has to be determined. Depending on type, vegetation tends to clump differently so that its characterization must depend on the PFT considered. Roughly, the size of gaps is related to the leaf area index (LAI) of vegetation. For a characterization independent of seasonality, JSBACH uses the maximum LAI (denoted by  $LAI^{max}$ ) prescribed for the phenology of the various PFTs (see section 3.2.1) and not the actual LAI. Loosely modeled in analogy to Beer’s law (see e.g. [70]) it is assumed that gaps reduce vegetation cover by a factor  $1 - \exp(-LAI^{max}/\kappa)$ , where  $\kappa$  characterizes the structure of the canopy. The standard value for all PFTs is  $\kappa = 2$ , which represents random orientation of leaves [96], whereas  $\kappa = 3$  is chosen for crops to account for the control of light limitation between neighboring plants by the way farmers plant crops, e.g. in rows leaving large gaps in the canopy. The values of  $LAI^{max}$  and  $\kappa$  are defined for each PFT in the lctlib-file (see C.3). Hence, defining the PFT-dependent correction factor accounting for clumping by

$$(1.7) \quad f_i^{corr} = 1 - e^{-LAI_i^{max}/\kappa_i},$$

the fraction  $f_i^{can}$  of *closed canopy* of a given PFT is smaller than the associated fraction  $f_i$  of vegetated area according to

$$(1.8) \quad f_i^{can} = f_i f_i^{corr}.$$

Hence, the area  $V_{can}$  of closed canopy in a grid box made up by all PFTs in a grid box is obtained as

$$(1.9) \quad V_{canopy} = A \sum_{i=1}^K f_i^{can} = A \text{veg}_{max} \sum_{i=1}^K c_i f_i^{corr},$$

where  $i$  counts over all tiles, and  $K$  is the number of tiles. In the JSBACH code,  $f_i^{corr}$  is called `veg_fract_correction`.

In summary, three types of area are distinguished in JSBACH:

**grid box area** : area of a whole grid box (also called “box area” or “ground area”).

**vegetated area** : area covered with vegetation including canopy gaps.

**canopy area** : area covered densely with vegetation, i.e. excluding canopy gaps.

The cover fractions  $f_i$  and  $f_i^{can}$  are relative to grid box area while the cover fractions  $c_i$  refer to vegetated area. Analogous to (1.8) one can thus define

$$(1.10) \quad c_i^{can} = c_i f_i^{corr},$$

which is the fraction of the vegetated part of the grid box covered by a closed canopy of PFT  $i$ .

Since it depends on the particular process which type of area simulation results refer to, it is important to know how to convert them to values referring to grid box area. The conversions are obtained from the above considerations as follows. Assume, for a particular PFT  $i$ , a quantity  $X_i$  is per vegetated area, then the associated value per grid box area (lab slang: “box value”) is given as

$$(1.11) \quad X_i^{box} = \text{veg}_{max} c_i X_i.$$

If instead a quantity  $Y_i$  is given per canopy area then the associated value per grid box area is given as

$$(1.12) \quad Y_i^{box} = \text{veg}_{max} c_i^{can} Y_i.$$

Such conversions can be done in the post processing and, for that purpose, the standard output provides not only  $c_i$  and  $\text{veg}_{max}$  but also the correction factor  $f_i^{corr}$ ; these variables are found in the `jsbach` output stream (see appendix D.2) under the names `cover_fract`, `veg_ratio_max`, and `veg_fract_correction`. For some variables, JSBACH computes box values already during runtime in addition to the originally computed values. These output fields are easily recognized by their names starting with `box` (compare appendix D.6). Box values are particularly useful to compute regional or global totals, since one needs simply to multiply with the grid box area and then sum over the region (e.g. to compute global NPP from grid box values). But, typically, the canopy values must be used for comparison with locally measured data, e.g. to compare GPP measured locally in a forest with simulated values.

Finally it should be noted that, for technical reasons, cover fractions are not allowed to drop below a certain very small value<sup>1</sup> throughout JSBACH. The only exceptions are tiles in a grid box with glaciers: It is assumed throughout JSBACH that glaciers either cover a grid box fully or are completely absent. Accordingly, in grid boxes with glacier, the cover fraction of the various PFTs is set to zero, and  $c_i = 1$  for glacier.

<sup>1</sup>This minimum value is called `fract_small` in the JSBACH code. It is defined in module `mo_land_surface`. Its standard value is  $10^{-10}$ .

## 1.4 Technical aspects

### 1.4.1 The JSBACH main routine

As part of ECHAM, the JSBACH main routine `jsbach_inter_1d` is called once during each time step while solving the equations for vertical heat and moisture transport. Numerically, the vertical transport equations are solved using an implicit scheme [120] – a Richtmyer-Morton scheme – as developed for the SECHIBA land surface scheme [103]. Therefore, JSBACH needs as input not only the standard inputs of a land surface scheme, like incoming radiation, precipitation, wind velocity etc., but also the so-called *Richtmyer-Morton coefficients* and the turbulent heat transfer coefficient characterizing the turbulent diffusion in the boundary layer. Table 1.1 lists the essential input variables of the JSBACH main routine. Using this information, JSBACH solves the surface energy balance and returns the necessary land boundary conditions to ECHAM. The essential output variables are listed in Table 1.2.

Table 1.1: Essential input variables of the JSBACH interface routine `jsbach_inter_1d`

variable	units	meaning
<i>Hydrological conditions:</i>		
<code>qair</code>	1	specific humidity of air at lowest level $q_N^{t-\Delta t}$
<code>precip_rain</code>	$kg/m^2s$	precipitation in form of rain
<code>precip_snow</code>	$kg/m^2s$	precipitation in form of snow
<i>Thermal conditions:</i>		
<code>temp_air</code>	$K$	air temperature at lowest level $T_N^{t-\Delta t}$
<i>Mechanical conditions:</i>		
<code>wind</code>	$m/s$	windspeed at lowest level
<code>wind10</code>	$m/s$	windspeed at 10m above ground
<code>cdrag</code>	1	surface drag coefficient (compare eq. (2.45))
<code>pressure</code>	$kg/m^2$	surface pressure
<i>Radiative conditions:</i>		
<code>lwdown</code>	$W/m^2$	downward longwave flux
<code>sw_vis_net</code>	$W/m^2$	net visible radiation flux at surface
<code>sw_nir_net</code>	$W/m^2$	net near infrared radiation flux at surface
<code>sw_par_down</code>	$W/m^2$	downward surface flux of photosynthetic active radiation
<code>sw_par_frac_diffuse</code>	1	diffuse fraction of <code>sw_par_down</code>
<code>czenith</code>	1	cosine of solar zenith angle
<i>Conditions for turbulent fluxes:</i>		
<code>etAcoef</code>	1	Richtmyer-Morton coefficient $E_N$ in (2.80)
<code>etBcoef</code>	$J/kg$	Richtmyer-Morton coefficient $\tilde{F}_N^s$ in (2.80)
<code>eqAcoef</code>	1	Richtmyer-Morton coefficient $\tilde{E}_N$ in (2.80)
<code>eqBcoef</code>	$kg/kg$	Richtmyer-Morton coefficient $\tilde{F}_N^q$ in (2.80)
<code>echam_zchl</code>	1	turbulent heat transfer coefficient $C_h$ (see eq. (2.45))
<i>Conditions for carbon cycle:</i>		
<code>CO2_concentration</code>	$kg(CO_2)/kg(air)$	lowest layer $CO_2$ concentration

### 1.4.2 Flux- versus parameter-aggregation

ECHAM needs land boundary conditions that are valid for a grid box as a whole. But many processes are calculated in JSBACH separately for each tile. Therefore, results obtained at

Table 1.2: Essential output variables of the JSBACH interface routine `jsbach_inter_1d`

variable	units	see eqn.	meaning
<i>Surface fluxes:</i>			
<code>evap_act</code>	$kg(H_2O)/m^2s$	(2.56)	total evapotranspiration
<code>sensible</code>	$W/m^2s$	(2.58)	sensible heat flux
<code>latent</code>	$W/m^2s$	(2.60)	latent heat flux
<code>CO2_flux</code>	$kg(CO_2)/m^2s$	(6.48), (8.20), (10.73) <sup>†</sup>	net land/atmosph. CO <sub>2</sub> flux
<i>Thermal surface conditions:</i>			
<code>temp_soil_new</code>	$K$	(2.97)	surface temperature
<code>t_soil_rad</code>	$K$	(2.98)	radiative surface temperature
<code>surf_dry_static_energy</code>	$J/kg$	(2.95)	mixed time dry static energy at surface
<code>tte_corr</code>	$K/s$	(2.10)	correction to temperature tendency (tte) for snow melt (see eqn.2.10)
<i>Hydrological surface conditions:</i>			
<code>csat</code>	1	(2.56)	weighting factor in latent heat flux related to <code>qsurf</code>
<code>cair</code>	1	(2.56)	weighting factor in latent heat flux related to air humidity
<code>qsurf</code>	1	(2.82)	mixed time saturation specific humidity at surface
<i>Radiative surface conditions:</i>			
<code>albedo_vis</code>	1	(4.5)	albedo of the visible spectrum
<code>albedo_nir</code>	1	(4.5)	albedo in the near infrared
<i>Mechanical surface conditions:</i>			
<code>z0m</code>	$m$	(4.15)	surface roughness length for turbulent momentum transport
<code>z0h</code>	$m$	(4.15)	surface roughness length for turbulent heat transport

<sup>†</sup> The combination of contributing carbon fluxes depends on the model configuration.

the tile level need to be aggregated to the grid box level. This is straight forward for quantities acting additively. Such additive quantities are flux densities (e.g. the CO<sub>2</sub>-flux measured in units  $kg(C)/m^2s$ ) and albedo. In these cases, the value for the whole grid box is obtained as a weighted average across all tiles where the weights are given by the cover fractions  $c_i$ . This is called “flux aggregation”.<sup>2</sup> A complication arises from the implicit handling of bare land in JSBACH. To prevent a separate tile for bare land – to calculate e.g. evaporation – a fraction  $c_i$  of the bare land fraction  $1 - veg_{max}$  is associated with each tile and calculations are performed in one go for the tile and the associated fraction of bare land.

Problematic are non-summable quantities, like roughness length, and, most prominently, surface temperature: For very different surface temperatures on different tiles it would be unclear how to obtain a grid box wide average temperature consistent with energy conservation. To remedy this problem of “parameter aggregation”, the calculation of temperature in JSBACH is made for the grid box as a whole. But since all heat and moisture fluxes are still computed separately for each tile, the problem of parameter aggregation has thereby only been shifted to the tile level: because the fluxes depend on the grid box wide surface

<sup>2</sup>JSBACH provides the generic routine `average_tiles` for this weighted averaging.

temperature, it must be assured that in particular the hydrological conditions do not get too inconsistent with that temperature. This is prevented by redistributing at the end of each time step the water between all tiles such that the water content for each reservoir type is equalized.<sup>3</sup> This means in particular that all vegetation types compete for water from the same soil reservoirs. Therefore, JSBACH implicitly assumes with respect to hydrology a *well mixed* vegetation. The other extreme would be a *well-separated* vegetation where processes on different tiles do not interfere.<sup>4</sup> This is in fact the case for light competition in JSBACH: Different PFTs do not compete for light in the canopy.

### 1.4.3 Basic data structures

JSBACH makes extensive use of structured data types. The top level structure `theLand` defined in `mo_jsbach_interface.f90` collects all substructures defined by the different components of JSBACH. E.g. the substructure `soil` collects all state information from the land physics, and the substructure `Cbalance` the state information of the vegetation and soil carbon. These substructures are declared and defined in the modules containing the code for the respective processes.

Although such a globally defined data structure `theLand` exists, the substructures remain private to the submodules. The implementation of JSBACH thus prevents direct exchange of data between submodules. This forces the programmer to pass state information between processes across their interfaces, thereby preventing error-prone side effects. Only general information on the underlying grid, on parallelization aspects, and calendar information is globally available via `theLand` to all sub-components.

Throughout, computations in JSBACH are performed only for land points. Therefore, in the JSBACH main routine, all latitude-longitude input arrays are first packed into a 1D-array containing only land points. For giving information back to the driver, these 1D arrays are again unpacked to 2D latitude-longitude arrays covering the grid points of the whole globe.

### 1.4.4 Infrastructure

JSBACH makes extensive use of the infrastructure provided by ECHAM. This concerns the parallelization support, the memory and I/O handling, as well as the time stepping control and calendar computations. These features are partially described in the ECHAM6 user manual [109].

---

<sup>3</sup>This equalizing is also performed by applying the generic routine `average_tiles`.

<sup>4</sup>See [79] for a discussion of a “mixed” vs. a “mosaic” representation of subgrid scale heterogeneity.

# Chapter 2

## Land physics

### Contents

---

2.1	Introduction . . . . .	12
2.2	Surface energy balance and soil heat budget . . . . .	12
2.2.1	Land surface energy balance and surface temperature . . . . .	12
2.2.2	Soil temperatures and ground heat flux . . . . .	13
2.3	Water budget . . . . .	14
2.3.1	Above ground water budget . . . . .	14
2.3.1.1	Interception of snow by the canopy . . . . .	14
2.3.1.2	Snow at the surface . . . . .	15
2.3.1.3	Interception of rain by vegetation and bare soil . . . . .	15
2.3.2	Soil water budget . . . . .	16
2.3.2.1	Vertically resolved soil moisture dynamics . . . . .	16
2.3.2.2	Infiltration and surface runoff . . . . .	18
2.3.2.3	Drainage $R_d$ . . . . .	22
2.4	Surface exchange Fluxes . . . . .	22
2.4.1	Moisture Fluxes . . . . .	22
2.4.1.1	Basic types of moisture fluxes . . . . .	22
2.4.1.2	Accounting for hydrological surface structure . . . . .	23
2.4.2	Heat Fluxes . . . . .	25
2.5	Coupling to the atmosphere . . . . .	25
2.5.1	Numerical solution of vertical turbulent transport in ECHAM . . . . .	26
2.5.2	Calculation of surface temperature and surface heat exchange . . . . .	30
2.5.3	Emulation of coupling in JSBACH standalone . . . . .	35
2.6	Lakes . . . . .	35
2.7	Glaciers . . . . .	36
2.8	River runoff: The Hydrological Discharge Model . . . . .	36
2.8.1	Linear reservoirs . . . . .	36
2.8.2	Model structure . . . . .	37
2.9	Diagnostics . . . . .	38
2.10	Implementation details . . . . .	38
2.10.1	General setup . . . . .	38

2.10.2	Implementation details: Thermal land processes . . . . .	39
2.10.2.1	Thermal soil parameters . . . . .	39
2.10.2.2	Numerics for solving the thermal diffusion equation . . . . .	39
2.10.3	Implementation details: Soil hydrology . . . . .	40
2.10.3.1	Hydrological input fields and model parameters . . . . .	40
2.10.3.2	Structure of calculations . . . . .	40
2.10.4	Implementation details: River routing in the HD model . . . . .	41

---

## 2.1 Introduction

Originally JSBACH development started with the land physics of ECHAM5. And, despite more than a decade of model development, still large parts of the codes for the land physics are identical to the original ECHAM routines. This concerns in particular the soil heat budget and the way in which the land physics couples to the atmospheric processes. Accordingly, the following description of these processes has been largely taken from the ECHAM5 documentation [116]. A major difference to ECHAM5 is the replacement of the original single-bucket soil hydrology by a hydrology based on Richard’s equation describing explicitly the vertical movement of water in the soils. The Richards equation and the heat conduction equation are discretized using the same vertical spacing. In principle, this allows to link thermal and hydrological soil properties, e.g. to represent the phase change of water in the soil, but this is not yet realized in JSBACH.

Although JSBACH represents sub-grid scale heterogeneity with respect to certain vegetation characteristics (by tiles, see section 1.3.1), the land physics is (largely) treated as being uniform within a grid box. Here, some processes, such as the surface energy balance and the turbulent surface fluxes, are computed based on parameters that represent the grid-box average, while other processes, such as the subsurface heat and moisture fluxes, are calculated for each tile separately although at the end of each time step those variables are averaged across all tiles to maintain consistency between the variables representing the state of the grid-box and those representing the state of individual tiles. Thus, JSBACH implicitly assumes a perfect mixture of all vegetation and bare land in a grid box, meaning e.g. that all PFTs access the same water reservoir. The reason for this aggregation of variables to the grid-box level is that the tile structure is absent in ECHAM and all land-atmosphere interaction must happen with respect to the grid box as a whole. Generally, averaging parameters and state variables across all tiles is only one way to achieve grid-box-wide communication with the atmosphere. An alternative to this “parameter averaging” would be “flux accumulation” as described by Best et al. [9].

## 2.2 Surface energy balance and soil heat budget

### 2.2.1 Land surface energy balance and surface temperature

The surface energy balance combines all energetic fluxes at the surface and thereby constrains the exchange of energy between ground and atmosphere. The following is an overview on how these fluxes combine in the energy balance. The energy balance is particularly important for calculating the surface temperature and closing the calculation of atmospheric turbulent latent and sensible heat fluxes – this aspect is described in detail in section 2.5.

The interface between the land surface and the atmosphere can be understood as a ‘layer’ at the surface which is in contact with the atmosphere. It is taken as the topmost layer of

the vertical discretisations used by the soil thermal and soil hydrology models. The energy balance for this layer is given as

$$(2.1) \quad C \frac{\partial T_{surf}}{\partial t} = R_{net} + H_{sensible} + H_{latent} + G$$

where  $T_{surf}$  is surface temperature,  $C = hC_s$  is the heat capacity of this topmost soil layer, with the volumetric heat capacity  $C_s$ , [ $\text{Jm}^{-3}\text{K}^{-1}$ ], and the thickness  $h$ , [m].  $H_{sensible}$  is the sensible heat flux,  $H_{latent}$  the latent heat flux from evapotranspiration, condensation, and sublimation of water,  $G$  is the ground heat flux and  $R_{net}$  the net radiation.  $C_s$  depends on soil type (see section 2.10.2.1) and is modified in the presence of snow.  $R_{net}$  is a combination of several components:

$$(2.2) \quad R_{net} = (1 - \alpha_{vis})R_{vis} + (1 - \alpha_{nir})R_{nir} + \epsilon R_{ld} - \epsilon \sigma T_{surf}^4$$

where  $R_{vis}$  and  $R_{nir}$  are the downwelling solar radiation in the visible and near infrared range,  $\alpha_{vis}$  and  $\alpha_{nir}$  are the respective surface albedo values (compare section 4.1),  $R_{ld}$  is the downwelling longwave radiation,  $T_{surf}$  is surface temperature,  $\epsilon$  is surface emissivity, and  $\sigma$  the Stefan-Boltzmann constant. Note that in (2.1) downward fluxes are considered positive. Due to the strong coupling between the surface and the atmosphere, the numerical solution of (2.1) is closely linked to the vertical heat and moisture transfer within the atmosphere. Following Polcher et al. [103], an implicit coupling scheme is used; the implementation is described in detail in section 2.5. This scheme uses a kind of leapfrog discretization together with an Asselin time filter, it is unconditionally stable and allows to synchronously calculate the prognostic variables in the atmosphere and the surface fluxes. To avoid iterations, the nonlinear terms appearing in the expressions for upward longwave radiation and saturated humidity at the surface are replaced by truncated Taylor expansions around their values at the last time step. This setup ensures energy conservation in the coupled system land-atmosphere because atmosphere and land component use the same surface fluxes in their calculations.

### 2.2.2 Soil temperatures and ground heat flux

The temperature profile within the soil is calculated from the thermal diffusion equation

$$(2.3) \quad C_s(z) \frac{\partial T}{\partial t} = - \frac{\partial}{\partial z} \left( -\lambda_s(z) \frac{\partial T}{\partial z} \right)$$

where  $z$  the depth [m],  $C_s(z)$  is the depth-dependent volumetric heat capacity of the soil, and  $\lambda_s(z) = C_s(z)\kappa_s(z)$  is the thermal conductivity [ $\text{W m}^{-1}\text{K}^{-1}$ ] with  $\kappa_s(z)$  denoting thermal diffusivity [ $\text{m}^2\text{s}^{-1}$ ] at different depth. These soil properties are prescribed from FAO data (see 2.10.2.1) and are modified in the presence of snow. The heat conduction equation (2.3) is solved by a Richtmyer-Morton scheme [114] down to a prescribed depth, assuming zero flux conditions at the bottom and temperature  $T_{surf}$  at the top (surface) as obtained from the calculation of dry static energy at the surface (see section 2.5.2). Numerical details are found in section 2.10.2.2.

By solving the heat conduction equation (2.3) one also obtains the ground heat flux (positive downward) from discretizing its defining equation

$$(2.4) \quad G := - \lambda_s \frac{\partial T}{\partial z} \Big|_{z_{surf}} .$$

This is the soil heat flux at the surface. Note that  $G$  is needed to close the surface energy balance (2.1).

## 2.3 Water budget

This section describes the land water budget. It is described separately for the processes above ground (section 2.3.1), the soil hydrology (section 2.3.2) and river runoff (section 2.8).

### 2.3.1 Above ground water budget

The above ground water budget is modeled using in each tile three reservoirs :

$h_{snc}$ : Snow on the canopy [m] (water equivalent).

$h_{sn}$ : Snow on the surface [m] (water equivalent).

$h_{wsk}$ : Water on the canopy and the surface [m] (“wet skin reservoir”).

Reservoirs are measured in  $[\text{m}^3(\text{H}_2\text{O})/\text{m}^2(\text{grid box area})]$ , i.e. in [m]. The dynamics determining these reservoirs is described in the next subsections.

#### 2.3.1.1 Interception of snow by the canopy

The amount of snow on the canopy  $h_{snc}$  is determined by several processes, namely snowfall, sublimation, melting, and wind-blow. The JSBACH implementation of these processes goes back to Roesch et al. [118]. See there in particular for a justification of the values of the various model parameters.

It is assumed that only the fraction  $f_v = 0.25$  of the snowfall  $S$  [ $\text{kg}/\text{m}^2\text{s}$ ] is intercepted by the canopy. With sublimation of snow from the canopy  $E_{snc} < 0$ , [ $\text{kg}/\text{m}^2\text{s}$ ], and unloading of snow due to melting  $U_1(T_c)$ , [1/s], as well as wind-blow  $U_2(v_c)$ , [1/s], where  $T_c$ , [K], and  $v_c$ , [m/s], are temperature and wind speed in the canopy, respectively, the balance for snow on canopy can be written as

$$(2.5) \quad \rho_w \frac{\partial h_{snc}}{\partial t} = f_v S + E_{snc} - \rho_w h_{snc} [U_1(T_c) + U_2(v_c)].$$

Here  $\rho_w$ , [ $\text{kg}/\text{m}^3$ ], is the density of water. The accumulation of snow on the canopy is limited by the capacity of the interception reservoir  $h_{snc}^{max}$ , i.e.  $h_{snc} \leq h_{snc}^{max}$ , where the capacity is a function of the time dependent leaf area index  $\Lambda$  (see chapter 3),

$$(2.6) \quad h_{snc}^{max} = h_0 \Lambda veg_{max}$$

with  $h_0 = 2 \cdot 10^{-4}$  m and  $veg_{max}$  as explained in section 1.3.2. The unloading processes are parameterized according to

$$(2.7) \quad U_1(T_c) = \begin{cases} (T_c - c_1)/c_2 & \text{for } T_c > c_1 \\ 0 & \text{otherwise.} \end{cases}$$

$$(2.8) \quad U_2(v_c) = v_c/c_3 \geq 0$$

with  $c_1 = T_0 - 3 K$ , where  $T_0$ , [K], is the freezing temperature of water,  $c_2 = 1.87 \cdot 10^5$  Ks and  $c_3 = 1.56 \cdot 10^5$  m. For further reference the snow melt flux from the canopy is abbreviated as

$$(2.9) \quad M_{snc} = \rho_w h_{snc} U_1(T_c).$$

Because the canopy is not vertically resolved,  $T_c$  and  $v_c$  are not available in the model. Instead, the respective values at the lowest atmospheric model level are used. Consistent

with this assumption, the snow melt parameterized according to (2.7) results in a cooling of the lowest model layer by extraction of heat according to

$$(2.10) \quad C_p \frac{\Delta p}{g} \frac{\partial T_c}{\partial t} = -\rho_w h_{snc} U(T_c) L_{sn}.$$

Here  $\Delta p$  is the pressure thickness of the lowest model layer, and the  $g$  gravitational acceleration so that  $\Delta p/g$  is the air mass per  $m^2$  that is cooled by snow melt.  $C_p$  is the specific heat of air, and  $L_{sn}$  the latent heat for melting snow. This equation is solved for  $\partial T_c/\partial t$  and given as temperature tendency correction via the JSBACH interface to ECHAM (compare table 1.2).

For albedo calculations (section 4.1) also the fraction of the canopy covered by snow is of importance (denoted by  $c_{snc}$ ). It is assumed that this fraction is given by the relative filling of the canopy skin reservoir with snow:

$$(2.11) \quad c_{snc} = \frac{h_{snc}}{h_{snc}^{max}}.$$

### 2.3.1.2 Snow at the surface

The snow budget at the surface is given by

$$(2.12) \quad \rho_w \frac{\partial h_{sn}}{\partial t} = (1 - f_v)S + E_{sn} - M_{sn} + \rho_w h_{snc} U_2(v_c).$$

The first term on the right hand side represents the snowfall not intercepted by the canopy, i.e. the snow reaching the surface. The next term  $E_{sn} < 0$ , [kg/m<sup>2</sup>s], stands for sublimation of snow, and  $M_{sn}$ , [kg/m<sup>2</sup>s], is the snow melt rate

$$(2.13) \quad M_{sn} = \frac{C_s T^* - T_0}{L_f \Delta t}$$

where  $T_0 = 0^\circ\text{C}$  is the melting temperature of snow,  $C_s$  is the heat capacity of the upper soil layer modified for the presence of snow,  $\rho_w$  the density of water,  $L_{sn}$  the latent heat for melting the snow, and  $T^*$  is the ‘‘unfiltered’’ surface temperature obtained from the surface energy balance equation (2.1) where contributions from snow melt were not considered (see end of section 2.5.2). The ‘final’ surface temperature, i.e. including the cooling due to snow melt, is given by<sup>1</sup>

$$(2.14) \quad T_{surf} = T^* - \frac{L_f}{C_s} M_{sn} \Delta t.$$

For the special case of complete melting during one time step  $M_{sn} \Delta t$  is limited by the available snow amount  $\rho_w h_{sn}$  so that  $T^* > T_{surf} \geq T_0$ . Over ice sheets and glaciers, snow processes are neglected, i.e.  $h_{sn} = h_{snc} = 0$ . However, a melting term analogous to (2.13) is diagnosed for  $T^* > T_0$  and  $T_{surf}$  is set to  $T_0$  in this case.

### 2.3.1.3 Interception of rain by vegetation and bare soil

Analogous to snowfall, a fraction  $f_v R$ , [kg/m<sup>2</sup>s], of the incoming rain  $R$  is intercepted on the canopy and on bare soil. With evaporation  $E_{wsk} < 0$  from the skin reservoir, dew deposition

<sup>1</sup>This correction to surface temperature is done in `mo_soil_hydrology::update_surface_hydrology()`.

$E_{wsk} > 0$ , and melted snow from the canopy (compare (2.5) and (2.7)) the interception reservoir changes according to

$$(2.15) \quad \rho_w \frac{\partial h_{wsk}}{\partial t} = f_v R + E_{wsk} + \rho_w h_{snc} U_1(T_c),$$

unless the capacity

$$(2.16) \quad h_{snc}^{max} = h_0(1 + \Lambda veg_{max})$$

of the interception reservoir is exceeded, where  $1 + \Lambda veg_{max}$  stands for the total grid box surface ('1') plus the surface area on the leaves accounting for the limited extent of vegetation in the grid box (' $veg_{max}$ '); the excess water contributes to the soil water budget through the term  $M_{snc}$  in (2.17).

### 2.3.2 Soil water budget

Ignoring water exchange between grid boxes, in JSBACH the change in total soil water content  $h_{tot}^{(i)}$  (measured in [m]) of the  $i$ -th tile<sup>2</sup> is described by

$$(2.17) \quad \rho_w \frac{\partial h_{tot}^{(i)}}{\partial t} = (1 - f_v)R - E_{bs}^{(i)} - E_{tr}^{(i)} + M_{sn}^{(i)} + M_{snc}^{(i)} - R_{surf}^{(i)} - R_d^{(i)}.$$

Here,  $(1 - f_v)R$  is the fraction of rainfall  $R$  not intercepted by the canopy,  $E_{bs}$  is bare soil evaporation,  $E_{tr}$  transpiration,  $M_{sn}$  snow melt at the surface,  $M_{snc}$  snow melt on canopies,  $R_{surf}$  surface runoff and  $R_d$  drainage. The fluxes  $E_{bs}^{(i)}$  and  $E_{tr}^{(i)}$  (see (2.53) and (2.54)) are considered positive when directed into the atmosphere, while negative values represent dew formation. Note that not equation (2.17) is solved in JSBACH, because the in- and out-fluxes partially depend on the soil water content at different depths in the soil. Instead a similar equation that accounts for a vertically resolved soil moisture dynamics is solved from which (2.17) follows by vertical integration. For an evaluation of this part of JSBACH see [61]. – In the following, first this vertically resolved dynamics is introduced, then in subsequent sections the right hand side fluxes of (2.17) are specified, namely the fluxes for bare soil evaporation  $E_{bs}$ , transpiration  $E_{tr}$ , surface runoff  $R_{surf}$ , and drainage  $R_d$ , which completes the description of the JSBACH soil water dynamics

#### 2.3.2.1 Vertically resolved soil moisture dynamics

The state variable for the vertically resolved soil water dynamics is the *volumetric soil water content*  $\theta(z, t)$  at depth  $z$  (negative since below surface) and time  $t$ , where ‘‘volumetric’’ means volume soil water per volume soil. Assuming homogeneity in the horizontal directions, its vertical dynamics can be described by the one-dimensional Richards equation<sup>3</sup> [113]

$$(2.18) \quad \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} \left( D \frac{\partial \theta}{\partial z} \right) + \frac{\partial K}{\partial z} + S,$$

where the first term on the right hand side describes vertical diffusion ( $D = D(z, t)$  is the ‘soil water diffusivity’), while the second term models percolation from gravitational drainage

<sup>2</sup>Note that equation (2.17) is valid for every tile separately, meaning that the right hand side fluxes are calculated per square meter canopy. To indicate this, here for a moment a tile index  $i$  is added, while further below this index is once more omitted.

<sup>3</sup>Below surface the coordinate  $z$  is assumed to be negative.

( $K = K(z, t)$  is the ‘soil hydraulic conductivity’). The third term  $S = S(z)$  represents all water inputs and losses from the terms on the right hand side of Eq. (2.17), acting at different depths (see below). Eq. (2.18) is related to budget equation (2.17) by vertical integration from the bottom of the soils at  $z = z_{bot}$  to the surface at  $z = 0$ . In particular the total amount of soil water is obtained from  $\theta$  by

$$(2.19) \quad h_{tot} = \int_{z_{bot}}^0 dz \theta(z).$$

We specify the source term  $S(z)$  in (2.18) as

$$(2.20) \quad S = \delta(z) [(1 - f_v)R - E_{bs} + M_{sn} + M_{snc} - R_{surf}] - E_{tr}(z) - \delta(z - z_{bot})R_d.$$

Here the  $\delta$ -functions assure that water inputs and losses appear either at the surface  $z = 0$  or at depth  $z = z_{bot}$  where the explicit description of the soil water dynamics ends.<sup>4</sup> The only volume extraction takes place by depth-dependent transpiration  $E_{tr}(z)$  that adds up to  $E_{tr}$  appearing by vertical integration across the whole soil column. Roots are assumed to reach down to depth  $z_{root}$  that is fixed geographically (see section 2.10.3.1). Total transpiration  $E_{tr}$  is computed according to eq. (2.54) and water extraction from the soils takes place homogeneously down to the root depth, but only for soil moisture values  $\theta$  above the permanent wilting point<sup>5</sup>  $\theta_{pwp,veg}$ , i.e.

$$(2.21) \quad E_{tr}(z) = \begin{cases} \frac{E_{tr}}{|z_{root}|} & \text{for } z > z_{root} \text{ and } \theta > \theta_{pwp,veg} \\ 0 & \text{otherwise.} \end{cases}$$

According to this model for transpiration, if somewhere in the soil the water level is at or below the wilting point, the full amount of  $E_{tr}$  cannot be extracted from the soil, and  $E_{tr}$  reduces accordingly.<sup>6</sup>

The hydrological soil depth  $z_{bot}$  is either limited by a fixed maximum depth identical with that of the heat conduction model (section 2.10.2.2) or by the possibly shallower bedrock depth (see section 2.10). In the model there is no percolation of water into grounds deeper than this soil depth so that deep groundwater components like aquifers below the bedrock are not considered. Nevertheless, in a simplified way such flows are included in the drainage, as described below.

<sup>4</sup>Instead of interpreting surface and bottom fluxes as part of the source term  $S$ , one could also take these fluxes as boundary conditions to solve equ. (2.18).  $S$  would then contain only the volume losses by transpiration. Vertical integration of (2.18) shows that by considering the surface fluxes as part of  $S$  implies (i) that (2.18) must be solved with zero-flux boundary conditions, i.e.  $\partial\theta/\partial t|_{z=0} = \partial\theta/\partial t|_{z=z_{bot}} = 0$ , and (ii) that  $K(z=0) = K(z=z_{bot}) = 0$ . This is the approach taken in JSBACH to solve (2.18) numerically (see section 2.10.3).

<sup>5</sup>The permanent wilting point is the volumetric soil moisture value below which plants cannot extract water from the soil because the suction forces of the soil are larger than the transpiration forces plants can maximally develop for water uptake via roots. Note, however, that JSBACH uses two wilting points,  $\theta_{pwp,veg}$  and  $\theta_{pwp}$ .  $\theta_{pwp,veg}$  is the point at which plants in JSBACH stop transpiration. It is defined as a certain fraction of the field capacity  $\theta_{cap}$ , of a given layer within the root-zone. (For the fraction, typically 35%, see variable `moist_wilt_fract` in namelist `soil_ctl1`, appendix C.2.12). In contrast,  $\theta_{pwp}$  is the moisture level at which all vertical water movement in the soil ceases. This parameter is derived from [102] and is read in via the JSBACH initial file as variable `wilting_point`; see table C.22 in the appendix.

<sup>6</sup>In the current version of JSBACH this reduction of  $E_{tr}$  is artificially minimized by further extracting water from all layers also below root depth and below the wilting point. But still the full value of  $E_{tr}$  may in some rare cases not be extractable. This reduced value is neither played back to the transpiration flux into the atmosphere, nor to plant productivity that crucially depends on water availability (compare 5.3.1.4).

To compute the percolation by gravitational drainage, the hydraulic conductivity  $K$  is approximated as a function of  $\theta$  alone employing the van Genuchten model [141] as proposed by Disse [36]: Let

$$(2.22) \quad \Theta_{cap} := \begin{cases} 1 & \text{for } \theta \geq \theta_{cap} \\ \frac{\theta - \theta_{pwp}}{\theta_{cap} - \theta_{pwp}} & \text{for } \theta_{cap} > \theta > \theta_{pwp} \\ 0 & \text{for } \theta_{pwp} \geq \theta, \end{cases}$$

where  $\theta_{cap}$  is the value of  $\theta$  at ‘field capacity’; note that  $\theta$  may be larger than  $\theta_{cap}$  since the latter is typically smaller than the storage capacity at saturation (denoted as  $\theta_{sat}$  in the following) that is determined by the pore volume of the soils.  $\Theta_{cap}$  ranges between 0 and 1 and describes the water filling relative to the range between the wilting point  $\theta_{pwp}$  and field capacity  $\theta_{cap}$ . Then  $K$  can be approximated as function of this ‘residual water content’  $\Theta_{cap}$  by<sup>7</sup>

$$(2.23) \quad K = K_{sat} \sqrt{\Theta_{cap}} \left(1 - (1 - \Theta_{cap}^{1/m})^m\right)^2,$$

where the parameter  $m$  is obtained from the ‘pore size index’  $PSI$  as

$$(2.24) \quad m = \frac{PSI}{PSI + 1}.$$

Eq. (2.23) is constructed such that for water saturated soils (i.e. when  $\theta \geq \theta_{cap}$ ) one has  $K = K_{sat}$ . In the other extreme, when the soil water is at the wilting point or below, i.e.  $\Theta = 0$ , hydraulic conductivity vanishes. This is not fully realistic, but a very good approximation since gravitational percolation forces are typically much smaller than those arising from transpiration so that below the wilting point percolation fluxes must be negligible. But note that diffusion takes place also below the wilting point down to  $\theta = 0$  (see (2.25)).

Following Clapp and Hornberger [23], the soil water diffusivity can be approximated as a function of  $\theta$  by setting

$$(2.25) \quad D = b_{Clapp} K_{sat} \frac{\Psi_{sat}}{\theta} \left(\frac{\theta}{\theta_{sat}}\right)^{3+b_{Clapp}},$$

with Clapp&Hornberger parameter  $b_{Clapp}$ , and the following additional parameters of water saturated soils: the volumetric water content at saturation  $\theta_{sat}$ , and the ‘saturated moisture potential’  $\Psi_{sat}$ . – How those equations are solved numerically is described in section 2.10.3.2.

To complete the description of the soil water dynamics, in particular the infiltration and drainage fluxes must be specified – this will be done in the following two subsections. The remaining fluxes seen in the source term (2.20) either have already been described above (for  $M_{sn}$  and  $M_{snc}$  see eqs. (2.13) and (2.9)), or will be described below as part of an introduction to the general handling of surface fluxes in section 2.4 below; this concerns bare soil evaporation  $E_{bs}$  (see eq. (2.47)) and transpiration  $E_{tr}$  (see (2.54)).

### 2.3.2.2 Infiltration and surface runoff

Only part of the water reaching the surface (rain, snowmelt, and dew) is taken up by the soils – this is the infiltration flux called  $I$  below. The rest leaves the grid box as surface runoff  $R_{surf}$ . To solve the Richards equation (2.18) one needs the infiltration flux as boundary

<sup>7</sup>Formally we set also  $K$  to zero at  $z = 0$  and  $z = z_{bot}$ ; compare footnote 4.

condition. Hence, as a preparation to solve the Richards equation, the task is to split the water intercepted at the surface into infiltration and surface runoff.<sup>8</sup> For this purpose in JSBACH the Arno scheme [135], originally developed for the chinese Xinanjiang river catchment [112], is employed. The implementation in JSBACH follows [37]. The basic equation implemented is (2.37) below. In the following the derivation of this equation is described in some detail, because in this way the underlying assumptions become obvious. Moreover, the published explanations for this equation are quite cryptic<sup>9</sup> so that it seems worth to justify the implementation in JSBACH in some detail.

Picture the soil of a grid box to be composed of many small local water storages<sup>10</sup>. The storage depths  $w$  (i.e. stored cubic meter water per square meter) of the local storages are assumed to be statistically distributed according to the cumulated distribution<sup>11</sup>

$$(2.26) \quad f(w) = 1 - (1 - w/w_{max})^b$$

where  $w_{max}$  is the grid box wide maximum depth of the local storages<sup>12</sup> and the parameter  $b$  accounts for sub-grid scale characteristics of the surface, namely steepness: Following [37] the parameter  $b$  in (2.26) and (2.37) is a function of the steepness of the sub-grid scale terrain expressed as standard deviation  $\sigma_z$  [m] of topography height in the grid box according to

$$(2.27) \quad b = \begin{cases} 0.5 & \text{for } \sigma_{max} < \sigma_z \\ \frac{\sigma_z - \sigma_0}{\sigma_z + \sigma_{max}} & \text{for } \sigma_0 \leq \sigma_z \leq \sigma_{max} \\ 0.01 & \text{for } \sigma_z < \sigma_0 \end{cases}$$

where  $\sigma_z$  [m] is the standard deviation of topography height in the grid box, and  $\sigma_0$ ,  $\sigma_{max}$  are parameters, the latter being chosen as  $1000\text{m} \times 64/nlat$ , where  $nlat$  denotes the number of latitudes from North to South pole at the considered grid resolution.

Denoting by  $p(w)$  the probability to have within a grid box a local water storage of depth  $w$ , it is formally related to  $f(w)$  by

$$(2.28) \quad f(w) = \int_0^w dx p(x).$$

Let  $Q$  [m] denote the water available for infiltration and/or runoff during a time interval  $\Delta t$ . Runoff is produced from  $Q$  where the local storages are filled so that no further water can be taken up. A basic assumption is that from one time interval  $\Delta t$  to the next all the local storages in the grid box equalize their water levels. This level is denoted by  $w$ . Note that in this situation those local storages with depth less than  $w$  are completely filled and contain less water than indicated by  $w$ . To compute the surface runoff assume first that  $w + Q \leq w_{max}$ .

<sup>8</sup>The exact form of this splitting is seen from eqs. (2.38) and (2.39), namely  $(1 - f_v)R + M_{sn} + M_{snc} - \min(0, E_{bs}) - \min(0, E_{tr}) = I + R_{surf}$ .

<sup>9</sup>It seems that there is no publication with the final formula (2.37) printed correctly. Interestingly, the same sign error in [135] – erroneously the  $[\dots]^{1+b}$  term in eq. (2.37) is printed with a minus sign in front – is seen printed also in later publications on this subject.

<sup>10</sup>Todoni [135] calls these local water storages ‘elementary’.

<sup>11</sup>According to [37] and [135] this distribution was proposed in 1977 by R.J. Zhao in a paper named ‘Flood forecasting method for humid regions of China’. Since the cited reference is not accessible (but nevertheless massively cited) the empirical basis of this formula remains obscure.

<sup>12</sup>This ‘maximum depth’ can be understood as the maximum depth for infiltration during a certain time interval, e.g. the model time step. Technically,  $w_{max}$  is the field `maxmoist` read in during initialization (see table C.22).

Measuring runoff  $R_{surf}$  in  $[\text{kg}/\text{m}^2 \text{ s}]$ , the runoff produced during  $\Delta t$  by adding the amount of water  $Q$  is obtained from

$$(2.29) \quad \frac{\Delta t}{\rho_w} R_{surf}(w) = Qf(w) + \int_w^{w+Q} dx p(x) (w + Q - x), \quad \text{for } w + Q \leq w_{max}.$$

The first right hand side term is the runoff produced from that fraction  $f(w)$  of the grid box where all the local storages are completely filled even before  $Q$  is added. The second term describes the amount of water  $w + Q - x$  that cannot be added to the local storages already filled up to level  $x$ , weighted by the fraction of the grid box that is saturated at that level  $x$ . By partial integration of the second term while accounting for (2.28) one finds the much simpler formula

$$(2.30) \quad \frac{\Delta t}{\rho_w} R_{surf}(w) = \int_w^{w+Q} dx f(x), \quad \text{for } w + Q \leq w_{max}.$$

In the second case  $w + Q > w_{max}$ , the upper integration limit in (2.29) must be replaced by  $w_{max}$  since this is the maximum that can be stored in the local storages. Partial integration then gives

$$(2.31) \quad \frac{\Delta t}{\rho_w} R_{surf}(w) = \int_w^{w_{max}} dx f(x), \quad \text{for } w + Q > w_{max}.$$

Entering (2.26) these two equations for runoff can be easily integrated, but the result is in this form not very helpful since the sub-grid scale values  $w$  and  $w_{max}$  are not known. But these two values can be related to grid-box-wide values, i.e. to values at the description level of JSBACH. The relation is obtained as follows. For given  $w$  the total water  $W$  stored in all local storages of the considered grid box is

$$(2.32) \quad W(w) = w(1 - f(w)) + \int_0^w dx p(x) x = w - \int_0^w dx f(x),$$

where the first right hand side term is the amount of water in all local storages in that fraction of the grid box where the water holding capacity is less or equal  $w$ . The second term integrates the amount of water in the rest of the grid box where all local storages are filled up to  $x \leq w$ . The expression after the second equal sign is obtained by partial integration. Invoking the particular distribution (2.28) in this expression one finds

$$(2.33) \quad W(w) = -\frac{w_{max}}{b+1} \left[ \left(1 - \frac{w}{w_{max}}\right)^{b+1} - 1 \right].$$

Evaluating this equation for maximum filling, i.e.  $w = w_{max}$ , one obtains the maximum water holding capacity  $W_{max}$  from all local storages in the grid box

$$(2.34) \quad W_{max} = \frac{w_{max}}{b+1}.$$

Using this in (2.33) one obtains the simple relation

$$(2.35) \quad 1 - \frac{w}{w_{max}} = \left(1 - \frac{W}{W_{max}}\right)^{1/(b+1)}.$$

Using these equations one can translate the condition  $w + Q \leq w_{max}$  that distinguishes the two cases for computing runoff above into a relation involving  $W$  and  $W_{max}$  only:

$$(2.36) \quad w + Q \leq w_{max} \Leftrightarrow Q \leq (b + 1)W_{max} \left(1 - \frac{W}{W_{max}}\right)^{1/(b+1)}.$$

Integrating now eqs. (2.30) and (2.31) for the distribution (2.28) and invoking the relations just derived, one finally finds

$$(2.37) \quad \frac{\Delta t}{\rho_w} R_{surf} = Q - (W_{max} - W) + \frac{1}{(b + 1)^{b+1} (W_{max})^b} \begin{cases} \left[ (b + 1)W_{max} \left(1 - \frac{W}{W_{max}}\right)^{1/(1+b)} - Q \right]^{b+1} & \text{for } [\dots] \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

The infiltration related to this runoff is

$$(2.38) \quad I = \frac{\rho_w}{\Delta t} Q - R_{surf}.$$

These are the equations for runoff and infiltration implemented in JSBACH, but with the following modifications:

- $R_{surf} = 0$ , i.e.  $I = Q$ , for  $W < 0.05W_{max}$ .
- $R_{surf} = Q$ , i.e.  $I = 0$ , for frozen soil, i.e. when surface temperature drops below zero.
- The value of  $I$  calculated from (2.38) might lead to water levels  $\theta$  above field capacity. In that case infiltration is effectively reduced by adding the surplus water to drainage. The value of  $R$  remains unchanged.

The water influx needed for evaluation of (2.37) is obtained by integrating all right hand side fluxes in (2.17) over the time interval  $\Delta t$ :

$$(2.39) \quad Q = \frac{\Delta t}{\rho_w} [(1 - f_v)R + M_{sn} + M_{snc} - \min(0, E_{bs}) - \min(0, E_{tr})],$$

where the minimum values are taken to separate the water uptake from the formation of dew from the evapotranspiration losses.

It remains to be clarified what exactly is meant by  $\Delta t$ ,  $W_{max}$  and  $W$ . In JSBACH  $\Delta t$  is the time step of the model, which in simulations coupled to the atmosphere varies with spatial resolution (e.g.  $\approx 7.5$  minutes at T63, or even less at T123). The Arno model assumes that the local water levels equilibrate during this time interval. Hence this sets a minimum length for the time step of integration. Moreover, applying equation (2.37) twice at half the time step is not identical to applying it at the full time step. This dependence comes in only via the bracket term  $[\dots]$  in the first case, while the first right hand side term is due to its linearity independent from a separation into sub-time steps. Despite the wide usage of the Arno model, the strength of this time step dependence does not appear to be fully investigated so far.

The active volume  $W_{max}$  determining runoff is taken in JSBACH from a world-wide map of plant available water [61], while  $W$  is the actual soil water content down to root depth  $z_{root}$  (also a world wide map [61]), i.e.

$$(2.40) \quad W = \int_{z_{root}}^0 dz \theta(z).$$

Taking  $z_{root}$  as characterizing depth for the active volume  $W_{max}$  is a compromise upon lack of empirical data for  $W$ .

In later sections the relative amount of water in the root zone is needed. This is defined as

$$(2.41) \quad w_{rel} = \frac{W}{W_{max}}.$$

### 2.3.2.3 Drainage $R_d$

Drainage describes the loss of water at the bottom boundary of the modeled soil column, i.e. at  $z = z_{bot}$ . The model for drainage  $R_d$  is a slight modification [61] of the ECHAM4 formulation [115] that follows Dümenil and Todini [37]. Analogously to  $\Theta_{cap}$  from above, another ‘residual’ soil moisture  $\Theta_{sat}$  can be defined relative to saturation capacity  $\theta_{sat}$  instead of field capacity  $\theta_{cap}$  by

$$(2.42) \quad \Theta_{sat} := \begin{cases} \frac{\theta - \theta_{pwp}}{\theta_{sat} - \theta_{pwp}} & \text{for } \theta > \theta_{pwp} \\ 0 & \text{for } \theta_{pwp} \geq \theta. \end{cases}$$

Using both,  $\Theta_{cap}$  and  $\Theta_{sat}$ , drainage is implemented in JSBACH as

$$(2.43) \quad R_d = \begin{cases} a_{min} \Theta_{cap} + (a_{max} - a_{min}) \left[ \frac{\Theta_{sat} - 0.9}{0.1} \right]^{a_{exp}} & \text{for } \Theta > 0.9 \\ a_{min} \Theta_{cap} & \text{otherwise.} \end{cases}$$

For the function parameters  $a_{min}$ ,  $a_{max}$  and  $a_{exp}$  see table 2.4. In this modified model, drainage appears only as long as soil moisture exceeds the permanent wilting point<sup>13</sup> i.e. for  $\theta > \theta_{pwp}$ . Although the wilting point is a characteristic of plants, this value is used here because suction forces of plants are usually stronger than the forces of gravity (e.g. [97]) so that the wilting point provides a general lower limit for the occurrence of drainage.

## 2.4 Surface exchange Fluxes

This section summarizes the moisture and heat fluxes entering the energy balance (2.1). The energy balance is not performed separately for each tile, but for each grid box as whole. Accordingly, the surface fluxes entering the energy balance are from all tiles together. And since they enter in the same way, in the following a simplified notation used, where the tile index  $i$  is omitted.

### 2.4.1 Moisture Fluxes

#### 2.4.1.1 Basic types of moisture fluxes

Potential evaporation: From the wet or snow covered parts of a tile (which includes wet surfaces on vegetation) evaporation happens at potential rate

$$(2.44) \quad E_{pot} = \rho \frac{q_a - q_{sat}(T_{surf}, p_{surf})}{r_a},$$

<sup>13</sup>In the original ECHAM4 bucket soil water model drainage happened whenever total water content exceeded 5% of the water content at saturation.

where  $\rho$  is air density,  $q_a$  specific humidity in lowest atmospheric level,  $q_{sat}$  saturation specific humidity at surface temperature  $T_{surf}$  and pressure  $p_{surf}$ . Note that in case of dew formation, i.e. when  $q_a > q_{sat}$ ,  $E_{pot}$  gets negative. Aerodynamic resistance<sup>14</sup>

$$(2.45) \quad r_a = (C_h |\vec{v}_h|)^{-1}$$

depends on the absolute value of horizontal velocity  $\vec{v}_h$  and the heat transfer coefficient  $C_h$  obtained from surface roughness and atmospheric stability (see ECHAM 6 documentation [55, section 2.4.3.]).<sup>15</sup>

Transpiration: Transpiration from the dry vegetation is computed as

$$(2.46) \quad E_{tr} = \rho \frac{q_a - q_{sat}(T_{surf}, p_{surf})}{r_a + r}$$

where the stomatal resistance  $r$  is the inverse of the stomatal canopy conductance  $g_{C, stress}^{H_2O}$  obtained from the models for photosynthesis and stomatal control (see eq. (5.49)).

Bare soil evaporation: Evaporation from dry bare soil is determined from

$$(2.47) \quad E_{bs} = \rho \frac{q_a - h q_{sat}(T_{surf}, p_{surf})}{r_a}$$

where relative humidity  $h$  at the surface is related to volumetric soil moisture  $\theta$  in the top soil by

$$(2.48) \quad h = \max \left[ 0.5 \left( 1 - \cos \left( \pi \frac{\theta(z=0)}{\theta_{cap}} \right) \right), \min \left( 1, \frac{q_a}{q_{sat}(T_{surf}, p_{surf})} \right) \right].$$

### 2.4.1.2 Accounting for hydrological surface structure

The moisture fluxes introduced above occur at different types of surfaces. Accordingly, the total evaporation from a tile is a weighted average of these fluxes, where the weights account for the fraction of the particular type of surface in the tile. To obtain these weights it is useful to distinguish three types of surfaces with the following surface fractions:

- $c_{sn}$ : fraction of tile covered with snow (“snow fraction”),
- $c_{veg}$ : fraction of tile covered with vegetation,
- $c_{wet}$ : fraction of tile covered with water (“wet fraction”).

The snow fraction is computed following Roesch et al. [118] as

$$(2.49) \quad c_{sn} = 0.95 \tanh(100 h_{sn}) \left( 1 + 0.15 \frac{\sigma_z}{1000 h_{sn}} \right)^{-\frac{1}{2}}$$

where  $h_{sn}$  is the snow reservoir filling (2.12) in the particular tile and  $\sigma_z$  the standard deviation of orography in the grid box (compare table 2.3);  $\sigma_z$  enters the equation to account for less snow coverage in strongly sloped regions. The wet fraction is obtained from the relative filling of the interception reservoir:

$$(2.50) \quad c_{wet} = h_{wsk} / h_{wsk}^{max},$$

<sup>14</sup>In the code often the inverse  $1/r_a = c_{30} * cdrag$  is used, where  $c_{30} = 1/\alpha g 2 \Delta t$  and  $cdrag = c_{30} \rho C_h |\vec{v}|$  is calculated in the ECHAM routine `mo_surface::precalc_land()` as variable `zcfhl`.

<sup>15</sup>Technically,  $C_h$  is the variable `echam_zchl` in the JSBACH interface (see `tabletab:ch-1.JSBACH-interfaceInput.`) and calculated as `zchl` in the ECHAM routine `mo_surface_land::precalc_land()`.

and the fraction of a tile covered with vegetation is given as

$$(2.51) \quad c_{veg} = veg_{max}(1 - e^{-\Lambda/\kappa})$$

where the term in brackets accounts for gaps in the canopy in the same spirit as discussed in the context of eq. (1.7) although here the actual time dependent leaf area index  $\Lambda = \Lambda(t)$  as computed by the phenology model (chapter 3) is taken.

On the basis of these surface fractions, the moisture flux from a square meter ground in tile  $i$  is computed as follows. Potential evaporation  $E_{pot}$  happens only at those surface fractions of a tile that are either snow covered (fraction  $c_{sn}$  of the tile) or not snow covered but with a wet surface (fraction  $(1 - c_{sn})c_{wet}$  of the tile). Therefore evaporation of this type from tile  $i$  is given as

$$(2.52) \quad E_{pot}^{(i)} = [c_{sn} + (1 - c_{sn})c_{wet}]E_{pot}.$$

Evaporation from the top soil layer (“bare soil evaporation”) happens only where the surface is neither snowed, neither wet, nor covered by vegetation. In addition the atmosphere must be able to take up moisture, i.e. relative surface humidity  $h$  must be sufficiently large. This gives

$$(2.53) \quad E_{bs}^{(i)} = \begin{cases} [(1 - c_{sn})(1 - c_{wet})(1 - c_{veg})]E_{bs} & \text{for } h > q_a/q_{sat} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, transpiration from vegetation can not happen from vegetation whose surface is wet or snow covered, and the soil water level  $W$  (see equ. (2.40)) must be above the permanent wilting point  $W_{wilt} = f_{wilt}W_{max}$ , with  $f_{wilt}$  as described in the context of eq. (5.48) and the  $W_{max}$  the maximum root zone water content (compare table 2.3). Therefore

$$(2.54) \quad E_{tr}^{(i)} = \begin{cases} [(1 - c_{sn})(1 - c_{wet})c_{veg}]E_{tr} & \text{for } W_{tot} > W_{wilt} \\ 0 & \text{otherwise.} \end{cases}$$

Accordingly, total evapotranspiration from a grid box is

$$(2.55) \quad E = \sum_{i=1}^K c_i(E_{pot}^{(i)} + E_{bs}^{(i)} + E_{tr}^{(i)}),$$

where  $c_i$  is the cover fraction of tile  $i$  (compare (1.5)). Noting further that the formulas for potential evaporation, transpiration, and bare soil evaporation all involve (up to a factor) the difference between specific humidity  $q_a$  in the lowest atmospheric layer and saturated specific humidity  $q_{sat}$  at the surface (compare eqs. (2.44), (2.46), and (2.47)), total evapotranspiration can as well be written as

$$(2.56) \quad E = \rho C_h |\vec{v}| (C_{air} q_a - C_{sat} q_{sat}),$$

where  $C_{air}$  and  $C_{sat}$  can be identified from the foregoing equations.<sup>16</sup> This representation of total evapotranspiration is used to calculate the exchange of moisture with the atmosphere (compare eq. (2.68)).

<sup>16</sup>In the ECHAM/JSBACH code  $C_{atm}$  and  $C_{surf}$  are named `zcair` and `zcsat`.

### 2.4.2 Heat Fluxes

The ground heat flux  $G$  is obtained from the solution of the thermal diffusion equation (2.3). For sensible heat one needs the dry static energy defined by

$$(2.57) \quad s := c_p T + g(z - z_{surf}),$$

where  $c_p$  is the heat capacity of the air at constant pressure from eq. (2.84), and  $g(z - z_{surf})$  is the gravitational potential at height  $z$  relative to the height at the surface  $z_{surf}$ .<sup>17</sup> Thereby the sensible heat flux  $H$  is obtained from the gradient of dry static energy according to

$$(2.58) \quad H_{sensible} = \rho \frac{s_a - s_{surf}}{r_a}.$$

Here  $s_a$  and  $s_{surf}$  are dry static energy in the lowest atmospheric layer and at the surface,  $\rho$  is the density of dry air and  $r_a$  is the aerodynamic resistance (2.45).

To derive the latent heat flux from evapotranspiration it is convenient to first write down an expression for the total evaporation from snowed surfaces in a grid box:

$$(2.59) \quad E_{sn} = c_{sn} E_{pot}.$$

This expression is independent of the tile structure of the grid boxes, because it is assumed that all tiles participate in snow cover with the same fraction  $c_{sn}$ . Hence the latent heat flux can be written as

$$(2.60) \quad \begin{aligned} H_{latent} &= L_{evap}(E - E_{sn}) + L_{subl}E_{sn} \\ &= \rho C_h |\vec{v}| (\mathcal{D}_{air} q_a - \mathcal{D}_{sat} q_{sat}), \quad \text{with} \quad \begin{cases} \mathcal{D}_{air} := L_{evap}(C_{air} - c_{sn}) + L_{subl} c_{sn} \\ \mathcal{D}_{sat} := L_{evap}(C_{sat} - c_{sn}) + L_{subl} c_{sn} \end{cases} \end{aligned}$$

where in the second line eqs. (2.56) and (2.44) have been invoked, so that finally by introducing the abbreviations  $\mathcal{D}_{air}$  and  $\mathcal{D}_{sat}$  this expression for latent heat assumes a similarly simple form as that for total evaporation (2.56).<sup>18</sup>

## 2.5 Coupling to the atmosphere

JSBACH provides the lower boundary conditions for ECHAM. This concerns albedo, surface roughness, and the fluxes of sensible and latent heat. While the calculation of albedo and roughness is described in chapter 4, the present section describes how JSBACH provides those flux boundary conditions to ECHAM. This involves in particular the calculation of surface temperature by employing the surface energy balance.

From the viewpoint of ECHAM, the calculation of surface temperature in JSBACH is part of the calculations of the vertical distribution of temperature and humidity in the atmosphere. Such calculations are performed for each column of the atmospheric grid with different calculations for lower boundary conditions over land, ocean, and sea ice.<sup>19</sup> The

<sup>17</sup>This convention of vanishing gravitational potential at the surface is made possible by ECHAM's usage of hybrid orography-following pressure level coordinates [55].

<sup>18</sup>In the subroutine `update_surfacetemp()`,  $\mathcal{D}_{air}$  and  $\mathcal{D}_{sat}$  are represented by the variables `zca` and `zcs`.

<sup>19</sup>In ECHAM this numerical scheme for vertical turbulent transport is implemented in the subroutine `vdiff()`. In `vdiff()` the call to the JSBACH interface happens indirectly via calling the routine `update_surface()`, where first atmospheric arrays are split for calculations over the different surface types distinguished by ECHAM (ocean, sea ice, land). These arrays are then passed within `update_surface()` to the respective routines (like JSBACH for land) to update the respective surface conditions so that back in ECHAM the calculation of vertical turbulent transport can be completed.

implicit numerical scheme for solving the respective diffusion equations for vertical turbulent transport (following Richtmyer and Morton [114]) leads to a tri-diagonal system of linear equations. It is solved iteratively by a variant of the Thomas algorithm (see e.g. [107]). The iterative solution starts at the top of the atmosphere assuming zero boundary conditions for heat and moisture fluxes. Working downwards, a sequence of coefficients – known as Richtmyer-Morton coefficients – is calculated for lower and lower layers until the surface is reached. Knowing the coefficients at the surface, JSBACH then calculates with help of the surface energy balance the surface values for saturated humidity and dry static energy, which gives surface temperature. Providing this information to ECHAM, finally dry static energy and specific humidity of all atmospheric layers are updated according to Thomas' algorithm, by working iteratively through all layers from the surface to the top of the atmosphere.

This gives a first idea how JSBACH is embedded into ECHAM's calculations of vertical turbulent transport. And it makes clear that in order to understand how JSBACH calculates the flux boundary conditions, it is first necessary to present how in ECHAM the diffusion equation for vertical turbulent transport is solved (section 2.5.1 below) before then the calculations in JSBACH that lead in particular to the surface temperature can be presented (section 2.5.2 below). The theory for vertical turbulent transport is described in the ECHAM6 documentation [55, section 2.4.4] and the land-atmosphere coupling is described in some detail in [120, 121].

### 2.5.1 Numerical solution of vertical turbulent transport in ECHAM

This section describes how in ECHAM the diffusion equation for vertical turbulent transport of its prognostic variables is solved. Denoting the considered prognostic variable by  $X$ , which in the present context can be dry static energy or specific humidity (see next section), the diffusion equation to be solved is<sup>20</sup>

$$(2.61) \quad \frac{\partial X}{\partial t} = \frac{1}{\rho} \frac{\partial F}{\partial z} = \frac{1}{\rho} \frac{\partial}{\partial z} \left( \rho K \frac{\partial X}{\partial z} \right),$$

Here  $z$  measures height,  $F(z, t)$  is the flux of the considered quantity  $X(z, t)$ ,  $\rho(z, t)$  is the density of air, and  $K(z, t)$  is the diffusion coefficient, also denoted as 'exchange coefficient' [121] or 'turbulent viscosity' [55]. In ECHAM, using a TKE closure scheme,  $K$  is calculated from

$$(2.62) \quad K = l S_X \sqrt{E},$$

with  $l$  being the turbulent mixing length,  $S_X$  the vertical stability of  $X$ , and  $E$  kinetic energy [55, eq. (2.152)]. ECHAM is not using (2.61) directly, but only after transformation to pressure level coordinates. Implicit to the usage of such coordinates is the hydrostatic approximation  $dp/dz = -g\rho$ , where  $p$  is pressure and  $g$  is Earth's gravitational acceleration. Accordingly, the vertical derivatives in (2.61) transform as  $\partial/\partial z = -g\rho\partial/\partial p$  so that the transformed equation can be written as<sup>21</sup>

$$(2.63) \quad \frac{\partial X}{\partial t} = -g \frac{\partial}{\partial p} \left( \rho K \frac{\partial X}{\partial z} \right),$$

<sup>20</sup>Note that by convention downward fluxes are considered positive, i.e. against the positive direction of the  $z$  coordinate. This explains why  $\partial X/\partial t$  has the same sign as the flux divergence, and why the flux has the same sign as the gradient in  $z$ .

<sup>21</sup>Formally, the transformation from spatial coordinates to pressure level coordinates gives an additional term on the left hand side, since  $\partial X(z, t)/\partial t|_t = \partial X(p, t)/\partial t|_t + \partial X(p, t)/\partial p|_t dp/dt$ . Here, the second term arises from the time dependence of pressure level heights and is handled as part of the advection in ECHAM. For the present discussion of vertical turbulent fluxes it can thus be omitted.

where for convenience only the first vertical derivative was replaced.

To solve this equation numerically, a semi-implicit leap frog scheme is employed by introducing the mixed-time quantity

$$(2.64) \quad \widehat{X}^t := \alpha X^{t+\Delta t} + (1 - \alpha) X^{t-\Delta t},$$

where  $\alpha := 3/2$  controls the relative implicit and explicit contributions of the solution scheme. Since thereby

$$(2.65) \quad \frac{X^{t+\Delta t} - X^{t-\Delta t}}{2\Delta t} = \frac{\widehat{X}^t - X^{t-\Delta t}}{2\alpha\Delta t}$$

the discretization in time of (2.63) can be written as

$$(2.66) \quad \widehat{X}^t - X^{t-\Delta t} = -2\alpha g \Delta t \frac{\partial}{\partial p} \left( (\rho K)^{t-\Delta t} \frac{\partial \widehat{X}^t}{\partial z} \right),$$

where  $X$  at the right hand side is chosen to be evaluated by the mixed time quantity  $\widehat{X}^t$ . Note that  $\rho$  and  $K$  are evaluated at time  $t - \Delta t$  because for time  $t$  their values are not known when doing the leap from  $t - \Delta t$  to  $t + \Delta t$  (see footnote 28).

Next the vertical discretization is considered. For this purpose the notation depicted in Fig.2.1 is used: Pressure layers are numbered by  $n = 1, 2, \dots, N$  from the highest ( $n = 1$ ) to the lowest ( $n = N$ ) layer. Mid level values are indexed by  $n$ , while indices  $n - 1/2$  and  $n + 1/2$  indicate that quantities are taken at the upper or lower layer boundaries, respectively. Pressure and height differences are always taken to be positive, i.e.  $\Delta p_n := p_{n+1/2} - p_{n-1/2}$  and  $\Delta z_{n+1/2} := z_n - z_{n+1}$ . Using this notation, and taking boundary conditions into account, namely zero flux at the top of the atmosphere ( $n = 1$ ) and given flux  $\mathcal{F}_{surf}$  at the surface ( $n = N$ ), eq. (2.66) can be discretized as

$$(2.67) \quad \begin{aligned} \widehat{X}_n^t - X_n^{t-\Delta t} &= -\frac{2\alpha g \Delta t}{\Delta p_n} (\rho K)_{n+1/2}^{t-\Delta t} \left. \frac{\partial \widehat{X}^t}{\partial z} \right|_{z_{n+1/2}} \\ &= -\frac{2\alpha g \Delta t}{\Delta p_n} (\rho K)_{n+1/2}^{t-\Delta t} \frac{\widehat{X}_n^t - \widehat{X}_{n+1}^t}{\Delta z_{n+1/2}} && \text{for } n = 1, \\ \widehat{X}_n^t - X_n^{t-\Delta t} &= \frac{2\alpha g \Delta t}{\Delta p_n} \left( (\rho K)_{n-1/2}^{t-\Delta t} \left. \frac{\partial \widehat{X}^t}{\partial z} \right|_{z_{n-1/2}} - (\rho K)_{n+1/2}^{t-\Delta t} \left. \frac{\partial \widehat{X}^t}{\partial z} \right|_{z_{n+1/2}} \right) \\ &= \frac{2\alpha g \Delta t}{\Delta p_n} \left( (\rho K)_{n-1/2}^{t-\Delta t} \frac{\widehat{X}_{n-1}^t - \widehat{X}_n^t}{\Delta z_{n-1/2}} - (\rho K)_{n+1/2}^{t-\Delta t} \frac{\widehat{X}_n^t - \widehat{X}_{n+1}^t}{\Delta z_{n+1/2}} \right) \\ &&& \text{for } n = 2, 3, \dots, N - 1, \\ \widehat{X}_n^t - X_n^{t-\Delta t} &= \frac{2\alpha g \Delta t}{\Delta p_n} \left( (\rho K)_{n-1/2}^{t-\Delta t} \left. \frac{\partial \widehat{X}^t}{\partial z} \right|_{z_{n-1/2}} - \mathcal{F}_{surf}^t \right) \\ &= \frac{2\alpha g \Delta t}{\Delta p_n} \left( (\rho K)_{n-1/2}^{t-\Delta t} \frac{\widehat{X}_{n-1}^t - \widehat{X}_n^t}{\Delta z_{n-1/2}} - \mathcal{F}_{surf}^t \right) && \text{for } n = N. \end{aligned}$$

<sup>22</sup> Associating  $X$  with level mids and fluxes with level boundaries is consistent with the convention used throughout ECHAM (see [55, section 2.1.3.] and in particular Fig. 3 in [54]). In contrast, the figure given in [120], associating quantities  $X$  with layer boundaries and fluxes with layer mids, is inconsistent with the ECHAM convention; also inconsistent is the figure shown in [103].

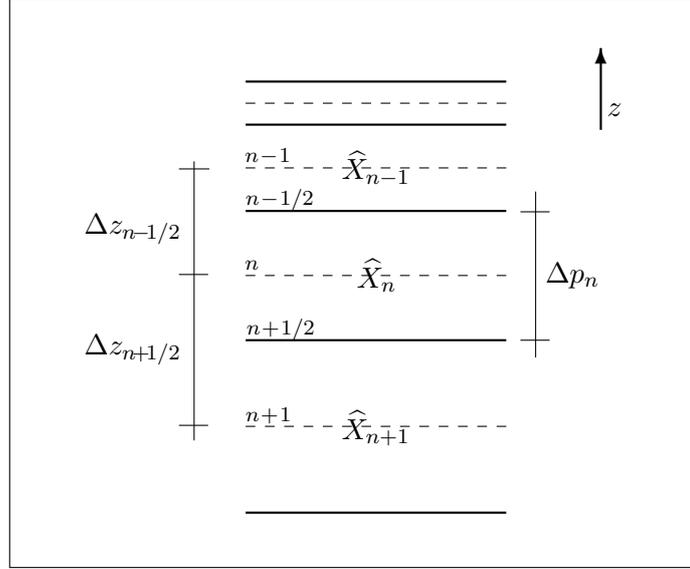


Figure 2.1: Notation used for the vertical discretization of the diffusion equation (2.66). Solid lines depict boundaries of atmospheric layers, while broken lines indicate layer mids. Layers are numbered  $n = 1, 2, \dots, N$  down from top of atmosphere. Transported quantities  $\widehat{X}_n$  are defined at layer mids while the associated fluxes are defined at layer boundaries and thus are indexed by  $n - 1/2$  (top layer boundary) and  $n + 1/2$  (bottom layer boundary). Pressure and height differences are defined such they are positive.<sup>22</sup>

For solving this system of equations the surface flux  $\mathcal{F}_{surf}^t$  appearing for  $n = N$  and taken at time  $t$  needs to be specified. In connection with JSBACH only the cases of latent and sensible heat flux are of importance.<sup>23</sup> In these cases the flux is proportional to the gradient of the transported quantity  $X$  at the surface and can generally be written as

$$(2.68) \quad \mathcal{F}_{surf}^t = \rho (C_X |\vec{v}|)^{t-\Delta t} \left( C_{atm} \widehat{X}_N^t - C_{surf} \widehat{X}_{surf}^t \right),$$

where  $\vec{v}$  is the horizontal wind velocity in the lowest atmospheric layer and  $C_X$  is the bulk transfer coefficient for quantity  $X$  calculated by ECHAM (see [55, eq. (2.173)]).  $C_X$  depends on the stability conditions of the lower atmosphere and in particular on roughness length  $z_0$ , calculated every time step by JSBACH (see section 4.2). The factors  $C_{atm}$  and  $C_{surf}$  turn out to be needed when applying this formalism to the transport of moisture in the next section (compare also (2.56)), in other cases  $C_{atm} = C_{surf} = 1$ .

While the above discretization scheme is implicit by construction, in addition by the way the surface flux is written in (2.68) also the *coupling* is implicit<sup>24</sup> because  $\widehat{X}$  involves  $X$  at time  $t + \Delta t$  (compare (2.64)). Nevertheless, the coefficients are evaluated at time  $t - \Delta t$ , consistent with the handling of all other coefficients in the system of equations (2.67). Note that the difference representing the gradient in (2.68) involves a value of  $\widehat{X}$  at a layer boundary, namely its value at the surface. Accordingly, in this expression the meaning of  $\Delta z_{N+1/2}$  is not – as would be expected from the notation introduced in Fig. 2.1 – the distance

<sup>23</sup>For the vertical turbulent transport of momentum the surface flux is given as  $\vec{\mathcal{F}}_{surf}^t = \rho (C_m |\vec{v}| \vec{v})^{t-\Delta t}$ . Here the exchange coefficient  $C_m$  depends on the roughness length  $z_0$  calculated in JSBACH (see section 4.2), but otherwise this flux is independent from calculations in JSBACH and is thus calculated directly in ECHAM to solve in the subroutine `vdiff()` the system of equations (2.67).

<sup>24</sup>In [103] Polcher distinguishes four types of coupling: Writing the difference in (2.68) in simplified form as  $(X_N^{t'} - X_{surf}^{t''})$ , he calls the coupling *implicit* for  $(t', t'') = (t + \Delta t, t + \Delta t)$ , *semi-explicit* for  $(t', t'') = (t + \Delta t, t - \Delta t)$ , *explicit* for  $(t', t'') = (t - \Delta t, t + \Delta t)$ , and *open-explicit* for  $(t', t'') = (t - \Delta t, t - \Delta t)$ .

between two mid-levels, but the distance between the mid level of the lowest layer and its lower boundary, i.e.  $\Delta z_{N+1/2}$  is only half the thickness of the lowest level.

By the way the time dependence is written in (2.68), the problem of not knowing the surface flux needed to solve the system of eqs. (2.67) is shifted to the problem of not knowing the value of  $X$  at the surface (i.e.  $\widehat{X}_{surf}^t$ ). As will be described below, the energy balance at the surface will be invoked to obtain this quantity.

But first the system of eqs. (2.67) needs to be rewritten in a form that can be efficiently solved. For this purpose it is convenient to set artificially the value of the turbulent diffusion coefficient  $K$  at the top of the atmosphere ( $n = 1/2$ ) and at the surface ( $n = N + 1/2$ ) to the values

$$(2.69) \quad \begin{aligned} (\rho K)_{1/2}^{t-\Delta t} &:= 0 \\ (\rho K)_{N+1/2}^{t-\Delta t} &:= \Delta z_{N+1/2} \rho \mathcal{C}_{atm} (C_X |\vec{v}|)^{t-\Delta t}. \end{aligned}$$

These definitions can be made without running into conflicts with  $K$  obtained for the inner vertical column from (2.62) because these two coefficients describing the situation at the boundaries are not used in the above system of eqs. (2.67). Introducing next the abbreviation

$$(2.70) \quad A_k^t := 2\alpha g \Delta t \frac{(\rho K)_k^t}{\Delta z_k}, \quad \text{for } k = n - 1/2 \text{ or } k = n + 1/2, \quad n = 1, 2, \dots, N,$$

and introducing formally the value of  $\widehat{X}$  at the surface as the value of an imaginary  $N$ -plus-first layer by setting

$$(2.71) \quad \widehat{X}_{N+1}^t := \frac{\mathcal{C}_{surf}}{\mathcal{C}_{atm}} \widehat{X}_{surf}^t,$$

the system of equations (2.67) reads<sup>25</sup>

$$(2.72) \quad X_n^{t-\Delta t} = -\frac{A_{n-1/2}^{t-\Delta t}}{\Delta p_n} \widehat{X}_{n-1}^t + \left( 1 + \frac{A_{n-1/2}^{t-\Delta t}}{\Delta p_n} + \frac{A_{n+1/2}^{t-\Delta t}}{\Delta p_n} \right) \widehat{X}_n^t - \frac{A_{n+1/2}^{t-\Delta t}}{\Delta p_n} \widehat{X}_{n+1}^t. \\ n = 1, 2, \dots, N,$$

a formulation that now covers by a single equation also the relations of (2.67) at the upper ( $n = 1$ ) and lower ( $n = N$ ) boundaries thanks to the definitions (2.69) and (2.71). Further simplification is obtained by introducing the additional abbreviations

$$(2.73) \quad a_n := \frac{A_{n+1/2}^{t-\Delta t}}{\Delta p_n}, \quad c_n := \frac{A_{n-1/2}^{t-\Delta t}}{\Delta p_n} \quad n = 1, 2, \dots, N.$$

so that (2.72) reads even more compactly

$$(2.74) \quad X_n^{t-\Delta t} = -c_n \widehat{X}_{n-1}^t + (1 + c_n + a_n) \widehat{X}_n^t - a_n \widehat{X}_{n+1}^t, \quad n = 1, 2, \dots, N.$$

In this way the whole system of eqs. (2.67) describing the vertical turbulent transport has been converted into a two-step recursion that can be solved by the ansatz

$$(2.75) \quad \widehat{X}_n^t = E_n \widehat{X}_{n+1}^t + \alpha F_n, \quad n = 1, 2, \dots, N.$$

<sup>25</sup>Compare eq. (A4) in [121] and eqs. (2.196), (2.198) in [55].

where  $E_n$  and  $F_n$  are two new parameters, called Richtmyer-Morton coefficients.<sup>26</sup> To determine their values, one can use this ansatz to replace  $\widehat{X}_{n-1}^t$  in (2.74) so that the resulting equation is a one-step recursion in  $\widehat{X}_n^t$  and  $\widehat{X}_{n+1}^t$  that can be compared with (2.75) to obtain equations for  $E_n$  and  $F_n$ .<sup>27</sup>

$$(2.76) \quad \left. \begin{aligned} E_n &= \frac{a_n}{1 + c_n(1 - E_{n-1}) + a_n}, \\ F_n &= \frac{x_n^{t-\Delta t}/\alpha + c_n F_{n-1}}{1 + c_n(1 - E_{n-1}) + a_n}, \end{aligned} \right\} \text{ for } n = 1, \dots, N \quad \text{with} \quad \left\{ \begin{aligned} E_1 &= \frac{a_1}{1 + a_1} \\ F_1 &= \frac{x_1^{t-\Delta t}}{\alpha(1 + a_1)}, \end{aligned} \right.$$

where  $E_1$  and  $F_1$  are obtained from the two recursion formulas in (2.76) by noting that  $c_1 = 0$  because  $(\rho K)_{1/2}$  was set to zero in (2.69).

With these equations it can now be seen how in ECHAM  $X^{t+\Delta t}$  can be obtained for all levels of the vertical column: Using the known values of  $E_1$  and  $F_1$ , eqs. (2.76) can be used to recursively calculate all other  $E_n$  and  $F_n$  down to the lowest layer  $n = N$ . Provided the surface value  $\widehat{X}_{N+1}^t = \widehat{X}_{surf}^t \mathcal{C}_{surf} / \mathcal{C}_{atm}$  is known (compare (2.71)), eq. (2.75) can be used to recursively determine all other values of  $\widehat{X}_n^t$  throughout the vertical column. Using then the formula (2.64) in the re-arranged form

$$(2.77) \quad X^{t+\Delta t} = \frac{\widehat{X}^t}{\alpha} + \left(1 - \frac{1}{\alpha}\right) X^{t-\Delta t},$$

one finally obtains the desired new state  $X^{t+\Delta t}$  at all vertical layers.<sup>28</sup> Hence the problem remains to determine  $\widehat{X}_{surf}^t$ . How JSBACH solves this problem is described in the next section.

## 2.5.2 Calculation of surface temperature and surface heat exchange

To obtain the vertical distribution of temperature and humidity, ECHAM solves the diffusion equation (2.61) by the numerical scheme described in the previous section. For this purpose, the diffusion equation is solved for heat flux boundary conditions at the surface. But by writing the surface fluxes in the form of eq. (2.68) not the fluxes themselves must be provided by JSBACH to ECHAM, but surface dry static energy and saturated surface humidity, as will get clear from the description below. The main purpose of the present section is to explain how these two quantities – which represent  $\widehat{X}_{surf}^t$  from the last section – are computed in

<sup>26</sup>The  $\alpha$  used here is that from eq. (2.64). It is not seen why one should use it here, but its use here is consistent with the implementation in ECHAM.

<sup>27</sup>For  $n = 1, 2, \dots, N-1$  these recursions for  $E_n$  and  $F_n$  are implemented in the ECHAM routine `vdiff()` in the code section “5.3 Elimination for middle layers”. Here  $E_n$  is the variable `zebsh` (which is the same for all transported quantities as can be seen from the recursion), and `ztdif` and `zqdif` represent  $F_n$  for dry static energy and humidity, respectively. But note that confusingly `ztdif` and `zqdif` have this meaning only after they are updated via (2.76); before their meaning is that of  $x_n^{t-\Delta t}$ , i.e. dry static energy and humidity, respectively, at time  $t - \Delta t$ . And even more confusingly, in the backsubstitution via eq. (2.75) (code section “5.5 Backsubstitution”), this change in meaning is once more reverted. For  $n = N$ , i.e. to calculate  $E_N$  and  $F_N$ , the values given from ECHAM to JSBACH, these recursions are evaluated in `mo_surface_land::richtmyer_land()`.

<sup>28</sup>Actually, in ECHAM an additional filtering of the time series  $X^t$  is employed (“Asselin filter”) to suppress artificial high frequency oscillations that may arise from the leap frog discretization (for details see [55, section 2.1.4.]). Thereby every time step  $X^t$  is replaced by a filtered value (obtained from  $X^{t+1}$  and  $X^{t-1}$ ) and it is this filtered value that is then used as  $X^{t-\Delta t}$  in the next time step. This is also the reason why in (2.66)  $\rho$  and  $K$  are not evaluated at time step  $t$ , but at time step  $t - \Delta t$ : in this combination of leap frog and filtering the values at time  $t$  are not known when calculating the (unfiltered) value  $X^{t+1}$ .

JSBACH. These quantities are also the key for the calculation of the latent and sensible heat fluxes at the surface.<sup>29</sup>

In ECHAM the vertical turbulent transport of latent and sensible heat is described by means of moisture  $q$  and dry static energy  $s$ . Hence, the formalism from the previous section is applied here to  $X$  equal  $q$  or  $s$ . To calculate their vertical turbulent transport, ECHAM needs from JSBACH the associated values representing  $\widehat{X}_{N+1}^t$  (see the discussion at the end of last section). A glance at the expression for the surface flux (2.68) reveals that for the dry static energy eq. (2.71) reduces to<sup>30</sup>

$$(2.78) \quad \widehat{X}_{N+1}^t = \hat{s}_{surf}^t \quad \text{with} \quad C_{atm} = C_{surf} = 1.$$

For humidity the situation is different. As seen from eq. (2.56), the total evaporation from the different surfaces can be written as a weighted difference between specific humidity  $q_a$  (called  $q_N$  here) in the lowest atmospheric layer and saturated specific humidity  $q_{sat}$  at the surface. Hence comparison with (2.68) reveals that here one has to write eq. (2.71) as

$$(2.79) \quad \widehat{X}_{N+1}^t = \frac{C_{sat}}{C_{air}} \hat{q}_{sat}^t \quad \text{i.e.} \quad C_{atm} = C_{air}, \quad C_{surf} = C_{sat}.$$

By eq. (2.75) humidity  $q$  and dry static energy  $s$  in the first layer can thus be obtained from<sup>31</sup>

$$(2.80) \quad \begin{aligned} \hat{s}_N^t &= E_N \hat{s}_{surf}^t + \widetilde{F}_N^s & \text{with} \quad \widetilde{E}_N &:= E_N \frac{C_{sat}}{C_{air}}, \quad \widetilde{F}_N^s := \alpha F_N^s, \quad \widetilde{F}_N^q := \alpha F_N^q, \\ \hat{q}_N^t &= \widetilde{E}_N \hat{q}_{sat}^t + \widetilde{F}_N^q \end{aligned}$$

where one has to distinguish between  $F_N$  values for dry static energy ( $F_N^s$ ) and for humidity ( $F_N^q$ ), while the values  $E_N$  are the same for both quantities; the latter is seen from the expressions (2.76) by noting that specificity concerning  $s$  or  $q$  enters via  $x_n^{t-\Delta t}$  only into the  $F_n$  recursion.<sup>32</sup> For comparison with the implementation of JSBACH it is useful to rewrite  $E_N$  and  $\widetilde{E}_N$  more explicitly by invoking eqs. (2.76), (2.73), (2.70) and (2.69) so that<sup>33</sup>

$$(2.81) \quad \begin{aligned} E_N &= \frac{a_N^s}{1 + c_N(1 - E_{N-1}) + a_N^s} & \text{with} \quad a_N^s &:= 2\alpha g \Delta t \frac{(\rho C_h |\vec{v}|)^{t-\Delta t}}{\Delta p_N}, \\ \widetilde{E}_N &= \frac{C_{sat} a_N^s}{1 + c_N(1 - E_{N-1}) + C_{air} a_N^s} & c_N &= 2\alpha g \Delta t \frac{(\rho K)^{t-\Delta t}}{\Delta z_{N-1/2} \Delta p_N}. \end{aligned}$$

Note that  $c_N$  has the same value for the  $q$  and  $s$  cases.

For the following, eqs. (2.80), (2.81) are crucial because in the calculations below they will allow to eliminate the lowest layer values  $\hat{s}_N^t$  and  $\hat{q}_N^t$  so that one is left with equations determining the surface values  $\hat{s}_{surf}^t$  and  $\hat{q}_{sat}^t$  needed to obtain from (2.75) the values of  $s$  and  $q$  for all atmospheric layers.

<sup>29</sup>The following description largely draws from personal notes of Erich Roeckner [117].

<sup>30</sup>Note that in order to be consistent with the formalism described in the previous section, all calculations must be performed for mixed-time quantities defined via (2.64), indicated as before by a ‘‘hat’’ over the symbol.

<sup>31</sup>These equations for the lowest level values  $\hat{s}_N^t$  and  $\hat{q}_N^t$  are evaluated in the ECHAM routine `mo_surface_land::update_land()`, and also for diagnosing surface fluxes in the JSBACH routine `mo_soil::update_soil()`.

<sup>32</sup>In `mo_soil::update_soil()` the Richtmyer-Morton coefficients  $E_N$ ,  $\widetilde{F}_N^s$ ,  $\widetilde{E}_N$ ,  $\widetilde{F}_N^q$  are given by the variables `t.Acoef`, `t.Bcoef`, `q.Acoef`, `q.Bcoef`, respectively.

<sup>33</sup>The Richtmyer-Morton coefficients are calculated in subroutine `mo_surface_land::richtmyer_land()`, where  $E_N$ ,  $\widetilde{F}_N^s$ ,  $\widetilde{E}_N$ ,  $\widetilde{F}_N^q$  are the variables `zftnl`, `zftnl`, `zeqnl`, `zfqnl`, respectively, and  $C_{sat}$ ,  $C_{air}$  are `zsat`, `zair`, while  $c_N$  is `zfac`, and  $a_N^s$  is `zcfhl*zqdp`.

Considering first dry static energy, expansion of the known dependence of saturated specific humidity on temperature around its value at the previous time step  $t - \Delta t$  reveals to first order

$$(2.82) \quad \hat{q}_{sat}^t = q_{sat}^{t-\Delta t} + \left. \frac{\partial q_{sat}}{\partial T} \right|_{T_{surf}^{t-\Delta t}} (\hat{T}_{surf}^t - T_{surf}^{t-\Delta t}) = q_{sat}^{t-\Delta t} + Q' \frac{\hat{s}_{surf}^t - s_{surf}^{t-\Delta t}}{c_p},$$

where  $Q'$  abbreviates the derivative of  $q_{sat}$  and the definition (2.57) was invoked to convert temperature  $T$  to dry static energy  $s$ . This equation shows that  $\hat{q}_{sat}^t$  is known once  $\hat{s}_{surf}^t$  has been obtained. Hence the remaining problem to be tackled in this section reduces to determining  $\hat{s}_{surf}^t$ .

In translating temperature into dry static energy in eq. (2.82) one needs to know the value of the heat capacity of the air  $c_p$ , which because of the changing moisture content is time dependent. Here one needs the value of  $c_p$  only at the surface. To find the dependence of  $c_p$  on surface humidity, it is useful to introduce an *effective* surface humidity  $q_{surf}$  by imagining total evaporation (2.56) to occur from an equivalent potential evaporation (2.44). Equating those two expressions, one sees that  $q_N - q_{surf} = \mathcal{C}_{air} q_N - \mathcal{C}_{sat} q_{sat}$ , or

$$(2.83) \quad q_{surf} = \mathcal{C}_{sat} q_{sat} + (1 - \mathcal{C}_{air}) q_N.$$

Noting further that specific humidity  $q$  is the concentration of water vapor in moist air, one can write  $c_p$  as a weighted average of the heat capacities of dry air  $c_p^{dry}$  and water vapor  $c_p^{vap}$  to find at the surface<sup>34</sup>

$$(2.84) \quad c_p = (1 - q_{surf}) c_p^{dry} + q_{surf} c_p^{vap} = c_p^{dry} + (c_p^{vap} - c_p^{dry}) (\mathcal{C}_{sat} q_{sat} + (1 - \mathcal{C}_{air}) q_N).$$

This expression is used for the conversion between dry static energy and temperature at the surface by evaluating it for time  $t - \Delta t$ .

The problem to obtain  $\hat{s}_{surf}^t$  is solved by invoking the equation of the surface energy balance (2.1). Switching via (2.57) from temperature to dry static energy and approximating the time derivative according to (2.65), the discretized energy balance reads

$$(2.85) \quad \frac{C}{c_p \delta t} (\hat{s}_{surf}^t - s_{surf}^{t-\Delta t}) = \hat{R}_{net}^t + \hat{H}_{sensible}^t + \hat{H}_{latent}^t + G^{t-\Delta t},$$

where the abbreviation

$$(2.86) \quad \delta t := 2\alpha \Delta t$$

was used. In this way, the only unknown on the left hand side is  $\hat{s}_{surf}^t$ . Hence to make it an equation determining  $\hat{s}_{surf}^t$ , all right hand side terms of the energy balance must as well be transformed to expressions that contain  $\hat{s}_{surf}^t$  as the only unknown. In the following this is done term by term. Note that more properly one had to use in (2.85) the mixed time ground heat flux  $\hat{G}^t$  instead of the known ground heat flux  $G^{t-\Delta t}$  from the last time step; this is an additional approximation.

Net radiation: In the eq. for net radiation (2.2) the Boltzmann term for outgoing longwave radiation can be expressed by  $\hat{s}_{surf}^t$  as follows. Expansion gives to first order in  $\hat{T}_{surf}^t - T_{surf}^{t-\Delta t}$

$$(2.87) \quad \begin{aligned} (\hat{T}_{surf}^t)^4 &= (T_{surf}^{t-\Delta t})^4 + 4 (T_{surf}^{t-\Delta t})^3 (\hat{T}_{surf}^t - T_{surf}^{t-\Delta t}) \\ &= (T_{surf}^{t-\Delta t})^4 + 4 \frac{(T_{surf}^{t-\Delta t})^3}{c_p} (\hat{s}_{surf}^t - s_{surf}^{t-\Delta t}). \end{aligned}$$

<sup>34</sup> $c_p$  is the variable `zcpq` in `mo_soil::update_soil()`.

Hence net radiation can in this linear approximation be written as<sup>35</sup>

$$(2.88) \quad \widehat{R}_{net}^t = R_{net}^{t-\Delta t} - 4\epsilon\sigma \frac{(T_{surf}^{t-\Delta t})^3}{c_p} (\hat{s}_{surf}^t - s_{surf}^{t-\Delta t}),$$

where in view of (2.2) one finds

$$(2.89) \quad R_{net}^{t-\Delta t} = (1 - \alpha_{vis}^{t-\Delta t})R_{vis}^{t-\Delta t} + (1 - \alpha_{nir}^{t-\Delta t})R_{nir}^{t-\Delta t} + \epsilon R_{ld}^{t-\Delta t} - \epsilon\sigma \left(T_{surf}^{t-\Delta t}\right)^4.$$

Sensible heat flux: Using the definition of dry static energy (2.57) the equation for the sensible heat flux (2.58) can be written as

$$(2.90) \quad \begin{aligned} \widehat{H}_{sensible}^t &= C_H(\hat{s}_N^t - \hat{s}_{surf}^t) \\ &= C_H((E_N - 1)\hat{s}_{surf}^t + \widetilde{F}_N^s) \end{aligned}$$

where (2.80) was used to eliminate the atmospheric value of dry static energy and the coefficient<sup>36</sup>

$$(2.91) \quad C_H := \rho C_h |v_h|$$

is identified by comparison with (2.58) and (2.45).

Latent heat flux: To express the latent heat flux as a function of the unknown  $\hat{s}_{surf}^t$  one can insert first (2.80) into (2.60) and then express via (2.82) the unknown  $\hat{q}_{surf}^t$  by  $\hat{s}_{surf}^t$  which gives

$$(2.92) \quad \begin{aligned} \widehat{H}_{latent}^t &= C_H (\mathcal{D}_{air} \hat{q}_N^t - \mathcal{D}_{sat} \hat{q}_{sat}^t) \\ &= C_H \left( (\mathcal{D}_{air} \mathcal{C}_{sat} \widetilde{E}_N - \mathcal{D}_{sat}) \hat{q}_{sat}^t + \mathcal{D}_{air} \widetilde{F}_N^q \right) \\ &= C_H \left( (\mathcal{D}_{air} \mathcal{C}_{sat} \widetilde{E}_N - \mathcal{D}_{sat}) \frac{Q'}{c_p} \hat{s}_{surf}^t + \right. \\ &\quad \left. + (\mathcal{D}_{air} \mathcal{C}_{sat} \widetilde{E}_N - \mathcal{D}_{sat}) \left( q_{sat}^{t-\Delta t} - \frac{Q'}{c_p} s_{surf}^{t-\Delta t} \right) + \mathcal{D}_{air} \widetilde{F}_N^q \right). \end{aligned}$$

Having written the surface fluxes as a function of the unknown  $\hat{s}_{surf}^t$  one can now insert these fluxes into the energy balance (2.85) to obtain a linear equation for this unknown. This gives

$$(2.93) \quad \begin{aligned} \frac{C}{c_p \delta t} (\hat{s}_{surf}^t - s_{surf}^{t-\Delta t}) &= G^{t-\Delta t} + \\ &+ R_{net}^{t-\Delta t} - 4\epsilon\sigma \frac{(T_{surf}^{t-\Delta t})^3}{c_p} (\hat{s}_{surf}^t - s_{surf}^{t-\Delta t}) + C_H ((E_N - 1)\hat{s}_{surf}^t + \widetilde{F}_N^s) + \\ &+ C_H \left( (\mathcal{D}_{air} \mathcal{C}_{sat} \widetilde{E}_N - \mathcal{D}_{sat}) \frac{Q'}{c_p} \hat{s}_{surf}^t + \right. \\ &\quad \left. + (\mathcal{D}_{air} \mathcal{C}_{sat} \widetilde{E}_N - \mathcal{D}_{sat}) \left( q_{sat}^{t-\Delta t} - \frac{Q'}{c_p} s_{surf}^{t-\Delta t} \right) + \mathcal{D}_{air} \widetilde{F}_N^q \right). \end{aligned}$$

<sup>35</sup>This discretization ignores changes between the values of albedo and incoming radiation at time  $t - \Delta$  and their mixed-time values.

<sup>36</sup>Up to a constant factor,  $C_H$  is the variable `cdrag` in the JSBACH interface (compare table 1.1); more details in footnote 14.

Introducing the abbreviations<sup>37</sup>

$$(2.94) \quad \begin{aligned} A_1 &:= \frac{C}{c_p} + \frac{\delta t}{c_p} \left[ 4\epsilon\sigma (T_{surf}^{t-\Delta t})^3 - C_H(\mathcal{D}_{air}\mathcal{C}_{sat}\tilde{E}_N - \mathcal{D}_{sat})Q' \right] \\ A_2 &:= -\delta t C_H(E_N - 1) \\ A_3 &:= \delta t \left[ G^{t-\Delta t} + R_{net}^{t-\Delta t} + C_H\tilde{F}_N^s + C_H(\mathcal{D}_{air}\mathcal{C}_{sat}\tilde{E}_N - \mathcal{D}_{sat})q_{sat}^{t-\Delta t} + C_H\mathcal{D}_{air}\tilde{F}_N^q \right] \end{aligned}$$

the solution to this equation is found as<sup>38</sup>

$$(2.95) \quad \hat{s}_{surf}^t = \frac{A_1 s_{surf}^{t-\Delta t} + A_3}{A_1 + A_2}.$$

Since the right hand side of this equation contains only known values, this equation completes the set of equations needed to solve for surface temperature and also the vertical distribution of temperature and humidity in the atmosphere: Entering  $\hat{s}_{surf}^t$  from (2.95) into (2.77) and noting that by (2.57) at the surface  $T = s/c_p$ , one finds the new surface temperature from

$$(2.96) \quad T_{surf}^{t+\Delta t} = \frac{1}{c_p} \left( \frac{1}{\alpha} \hat{s}_{surf}^t + \left(1 - \frac{1}{\alpha}\right) s_{surf}^{t-\Delta t} \right).$$

Note that this is not the final result for surface temperature, but (in the language of ECHAM) the “unfiltered” surface temperature. Reason for this naming is that the leap-frog type numerical integration scheme steps from time step  $t - \Delta t$  to future time  $t + \Delta t$  and is not calculating any value for current time  $t$ . This is “remedied” by applying an “Asselin filter” [2]. But before applying this filter,  $T_{surf}^{t+\Delta t}$  is first corrected for snowmelt (see section 2.3.1.2). Then applying the Asselin filter, the final “filtered” surface temperature is found for time  $t$  from

$$(2.97) \quad \bar{T}_{surf}^t = T_{surf}^t + \epsilon(\bar{T}_{surf}^{t-\Delta t} - 2T_{surf}^t + T_{surf}^{t+\Delta t}),$$

where filtered values are indicated by a bar over the symbol and the filtering parameter  $\epsilon$  has the value 0.1. This is the final surface temperature.

For completeness it shall be mentioned that

$$(2.98) \quad T_{rad} := \hat{s}_{surf}^t / c_p$$

is interpreted as *radiative* surface temperature and calculated only for usage in ECHAM (compare table 1.2).

To provide ECHAM with the necessary information to proceed with the calculation of temperature and humidity for all atmospheric layers, it remains to calculate saturated specific humidity  $\hat{q}_{sat}^t$  at the surface. This value is obtained from using the just determined value of  $\hat{s}_{surf}^t$  in eq. (2.82). But over snowed surfaces the value of  $\hat{s}_{surf}^t$  is first corrected to be consistent with the condition that for such surfaces temperature cannot be larger than 0°C there. This correction is done by reducing  $\hat{s}_{surf}^t$  to  $273.15\text{K} * c_p$  once  $\hat{s}_{surf}^t$  is larger than this value.

This finalizes the surface calculations needed to provide ECHAM with the necessary information to obtain dry static energy and specific humidity throughout the vertical column: Once ECHAM has received the values of  $\hat{s}_{surf}^t$  and  $\hat{q}_{sat}^t$  from JSBACH (compare 1.2), it calculates the values of  $\hat{s}_N^t$  and  $\hat{q}_N^t$  in the lowest atmospheric layers from (2.80) and then all other values of the atmospheric column by recursively using (2.75).

<sup>37</sup>In subroutine `update_surfacetemp()`  $A_1$ ,  $A_2$ , and  $A_3$  are the variables `zcolin`, `zcohfl`, and `zcoind`, respectively.

<sup>38</sup>In subroutine `update_surfacetemp()` this equation reads `psnew = (zcolin * psold + zcoind)/(zcolin + zcohfl)`.

### 2.5.3 Emulation of coupling in JSBACH standalone

As seen from table 1.1, JSBACH needs several input variables that in the JSBACH standalone setup (see appendix A) are not part of the forcing fields (listed in table C.30). This concerns in particular all variables related to the vertical turbulent transport, namely the exchange coefficient  $C_h$  (compare eq. (2.91)) and the Richtmyer-Morton coefficients (compare eq. (2.80)). This section shortly describes how in this standalone configuration the coupling to an atmosphere is emulated by calculating these quantities from the forcing fields.

The calculation of  $C_h$  closely follows the calculations in ECHAM (see eq. (2.173) in [55]), although with some small modifications made necessary by the different information available. One such modification is the calculation of the Richardson number that in JSBACH standalone is calculated as the “dry” bulk Richardson number (eq. (4) in [13]). Moreover, since in the standalone setup the feedbacks with the atmosphere are missing, the calculated Richardson number may strongly vary from one time step to the next, resulting in wild jumps in surface temperature. To prevent this, an additional damping is applied to the Richardson number.<sup>39</sup>

Although there is no turbulent atmosphere in the JSBACH standalone setup, JSBACH nevertheless needs the Richtmyer-Morton coefficients  $E_N$ ,  $\tilde{F}_N^s$ ,  $\tilde{E}_N$ , and  $\tilde{F}_N^q$  to calculate from eqs. (2.96), (2.95), (2.94) the surface temperature. To obtain the Richtmyer-Morton coefficients the implicit coupling used in the coupled setup is changed into an explicit coupling in the standalone setup (compare footnote 24): Technically this means to use in (2.68) instead of the unknown mixed time quantity  $\hat{X}_N^t$  the quantity  $X_N^{t-\Delta t}$  known from the previous time step. Following the consequences through the equations, the heat fluxes then read

$$(2.99) \quad \begin{aligned} \hat{H}_{sensible}^t &= C_H(s_N^{t-\Delta t} - \hat{s}_{surf}^t) \\ \hat{H}_{latent}^t &= C_H(\mathcal{D}_{air} q_N^{t-\Delta t} - \mathcal{D}_{sat} \hat{q}_{sat}^t). \end{aligned}$$

These equations are obtained from (2.90) and (2.92) when redefining the index  $N$  Richtmyer-Morton coefficients by

$$(2.100) \quad E_N := 0, \quad \tilde{F}_N^s := s_N^{t-\Delta t}, \quad \text{and} \quad \tilde{E}_N := 0, \quad \tilde{F}_N^q := q_N^{t-\Delta t}.$$

To calculate these values one takes  $q_N^{t-\Delta t}$  directly from the forcing, while  $s_N^{t-\Delta t}$  is computed from (2.57) as

$$(2.101) \quad s_N^{t-\Delta t} = c_p T + g \Delta z,$$

with  $c_p$  calculated from (2.84) using  $q_N^{t-\Delta t}$ , while  $\Delta z$  is calculated from the barometric formula assuming this to be the height difference between the mid-level pressure height  $p_h$  and the surface in a 47-layer ECHAM model, where  $p_h/p_{surf} = 0.99615$ . In summary, with the tricky definitions (2.100), the whole apparatus developed in the previous section for the *implicit* coupling can be employed without any further changes to handle *explicit* coupling to derive in particular surface temperature (2.97). In this way JSBACH can be run without any modifications in standalone mode simply by calling the JSBACH interface (see table 1.1) with the emulated Richtmyer-Morton coefficients (2.100).

## 2.6 Lakes

The model for the lake water balance is part of ECHAM not of JSBACH. For details on the lake thermodynamics see the ECHAM documentation [55]. The water imbalance over lakes

<sup>39</sup>These calculations are performed in `mo_soil::update_cdrag()`.

is computed from precipitation, evaporation of open lake water, and sublimation from sea ice and snow, it enters as runoff the HD model (section 2.8) and is routed with the land fluxes towards the oceans.

## 2.7 Glaciers

In the current setup of JSBACH glaciers either fill the whole grid box, or there is no glacier in the grid box. In view of physical processes, glaciers differ in JSBACH from other land surfaces by the absence of vegetation (i.e. no transpiration) and the absence of infiltration. Otherwise glaciers are handled as other land surfaces. Accounting for these particularities, surface fluxes are computed as described in section 2.4, and surface runoff is obtained from the surface water balance as described in section 2.3.

## 2.8 River runoff: The Hydrological Discharge Model

The lateral water flows from the continents via rivers into the oceans are described by the Hydrological Discharge (HD) model [56, 59]. The purpose of the model is to compute the fresh water fluxes entering the oceans. Hence it collects surface runoff (2.37) and subsurface drainage (2.43), and routes these water fluxes across the land topography to the river mouths of the global river catchments. The model needs to be run only in combination with ocean simulations, or to diagnose the land hydrology, but otherwise has no effect on land processes in JSBACH. Internally, the model uses a 0.5 grid. By interpolating the input fluxes from the resolution used in a particular JSBACH simulation, the HD model is largely independent from the particular simulation setup used.

### 2.8.1 Linear reservoirs

Conceptually, the HD model is based on the dynamics of linear reservoirs. At every instant  $t$  the outflow  $Q$  from a reservoir is assumed to be proportional to its content  $S$ , i.e.

$$(2.102) \quad Q(t) = \frac{S(t)}{k},$$

where  $k$  is the retention time of the reservoir, also called 'retention coefficient'. With input flux  $I(t)$  the mass balance of a reservoir is

$$(2.103) \quad \frac{dS(t)}{dt} = I(t) - Q(t),$$

and with (2.102) an equation for the time development of  $S$  would follow. But since throughout the HD model the geographically varying  $k$  values are constant in time, one can instead use a completely fluxed based description of the dynamics:

$$(2.104) \quad k \frac{dQ(t)}{dt} = I(t) - Q(t).$$

This can be extended to a cascade of  $n$  linear reservoirs in series. In such a configuration the outflow of reservoir  $i$  enters reservoir  $i + 1$  as inflow, i.e.

$$(2.105) \quad k \frac{dQ_i}{dt} = I_i - Q_i, \quad i = 1, \dots, n$$

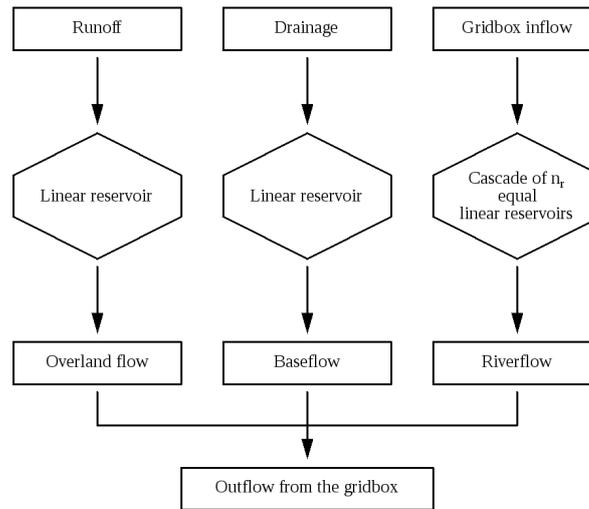
$$(2.106) \quad I_i = Q_{i-1}, \quad i = 2, \dots, n.$$

The input  $I_1(t)$  to this cascade is thereby delayed by approximately the time  $nk$  until it leaves the delay line as output  $Q_n(t)$ .

### 2.8.2 Model structure

Figure 2.2 shows the structure of the HD model. Lateral water flow is separated into the three flow processes of overland flow, base flow and river flow. Overland flow and base flow are represented by a single linear reservoir, while river flow is represented by a cascade of five equal linear reservoirs. In each grid box, overland flow is fed by the runoff from land (2.37), glaciers (section 2.7) and lakes (section 2.6), while base flow is fed only by the drainage from land (2.43) in the particular grid box. Their fluxes are discharged into the river flow in that grid box. But river flow is also fed by the river outflow from other grid boxes. The sum of the three flow processes makes the outflow from a grid box.

Figure 2.2: Structure of Hydrological Discharge Model.



For the linear reservoir ( $n = 1$ ) of overland flow, the retention coefficient is a function of average slope  $\phi$  and length  $\Delta x$  in a grid box,

$$(2.107) \quad k_{over} = 17.87 \cdot 10^{-2} \frac{\Delta x}{\phi^{0.1}},$$

where grid box length is defined as the distance between the centers of two adjacent grid boxes in the direction of the flow.

For river flow, the retention coefficient of the  $n = 5$  linear reservoirs is a function of  $\Delta x$  and the topography gradient  $\Delta h$  between two adjacent grid boxes in the direction of the flow and grid box length:

$$(2.108) \quad k_{river} = 9.92 \cdot 10^{-4} \frac{\Delta x}{\left(\frac{\Delta h}{\Delta x}\right)^{0.1}}.$$

The particular numbers in the latter two equations were obtained by adjustment to a large Swedish catchment (see [56]).

For base flow, the topography is less relevant. Accordingly, the retention time of the associated linear reservoir ( $n = 1$ ) is assumed to depend only on the size of a grid box as represented by the grid box length  $\Delta x$ , namely

$$(2.109) \quad k_{base} = \tau_{base} \frac{\Delta x}{d_0},$$

where  $d_0$  is the typical diameter of the underlying  $0.5^\circ$  grid boxes (50 km) and  $\tau_{base}$  is a typical retention time of baseflow in a grid box of that size (set to 300 days); the HD model is not very sensitive to small changes in  $t_B$  so that its exact value is not of much interest [56].

Basic to the HD model is a static routing map ('river direction map') that defines for each grid box to what neighboring grid box river outflow is transferred. This map also marks the grid boxes where the rivers discharge into the ocean. In coupled simulations, this discharge is the fresh water influx to the ocean. Some of the world wide river catchments are without ocean outflow. For such catchments there are grid boxes in the routing map without any outflow into a neighboring grid box. The river outflow computed for such grid boxes is equally distributed to all river discharges into the ocean. – For more details on the implementation of the HD model see section 2.10.4.

## 2.9 Diagnostics

Table 2.1 lists the most important output fields for diagnosis of the physical land processes. These are located in different output streams (see section D).

Table 2.1: Standard diagnostic output from the land physics. The output fields are located in different output streams as indicated in the third column (compare section D); note that the echam stream is not part of this documentation.

name in output	symbol	stream	meaning
<i>State variables:</i>			
snow	$h_{sn}$	echam	snow on ground (eq. (2.12))
snow_depth_canopy	$h_{snc}$	land	snow on canopy (eq.(2.5))
skin_reservoir	$h_{wsk}$	jsbach	water in skin reservoir (eq.(2.15))
soil_temperature	$T$	land	temperatures in the five thermal soil layers (eq. (2.3))
layer_moisture	$\int_{z_{i+1}}^{z_i} dz \theta(z)$	land	water content in the different soil layers (layer $i$ reaching from $z_{i+1}$ to $z_i$ )
<i>Surface characteristics:</i>			
surface_temperature	$T_{surf}$	land	surface temperature as computed from (2.1) including the snow melt correction (2.14)
ground_heat_flux	$G$	land	ground heat flux (see eq. (2.3))
sensible_heat_flux	$H_{sensible}$	land	sensible heat flux from eq. (2.58)
latent_heat_flux	$H_{latent}$	land	latent heat flux from eq. (2.60)
evapotranspiration	$E$	land	evaporation fluxes from all surface types including transpiration (eq. (2.55))
runoff	$R_{surf} + R_d$	land	surface runoff plus drainage from eqs. (2.37) and (2.43)
drainage	$R_d$	land	soil drainage from (2.43)
disch	–	hdjsbach	river discharge from the HD model

## 2.10 Implementation details

### 2.10.1 General setup

The entry routine for all land physics is `update_soil()` located in module `mo_soil`. This module also contains the definition of all data structures for the land physics and also all initialization routines. While the exchange with the atmosphere happens for each grid box as

a whole, most other calculations are performed for each tile separately, assuming that each tile has its share of the bare land part of the grid box (compare section 1.3.2). Nevertheless, concerning the hydrology and temperature all variables are averaged across the tiles to have a uniform physical state in exchange with the atmosphere. This means in particular, that the different landcover types including the implicit bare land part are assumed to be well mixed in a grid box.

## 2.10.2 Implementation details: Thermal land processes

### 2.10.2.1 Thermal soil parameters

The volumetric heat capacities  $C_s$  in (2.1) and (2.3) and the heat conductivity  $\lambda_s = C_s \kappa_s$  needed in (2.3), where  $\kappa_s$  is the thermal diffusivity, are computed for snow-free conditions from a soil map prepared under the auspices of FAO. This soil map is part of the JSBACH initial data file (see appendix C.4.2). It describes soils by 5 classes (compare appendix C.4.7), and each soil class is associated with a particular value for  $C_s$  and  $\lambda_s$ . These values are defined as parameters in the routine `init_soil` (see table 2.2). Similar values exist for snow and ice, defined as parameters in `mo_jsbach_constants`.

Table 2.2: Values for heat capacity  $C$  and thermal diffusivity  $\kappa$  used in JSBACH. The top 5 entries refer to the 5 classes of the FAO soil map; in the absence of snow these are the values  $C_s$  and  $\kappa_s$  in eq. (2.3). The values for ice are used in glacier grid boxes, while the values for snow are used to modify the soil values in the presence of snow. Thermal conductivity is computed from  $\lambda = C\kappa$ .

material	$C$ [J/m <sup>3</sup> K]	$\kappa$ [m <sup>2</sup> /s]
<i>FAO soil classes:</i>		
sand	1.930 10 <sup>6</sup>	8.7 10 <sup>-7</sup>
sandy loam	2.100 10 <sup>6</sup>	8.0 10 <sup>-7</sup>
loam	2.250 10 <sup>6</sup>	7.4 10 <sup>-7</sup>
loam/clay mixture	2.360 10 <sup>6</sup>	7.1 10 <sup>-7</sup>
clay	2.480 10 <sup>6</sup>	6.7 10 <sup>-7</sup>
<i>Water phases:</i>		
snow	6.345 10 <sup>6</sup>	4.9 10 <sup>-7</sup>
ice	2.090 10 <sup>6</sup>	12.0 10 <sup>-7</sup>

### 2.10.2.2 Numerics for solving the thermal diffusion equation

The thermal diffusion equation (2.3) for the soil heat transport is solved in `mo_soil_temperature::update_soil_temperature()`. In the standard setup of JSBACH, it is discretized by dividing the upper 10 m of the soil into 5 unevenly spaced layers. Global values for layer thickness as defined in subroutine `soil_init_io` are from top to bottom 0.065, 0.254, 0.913, 2.902, and 5.700 [m]. Alternatively, geographically different values can be read in from the JSBACH initial file (see appendix C.4) via the field `soil_layer_depth`, but suitable data are currently not available. Somewhat counter-intuitively, the namelist parameter `nsoil` (see C.2.12), by which the number of soil layers can be chosen, has in the standard setup the value “1” although this indicates that the standard setup with 5 soil layers is used.

For snow-free land, spatially varying values for heat capacity  $C_s$  and thermal diffusivity  $\kappa_s$  are prescribed according to the above mentioned FAO soil map, while for grid boxes covered with ice sheets/glaciers values for ice are used (defined in `mo_jsbach_constants`). For snow

covered land  $C_s$  and  $\kappa_s$  are modified in `update_soil_temperature` for the presence of snow. These modifications are based on the geometric snow depth  $d_{sn} = h_{sn} \rho_{H_2O} / \rho_{sn}$ , calculated from snow at the surface  $h_{sn}$  (measured in meter water equivalent) and water and snow densities  $\rho_{H_2O}$  and  $\rho_{sn}$  defined in `mo_physical_constants` and `mo_jsbach_constants`. To calculate the modified values, a simple picture is applied: It is assumed that snow gradually replaces soil layers from the top, so that depending on soil depth,  $C_s$  and  $\kappa_s$  are either fully replaced by their snow values, or, if snow reaches only partially into a soil layer, a depth weighted average between soil and snow values is used for  $C_s$  and  $\kappa_s$  in that layer.

To solve the heat conduction equation (2.3) the implicit Richtmyer and Morton time integration scheme is used [114], which is technically similar to the integration for turbulent vertical diffusion described in section 2.5. Thereby the discretized heat conduction equation leads to a tri-diagonal linear system for the temperatures in all soil layers that can be solved by the Thomas algorithm (see e.g. [107]). Similarly to (2.75), this algorithm leads to an iterative equation by which the temperatures of the different soil layers can be updated iteratively via

$$(2.110) \quad T_{k+1}^{n+1} = A_k^n + B_k^n T_k^{n+1}.$$

Here  $n$  is the time step index and  $k$  the layer index. The coefficients  $A_k^n$  and  $B_k^n$  include the vertical increments  $\Delta z$ , the time step  $\Delta t$ , the volumetric heat capacity  $C_s$ , the thermal conductivity  $\lambda_s$  and the temperature at time step  $n$  in  $A_k^n$ . The Richtmyer-Morton coefficients  $A$  and  $B$  also obey a recursive formula by which their values can be obtained iteratively starting from the bottom of the soil, where the assumed zero heat flux boundary condition determines the lowest level coefficients as starting point. Having determined the coefficients  $A$  and  $B$ , the temperature profile is then found by iterating (2.110) from top to bottom by using surface temperature as determined from (2.1) as upper boundary condition, i.e.

$$(2.111) \quad T_1^{n+1} = T_s^{n+1}.$$

### 2.10.3 Implementation details: Soil hydrology

#### 2.10.3.1 Hydrological input fields and model parameters

In the standard setup, the soil parameter values for the different soil textures are derived from an improved FAO soil type dataset (see [61]). Thus, the different soil parameters including  $\theta_{cap}$ ,  $\theta_{pwp}$  and  $z_{root}$  are geographically varying datasets (see table 2.3). Other model parameters are summarized in table 2.4.

#### 2.10.3.2 Structure of calculations

To solve the Richards equation (2.18) together with the specifications for water diffusivity (2.25) and hydraulic conductivity (2.23), it is vertically discretized by introducing several soil layers of the same depths as used for the solution of the heat flux equation (2.3) (see section 2.10.2.2). The coupled process of percolation and diffusion is numerically represented as two separate processes using different discretizations: while for the diffusion part a Richtmyer and Morton scheme (see [114]) is used employing centered variables (also diffusivity is averaged between two layers), a midpoint time-discretization is used for the percolation part. Moreover, to overcome numerical instabilities occurring for large relative soil moisture values  $\Theta_{cap}$  and large time steps a maximum percolation flux is set.

In each time step, the calculations start by computing the changes in the water content of the different layers induced by upward fluxes of transpiration and bare soil evaporation. Next,

Table 2.3: Static global input fields used for the soil hydrology (compare C.4)

Parameter	name in code	units	field name
$\theta_{por}$	vpor	$\text{m}^3/\text{m}^3$	soil_porosity
$\theta_{cap}$	vfc	$\text{m}^3/\text{m}^3$	soil_field_cap
$\theta_{pwp}$	vpwp	$\text{m}^3/\text{m}^3$	wilting_point
$PSI$	spsi	$\text{J}/\text{m}^3 \text{ K}$	pore_size_index
$b_{Clapp}$	bclapp	–	bclapp
$K_{sat}$	fksat	$\text{m}/\text{s}$	hyd_cond_sat
$\Psi_{sat}$	fpot	$\text{m}$	moisture_pot
$\sigma_z$	porostd	$\text{m}$	orography_std_dev
$z_{root}$	dzr	$\text{m}$	root_depth
$W_{max}$	ws_max	$\text{m}$	maxmoist

Table 2.4: Parameters used for the land physics

Parameter	name in code	units	defined where	typical value
$a_{min}$	zdrmin	$\text{kg}/\text{m}^2 \text{ s}$	code	$2.810^{-7}$
$a_{max}$	zdrmax	$\text{kg}/\text{m}^2 \text{ s}$	code	$2.810^{-5}$
$a_{exp}$	zdrexp	–	code	1.5
$\sigma_0$	zorvari	$\text{m}$	code	100
$\sigma_{max}$	zorvars	$\text{m}$	code	1000

infiltration is computed. Technically, half of the infiltration is applied before computation of the vertical soil water fluxes from percolation and diffusion, and the second half afterwards. This accounts better for the simultaneously occurring fluxes from infiltration and vertical water transport because it prevents to some extent that saturation limits are artificially hit. Percolation is calculated before accounting for the diffusive water transport. Except for the lowest layer above the bedrock in which no percolation fluxes appear, lateral drainage is extracted from the soil moisture before diffusion occurs. After diffusion has changed the soil moisture contents, the pre-computed percolation fluxes are used to change the soil moisture contents. Then the mentioned second half of infiltration is added to the soil water. Finally the soil moisture values obtained in this way for each tile are averaged across all tiles to obtain the final values for soil moisture in the considered time step.<sup>40</sup>

Optionally, bare soil evaporation can be modified by setting `LBSOIL=.TRUE.` in the soil namelist (default is `LBSOIL=.FALSE.`, compare appendix C.2.12). In that case water in the top layer is tracked separately for the bare soil and vegetated parts of the grid box with some diffusion between the two reservoirs. Baresoil evaporation is then computed only from that particular top soil water reservoir using a different formula than (2.47). For results from this modified setup see [61].

Sometimes its useful to check whether water is conserved in the soil water dynamics: When setting the namelist switch `ldiag_soil=.TRUE.`, the simulation output contains a field `water_balance` that should be zero if no water gets lost.

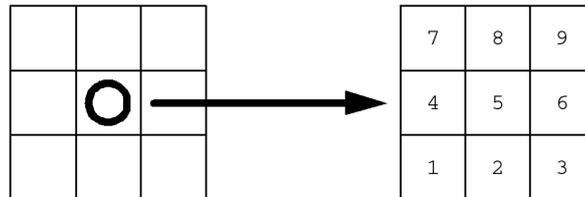
#### 2.10.4 Implementation details: River routing in the HD model

The HD model is found in `mo_hydrology` as subroutine `hydrology_model` that is called from the JSBACH main routine. As a general strategy the HD model computes the discharge at

<sup>40</sup>Note that the process of averaging across tiles conserves water mass.

0.5 resolution. All model input fields (such as runoff and drainage) from the various GCM resolutions are therefore interpolated to the same 0.5 grid. The origin of the model grid is located at the North Pole and the date line, while the center of the corresponding grid box has the coordinates -179.75 W and 89.75 N. Although the HD model uses a daily time step, the maximum time step is six hours to pay regard to the minimum travel time across a 0.5 grid box which is limited by the time step chosen.

Figure 2.3: Encoding of river directions in the HD parameter file (see table C.24).



For each grid box of the HD model, eight possible outflow directions are assumed: the four main directions North (N), East (E), South (S) and West (W), and the four diagonal directions NE, SE, SW and NW. For each grid box only a single outflow direction is permitted. The outflow along this direction adds to the river flux of the neighbouring grid box located in that direction. This directional information has to be provided by the so called 'river direction file' (RDF) that is read in during initialization of JSBACH. It is constructed such that the outflow direction of a grid box points to that neighboring box having the lowest topographical height. The structure of this data set is shown in Fig. 2.3.

The skill of the river discharge simulation depends not only on the formulation of the model equations, but also on the precise definition of the boundaries of the river catchments in the 'river direction file'. For the accurate construction of such a file see [56].

JSBACH also provides a driver to run the HD model in standalone mode (`hd_driver.f90`; see appendix A.5).

# Chapter 3

## Phenology

### Contents

---

3.1	Introduction . . . . .	43
3.2	The phenology model LoGro-P . . . . .	43
3.2.1	General dynamics . . . . .	44
3.2.2	Evergreen phenology . . . . .	45
3.2.3	Summergreen phenology . . . . .	45
3.2.4	Determining spring and autumn events . . . . .	45
3.2.5	The raingreen phenology . . . . .	48
3.2.6	The grass phenology . . . . .	48
3.2.7	The crop phenology . . . . .	48
3.3	Diagnostics . . . . .	50
3.4	Implementation details . . . . .	50

---

### 3.1 Introduction

In JSBACH two different phenology models are implemented. The only purpose these models is to provide at each time step the leaf area index (LAI), which is needed e.g. to compute the photosynthetic production (section 5), transpiration and albedo (section 4.1.1). The current standard phenology model is LoGro-P (*Logistic Growth Phenology*). LoGro-P has been evaluated by comparison with fAPAR data by Dahlke et al. [31] and also in the papers by Dalmonech et al. [32, 33] and Böttcher et al. [5]. Furthermore, the performance of LoGro-P is visible as part of MPI-ESM simulations in the studies [88, 1] where the phenology of several Earth system models participating in CMIP5 was evaluated. – During simulation, the state of this model switches from one phenological phase to the next. Since such discrete changes pose problems when trying to determine model parameters by data assimilation, Knorr et al. [78] invented another phenology model with a smoother state change during the year. This is the second phenology model implemented in JSBACH [72], but since it is not standard, it will not be described here.

### 3.2 The phenology model LoGro-P

It would be natural to link phenology closely to carbon allocation to leaves. But LoGro-P implements a much simpler approach, linking phenology directly to climate. An immediate

technical advantage is that thereby the carbon and vegetation dynamics submodels (that make up the CBALONE configuration of JSBACH) can be run independently from the phenology since there is no feedback from allocation to phenology.

In LoGro-P five phenology types are distinguished: evergreen, summergreen, raingreen, grasses, and crops, the latter consisting internally of two different phenology models for tropical and extra-tropical crops. Each PFT is in JSBACH linked to one of these phenology types.<sup>1</sup> A central parameter for LoGro-P is the maximum LAI a PFT can attain. Besides the maximum LAI, LoGro-P needs climatic conditions as input (temperature, soil moisture) and NPP (net primary productivity). All this is described in the following.

### 3.2.1 General dynamics

The key variable of LoGro-P is the LAI, denoted by the symbol  $\Lambda$  in the following. As already mentioned, the dynamics of the LAI is assumed to be limited by a prescribed maximum LAI (called  $\Lambda_{max}$ ). This maximum LAI is meant to describe a physiological limit of the considered vegetation, e.g. resulting from a finite number of leaf buds, grown in previous autumn. Or it could have its origin in the limited transport capacity of sapwoods, that limit the nutrient and water supply of leaves. Anyway, this limitation can be thought as a finite ‘‘carrying capacity’’ for the growth of leaves. Classically, growth under such conditions is modeled by the logistic equation, which is adopted here in the following form:<sup>2</sup>

$$(3.1) \quad \frac{d\Lambda}{dt} = k\Lambda\left(1 - \frac{\Lambda}{\Lambda_{max}}\right) - p\Lambda.$$

The term in brackets reduces growth when  $\Lambda$  approaches  $\Lambda_{max}$ .  $k$  and  $p$  are called *growth rate* and *shedding rate*, respectively.<sup>3</sup> Eq. (3.1) has a closed solution: Let

$$(3.2) \quad \Phi = \frac{\Lambda}{\Lambda_{max}}$$

denote the *phenological state*. Then in terms of  $\Phi$  integration over a time interval  $\tau$  gives

$$(3.3) \quad \Phi(t + \tau) = (k - p) \frac{\Phi(t)}{k\Phi(t) + \exp(-(k - p)\tau)(k - p - k\Phi(t))}.$$

The time interval  $\tau$  is currently set to the basic time step of JSBACH (usually a few minutes to half an hour) which is also used for most of the other processes (photosynthesis, energy balance, hydrology). In simulations with JSBACH coupled to an atmosphere model this time step is adopted from the atmosphere model. The temporal development of the LAI is thus simulated by iterating (3.3) from time step to time step during the simulation.

Environmental conditions enter the dynamics by the growth and shedding rate. Hence the more complicated part of LoGro-P is the modeling of their dependence on temperature, soil moisture etc., which is different for the various phenology types. For this purpose several different phenological phases are determined, which are updated only once per day.

<sup>1</sup>Technically, this link between a PFT and a phenology type is made in the `lctlib` file under the entry `PhenologyType` (see appendix C.3).

<sup>2</sup>The original logistic equation contains no shedding term  $-p\Lambda$ . Instead of introducing leaf shedding additively, one could imagine to reduce the growth rate in the logistic equation, i.e. replace  $k$  by  $k - p$ . But this is realistic only for small values of  $\Lambda$ , because then in the extreme case  $\Lambda = \Lambda_{max}$  the LAI could not decrease, whatever value the shedding rate  $p$  might have.

<sup>3</sup>The only effect of the additional leaf shedding term is to change the carrying capacity. This can be seen by rearrangement of Eq. (3.1):  $d\Lambda/dt = k\Lambda(1 - \frac{p}{k} - \Lambda/\Lambda_{max})$ , so that  $\Lambda$  is restricted to the range  $[0, 1 - p/k]\Lambda_{max}$ .

The logistic dynamics (3.1) is such that whenever  $\Lambda$  fell to zero it could never recover. Plants solve this problem by growing seeds or leaf buds containing biochemical resources sufficient to grow a leaf until it can grow further from doing photosynthesis. In LoGro-P this is modeled by assuming a minimum value of  $\Lambda$  – called “seed value” – when conditions are favorable for starting growth.

### 3.2.2 Evergreen phenology

Basic to the phenology model for evergreens is the assumption that the phenological dynamics can be divided into two subsequent phases:

- ★ a *growth phase* (spring), characterized by non-zero  $k$  and absence of leaf shedding ( $p = 0$ );
- ★ a *rest phase* (all other seasons), where growth is zero ( $k = 0$ ) and leaf shedding  $p$  is non-zero.

If not anyway zero, growth rate  $k$  and shed rate  $p$  are prescribed parameters (see table below).

### 3.2.3 Summergreen phenology

The summergreen phenology assumes three phases: Basic to the phenology model for summergreens is the assumption that the phenological dynamics can be divided into three subsequent phases:

- ★ a *growth phase* (spring), characterized by non-zero  $k$  and absence of leaf shedding ( $p = 0$ );
- ★ a *vegetative phase* (summer), where growth is zero ( $k = 0$ ) and leaf shedding  $p$  is small;
- ★ a *rest phase* (autumn and winter), also without growth ( $k = 0$ ), but more rapid leaf shedding.

Vegetative phase and rest phase differ only by the magnitude of the shed rate. The introduction of three phases instead of only those two of the evergreens is suggested by fAPAR data of summergreen vegetation, where typically after a short time of rapid growth a reduction of fAPAR is seen, much before in autumn leaves are shedded. – Also here, if not anyway zero, growth rate  $k$  and shed rate  $p$  are prescribed parameters (see table below).

### 3.2.4 Determining spring and autumn events

A main issue in modeling the phenological phases is to determine when the vegetation shifts from one phase to the next. Most simple is the modeling of the transition from the growth to the vegetative phase: it is assumed that the growth phase lasts only for a fixed time (typically some weeks), so that after that time automatically the vegetative phase starts. But for the shift from the rest phase to the growth phase (called *spring event* in the following) and the shift from the vegetative phase to the rest phase (*autumn event*) the dependence on environmental conditions cannot be neglected.<sup>4</sup> This is the topic of the following sections.

---

<sup>4</sup>Technically, for operating the dynamical phenology it needs only be known in what phase a particular plant functional type at a particular grid box currently is, the transition dates themselves are not needed. Hence, the routine `update_growth_phase()`, which handles the transition events and in particular detects the spring and autumn event, does not provide these dates. Instead, it updates the field `growth_phase`: A positive entry means that at the particular grid box the evergreens and summergreens are growing, a negative entry means that they are at rest, and Zero denotes the vegetative phase.

### The spring event

To determine the spring event the *alternating model* of Murray et al. [93], also discussed in [11], is used: Let  $S(d)$  denote the value of the heat sum at day  $d$ :<sup>5</sup>

$$(3.4) \quad S(d) = \sum_{d'=d_0}^d \max(T(d') - T_{alt}, 0),$$

where  $T(d)$  is the mean day temperature at day  $d$ ,  $T_{alt}$  is a parameter called *alternating temperature* (which has the function of a cutoff temperature in the heat sum), and  $d_0$  is the starting date for temperature summation. In the model it is taken care that this starting date falls in times after the buds have already experienced chilling (see below).<sup>6</sup>

Another key quantity of the alternating model is the number of chill days  $C(d)$ : this is the number of days with a mean day temperature *below* the alternating temperature  $T_{alt}$ , where counting starts at the day  $d_a$  of the last autumn event:

$$(3.5) \quad C(d) = \sum_{d'=d_a}^d \Theta(T_{alt} - T(d));$$

here  $\Theta()$  is the Heaviside step function. To prevent the number of chill days to grow beyond any limit in regions where the temperature is permanently below  $T_{alt}$  (e.g. in polar regions) and to assure that  $C(d)$  influences phenology only for the current growth period, the above counting of chill days is limited to a maximum number of chill days  $C_{max}$ . From  $C(d)$  a critical heatsum  $S_{crit}(d)$  is computed:

$$(3.6) \quad S_{crit}(d) = S_{crit}^{min} + S_{crit}^{range} e^{-C(d)/C_{decay}},$$

where  $S_{crit}^{min}$ ,  $S_{crit}^{range}$  and  $C_{decay}$  are parameters:  $S_{crit}^{min}$  and  $S_{crit}^{range}$  define minimum value and maximum range of the critical heatsum, whereas  $C_{decay}$  determines how fast  $S_{crit}(d)$  decreases with increasing number of chill days. Finally, the spring event happens at the first day for which during the rest phase<sup>7</sup>

$$(3.7) \quad S(d) \geq S_{crit}(d).$$

If at the spring event  $\Lambda$  is less than a given “seed value”,  $\Lambda$  is set to this seed value to allow for sufficiently fast growth.

### The autumn event

At the autumn event the vegetative phase of summergreens switches to the rest phase with rapid leaf shedding. For evergreens there is no such autumn event.

The autumn event is calculated from a pseudo soil temperature: “soil” temperature because there are hints that the signal for leaf abscission comes from the roots (via the ‘abscissic

<sup>5</sup>In the plant physiological literature, the value of  $S(d)$  is also called ‘Degree Days’ or ‘Growing Degree Days’, and typically abbreviated as ‘GDD’. See e.g. [70].

<sup>6</sup>For implementation purposes the information, whether summation has started or not, is kept also in the field representing  $S(d)$  (called `heat_sum`), which is set to a value smaller than -1 during times where no heat summation takes place.

<sup>7</sup>Technically, at this date for the considered grid point the associated entry of the array `growth_phase()` is set to +1 and `springEvent_flag` is set to “true”. Moreover the chill days count is reset to zero and `heat_sum` set to -99 to indicate that heat summation has stopped.

acid'); “pseudo” because not the real soil temperature is used, but a weighted running mean of the air temperature. Soil temperature itself is not used because the temperature of the uppermost soil layer is much too variable to give a stable signal and using alternatively the temperature of lower soil layers their seasonality would introduce a dependence on local soil properties (heat conductivity and heat capacity) preventing global tuning. How the pseudo soil temperature is calculated is described in a separate subsection below.

The autumn event happens when during the vegetative phase the pseudo soil temperature  $T^{(s)}(t)$  falls below a critical soil temperature  $T_{crit}^{(s)}$ . To prevent that this event is detected in spring, the condition is added that the mean day air temperature  $T(d)$  should be smaller than the soil temperature, i.e. the autumn event happens, when first

$$(3.8) \quad T(d) < T^{(s)}(t) < T_{crit}^{(s)}.$$

Alternatively one could use a daylength criterion to assure the beginning of the rest phase, but this has not been tried yet.

### Starting date for heat summation

Heat summation starts on the northern hemisphere always on January 1, and on the southern hemisphere on day 183. Since there are no PFTs with a summergreen or evergreen phenology at the equator there is no continuity problem with this mechanism.

### Pseudo soil temperature

The pseudo soil temperature at time step  $n$  is defined as

$$(3.9) \quad T^{(s)}(n) = \frac{1}{N} \sum_{n'=-\infty}^n T(n') e^{-(n-n')\frac{\tau}{\tau^{(s)}}},$$

where  $T(n)$  is the air temperature,  $\tau$  the length of a time step,  $\tau^{(s)}$  the characteristic time for memory loss and  $N$  the normalization constant

$$(3.10) \quad N = \sum_{n'=-\infty}^n e^{-(n-n')\frac{\tau}{\tau^{(s)}}} = \frac{1}{1 - \exp(-\tau/\tau^{(s)})}.$$

It follows from (3.9) that the computation of the pseudo soil temperature can be done iteratively:

$$(3.11) \quad T^{(s)}(n+1) = \frac{T(n)}{N} + T^{(s)}(n) e^{-\frac{\tau}{\tau^{(s)}}}$$

### Growth and shedding rates

Growth rate  $k$  is non-zero only during the growth phase. This non-zero value is a fixed parameter of the model. The shedding rate  $p$  is zero during the growth phase, and has prescribed values in the two other phases. Summergreens and evergreens differ significantly in the value of  $p$  during the rest phase: for evergreens  $p$  is such that leaves last for one or two years, whereas for summergreens  $p$  is so large that leaves are completely shedded within one or two weeks. For summergreens the value of  $p$  during the vegetative phase is chosen such that only moderate leaf shedding happens, most leaves are shedded during the rest phase.

### 3.2.5 The raingreen phenology

The phenology for raingreen vegetation is modeled without distinct phases. Raingreens grow whenever

- ★ the average soil moisture level (average over all soil layers) is above the wilting point, and
- ★ the average  $NPP$  is positive<sup>8,9</sup>

In this situation  $k$  is set to a prescribed value and the shedding rate  $p$  is computed from the relative soil water level  $w_{rel}$  in the root zone (compare eq. (2.41)) by

$$(3.12) \quad p = p_{aging} + k \begin{cases} 1 & \text{for } w_{rel} \leq w_{wilt} \\ \frac{w_{crit} - w_{rel}}{w_{crit} - w_{wilt}} & \text{for } w_{wilt} < w_{rel} < w_{crit} \\ 0 & \text{for } w_{crit} < w_{rel}. \end{cases}$$

The idea behind this parametrization is that at high water levels, i.e. above a critical water level  $w_{crit}$ , leaves are shedded only because of their aging (at rate  $p_{aging}$ ). Below this critical value the shedding rate is enhanced. One can understand this as an adaptation of the LAI to the ruling moisture availability. When  $w$  approaches the wilting point the shed rate increases until it even over-compensates the fixed growth rate  $k$  at the wilting point  $w_{wilt}$  and below. In this latter case ultimately all leaves are lost.

When any of the above conditions for growth get false, the growth rate  $k$  is set to zero and the shedding rate is set to a prescribed value for the dry season ( $p = p_{dry}$ ). Depending on the development of soil moisture, during a year the phase may thus switch several times from rest to growth and back. During the dry season  $\Lambda$  typically approaches zero. To prevent growth upon spurious rains before the wet season,  $\Lambda$  is re-set to its seed value only when relative soil moisture has reached a minimum value  $w_{leafout}$ , chosen well above the wilting point value.

### 3.2.6 The grass phenology

Grasses grow whenever sufficient soil moisture is available and, in addition, temperature is high enough. Therefore we model grasses like raingreens, but include as third criterion for the growth phase that

- ★ air temperature close to ground is larger than a critical temperature.

But even when all these conditions are fulfilled, leaves are shedded at a moderate rate when  $NPP$  is negative. This accounts for the fact that in contrast to raingreens, which are restricted to warm climates, grasses grow in all types of climates, but only when environmental conditions (mainly temperature, incoming photosynthetically active radiation, atmospheric CO<sub>2</sub> concentration, soil moisture) are all together in a suitable range for growth. And this is reflected in  $NPP$ . Another difference of this grass phenology to the raingreen phenology is that the soil moisture criterion depends only on the filling of the upper soil layer, not the average filling of all layer layers. This accounts for the shallower roots of grasses compared to raingreen vegetation.

### 3.2.7 The crop phenology

By choosing planting and harvesting dates the phenology of crops is largely controlled by farmers. So in principle one would need a model for farming practices to describe the phenology of crops. This is beyond the current scope of JSBACH. For the crop phenology

<sup>8</sup> $NPP$  is always the average value from the day before.

<sup>9</sup>This  $NPP$  criterion on positive yesterday's  $NPP$  was introduced to prevent negative  $NPP$  over larger times, that we observed in simulations, when using only the criterion for soil moisture.

implemented in JSBACH it is taken advantage of the fact that farmers cannot choose planting and harvesting dates independently of the ruling environmental conditions. Accordingly, to first approximation the behaviour of farmers is “controlled” environmentally so that an explicit modeling of farming practices is not needed. This idea underlies the crop phenology in JSBACH.

Due to climate conditions, farming practices and crop types, the phenology of crops is quite different in the tropics and extra-tropics. Therefore in JSBACH, crops in these two climate zones are modeled differently. Nevertheless, technically both crop phenologies belong to the single phenology type “crops” but the behaviour of this phenology type depends on the latitude (tropics vs. extra-tropics).

In contrast to the other phenology types, the LAI growth rate is for crops not a prescribed constant but is modeled as function of  $NPP$ . Thereby it is emphasized that most crops are annual plants, i.e. they grow every year anew from seeds and early after leaf-out they must grow from their current photosynthetic gains, in contrast to many other vegetation types that can grow from resources stored during previous growth phases. More precisely, their growth rate during growth phase is computed as

$$(3.13) \quad k = \begin{cases} f_{alloc} \frac{sla NPP}{\Lambda} & \text{for } NPP > 0 \\ 0 & \text{otherwise,} \end{cases}$$

where  $sla$  is the specific leaf area of crops (ratio of leaf area to carbon content), and  $f_{alloc}$  the fraction of NPP allocated to leaf growth.<sup>10</sup> The condition on the positivity of  $NPP$  assures that growth happens only when photosynthesis produces surplus sugars available for growth.

### Tropical crops

The phenology for tropical crops operates in the latitude band 30° South to 30° North. Like grasses, tropical crops are modeled by a continuous LAI development, i.e. without discrete phase switches. The LAI increases whenever pseudo soil temperature is sufficiently high, soil moisture of the upper soil layer is above the wilting point, and NPP is positive. Otherwise the LAI decreases at a fixed rate. In addition, this phenology model accounts for the particular sensitivity of the lack of water: Once the LAI falls below its seed value, it is reset to this value only if the upper soil moisture layer is filled beyond a critical value that lies above the wilting point. This prevents early growth before the wet season.

By this model, distinct decreases in LAI are clearly visible in regions with an expressed seasonality, mimicking well defined harvest dates. In regions with favorable growth conditions throughout the year, the LAI shows only small variations close to its maximum value. This reflects the absence of clear harvest dates and is consistent with random harvests in different parts of the grid box during the year.

### Extra-tropical crops

The phenology of crops outside the tropics distinguishes *winter crops* and *summer crops*, and accounts also for the possibility of two harvests of summer crops during a single year (“double cropping”). Accordingly, this phenology has five phenological phases, three for summer crops (rest, first growth, second growth), and two for winter crops (rest, growth). Depending on environmental conditions, the crop may switch from winter to summer crop and back, but

<sup>10</sup>Technically,  $sla$  is defined in the `lctlib`-file (see C.3), and  $f_{alloc}$  is a phenology parameter hard coded in `mo_phenology.f90`.

its always the same PFT that exhibits these different phenological behaviours. Accordingly, the phenology for crops also mimics cultivation choices by farmers based on the climate of the last year.<sup>11</sup>

The phenology model for extra-tropical crops determines harvest dates by a heat sum similar to (3.4). For summer crops heat summation is done as long as NPP is positive, for winter crops heat summation is bound in addition to the condition that they are in the growth phase. Harvest happens once the respective heat sum has reached a prescribed threshold value.<sup>12</sup> When this happens crops enter the rest phase and in this phase the LAI drops fast at a prescribed rate to zero to mimic harvest.

Every day it is checked, whether crops can switch back to the growth phase. For this the pseudo soil temperature that is also used for the autumn event of the summergreen phenology (see section 3.2.4) must be higher than  $T_{crit}^{(s)}$ . Additionally, the heat sum has to be below the prescribed harvest threshold. In the winter crop mode it is also not possible to go back to the growth phase between the summer event and the autumn event.

In fact, the model operates with three fixed calendar dates in a year: the days 70 (252), 172 (354), and 289 (105) in the northern (southern) hemisphere.<sup>13</sup> In the following these dates are called *spring*, *summer*, and *autumn event*. The following happens at these dates:

- ★ At the spring event heat summation of summer crops starts. In the case summer crops have not been harvested they enter the rest phase.
- ★ At the summer event winter crops switch into their rest phase. Summer crops that already have been harvested start a second growing phase and their heat summation starts again.
- ★ At the autumn event the heat summation of winter crops is started. This date also serves to check the success of the cropping strategy (winter/summer crop) for possibly changing the type of crop. The success is checked by testing whether the respective heat sum has been reached.

### 3.3 Diagnostics

Some variables interesting to diagnose the behaviour of LoGro-P are collected in table 3.1.

### 3.4 Implementation details

The LoGro-P phenology model is found in `mo_phenology.f90`. The central parts of the model are contained in the subroutines `update_phenology()` and `update_growth_phase()`, where the latter is called from the former. As all basic model components, also the update routine of the phenology model is called from the JSBACH main routine `jsbach_inter_1d()`.

Some model parameters needed by LoGro-P are PFT specific. These are prescribed in the `lctlib`-file (see C.3): The phenology type of the different PFTs (entry `PhenologyType`), the

<sup>11</sup>Technically, the different phenological phases are represented in the variable `growth_phase_CRP` in `mo_phenology.f90`. It takes the values -1, 0, and 1 for summer crops with the phases “rest”, “1st growth phase”, and “2nd growth phase”, respectively, and the values -2 and 2 for winter crops with phases “rest” and “growth”, respectively. Hence, `growth_phase_CRP` does not represent a phenological phase in a strict sense, but denotes also the type of crop (winter, summer).

<sup>12</sup>The chosen base temperature  $T_{alt}$  of 6°C is a compromise between a typical value for extratropical crops grown under warm condition (e.g. corn: 10°C) and cool conditions (e.g. wheat: 0°C). Similarly, the chosen threshold value  $S_{crit}$  for the heat sum is also a compromise between the respective values of several common crops (corn, wheat, rye,).

<sup>13</sup>The calendar dates have been chosen to match observed seasonality in MODIS fapar data for extended agricultural reagrions.

Table 3.1: Standard output variables useful for diagnosis of the functioning of the LoGro-P model.

name in output	symbol	stream	meaning
LAI	$\Lambda$	jsbach	leaf area index
growth_phase_EG	–	veg	growth phase of evergreens: rest (-1), growth (+1)
growth_phase_SG	–	veg	growth phase of summergreens: rest (-1), growth (0), vegetative (+1)
growth_phase_CRP	–	veg	growth phase of extra-tropical crops: summer crops: rest (-1), 1 <sup>st</sup> growth (0), 2 <sup>nd</sup> growth (1); winter crops: rest (-2), growth (+2)
heat_sum_EG	$S(d)$	veg	heat sum for evergreens
heat_sum_SG	$S(d)$	veg	heat sum for summergreens
heat_sum_CRP	$S(d)$	veg	heat sum for summer crops
heat_sum_winter	$S(d)$	veg	heat sum for winter crops
chill_days_EG	$C(d)$	veg	number of chill days for evergreens
chill_days_SG	$C(d)$	veg	number of chill days for summergreens
previous_day_NPP_rate	$NPP$	veg	average NPP rate at the day before current
previous_day_temp	$T(d)$	veg	average air temperature at the day before current
pseudo_soil_temp	$T^{(s)}(t)$	veg	pseudo soil temperature
layer_moisture	–	echam	moisture filling at different soil layers

maximum LAI  $\Lambda_{max}$  (entry `MaxLAI`), and the specific leaf area  $sla$  (`specificLeafArea_C`). The other PFT independent parameters are almost all hard coded in the initialization routine `init_pheno_params()`. These hard coded parameters are compiled in table 3.2 together with typical values. Technically these parameters are coded as elements of substructures of the structure `phenParams`.

Table 3.2: Parameters of the LoGro-P phenology model).

name in code	symbol	typical value	units	meaning
..... Parameters applying to all phenology types ( <code>phenParams%all</code> ) .....				
LAI_negligible	–	$10^{-5}$	$m^2/m^2$	below this value the LAI is considered to be zero.
laiSeed	–	0.4	$m^2/m^2$	when during a growth phase the LAI $\Lambda$ is detected to be smaller than this seed value, $\Lambda$ is reset to this value to allow sufficiently fast re-growth.
wilt_point	$w_{wilt}$	0.35	m/m	relative soil moisture content below which growth stops and shedding sets in
..... Common to evergreen and summergreen phenology ( <code>phenParams%EG_SG</code> ) .....				
tau_pseudo_soil	$\tau^{(s)}$	10.0	days	characteristic time for the memory loss for computing the pseudo soil temperature from air temperature
max_chill_days	$C_{max}$	365	days	at this value the counting of chill days stops. (This is a technical value that prevents the number of chill days to get “infinite” in permanently cold regions)

table continued on next page

Table 3.2: Parameters of the LoGro-P – continued

name in code	symbol	typical value	units	meaning
..... <i>Evergreen phenology</i> (phenParams%EG) .....				
growthRate	$k$	0.015	1/days	growth rate
alternation_temp	$T_{alt}$	4.0	$^{\circ}C$	critical temperature of the alternating model
heat_sum_min	$S_{crit}^{min}$	10	days $\cdot^{\circ}C$	minimum value of critical heat sum
heat_sum_range	$S_{crit}^{range}$	150	days $\cdot^{\circ}C$	range of critical heat sum
chill_decay_const	$C_{decay}$	150	days	number of chill days after which critical heat sum drops to $1/e$
growthPhaseLength	–	60	days	Length of growth phase
shedRate_rest	$p$	0.0008	1/days	leaf shedding rate $p$ in vegetative phase
..... <i>Summergreen phenology</i> (phenParams%SG) .....				
growthRate	$k$	0.087	1/days	growth rate
alternation_temp	$T_{alt}$	4.0	$^{\circ}C$	critical temperature of the alternating model
heat_sum_min	$S_{crit}^{min}$	30	days $\cdot^{\circ}C$	minimum value of critical heat sum
heat_sum_range	$S_{crit}^{range}$	200	days $\cdot^{\circ}C$	range of critical heat sum
chill_decay_const	$C_{decay}$	25	days	number of chill days after which chilling influence on critical heat sum drops to $1/e$ .
growthPhaseLength	–	60	days	length of growth phase
autumn_event_temp	$T_{crit}^{(s)}$	10.0	$^{\circ}C$	critical pseudo-soil-temperature that determines the autumn event
shedRate_veget	$p$	0.004	1/days	leaf shedding rate $p$ in vegetative phase
shedRate_rest	$p$	0.1	1/days	leaf shedding rate $p$ in rest phase
maxLength_gvPhase	–	270	days	number of days for growth plus vegetative phase after which switch to rest phase is forced (prevents never ending vegetative phase in boreal regions)
minLength_gvPhase	–	90	days	minimum number of days for growth plus vegetative phase before rest phase can start (prevents early switches to rest phase)
..... <i>Raingreen phenology</i> (phenParams%RG) .....				
growthRate	$k$	0.08	1/days	growth rate
shedRate_drySeason	$p_{dry}$	0.120	1/days	shed rate
shedRate_aging	$p_{aging}$	0.005	1/days	minimal leaf shedding rate due to leaf aging (inverse leaf longevity)
bucketFill_leafout	$w_{leafout}$	0.4	m/m	minimum relative soil moisture for re-set of LAI to its seed value
bucketFill_critical	$w_{crit}$	0.65	m/m	above this relative soil moisture level leaf shedding happens only because of leaf aging
..... <i>Grass phenology</i> (phenParams%GRS) .....				
growth_rate	$k$	0.09	1/days	growth rate
shedRate_growth	$p$	0.015	1/days	leaf shedding rate due to aging of leaves (inverse leaf longevity)
shedRate_drySeason	$p$	0.015	1/days	leaf shedding rate in absence of growth
crit_temp	–	4.0	$^{\circ}C$	Below this temperature growth of grasses stops
..... <i>Common parameters of crop phenology</i> (phenParams%CRP) .....				
crit_temp	$T_{crit}^{(s)}$	10	$^{\circ}C$	Below this pseudo soil temperature growth stops
shedRate_growth	$p$	0.033	1/days	Leaf shedding rate in growth phase in cold climates

table continued on next page

Table 3.2: Parameters of the LoGro-P – continued

name in code	symbol	typical value	units	meaning
shedRate_rest	$p$	0.14	1/days	Leaf shedding rate $p$ in rest phase in cold climates
leafAlloc_frac	$f_{alloc}$	0.8	–	Fraction of NPP invested into leaves during growth phase
sprout	–	0.37	m/m	fraction of soil water moisture for re-setting of LAI to its seed value
..... <i>Extra-tropical crops</i> (phenParams%CRP) .....				
heat_sum_harvest	$S_{crit}$	1300	days $\cdot^{\circ}C$	critical value of sum determining the harvest date
gdd_temp	$(T_{alt})$	6	$^{\circ}C$	minimum temperature for contribution to heat sum (“growing degree days”)



# Chapter 4

## Biophysical land surface properties

### Contents

---

4.1	Surface albedo . . . . .	55
4.1.1	Dynamic land surface albedo . . . . .	56
4.1.2	Static ECHAM5 albedo scheme . . . . .	59
4.1.3	Albedo of ice sheets . . . . .	60
4.2	Surface roughness . . . . .	60
4.3	Diagnostics . . . . .	61
4.4	Implementation details . . . . .	62
4.4.1	Albedo . . . . .	62
4.4.2	Roughness . . . . .	63

---

Changes in vegetation alter the physical properties of the land surface with consequences for the exchange of heat, water, and momentum with the atmosphere. This section describes how two of such biophysically modified surface properties – namely albedo and roughness – are calculated in JSBACH. Possible changes in surface emissivity are not considered in JSBACH.

### 4.1 Surface albedo

As inherited from ECHAM5, grid boxes in JSBACH are covered either to 100% or to 0% with ice sheets. Accordingly, the albedo is calculated differently for these two types of grid boxes. For grid boxes with ice sheets the albedo calculations are done as in ECHAM5 [116] (section 4.1.3), while for the grid boxes free of ice sheets a new method was developed (section 4.1.1). In this new scheme, land surface albedo is calculated from the current state of snow cover, LAI, vegetation distribution, and the spectral composition of solar downward radiation for each grid box. Basic for this albedo scheme is to consider within a grid box separately the albedo of snow covered surfaces, the albedo  $\alpha_{leaf}$  of the green leaves of vegetation canopies, and the albedo  $\alpha_{soil}$  of other surfaces, mostly the surface of the soil. Maps of  $\alpha_{leaf}$  and  $\alpha_{soil}$  are derived in a preprocessing step by a regression method (see appendix F) from observational albedo and fAPAR products that are based on satellite measurements. During simulations, these albedo fields, given separately for the visible and near infrared range, are dynamically combined depending on the current state of vegetation and snow cover to obtain

the grid box wide albedo in these two bands.<sup>1</sup> In addition, the albedo scheme inherited from ECHAM5 [116] is still optionally available as a part of JSBACH. This scheme computes albedo exclusively from a static map of background albedo, a static map of forest fraction, and snow cover. For completeness, this albedo scheme is also described below (subsection 4.1.2).

### 4.1.1 Dynamic land surface albedo

For grid boxes without ice-sheets, four different surface types are distinguished according to their albedo (compare fig. 4.1): land not covered by a canopy and not covered by snow (albedo  $\alpha_{soil}$ ), land not covered by a forest canopy, but covered with snow (albedo  $\alpha_{snow}$ ), a vegetation canopy without snow (albedo  $\alpha_{leaf}$ ), and forests with snow on the canopy (albedo  $\alpha_{snow,c}$ ). In addition it is assumed that for forested lands also the stems, branches, and twigs contribute to albedo – taking their albedo to be equal to the underlying ground as  $\alpha_{soil}$ .<sup>2</sup> This contribution to albedo is particularly important in regions with deciduous trees in the presence of snow, where the dark stems stand out from the white snow below.

All albedo calculations are performed separately for albedo in the visible and the near infrared band, meaning that all albedo parameters and albedo input maps must be provided individually for those two bands. Since the calculations are identical, the following description ignores this distinction.

Accounting for snow is highly important for modeling albedo, because of its unsteady appearance and its extraordinary high reflectivity. The fraction of a grid box  $c_{sn}$  covered by snow is calculated from the amount of snow on the surface (see eq. (2.49)). For the calculation of albedo,  $c_{sn}$  applies only to that part of the grid box neither covered by leaves, stems, branches, or twigs of trees because the snow on the surface is considered to cover only the soil and low vegetation but not the trees.

The albedo calculation uses two types of snow albedo: snow on ground  $\alpha_{snow}$  and snow on canopy  $\alpha_{snow,c}$ . Both are calculated by a weighted average of albedo values obtained from two different models accounting for different aspects affecting snow albedo, namely the temperature dependent snow albedo model from ECHAM5 [116, p. 43], and the snow-age dependent snow albedo model from the Biosphere-Atmosphere-Transfer-Scheme (BATS) [35, p. 23ff]:

$$(4.1) \quad \alpha_{snow} = w_{\alpha} \alpha_{snow}^{BATS} + (1 - w_{\alpha}) \alpha_{snow}^{ECHAM},$$

$$(4.2) \quad \alpha_{snow,c} = w_{\alpha} \alpha_{snow,c}^{BATS} + (1 - w_{\alpha}) \alpha_{snow,c}^{ECHAM},$$

where  $w_{\alpha}$  is the weight.<sup>3</sup> For the canopy albedo the two values  $\alpha_{snow,c}^{BATS}$  and  $\alpha_{snow,c}^{ECHAM}$  are simply model specific parameters,<sup>4</sup> while for the snow albedo at ground the calculation of the two values  $\alpha_{snow}^{BATS}$  and  $\alpha_{snow}^{ECHAM}$  is based on more detailed modelling concepts, as described next.

<sup>1</sup>A prior version of this albedo scheme is described in the appendix of Otto et al. [99].

<sup>2</sup>This assumption is consistent with the nature of the  $\alpha_{soil}$  albedo maps used in JSBACH: These maps are constructed by the regression method described in appendix F. The method is based on the separation of solar radiation fluxes related to green leaves from radiation fluxes at other surfaces. By construction these 'other surfaces' contain the radiative properties of stems, branches, and twigs, so that their reflectivity is part of the  $\alpha_{soil}$  maps. This is why  $\alpha_{soil}$  must be used also for the stem-part of the vegetation covered ground.

<sup>3</sup>This weight, with a typical value of 0.5, is set by the namelist parameter `albedo_age_weight`; see appendix C.2.1.

<sup>4</sup>They are hardcoded in the subroutine `update_albedo_snowage_temp()` as parameters `AlbedoCanopySnow_age` and `AlbedoCanopySnow_temp`.

In the ECHAM model, snow albedo is decreasing linearly with surface temperature ( $T$ ) rising from  $-5\text{ }^\circ\text{C}$  to  $0\text{ }^\circ\text{C}$ :

$$(4.3) \quad \alpha_{snow}^{ECHAM} = \alpha_{snow}^{min} + (\alpha_{snow}^{max} - \alpha_{snow}^{min}) \begin{cases} 0 & \text{for } T > 0\text{ }^\circ\text{C} \\ \frac{0\text{ }^\circ\text{C} - T}{5\text{ }^\circ\text{C}} & \text{for } -5\text{ }^\circ\text{C} \leq T \leq 0\text{ }^\circ\text{C} \\ 1 & \text{for } T < -5\text{ }^\circ\text{C}, \end{cases}$$

where  $\alpha_{snow}^{max}$  and  $\alpha_{snow}^{min}$  are prescribed PFT-specific maximum and minimum values for  $-5\text{ }^\circ\text{C}$  and  $0\text{ }^\circ\text{C}$ , respectively.<sup>5</sup>

The BATS model accounts besides temperature for additional aspects influencing snow albedo, namely a decrease of snow albedo with aging due to increased grain size and accumulation of dust with time after snow fall, and for the incidence angle of solar radiation. Here only a rough idea of the calculations is given. The final expression is

$$(4.4) \quad \alpha_{snow}^{BATS} = \alpha_{snow}(\tau_{snow}) - r(\theta_{zenith})(1 - \alpha_{snow}(\tau_{snow})),$$

where  $\alpha_{snow}(\tau_{snow})$  is the albedo for 'aged' snow at zenith angle zero decreasing with increasing 'snow age'  $\tau_{snow}$ , and  $r(\theta_{zenith})$  is a factor reducing the aged snow albedo for zenith angles  $\theta_{zenith}$  smaller than  $60^\circ$ . Snow age is updated recursively every time step such that aging increases with temperature and decreases with snowfall rate (called  $S$  in section 2.3.1.1).

Coming back to the calculation of the albedo  $\alpha$  of a grid box as a whole, it is composed from the albedo values  $\alpha_k$  of the different tiles:

$$(4.5) \quad \alpha = \sum_k^K c_k \alpha_k,$$

where  $c_k$  is the cover fraction of tile  $k$ . In this formula the unvegetated part of the grid box (compare e.g. fig. 1.2) is not showing up, because the grid box is assumed here to be divided into tiles with the unvegetated part equally distributed between them, to the consequence that the values of the cover fractions  $c_k$  of the vegetated part (see eq. (1.5)) are equal to the values of the cover fractions of these extended tiles. Accordingly, for the calculation of albedo by (4.5) the unvegetated part must be accounted for when calculating the albedo values  $\alpha_k$  for the different tiles (see below).

First the albedo calculation for a non-forest tile is described (tiles  $K-1$  and  $K$  in Fig. 4.1). On such tiles, vegetation is either gras, shrub or crop. The albedo of such a tile is calculated by

$$(4.6) \quad \begin{aligned} \alpha_k &= (1 - c_{sn}) [f_{bare} + (1 - f_{bare}) e^{-\Lambda_k/2}] \alpha_{soil} \\ &+ (1 - c_{sn}) (1 - f_{bare}) (1 - e^{-\Lambda_k/2}) \alpha_{leaf} \\ &+ c_{sn} \alpha_{snow}. \end{aligned}$$

The first line collects the contribution from all surfaces having albedo  $\alpha_{soil}$ . Firstly, this is the fraction of the snow-free and unvegetated part of the tile, i.e.  $(1 - c_{sn})f_{bare}$ . Here  $f_{bare}$  is the fraction of the grid box being unvegetated, and  $c_{sn}$  is the fraction of the grid box under snow from eq. (2.49); since bare land and snow cover are assumed to be equally distributed between the tiles of a grid box, these values also give the fractions within a tile. But also for canopy gaps the ground is seen from the sky. Hence also canopy gaps are given the albedo

<sup>5</sup>In the lctlib file (see appendix C.3) these are the entries for `AlbedoSnowVisMin`, `AlbedoSnowVisMax` (visible range) and `AlbedoSnowNirMin`, `AlbedoSnowNirMax` (near infrared range).

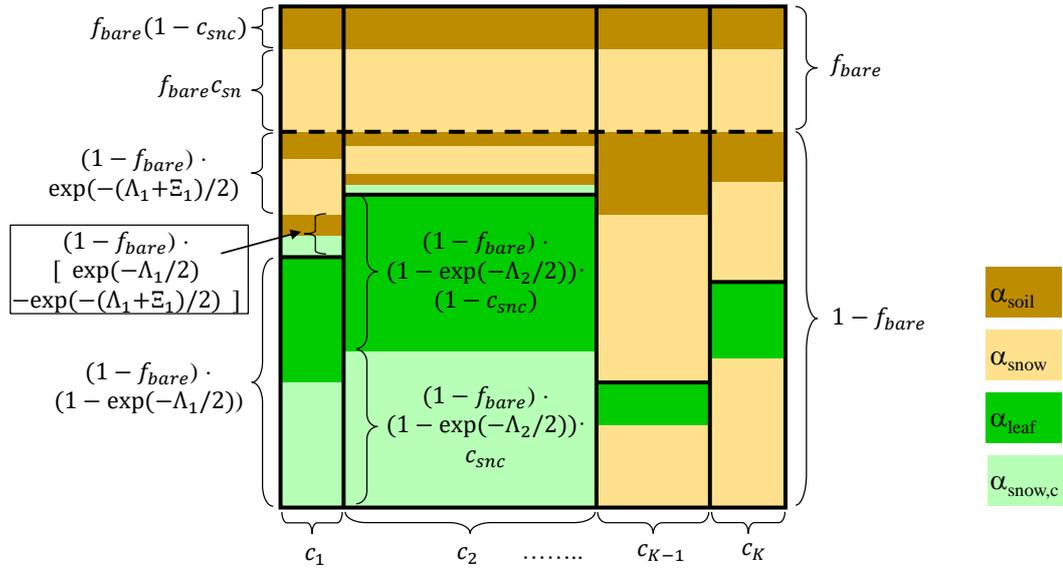


Figure 4.1: Illustration of the terms in the equations 4.6 and 4.10. In this example tile 1 and tile 2 are covered by forest, whereas the tiles  $K - 1$  and  $K$  are occupied by grasses, shrubs or crops.

value  $\alpha_{soil}$ . Since  $1 - f_{bare}$  is the fraction of the tile covered with vegetation, these gaps occupy the fraction  $(1 - f_{bare}) e^{-\Lambda_i/2}$ , where the exponential factor is an estimate for the size of canopy gaps as discussed in connection with eq. (1.7), although here  $\Lambda$  is the value of LAI at the time step of calculations (see chapter 3) so that the contribution from canopy gaps causes albedo to vary in time in addition to variations from snow cover.

The second line adds the contribution from surfaces having albedo  $\alpha_{leaf}$ . For the non-forest tile considered, only the part with snowfree vegetation contributes here, being the vegetated part of the grid box except the canopy gaps (therefore  $1 - e^{-\Lambda_i/2}$ ).

The third line is the contribution from snow albedo  $\alpha_{snow}$ , where it is assumed that all surfaces so far not accounted for in the first two lines are covered by snow. This concerns in particular the rest of the vegetated lands that is assumed to be completely snowed under because the tile discussed here contains only low vegetation.

For forested tiles the calculation of albedo is a bit more complicated (tile 1 and 2 in Fig.4.1). First, because snow may not completely fill the interception reservoir so that even in the presence of snow part of the canopy is still visible from sky (contributing to 'snow masking'). Accordingly, the albedo value of forest canopies lies in between the albedo of the green leaves ( $\alpha_{leaf}$ ) and the albedo of snow on canopies ( $\alpha_{snow,c}$ ), namely

$$(4.7) \quad \alpha_{canopy} = (1 - c_{snc})\alpha_{leaf} + c_{snc}\alpha_{snow,c},$$

where  $c_{snc}$  is the fraction of canopy under snow given by eq. (2.11) and  $\alpha_{snow,c}$  is a model parameter. Second, also the albedo contribution from branches and stems is accounted for. This is particularly important for deciduous forests in snowy winters, where the dark wood of trees clearly stand out from the bright snow on the ground (also contributing to 'snow masking'). The albedo of stems is set to the value  $\alpha_{soil}$  of the underlying ground (see above) so that the value of the 'stem' albedo varies between  $\alpha_{soil}$  and that of snow on the canopy, namely

$$(4.8) \quad \alpha_{stem} = (1 - c_{snc})\alpha_{soil} + c_{snc}\alpha_{snow,c},$$

where it is assumed that  $c_{snc}$  also describes what fraction of the branches and stems is covered by snow. While  $\alpha_{canopy}$  and  $\alpha_{stem}$  refer to the trees of the forested tile, it remains to consider the albedo of the ground. This can be either under snow or free of snow. Therefore this albedo is given by

$$(4.9) \quad \alpha_{ground} = (1 - c_{sn})\alpha_{soil} + c_{sn}\alpha_{snow}.$$

With these three albedo values, the total albedo of a forested tile is calculated by

$$(4.10) \quad \begin{aligned} \alpha_k = & \left( f_{bare} + (1 - f_{bare}) e^{-(\Lambda_k + \Xi_k)S(\theta_{zenith})} \right) \alpha_{ground} \\ & + (1 - f_{bare}) \left( 1 - e^{-\Lambda_k/2} \right) \alpha_{canopy} \\ & + (1 - f_{bare}) \left( e^{-\Lambda_k/2} - e^{-(\Lambda_k + \Xi_k)S(\theta_{zenith})} \right) \alpha_{stem} \end{aligned}$$

The first line gives the contribution from ground, i.e. from the fraction of bare land  $f_{bare}$  and from the canopy gaps in the forested part  $(1 - f_{bare})$  of the grid box. For the canopy gaps it is assumed that not only the LAI  $\Lambda$  is of importance to determine the closedness of the canopy, but also the presence of stems, branches and twigs. Therefore in analogy to LAI also a 'stem area index'  $\Xi$  is defined,<sup>6</sup> being the (top) surface area of stems, branches and twigs divided by ground area. For a top view the fraction of canopy gaps could therefore be modeled by  $\exp(-(\Lambda_k + \Xi_k)/2)$ , the division by 2 accounting for random orientation of leaves, branches, twigs, and stems, which is particularly not true for stems, but may be acceptable as an approximation. But in the presence of stems, the reflectivity of canopy gaps depends also on the position of the sun, because trees are erect and shed the gaps. To account for this the 'sky view factor'

$$(4.11) \quad S(\theta_{zenith}) = (2 - \cos(\theta_{zenith}))/2$$

is introduced into the exponential, taking the value 1/2 for the sun in zenith (zenith angle  $\theta_{zenith} = 0$ ) and 1 for the sun located at the horizon. Thereby the relevance of canopy gaps is decreased with increasing zenith angle. The second line in eq. (4.10) gives the contribution from the dense forest canopy. Therefore here the contribution from stems as represented by  $\Xi$  is absent. Finally, the third line gives the contribution only from the rest of the grid box that can be attributed to the 'stem' part as seen by the subtraction of the exponential for the size of canopy gaps without stems and the exponential for the size of canopy gaps with stems.

#### 4.1.2 Static ECHAM5 albedo scheme

As a legacy still the ECHAM5 albedo scheme is part of JSBACH.<sup>7</sup> Also here, the albedos  $\alpha_k$  from different tiles are combined to an albedo for the grid box using eq. (4.5). The tile albedo  $\alpha_k$  is combined from the albedo  $\alpha_{forest}$  of the forested fraction of a grid box (assuming the fraction to be equally distributed to all tiles) and the non-forested rest, accounting also for snow and canopy gaps. More precisely, denoting by  $c_{tree}$  the fraction of a grid box vegetated with tree PFTs,<sup>8</sup> the forest fraction of a tile  $k$  including canopy gaps is calculated by

$$(4.12) \quad c_{forest}^{(k)} = c_{tree}(1 - e^{-\max(\Lambda_k, 2)}),$$

<sup>6</sup>The values for this PFT-specific stem area index  $\Xi$  are defined in the land cover type library file (see appendix C.3).

<sup>7</sup>To use the ECHAM5 albedo scheme the namelist switch `use_albedo` must be set to false.

<sup>8</sup> $c_{tree}$  is a static input field with name `forest_fract`; see table C.22 in the appendix.

following a similar logic for calculating canopy gaps from leaf area index  $\Lambda_k$  as described in connection with eq. (1.7), although here a maximum size of canopy gaps is assumed by using the maximum of  $\Lambda_k$  and 2 in the exponential.<sup>9</sup> Using this canopy fraction, the tile albedo is then calculated from

$$(4.13) \quad \alpha_k = \max(\alpha_{bg}, (1 - c_{forest})(c_{sn}\alpha_{snow} + (1 - c_{sn})\alpha_{bg}) + c_{forest}(c_{snc}\alpha_{snow,c} + (1 - c_{snc})\alpha_{bg})).$$

In this expression, by using the *max*-function, tile albedo is forced not to be less than the background albedo  $\alpha_{bg}$ , which is a global albedo map for snowfree land [60].<sup>10</sup> Values larger than  $\alpha_{bg}$  are constructed in by combining the albedos of the forested ( $c_{forest}$ ) and non-forested ( $1 - c_{forest}$ ) fractions of the tile, which are obtained by combining the albedo of the snow covered part ( $\alpha_{snow}$  in absence of forest and  $\alpha_{snow,c}$  for forest canopies) with the background albedo  $\alpha_{bg}$  for the snowed part. Snow fractions in absence ( $c_{sn}$ ) and presence ( $c_{snc}$ ) of forests are obtained from eqs. (2.49) and (2.11).  $\alpha_{snow}$  is calculated in dependence of surface temperature from (4.3), while the albedo for snow on canopies  $\alpha_{snow,c}$  is a model parameter.

### 4.1.3 Albedo of ice sheets

For ice sheets the ECHAM snow albedo model eq. (4.3) is applied, although with maximum and minimum albedo values appropriate for ice sheets.<sup>11</sup> Aging of snow covering the ice sheets is not considered, i.e  $w_\alpha = 0$  in eq. (4.1). The two parameters for the albedo in the NIR range are adjusted to minimize the near surface temperature bias over the ice sheets in summer.

## 4.2 Surface roughness

Surface roughness, defined as the length scale of obstacles that various surface types exhibit to the near surface flow, may differ by several orders of magnitude within a grid box. To derive the turbulence-driven exchange fluxes of heat and momentum between the land surface and the atmosphere, the characteristic lengths for the different surface types within a grid box must thus be appropriately aggregated into a single roughness length for momentum  $z_{0m}^{surf}$  and for heat  $z_{0h}^{surf}$ , respectively. In addition, depending on the way the blocked flow drag is handled by the atmospheric model, also orography must be included in the roughness length for momentum. In this case, the overall roughness length for momentum of a grid box is calculated from

$$(4.14) \quad z_{0m} = \sqrt{(z_{0m}^{surf})^2 + (z_0^{oro})^2}$$

where  $z_0^{oro}$  is the contribution to roughness length from the orography in the particular grid box. This enlarges the roughness length of momentum due to sub-grid scale orographic structures as described in [58].

<sup>9</sup>Note that because  $c_{tree}$  is a static input field independent from the vegetation distribution of the PFTs, the use of the ECHAM5 albedo scheme described here may introduce an inconsistency with other parts of JSBACH.

<sup>10</sup>In the JSBACH initial file  $\alpha_{bg}$  is given as variable `albedo` (compare table C.22 in the appendix).

<sup>11</sup>0 °C: 0.78 for visible, 0.4 for NIR; -5 °C: 0.9 for visible, 0.75 for NIR.

By invoking the blending height concept of [24], the roughness lengths for momentum and heat  $z_{0m,h}^{surf}$  are calculated by

$$(4.15) \quad z_{0m,h}^{surf} = l_b \exp \left( - \left( \sum_i \frac{f_i}{\left( \ln \left( l_b / z_{0m,h}^i \right) \right)^2} \right)^{-0.5} \right)$$

where the sum runs over all surface types  $i$  in the grid box,  $l_b$  is the blending height (set to  $100m$ ),  $z_{0m,h}^i$  are the roughness lengths for momentum resp. heat of surface type  $i$ , and  $f_i$  the fraction of the grid box with surface type  $i$ .<sup>12</sup> For simplicity, the  $m$  and  $h$  subscripts are dropped from  $z_{0m,h}^i$  in the following.

Considering first the roughness length for heat, three surface types  $i$  with different characteristic roughness lengths are distinguished within each tile (and included in the sum in eq. (4.15)): snow-covered surface ( $z_0^i = 1mm$ ), snow-free bare surface ( $z_0^i = 5mm$ ), and snow-free vegetated surface. To calculate the roughness length of snow-free vegetated surfaces two variants are implemented. In the first variant, a PFT-dependent value is taken for  $z_0^i$  (see the last column in table 4.2). In the second variant, the roughness length of snow-free vegetation is calculated by interpolating between two PFT-specific roughness length values  $z_0^i(\Lambda = 0)$  and  $z_0^i(\Lambda = \infty)$ , the first being the value in the absence of leaves, and the second the one for a fully developed canopy, where  $\Lambda$  is the leaf area index (compare table 4.2). With these two values the roughness length of snow-free vegetated surfaces is calculated by

$$(4.16) \quad z_0^i = z_{0,\Lambda=0}^i + (z_{0,\Lambda=\infty}^i - z_{0,\Lambda=0}^i) \tanh(\gamma\Lambda)$$

where the parameter  $\gamma$  controls how fast the roughness length saturates with increasing LAI.<sup>13</sup> It is assumed that snow is not able to stay on forest canopies but falls to the ground. Hence, even in the presence of snow, the roughness length of forests (including the snow-covered part) is calculated from equation (4.16).

For the roughness length for momentum, the same formulas as for heat are used except that only a bare and a vegetated part are distinguished within a tile. The presence of snow is therefore not modifying the roughness length for momentum.

In summary, with eqs. (4.14) to (4.16), the grid box-wide roughness lengths  $z_{0m}$  and  $z_{0h}$  can change with LAI, snow cover, orography, and in particular with the cover fractions of tiles so that natural (chapter 9) and anthropogenic (chapter 10) land cover change dynamically modify the surface exchange of heat and momentum in JSBACH via changing surface roughness (compare explanations in connection with eq. (2.68)).

## 4.3 Diagnostics

Table 4.1 lists useful output for diagnostics.

<sup>12</sup>This aggregation of  $z_{0m,h}$  across different surface types implies that the scale of patches of different cover types is assumed to be of the order of kilometers [24]. In contrast, most of the other parametrizations within JSBACH (e.g. hydrology) are based on the assumption that patches of different cover type are well mixed or separated at a much smaller spatial scale.

<sup>13</sup>A typical choice is  $\gamma = 0.4$  so that canopy closure is reached at approximately  $\Lambda = 5$ .

Table 4.1: Useful output diagnostics from the albedo and roughness calculations.

name in output	units	stream	meaning
<code>albedo_vis</code>	-	<code>jsbach</code>	surface albedo in visible range (eq. (4.5))
<code>albedo_nir</code>	-	<code>jsbach</code>	surface albedo in visible range (eq. (4.5))
<code>snow_age</code>	-	<code>jsbach</code>	snow 'age'
<code>az0lh</code>	m	<code>surf</code>	roughness length for heat over land (eq. (4.15))
<code>az0lm</code>	m	<code>surf</code>	roughness length for momentum over land <sup>1</sup> (eq. (4.15))
<code>az0</code>	m	<code>surf</code>	roughness length for heat globally (eq. (4.15))

<sup>1</sup>Not in standard output.

## 4.4 Implementation details

### 4.4.1 Albedo

The main routine for calculating albedo is `mo_land_surface::update_albedo()`. During the 'radiation time steps', it is called from the JSBACH main routine `mo_jsbach_interface::jsbach_inter_id()` before the radiation is recalculated in the atmosphere, because only during these time steps new albedo values are needed. When the standard JSBACH albedo scheme is used (section 4.1.1), `update_albedo()` calls `mo_land_surface::update_albedo_snowage_temp()` where the main calculations are done. These routines use the ground and canopy snow fractions  $c_{sn}$  and  $c_{snc}$ , and snow age  $\tau_{snow}$  that were calculated earlier in the time step in a call of `mo_soil::update_soil()` from the JSBACH main routine.

There are several namelist switches to activate the different albedo schemes. For the standard albedo scheme, the namelist parameter `use_albedo` must be set to `.TRUE.`, otherwise the static ECHAM5 albedo scheme is active. The standard albedo scheme needs input maps for ground albedo  $\alpha_{soil}$  separately for the visible and near infrared range that are part of the JSBACH initial data file `jsbach.nc` (see appendix C.4). Also for the vegetation albedo  $\alpha_{leaf}$  maps from `jsbach.nc` can be used (namelist switch `use_albedocanopy=.TRUE.`), but alternatively, by setting `use_albedocanopy=.FALSE.`, they can be replaced by PFT specific albedo values that are defined under the keywords `AlbedoCanopyVIS` and `AlbedoCanopyNIR` in the land cover type library file (see appendix C.3); typical values are listed in table 4.2. Usage of PFT-specific albedo values is particularly useful when model components for natural (chapter 9) or anthropogenic (chapter 10) land cover change are used, because then the static vegetation albedo maps might get inconsistent with the changing distribution of vegetation. How to derive the maps for  $\alpha_{soil}$  and  $\alpha_{leaf}$  from data is explained in appendix F. Another option concerns the parametrization of snow albedo. The namelist parameter `albedo_age_weight` defines the value of  $w_\alpha$  from equation (4.1) that linearly weights the snow albedo values calculated from the two snow albedo models.

In case the static ECHAM5 albedo scheme from section 4.1.2 is used (`use_albedo=.FALSE.`), the main routine for albedo calculations `update_albedo()` calls `mo_land_surface::update_land_surface_fast()`. This routine uses the background albedo map `albedo` from the JSBACH initial data file `jsbach.nc` (see appendix C.4).

Additionally, the code contains an experimental albedo scheme where soil surface albedo depends on litter and soil carbon [140] which is activated by setting in the albedo namelist besides `use_albedo=.TRUE.` also `use_albedosoil=.TRUE.`.

Table 4.2: Typical values of PFT specific parameters for: albedo of vegetation covered surfaces in the visible ( $\alpha_{leaf,vis}$ ) and near infrared range ( $\alpha_{leaf,nir}$ ); roughness length of vegetation covered surfaces without leaves ( $z_{0,\Lambda=0}$ ) and with closed canopy ( $z_{0,\Lambda=\infty}$ ), roughness length of vegetation covered surfaces for the model without LAI-dependence ( $z_0$ ). Technically, these values are specified in the land cover type library file (see appendix C.3).

PFT	$\alpha_{leaf,vis}$	$\alpha_{leaf,nir}$	$z_{0,\Lambda=0}$	$z_{0,\Lambda=\infty}$	$z_0$
	-	-	[m]	[m]	[m]
forest tropical evergreen	0.03	0.22	2.0	5.0	2.0
forest tropical deciduous	0.04	0.23	1.0	3.0	1.0
forest extra-tropical evergreen	0.04	0.23	1.0	3.0	1.0
forest extra-tropical deciduous	0.05	0.26	1.0	3.0	1.0
shrubs raingreen	0.05	0.25	0.3	1.0	0.5
shrubs cold	0.05	0.28	0.3	1.0	0.5
grass/pasture	0.08	0.33	0.005	0.1	0.05
crops	0.08	0.33	0.005	0.1	0.05

#### 4.4.2 Roughness

To calculate roughness lengths for heat and momentum the routine `mo_land_boundary::update_land_boundary_up()` is called every time step from the JSBACH main routine `mo_jsbach_interface::jsbach_inter_1d()`. The inclusion of LAI in the calculation of roughness length of vegetated surfaces according to equation (4.16) can be switched on or off with the namelist parameter `use_roughness_lai` of the main JSBACH namelist (see C.2.11). In the same namelist, the parameter `use_roughness_oro` controls whether orography is included in the calculations: setting it to `.TRUE.` means that eq. (4.14) is used for the grid box-wide roughness length for momentum, otherwise  $z_{0m}$  is set to  $z_{0m}^{surf}$  calculated from eq. (4.15).



## Chapter 5

# Plant productivity and stomatal control

### Contents

---

5.1	Introduction . . . . .	65
5.2	Radiation in the canopy . . . . .	66
5.2.1	The two-stream approximation . . . . .	66
5.2.2	Accounting for clumping of vegetation . . . . .	70
5.2.3	Implementation details . . . . .	70
5.3	Photosynthesis . . . . .	72
5.3.1	C3-photosynthesis: The Farquhar model . . . . .	72
5.3.1.1	Temperature dependency . . . . .	74
5.3.1.2	Nitrogen scaling . . . . .	75
5.3.1.3	Stomatal conductance . . . . .	76
5.3.1.4	Assimilation under water stress . . . . .	77
5.3.2	C4-photosynthesis: The Collatz model . . . . .	79
5.3.3	Implementation details . . . . .	80
5.4	Diagnostics . . . . .	83

---

## 5.1 Introduction

In terrestrial plants the assimilation of carbon by photosynthesis (“productivity”) and the loss of water through transpiration are closely connected: Plants need to open their stomata to gain  $CO_2$  for the buildup of sugars. The opening of the stomata is however associated with a loss of water. Therefore most plants actively control their stomata to achieve a high water use efficiency, i.e. a high carbon gain per  $H_2O$  molecule transpired. Through this control mechanism carbon cycle and hydrological cycle are closely coupled. Hence, within a land biosphere model as JSBACH, the estimation of productivity is combined with the estimation of stomatal conductivity, a measure for the transpirational ability of plants (see e.g. [80]).

To represent the dynamics of plant carbon uptake and water loss via the stomata, JSBACH uses components of the BETHY model [75, 76]: the BETHY canopy radiation code based on Sellers [124] and two photosynthesis modules, one, following Farquhar et al. [41], for the C3 pathway, and the other, following Collatz et al. [28], for the C4 pathway, both working in combination with the BETHY approach for the stomatal dynamics.

## 5.2 Radiation in the canopy

The canopy radiation model has the purpose to determine which portion of the incoming radiation is absorbed by the leaves and is thereby available for photosynthesis. Studies indicate that the radiative properties of leaves are such that practically all radiation in the photosynthetically active band (400-700 nm) is absorbed by chloroplasts and not by any other pigments of the leaves [124]. This simplifies the situation significantly. The key quantity to be determined by the canopy radiation model is the amount of radiation absorbed in the canopy. For practical reasons one does not determine the absolute amount, but the so called *fraction of absorbed photosynthetically active radiation* fAPAR. This is the quantity the canopy radiation model hands over to the photosynthesis model. – For a discussion of the JSBACH canopy radiation model in comparison to ray tracing simulations and implementations in other vegetation models see Loew et al. [87].

### 5.2.1 The two-stream approximation

The JSBACH radiation calculations are a prerequisite for subsequently computing the photosynthetic carbon fixation. Thus the radiation calculations must consider only that part of the solar spectrum relevant for photosynthesis (400-700 nm) which is the *photosynthetically active radiation*, abbreviated as PAR. PAR absorbed in the canopy depends on the direction of the incoming radiation, i.e. the zenith angle, the thickness of the canopy (i.e. the leaf area index  $\Lambda$ ), the optical properties of the leaves, their orientation, and the optical properties of the underlying soils. As proposed by Dickinson [34] and later on studied in more detail by Sellers [124], for closed canopies this complicated radiation problem can be simplified by using the so called *two stream approximation*. The approximation requires that scatterers (i.e. leaves) are distributed homogeneously in the canopy (“green jelly”), and the radiation distribution within the canopy is horizontally invariant. Hence - within the canopy - it is sufficient to consider vertically upward and downward radiation fluxes, the two ‘streams’ after which the approximation has been named.

The radiation is split into a *direct* part  $R$ , and into upward and downward *diffuse* components  $R_{\uparrow}$  and  $R_{\downarrow}$ . Considering first the direct beam, its way through the canopy is attenuated exponentially:

$$(5.1) \quad R(l) = R(0)e^{-kl},$$

Here,  $l$  denotes the cumulated leaf area index measured from the top of the canopy. It varies from 0 at the top to  $\Lambda$  at the bottom of the canopy. The leaf area index  $\Lambda$  is obtained from the phenology model (see chapter 3), although modified for clumping of vegetation (see section 5.2.2).  $R(0)$  is the direct radiation at the top of the canopy, and  $R(l)$  is the radiation at “depth”  $l$ . The extinction coefficient  $k$  of the canopy, also called “optical depth”, depends on the zenith angle of the sun  $\theta$ :

$$(5.2) \quad k = k(\cos(\theta)) = G(\cos(\theta))/\cos(\theta).$$

Here  $G(\cos(\theta))$  is the relative area shaded by a canopy element with LAI= 1 following the direction of the incoming beam. Since the orientation of the leaves is not known, it is assumed that they are oriented randomly with equal probability. In this case  $G(\cos(\theta)) = 1/2$  independent of the zenith angle (see [96]). Hence

$$(5.3) \quad k = \frac{1}{2 \cos(\theta)}.$$

Assuming diffuse radiation entering the canopy from all directions with equal intensity, its average inverse optical depth per unit leaf area index  $\bar{\mu}$  is given as

$$(5.4) \quad \bar{\mu} = \int_0^1 dx \frac{x}{G(x)},$$

which for random leaf orientations, i.e.  $G(x) = 1/2$ , reduces to

$$(5.5) \quad \bar{\mu} = 1.$$

Let  $\omega$  denote the scattering coefficient of leaves, i.e. the fraction of radiation transmitted (scattered forward) and reflected (scattered backward). Then  $(1 - \omega)$  is the fraction of radiation absorbed. Moreover, let  $\beta$  denote the fraction of radiation scattered backward. Then  $\beta\omega$  is the fraction scattered backward,  $(1 - \beta)\omega$  is the fraction of radiation scattered forward, and  $1 - (1 - \beta)\omega$  is the fraction of radiation not scattered forward, i.e. neither being absorbed nor scattered backward. Let, analogously, denote  $\beta_0$  the fraction of direct radiation scattered backward. With this, the budget for diffusive radiation in the canopy is obtained by the two stream approximation, describing separately the “streams” of upward diffuse radiation  $R_\uparrow$  and downward diffuse radiation  $R_\downarrow$  radiation by the two coupled equations

$$(5.6) \quad \begin{aligned} -\frac{dR_\uparrow}{dl} + (1 - (1 - \beta)\omega) \frac{R_\uparrow}{\bar{\mu}} - \omega\beta \frac{R_\downarrow}{\bar{\mu}} &= k\omega\beta_0 R(0)e^{-kl} \\ \frac{dR_\downarrow}{dl} + (1 - (1 - \beta)\omega) \frac{R_\downarrow}{\bar{\mu}} - \omega\beta \frac{R_\uparrow}{\bar{\mu}} &= k\omega(1 - \beta_0)R(0)e^{-kl}. \end{aligned}$$

The change in radiation flux (the first left hand side term<sup>1</sup>) is (i) the result of a gain of diffuse radiation from the scattering of the direct beam of sunlight at individual leaf elements (the right hand side terms), and (ii) the combined effect of absorption of diffuse radiation by the leaves and its redistribution between the two streams (all together represented by the second and third left hand side terms). More precisely, the scattering of the direct beam of radiation  $R(l)$ , distributed exponentially in the canopy (compare eq. (5.1)), generates diffuse radiation when it hits leaves. Its fraction  $\omega\beta_0$  is scattered upward, while the fraction  $(1 - \beta_0)\omega$  is scattered downward. Multiplied with the extinction coefficient  $k$  for direct radiation, this gives the contributions to diffuse radiation in the two streams (see right hand side terms). Considering the second left hand side terms, these describe the reduction of the respective stream by backward scattering and absorption (therefore the factor  $1 - (1 - \beta)\omega$ ) of diffuse radiation per unit change in LAI (therefore the scaling by the mean inverse optical depth  $\bar{\mu}$  of diffuse radiation). Part of this diffuse radiation is absorbed by the leaves, but the fraction  $\beta\omega$  is scattered back, i.e. it contributes to the respective other stream (see the third left hand side terms).

These equations need to be complemented by boundary conditions: At the top of the canopy the diffuse downward radiation flux  $R_\downarrow(0)$  is prescribed, either from external data (JSBACH standalone), or from the radiation calculations of ECHAM (variable `sw_par_down` in table 1.1). At the bottom of the canopy, upward and downward radiation fluxes are related by

$$(5.7) \quad R_\uparrow(\Lambda) = \alpha_{soil,PAR}(R(\Lambda) + R_\downarrow(\Lambda)),$$

with  $\alpha_{soil,PAR}$  being the soil albedo in the PAR band (compare appendix F for more information on soil albedo).

---

<sup>1</sup>The reason for the different signs of these first left hand side terms in the two equations is that  $l$  increases downwards, so that a downward increase in  $R_\downarrow$  implies  $dR_\downarrow/dl$  to be positive, while an upward increase in  $R_\uparrow$  implies  $dR_\uparrow/dl$  to be negative.

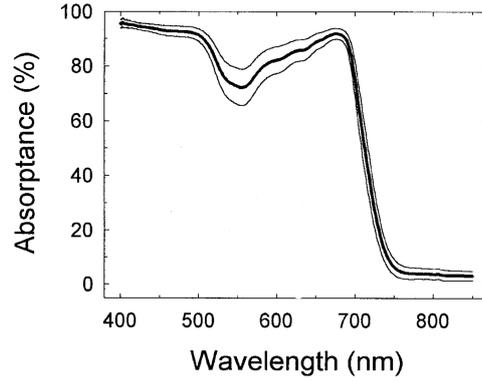


Figure 5.1: Light absorption  $1 - \omega$  of leaves from 26 species of a broad range of habitats  $\pm 1$  standard deviation (from [74]).

In the following it is additionally assumed that leaf reflectance and transmission are equal in the PAR spectral band, for both direct and diffuse radiation, i.e.  $\beta = \beta_0 = 1/2$ . With this, the only free parameter in (5.6) is the scattering coefficient  $\omega$ . An appropriate value can be obtained from Fig. 5.1: in the photosynthetically active region  $1 - \omega$  varies between 0.7 and 0.95, i.e.  $\omega$  is in the range from 0.3 to 0.05, largely independent of the vegetation type. Within the photosynthetically active spectral range, photosynthesis is most efficient in the high absorption bands (around 550 nm chlorophyll is inactive). Thus  $\omega$  should be chosen rather in the lower range. Accordingly, in JSBACH  $\omega = 0.12$  is used for all types of vegetation.

The system of differential equations (5.6) can be solved with the ansatz

$$(5.8) \quad \begin{aligned} R_{\uparrow} &= q_2 R(0) e^{-kl} + p_1 B_1 e^{hl} + p_2 B_2 e^{-hl} \\ R_{\downarrow} &= -q_1 R(0) e^{-kl} + B_1 e^{hl} + B_2 e^{-hl}. \end{aligned}$$

Using the above values  $\beta = \beta_0 = 1/2$  and  $\bar{\mu} = 1$  and entering (5.8) into (5.6) one finds

$$(5.9) \quad q_1 = \frac{\omega}{2} \frac{k(k+1)}{k^2 + \omega - 1} = \frac{\omega}{2} \frac{1 + 2 \cos(\theta)}{1 - 4 \cos^2(\theta)(1 - \omega)}$$

$$(5.10) \quad q_2 = \frac{\omega}{2} \frac{k(k-1)}{k^2 + \omega - 1} = \frac{\omega}{2} \frac{1 - 2 \cos(\theta)}{1 - 4 \cos^2(\theta)(1 - \omega)}$$

$$(5.11) \quad p_1 = \frac{2 + 2\sqrt{1 - \omega} - \omega}{\omega}$$

$$(5.12) \quad p_2 = 1/p_1$$

$$(5.13) \quad h = \sqrt{1 - \omega}.$$

And with the definitions

$$(5.14) \quad \gamma_1(\Lambda) := (p_1 - \rho_{soil}) e^{h\Lambda}$$

$$(5.15) \quad \gamma_2(\Lambda) := (p_2 - \rho_{soil}) e^{-h\Lambda}$$

$$(5.16) \quad \eta(\Lambda) := (\rho_{soil}(1 - q_1) - q_2) e^{-k\Lambda}$$

the boundary conditions (5.7) give

$$(5.17) \quad B_1 = B_1(\Lambda) = \frac{\eta(\Lambda)R(0) - \gamma_2(\Lambda)(R_\downarrow(0) + q_1R(0))}{\gamma_1(\Lambda) - \gamma_2(\Lambda)}$$

$$(5.18) \quad B_2 = B_2(\Lambda) = \frac{\eta(\Lambda)R(0) - \gamma_1(\Lambda)(R_\downarrow(0) + q_1R(0))}{\gamma_2(\Lambda) - \gamma_1(\Lambda)},$$

where for convenience the dependence of  $B_1$  and  $B_2$  on the total canopy leaf area index  $\Lambda$  is now made explicit.

With this solution the radiation absorbed by the canopy can be computed as follows. The total flux density (positive downwards) through a horizontal plane in the canopy at a given depth  $l$  of the cumulative LAI  $l$  is

$$(5.19) \quad \begin{aligned} I_{tot}(l) &= R(l) + R_\downarrow(l) - R_\uparrow(l) \\ &= (1 - q_1 - q_2)R(0)e^{-kl} + (1 - p_1)B_1(\Lambda)e^{hl} + (1 - p_2)B_2(\Lambda)e^{-hl}. \end{aligned}$$

The radiation absorbed in a canopy layer between the values  $l_1$  and  $l_2$ ,  $l_1 < l_2$ , is the difference between the radiation entering the layer at  $l_1$  and the radiation leaving it at  $l_2$ . Hence  $fAPAR$ , the fraction of PAR absorbed in such a layer relative to the total incoming radiation is

$$(5.20) \quad fAPAR(l_1, l_2) = \frac{I_{tot}(l_1) - I_{tot}(l_2)}{R(0) + R_\downarrow(0)}.$$

The above equations give the full solution for light absorption in the canopy at every depth. If photosynthesis operated homogeneously throughout the canopy, it would be sufficient to hand over the total light absorption in the canopy (i.e.  $fAPAR(0, \Lambda)$ ) to the photosynthesis module of JSBACH. But observations show that the nitrogen-rich enzyme Rubisco, which plays a key role in carbon assimilation, is allocated predominantly in well lit leaves, i.e. mostly in the top part of the canopy. In JSBACH, the resulting differences in photosynthetic capacity across the canopy are accounted for by a scaling ansatz (“nitrogen scaling”, see section 5.3). Under such non-uniform conditions, the photosynthesis routine needs information on light absorption at different depths in the canopy. In JSBACH the canopy is therefore split into several layers of equal depth (in terms of LAI). For each of these layers  $fAPAR$  is computed separately according to (5.20) and handed over to the photosynthesis routine.<sup>2</sup>

For comparison with observationally derived  $fAPAR$  data the special case of a deep canopy is of particular interest, because then the soil albedo  $\alpha_{soil, PAR}$  gets irrelevant because no light is reaching the ground and the only remaining model parameter is  $\omega$ . While practically this situation is already reached for values  $\Lambda$  as small as 5 or 6, formally this case is obtained by performing in the above equations the limit  $\Lambda \rightarrow \infty$ . This gives after some calculations

$$(5.21) \quad fAPAR_\infty := \lim_{\Lambda \rightarrow \infty} fAPAR(0, \Lambda) = \frac{I_{tot}(0)}{R(0) + R_\downarrow(0)} = 1 - \frac{(q_2 + p_2q_1)R(0) + p_2R_\downarrow(0)}{R(0) + R_\downarrow(0)},$$

where  $\lim_{\Lambda \rightarrow \infty} I_{tot}(\Lambda) = 0$ ,  $\lim_{\Lambda \rightarrow \infty} B_1(\Lambda) = 0$  and  $\lim_{\Lambda \rightarrow \infty} B_2(\Lambda) = R_\downarrow(0) + q_1R(0)$  has been used.

To get an idea of the magnitude of maximum  $fAPAR$  one can take advantage of the smallness of  $\omega$  and expand (5.21) in this parameter. This gives to first order

$$(5.22) \quad fAPAR_\infty = 1 - \frac{\frac{2}{1+2\cos(\theta)}R(0) + R_\downarrow(0)}{4(R(0) + R_\downarrow(0))}\omega + \mathcal{O}(\omega^2),$$

<sup>2</sup>Note that this splitting into several canopy layers is done for better process representation, namely nitrogen scaling, and not for numerical reasons to solve the equations (5.6): In the approach followed here,  $fAPAR$  is obtained from their *exact* solution in every layer!

where to linear order in  $\omega$  one has invoked  $p_2 = \omega/4$ ,  $q_1 = (1 - 2\cos(\theta))^{-1}\omega/2$  and  $q_2 = (1 + 2\cos(\theta))^{-1}\omega/2$ . With a typical value of  $\omega = 0.12$ , this gives for clear sky conditions (i.e.  $R_{\downarrow}(0) = 0$ ) at zenith angle zero  $fAPAR_{\infty} = 1 - \omega/6 = 0.98$ , and for fully clouded sky (i.e.  $R(0) = 0$ )  $fAPAR_{\infty} = 1 - \omega/3 = 0.96$ .

### 5.2.2 Accounting for clumping of vegetation

Radiation is only absorbed in canopies, not in canopy gaps. Following the BETHY approach [75], this is accounted for in JSBACH by using a slightly modified LAI  $\Lambda_{clump}$  in the above two-stream approximation. In contrast to the actual LAI  $\Lambda$  obtained from the phenology model  $\Lambda_{clump}$  is accounting for *clumping* of vegetation.  $\Lambda_{clump}$  is obtained by assuming that  $\Lambda$  represents the average LAI over the vegetated area of a tile *including* canopy gaps. With  $f_{clump}$  being the fraction of vegetated area and  $1 - f_{clump}$  being the fraction of gaps,  $\Lambda_{clump}$  is defined as

$$(5.23) \quad \Lambda_{clump} = \Lambda / f_{clump}.$$

This definition assures that  $\Lambda_{clump}$  integrated over the fraction  $f_{clump}$  of the vegetated area, is identical to  $\Lambda$  integrated over the full vegetated area including canopy gaps. Accounting for clumping, this approach thus leads to an increase of LAI of those areas with closed canopy. Using  $\Lambda_{clump}$ , equation (5.19) gives the downward flux per square meter of closed canopy, i.e. the area reduced by clumping. To scale it back to the whole vegetated area including gaps,  $I_{tot}(l)$  has to be multiplied by  $f_{clump}$ .

The clumping fraction itself is determined from the LAI via

$$(5.24) \quad f_{clump} = \begin{cases} \tilde{f}_{clump} \Lambda / \tilde{\Lambda}_{clump} & \text{for } \Lambda < \tilde{\Lambda}_{clump} \\ \tilde{f}_{clump} & \text{otherwise.} \end{cases}$$

Thus, above a critical clump LAI  $\tilde{\Lambda}_{clump}$  the clumping fraction has the value  $\tilde{f}_{clump}$ , and below this value is approached linearly.<sup>3</sup> The overall effect of the clumping scheme is that in the presence of canopy gaps, i.e. whenever  $f_{clump} < 1$ , the radiation absorbed in the canopy is reduced.

### 5.2.3 Implementation details

The canopy radiation model is located in routine `fapar1()` (module `mo_bethy_fapar.f90`). The routine is called from `update_bethy` and computes fAPAR for a given number of canopy layers `ncanopy` that are set in namelist `bethy_ctl` (compare C.3). In the standard setup `ncanopy=3`. For PFTs without nitrogen scaling (see section 5.3.1.2), the number of canopy layers has no effect on the simulation results because, as explained above, the equations for the canopy radiation balance (5.6) are solved analytically and not by using a numerical discretization scheme.

Table 5.1 summarizes the the interface of `fapar1()`. Note that neither the direct downward radiation flux  $R(0)$ , nor the downward diffuse flux  $R_{\downarrow}(0)$  is listed as input to the routine. The reason for this is that fAPAR is independent from the total incoming radiation ( $R(0) + R_{\downarrow}(0)$ ), but depends only on the relative fractions of direct and diffuse radiation content.<sup>4</sup> Moreover, since the fractional direct and diffuse radiation contents sum up to one,

<sup>3</sup>Since JSBACH 3.0 the standard values for these two parameters are  $\tilde{f}_{clump} = 1.0$  and  $\tilde{\Lambda}_{clump} = 3.0$ . In earlier versions  $\tilde{f}_{clump}$  had the different value 0.9, as used e.g. in the study [87].

<sup>4</sup>Reason: the radiation absorbed in the canopy is strictly proportional to the total incoming radiation, so this cancels out in (5.20).

it is sufficient to supply the fraction of the direct part in the total incoming radiation, as seen in table 5.1.

Table 5.1: Interface of the routine `fapar1()` (the canopy radiation model).

name in code	symbol	intent	meaning
<code>pLAI</code>	$\Lambda$	in	total leaf area index of the canopy
<code>rhos</code>	$\alpha_{soil,PAR}$	in	albedo of soil below vegetation canopy in the PAR-band
<code>coszen</code>	$\cos(\theta)$	in	cosine of solar zenith angle
<code>fdir</code>	$\frac{R(0)}{R(0) + R_{\downarrow}(0)}$	in	fraction of direct radiation in incoming PAR
<code>soil_reflectivity_par</code>	$\alpha_{soil,PAR}$	in	soil albedo in the PAR band
<code>aPAR</code>	$fAPAR$	out	fAPAR: fraction of incoming PAR absorbed in each canopy layer
Other technical information			
<code>ngrpts</code>	-	in	number of grid points to be handled in this call
<code>mask</code>	-	in	logical mask indicating the presence/absence of vegetation
<code>nnl</code>	-	in	number of canopy layers
<code>LAI1</code>	-	out	LAI within each canopy layer
<code>ddl</code>	-	in	canopy layer borders as fractions ( $\sum_{i=1}^{nnl} (ddl_i - ddl_{i-1}) = 1$ )

As described in section 4.1, depending on the switch `use_albedo` either the dynamic albedo scheme is used (`true`) or the original static ECHAM scheme (`false`). These two schemes are employing different observational albedo data, in particular for soil albedo. To stay consistent, the respective soil albedo is then also used in the canopy radiation scheme. More precisely, when the dynamic albedo scheme is active,  $\alpha_{soil,PAR}$  is taken as the input field `albedo_soil_vis` (see table C.22), which is the soil albedo in the visible range and thus a good approximation for the PAR band soil albedo. When instead the ECHAM albedo scheme is active,  $\alpha_{soil,PAR}$  is based on its soil albedo input field `albedo` (also table C.22), but only after appropriate adjustment because this field represents the albedo for the whole solar spectral range: The PAR band albedo is obtained from it via

$$(5.25) \quad \alpha_{soil,PAR} = \max(0, 0.92\alpha_s - 0.015),$$

where  $\alpha_s$  is the whole range albedo field `albedo`; for a justification of this formula see [75, eq. (122)].

It should be noted that there is a problem in directly comparing the calculated fAPAR with measured fAPAR-data: fAPAR is defined as a quotient of two values (compare (5.20)), each strongly varying during a day, and also on longer time scales. It thus does not make much sense to compute time averages (daily, weekly etc.) of modeled fAPAR for comparison with measurement data. Therefore, in addition to fAPAR, JSBACH provides the downward incoming radiation and the radiation absorbed in the canopy *averaged* over the output interval. Using these output fields, fAPAR suitable for comparison with observations can be calculated by first averaging them individually over a suitable time period (e.g. one month), and then dividing these averages.

### 5.3 Photosynthesis

The purpose of the photosynthesis module in JSBACH is twofold: to derive the plant productivity (for carbon issues), and to derive the stomatal conductance (for hydrological issues). The implementation of photosynthesis is taken over from BETHY [75, 76]. Peculiar to this approach is to proceed in two steps (compare Fig. 1.1): In a first step productivity is computed by ignoring a possible limitation in water availability (“water stress”). The main assumption here is that the leaf internal CO<sub>2</sub>-concentration is a fixed fraction of the ambient CO<sub>2</sub>-concentration (compare eq. (5.47)). The resulting *potential* productivity determines the *unstressed* stomatal conductance. Using this, the soil hydrology model (see section 2.3.2) is run to compute the potential water losses from transpiration. This loss may be larger than the water actually available to the plants from the storage in the soils. Considering this possible water deficit, the unstressed stomatal conductance is reduced to the *stressed* stomatal conductance. The latter is then used in a second call of the photosynthesis routine to compute the *actual* productivity. Implicit to this approach is the assumption that during one simulation time step the soil water situation is not much changing, so that the inconsistency between plant productivity and soil water changes introduced by using the unstressed instead of stressed stomatal conductivity when accounting for plant transpiration in the soil water budget remains small.

#### 5.3.1 C3-photosynthesis: The Farquhar model

For the representation of C3-plants the model of Farquhar et al. [41] is implemented in JSBACH (see also the book by von Caemmerer [16]). It is based on the observation that the assimilation rate in the chloroplasts is limited either by the carboxylation rate  $J_C$  of the RuBP molecules (Ribulose 1,5-biphosphate), or the transport rate  $J_E$  of the two electrons freed during the photoreaction. The total rate of carbon fixation (“productivity”)  $A$ , measured in [mol(C)/s m<sup>2</sup>(leaf area)], is thus expressed as

$$(5.26) \quad A = \min(J_C, J_E) - r_d,$$

where  $r_d$  is a special kind of “dark respiration” [41] (called “day respiration” in [16]) occurring in the presence of light to keep the photosynthetic apparatus running, but being different from photorespiration.

There is a process competing with the carboxylation of RuBP, namely its oxygenation. Hence a high O<sub>2</sub> concentration reduces the carboxylation rate. Accounting for both processes, the Farquhar model describes the carboxylation rate as

$$(5.27) \quad J_C = V_{max} \frac{c_i - \Gamma_*}{c_i + K_C(1 + O_i/K_O)}.$$

Here  $V_{max}$  is the maximum carboxylation rate,  $c_i$  and  $O_i$  are the leaf internal CO<sub>2</sub> and O<sub>2</sub> concentrations<sup>5</sup>.  $\Gamma_*$  is the so called CO<sub>2</sub> compensation point, which is the CO<sub>2</sub> concentration where assimilation breaks down, and  $K_C$  and  $K_O$  are Michaelis-Menten constants parametrizing the dependence on CO<sub>2</sub> and O<sub>2</sub> concentrations.<sup>6</sup> The electron transport rate is given by

<sup>5</sup>If not otherwise stated,  $c_i$  is in this chapter the leaf internal CO<sub>2</sub> concentration and not cover fraction.

<sup>6</sup>The original formulation by Farquhar [41, 16] is based on partial pressures instead of concentrations for CO<sub>2</sub> and O<sub>2</sub>. Since partial pressure  $p_{part}$  and concentration  $c$  are linearly related by  $p_{part} = p c$ , where  $p$  is total pressure, the structure of the original Farquhar equation is identical to the version used here, except that all parameters appear as divided by  $p$ . The formulation implemented in JSBACH can be understood as using everywhere  $p = 1$  bar (surface pressure), and this pressure value is absorbed into the parameters. Accordingly,

$$(5.28) \quad J_E = J(I) \frac{c_i - \Gamma_*}{4(c_i + 2\Gamma_*)}$$

with

$$(5.29) \quad J(I) = J_{max} \frac{\alpha I}{\sqrt{J_{max}^2 + \alpha^2 I^2}}.$$

Here

$$(5.30) \quad I = (R(0) + R_{\downarrow}(0)) * fAPAR$$

is the radiation absorbed by the leaves in the photosynthetically active band per  $\text{m}^2$  leaf area (compare eq. (5.20)), with fAPAR computed by the canopy radiation model described in the previous section (5.2).  $J_{max}$  is the maximum electron transport rate and  $\alpha$  the quantum efficiency for photon capture. Since (5.29) is an empirical formula, different formulations are in use [41, 43] – (5.29) is just one variant that meets the two necessary bounding conditions: slope  $\alpha$  for small  $I$ , and saturation at  $J_{max}$  for large  $I$ .

The above equations contain a number of parameters. The values used in JSBACH are listed in table 5.2; note that leaf internal  $\text{CO}_2$ -concentration  $c_i$  is calculated from ambient  $\text{CO}_2$ -concentration  $c_a$  following eq. (5.47) in water unstressed conditions, whereas for stressed conditions it is implicit to the solutions (5.51) and (5.52).  $V_{max}$  and  $J_{max}$  depend on the vegetation type. JSBACH reads these values at runtime from the file `lctlib.def` that is listed in appendix C.3.

Table 5.2: Parameters of the Farquhar model that are independent of vegetation type. Values are given for the reference temperature  $25^\circ\text{C}$ . The value of  $\gamma_d$  is taken from [41], for the origin of the other values see [75]. Except  $\Gamma_*$ , whose value is implicit to eq. (5.32), values are hard-coded in `mo.bethy_photosyn.f90`.

param.	meaning	value at $25^\circ\text{C}$
$\alpha$	efficiency of photon capture	0.28
$\Gamma_*$	$\text{CO}_2$ compensation point	$42.5^\dagger \mu\text{mol}(\text{CO}_2)/\text{mol}(\text{air})$
$O_i$	leaf internal $\text{O}_2$ concentration	$0.21 \text{ mol}(\text{O}_2)/\text{mol}(\text{air})$
$K_C$	Michaelis-Menten constant for $\text{CO}_2$	$460 \mu\text{mol}(\text{CO}_2)/\text{mol}(\text{air})$
$K_O$	Michaelis-Menten constant for $\text{O}_2$	$0.33 \text{ mol}(\text{O}_2)/\text{mol}(\text{air})$
$\gamma_d$	Fraction of dark respiration	0.011

<sup>†</sup>This is the value obtained from (5.32) for  $25^\circ\text{C}$ .

It remains to specify dark respiration  $r_d$ . Unlike the formulas above, which are valid for a whole temperature range, the following equation refers only to  $25^\circ\text{C}$  conditions, in the following indicated by index 0:

$$(5.31) \quad r_{d,0} = \gamma_d V_{max,0},$$

the equations given here get invalid for plants at higher altitudes. Looking at the Farquhar equations, only the carboxylation rate is pressure dependent, but not the electron transport rate. Assuming constant  $V_{max}$ , the carboxylation rate drops when lowering  $p$  (assuming the  $\text{CO}_2$  concentration to be well above the compensation point). Measurements however indicate that  $V_{max}$  increases with altitude (see [12]) so that this effect at least partially compensates the error we make by always assuming surface pressure.

Table 5.3: Activation energies for the temperature dependence of the parameters in the Farquhar and Collatz models in [J/mol]. For the origin of these values see [75]. The values are hard-coded in `mo_bethy_photosyn.f90`

param.	E	see Equation
$E_C$	59356	(5.33)
$E_O$	35948	(5.33)
$E_V$	58520	(5.34), (5.59)
$E_R$	45000	(5.37)
$E_K$	50967	(5.59)

with  $\gamma_d$  from table 5.2. This equation serves a reference to obtain  $r_d$  at temperatures different from 25°C (see next section). The parametrization (5.31) expresses the idea<sup>7</sup> that  $r_d$  strongly depends on the nitrogen content of leaves: The value of  $V_{max}$  reflects the Rubisco content of leaves because this enzyme catalyzes the carboxylation during photosynthesis. But the nitrogen of leaves is predominantly stored in the Rubisco molecules. Therefore, as expressed by eq. (5.31),  $r_{d,0}$  should roughly be proportional to  $V_{max,0}$ .

### 5.3.1.1 Temperature dependency

The above model equations need to be completed by specifying the temperature dependency. Following Farquhar [42], the CO<sub>2</sub> compensation point changes with temperature:

$$(5.32) \quad \Gamma_* = 1.7 T_C \quad (\Gamma_* \text{ in ppm and } T_C \text{ in } ^\circ\text{C}).^8$$

The Michaelis-Menten constants for the enzyme kinetics of Rubisco have an Arrhenius type temperature dependence [42]

$$(5.33) \quad \begin{aligned} K_C &= K_{C,0} \exp\left(\left(\frac{T}{298K} - 1\right)\frac{E_C}{RT}\right) \\ K_O &= K_{O,0} \exp\left(\left(\frac{T}{298K} - 1\right)\frac{E_O}{RT}\right), \end{aligned}$$

where  $K_{C,0}$  and  $K_{O,0}$  are the respective Michaelis-Menten constants at 25°C,  $E_C$  and  $E_O$  are the associated activation energies (table 5.3),  $R$  is the gas constant, and  $T$  the temperature in  $K$ . The temperature dependence of the maximum carboxylation rate is of the same type [42]:

$$(5.34) \quad V_{max} = V_{max,0} f(T) \exp\left(\left(\frac{T}{298K} - 1\right)\frac{E_V}{RT}\right),$$

with  $V_{max,0}$  the reference value at 25°C and  $E_V$  the associated activation energy (table 5.3). Following Collatz et al. [27] the additional function

$$(5.35) \quad f(T) = 1/(1 + e^{1.3(T-328K)})$$

accounts for an inhibition of the biochemical processes at high temperatures outside the validity range of the Farquhar model:  $f(T)$  assumes values between 0 and 1 with a sharp decrease from 1 to 0 around 55°C.

<sup>7</sup>By personal communication with Jens Kattge.

<sup>8</sup>This relation is obtained from measurements between 15°C and 35°C at spinach, wheat, cotton, beans, aspen and wayfaringtree; see [14]. For an alternative approximation to a slightly extended temperature range see [8].

Following once more [42], the temperature dependence of the maximum electron transport rate is

$$(5.36) \quad J_{max} = \begin{cases} J_{max,0} f(T) \frac{T_C}{25^{\circ}C} & \text{for } T_C > 0^{\circ}C \\ 0 & \text{otherwise.} \end{cases}$$

where  $J_{max,0}$  is the reference value at  $25^{\circ}C$ ; the function  $f(T)$  is once more the above “high temperature inhibition function” (5.35). – An alternative temperature dependence of  $V_{max}$  and  $J_{max}$  is discussed in [73] and further analyzed in the context of JSBACH in [51].

For the temperature dependence of the dark respiration an Arrhenius relationship is assumed as well[16]:

$$(5.37) \quad r_d = f(T)g(I)r_{d,0} \exp\left(\left(\frac{T}{298K} - 1\right)\frac{E_R}{RT}\right),$$

with the associated activation potential  $E_R$  (table 5.3). Dark respiration is known to decrease significantly already at very low solar light intensities where it reaches a constant value [14, 100]. Therefore, in addition to the high temperature inhibition via  $f(T)$ , the function

$$(5.38) \quad g(I) = 0.5(1 + e^{-I/10 \mu\text{mol m}^{-2}\text{s}^{-1}}),$$

has been introduced in (5.37) to reduce the dark respiration with increasing solar irradiance  $I$ . The function  $g(I)$  is constructed such that for irradiances larger than  $50 \mu\text{mol}(\text{photons})/\text{m}^2\text{s}$  dark respiration drops fastly to 50% of its uninhibited value.

Values for the various parameters in the above formulas can be found in [42, 8, 16]. Only  $V_{max,0}$  and  $J_{max,0}$  are assumed to depend on the PFT, all other constants are set to a single value for all vegetation types.

### 5.3.1.2 Nitrogen scaling

The vital use of nitrogen is very costly for plants, it can consume 25-40% of the total net photosynthetic carbon gain [20]. Hence the photosynthetic enzyme Rubisco, which contains the majority of leaf nitrogen, is allocated mainly at places where light gain is largest. Optimality considerations [43, 125] indicate that the Rubisco profile in the canopy follows the profile of light intensity. To account for this, the maximum carboxylation rate  $V_{max}$  (5.34) – known to be proportional to the Rubisco content – is modified here by a factor depending on the canopy depth. Following the implementation of BETHY [75], thus the modified quantity

$$(5.39) \quad V_{max}(l) = V_{max}e^{-k_{noon}l},$$

is used instead of  $V_{max}$  in the calculation of photosynthetic productivity, where  $l$  is the cumulated leaf area index from canopy top to the particular depth, and  $k_{noon}$  is the optical depth at local noon (compare eq. (5.3)).<sup>9</sup> Technically, fAPAR and thus carbon assimilation is

<sup>9</sup> $k_{noon}$  depends on the zenith angle at noon. This zenith angle is computed from declination and latitude. ECHAM computes declination only at time steps when radiation is calculated. When running JSBACH as part of ECHAM, JSBACH obtains the value for declination from ECHAM. Therefore JSBACH computes  $k_{noon}$  at most time steps from a slightly outdated declination. In contrast, in JSBACH standalone declination is computed every time step so that  $k_{noon}$  is always up-to-date. This is a source of slight numerical differences in plant productivity between JSBACH coupled to ECHAM and JSBACH standalone, even when the JSBACH interface is run with identical climate data. For numerical tests the switch `interface_test` can be used that sets  $k_{noon} = 1$  so that there is no dependence on declination.

computed separately for several canopy layers so that carboxylation rates depend via (5.39) on the assumed nitrogen distribution in the canopy.<sup>10</sup>

### 5.3.1.3 Stomatal conductance

Assimilation is accompanied by transpiration losses of water vapor: while CO<sub>2</sub> is taken up through the leaf stomata, the plant loses water through the same openings. The opening and closing of stomata is actively controlled by plants. The flux

$$(5.40) \quad J_{H_2O} = g_L^{H_2O} (h_i - h_o),$$

of water vapor through the stomata (i.e. the transpiration) is determined by the so called stomatal conductance  $g_L^{H_2O}$ , which characterizes the wideness of the stomatal opening. Here  $h_i$  and  $h_o$  are the (absolute) humidities inside and outside the leaf stomata<sup>11</sup>. A similar relation defines the CO<sub>2</sub> flux

$$(5.41) \quad J_{CO_2} = g_L^{CO_2} (C_a - C_i)$$

across the stomata, where  $g_L^{CO_2}$  is the stomatal conductance for CO<sub>2</sub>, with  $C_i$  and  $C_a$  denoting the the CO<sub>2</sub> densities inside and outside the stomata<sup>12</sup>. As the mass of water vapor molecules is much smaller than the mass of CO<sub>2</sub> molecules, the stomatal conductance for H<sub>2</sub>O is higher than for CO<sub>2</sub>. Approximately, the conductances are related by (see e.g. [70] p. 186)

$$(5.42) \quad g_L^{H_2O} \approx 1.6 g_L^{CO_2}.$$

It should be noted that eqs. (5.40) and (5.41) are understood in mass units. For the following it is however more convenient to rewrite the latter equation in molar units. Denoting the volume mixing ratios of CO<sub>2</sub> inside and outside the stomata by  $c_i$  and  $c_a$  eq. (5.41) results in the equivalent expression

$$(5.43) \quad J_{CO_2} = g_L^{CO_2} (c_a - c_i) M_{CO_2} \frac{p}{RT},$$

where  $M_{CO_2}$  is the molar mass of CO<sub>2</sub>, and the universal gas constant  $R$ , pressure  $p$  and the leaf temperature  $T$  (in  $K$ ) have entered by assuming CO<sub>2</sub> to behave like an ideal gas.

To obtain stomatal conductance, it is assumed that plants control the CO<sub>2</sub> uptake and water vapor losses in an optimal way, i.e. stomata are opened sufficiently wide to assimilate carbon according to the photosynthetic capacity at a given solar irradiation, but not wider to prevent unnecessary transpiration losses of water. Since the CO<sub>2</sub> uptake of leaves is determined by assimilated minus respired CO<sub>2</sub>, i.e. by the net assimilation rate  $A$  (see eq. (5.26)), it follows from the above equations that the stomatal conductance of water vapor can be obtained from leaf productivity by<sup>13</sup>

$$(5.44) \quad g_L^{H_2O} = \frac{1.6}{c_a - c_i} \frac{RT}{p} A.$$

<sup>10</sup>Whether a particular PFT takes part in nitrogen scaling is prescribed in the initialization file `lctlib.def` by the keyword `NitrogenScalingFlag` (see appendix C.3).

<sup>11</sup>The units in (5.40) are:  $[J_{H_2O}] = kg(H_2O)/m^2s$ ,  $[h_i, h_o] = kg(H_2O)/m^3$  so that  $[g_L^{H_2O}] = m/s$ . Note that in the literature different definitions of stomatal conductance are used. Instead of absolute humidity, sometimes the mass mixing ratio for water vapor is used, whereby the units for the stomatal conductance are different.

<sup>12</sup>Units:  $[J_{CO_2}] = kg(CO_2)/m^2(leaf)s$ ,  $[C_a, C_i] = kg(CO_2)/m^3(air)$ .

<sup>13</sup>Note that  $A$  is in molar flux units, while  $J_{CO_2}$  is in mass flux units. The conversion factor between them is the mass mixing ratio  $M_{CO_2}$  that in the transition from (5.43) to (5.44) is absorbed in  $A$ .

Equation (5.44) describes how to obtain the stomatal conductivity from the assimilation rate  $A$  at leaf level. To obtain the conductivity of a canopy as a whole one has to sum the fluxes through all stomata. To this end, let the total assimilation in the canopy be defined as the integral over all canopy layers, accounting for the local LAI from zero to its maximum value  $\Lambda$ :

$$(5.45) \quad A_C = \int_0^\Lambda dl A(l).$$

Here  $A(l)$  is the net assimilation from (5.26) computed at a particular canopy depth  $l$ , with  $l$  being the accumulated LAI from the top of the canopy. By integrating (5.44) over the whole canopy, one obtains the “unstressed” canopy conductivity:

$$(5.46) \quad g_C^{H_2O} = \frac{1.6}{c_a - c_i} \frac{RT}{p} A_C.$$

Hence, stomatal conductance in the absence of water stress is proportional to the total assimilation.

This is in accordance with the findings of Schulze et al. [122, Fig. 3C] across the global ecosystems. An analysis of the data from this study by Knorr [75, p. 41] gives for C3 plants the empirical relationship  $g_C \approx 0.883A_C$  (with units  $[g_C]=\text{mm/s}$  and  $[A_C]=\mu\text{mol}(\text{CO}_2)/\text{m}^2\text{s}$ ). Using this relationship in (5.46), one obtains  $c_a - c_i \approx 45\text{ppm}$  for standard conditions ( $25^\circ\text{C}$  and standard pressure), so that leaf internal  $\text{CO}_2$  concentration  $c_i$  is related to ambient  $\text{CO}_2$  concentration by

$$(5.47) \quad c_i \approx 0.87 c_a$$

when using  $c_a = 355$  ppm, which is consistent with observations [75]. This relationship is crucial for the formulation of photosynthetic production. It allows explicitly to evaluate the equations of the Farquhar model in the absence of water stress.

#### 5.3.1.4 Assimilation under water stress

As mentioned above, photosynthesis is computed twice in JSBACH: first to determine the unstressed canopy conductance (see eq. (5.46)), and second to determine from the stressed canopy conductance the actual carbon assimilation.

The stressed canopy conductance is obtained from the unstressed conductance by a simple scaling factor  $f_{ws}$  expressing a relative soil moisture in the root zone between an upper threshold  $W_{crit} = f_{crit}W_{max}$  and a lower threshold  $W_{wilt} = f_{wilt}W_{max}$ , with  $W_{max}$  being the maximum water storage in the root zone (compare table 2.3):

$$(5.48) \quad f_{ws} = \begin{cases} 0 & \text{for } W \leq W_{wilt} \\ \frac{W - W_{wilt}}{W_{crit} - W_{wilt}} & \text{for } W_{wilt} < W < W_{crit} \\ 1 & \text{otherwise.} \end{cases}$$

Here  $W$  is the amount of water in the root zone (compare (2.40) for the root zone definition). Parameter  $f_{wilt}$  characterizes the wilting point below which plants cannot extract water from the soils any more. Parameter  $f_{crit}$  stands for the relative filling of the root zone above which transpiration is unhindered<sup>14</sup>. Thereby the stressed canopy conductance is obtained as

$$(5.49) \quad g_{C,stress}^{H_2O} = \begin{cases} f_{ws} g_C^{H_2O} & \text{for } q_a \leq q_s \\ 0 & \text{otherwise,} \end{cases}$$

<sup>14</sup>Standard values  $f_{crit} = 0.75$  and  $f_{wilt} = 0.35$  are defined inline in `mo_soil.f90` as parameters `moist_crit_fract` and `moist_wilt_fract` respectively, but can be overwritten via the soil namelist C.13.

where in addition to the water stress arising from the lack of soil water, also water stress arising from a too humid air is accounted for by setting the stressed conductance to zero when  $q_a$ , the specific humidity of the ambient air (i.e. in the lowest atmospheric layer), is larger than the saturation humidity  $q_s$  at the surface (calculated from surface temperature and surface pressure).<sup>15</sup>

It is assumed that all canopy layers experience the same water stress. Therefore, for each canopy layer the stressed stomatal conductance is calculated from the unstressed stomatal conductance by applying the same stress factor experienced by the canopy as a whole.<sup>16</sup>

$$(5.50) \quad g_{L,stress}^{H_2O} = g_L^{H_2O} \frac{g_{C,stress}^{H_2O}}{g_C^{H_2O}}.$$

From this stressed stomatal conductance plant productivity  $A$  is obtained from the Farquhar model “reversely” as follows. To derive the unstressed canopy conductance (5.44) the leaf internal CO<sub>2</sub> concentration is prescribed according to (5.47). Note that under water stress this leaf internal CO<sub>2</sub> concentration changes. Using eq. (5.44) with the stressed conductance (5.50) instead of the unstressed one (now at leaf level), one has an expression for the assimilation  $A_{stress}$  under water stress that can replace the left and side of (5.26). Thereby, the unknowns to solve equations (5.26), (5.27) and (5.28) are the leaf internal CO<sub>2</sub> concentration  $c_i$  and the assimilation rate  $A_{stress}$  under water stress. Eliminating  $c_i$  from these equations gives for each of the two cases in eq. (5.26) a quadratic equation for the stressed rates  $J_{C,stress}$  and  $J_{E,stress}$  with the solutions<sup>17</sup>

$$(5.51) \quad J_{C,stress} = a_1 - \sqrt{a_1^2 - b_1}$$

with  $a_1 = \frac{1}{2} \left( V_{max} + r_d + g_0 \left( c_a + K_C \left( 1 + \frac{O_i}{K_O} \right) \right) \right)$   
 $b_1 = V_{max} (g_0 (c_a - \Gamma_*) + r_d),$

$$(5.52) \quad J_{E,stress} = a_2 - \sqrt{a_2^2 - b_2}$$

with  $a_2 = \frac{1}{2} \left( \frac{J(I)}{4} + r_d + g_0 (c_a + 2\Gamma_*) \right)$   
 $b_2 = \frac{J(I)}{4} (g_0 (c_a - \Gamma_*) + r_d)$

where the stressed stomatal conductance enters via the shorthand

$$(5.53) \quad g_0 = \frac{p}{RT} \frac{g_{L,stress}^{H_2O}}{1.6}.$$

Finally, in accordance with (5.26), the assimilation rate under water stress is then obtained for each canopy layer from

$$(5.54) \quad A_{stress} = \min(J_{C,stress}, J_{E,stress}) - r_d.$$

Summing the assimilation rates from all layers across the canopy, the resulting canopy assimilation rate per leaf area is the carbon input to the land carbon cycle (see eq. (6.2)).

<sup>15</sup>The stressed canopy conductance is computed via  $f_{ws}$  from the unstressed canopy conductance in the hydrology routine `mo_soil::update_soil()`.

<sup>16</sup>This spreading of the water stress from the canopy to the leaf level is done in `mo_bethy::update_bethy()`.

<sup>17</sup>The sign before the square root in the solution to the quadratic equations is given by the condition that leaf internal CO<sub>2</sub> concentration must be positive.

Table 5.4: Parameters of the Collatz model. For the origin of these values see [75]. The parameters  $V_{p,max,0}$  and  $k_0$  are standard values for C4 grasses and C4 crops, read in from the file `lctlib.def` under the keywords `CarboxRate` and `ETransport`; see appendix C.3.

param.	meaning	value at 25°C
$V_{p,max,0}$	maximum PEP carboxylation rate (C4 grasses)	8 $\mu\text{mol}(\text{CO}_2)/\text{m}^2\text{s}$
$k_0$	PEPcase $\text{CO}_2$ specificity (C4 grasses)	0.14 $\text{mol}(\text{air})/\text{m}^2\text{s}$
$\alpha_i$	integrated C4 quantum efficiency	0.04 $\text{mol}(\text{C})/\text{m}^2\text{s}$
$\theta_s$	curve parameter for $J_E$	0.83
$\gamma_d$	fraction of dark respiration	0.031

### 5.3.2 C4-photosynthesis: The Collatz model

For C4-plants the model proposed by Collatz et al. [28] is implemented into JSBACH, taking over the implementation from [75]. The structure of the Collatz model is almost identical to the Farquhar model described above, except that the equations for the carboxylation rate (5.27), and the electron transport rate (5.28),(5.29) are replaced by

$$(5.55) \quad J_C = k c_i,$$

$$(5.56) \quad J_E = \frac{1}{2\theta_s} \left( V_{p,max} + J_i - \sqrt{(V_{p,max} + J_i)^2 - 4\theta_s V_{p,max} J_i} \right),$$

$$(5.57) \quad \text{with } J_i = \alpha_i I,$$

where the parameter  $k$  is called 'PEPcase specificity',  $V_{p,max}$  is the C4-plant parameter for the maximum electron rate, and  $I$  is the light intensity from (5.30).

As for C3-plants, dark respiration for C4-plants is assumed to be proportional to the maximum carboxylation rate:

$$(5.58) \quad r_{d,0} = \gamma_d V_{p,max,0}.$$

The temperature dependence is the same as for C3-plants with the same activation energy (see eq. (5.37)), i.e. the parameters  $k$  and  $V_{p,max}$  in (5.55) and (5.57) follow the usual Arrhenius law

$$(5.59) \quad \begin{aligned} k &= k_0 \exp\left(\left(\frac{T}{298K} - 1\right) \frac{E_K}{RT}\right) \\ V_{p,max} &= f(T) V_{p,max,0} \exp\left(\left(\frac{T}{298K} - 1\right) \frac{E_V}{RT}\right), \end{aligned}$$

with the associated activation potentials  $E_K$  and  $E_V$  from table 5.3. In addition, as for C3 plants, nitrogen scaling (5.39) is applied to arrive at values specific for each canopy layer. For the parameters  $k_0$  and  $V_{p,max,0}$  see table 5.4.

The strategy to determine the assimilation rate is the same as described above for C3-plants: First, in absence of water stress a fixed relation between ambient and leaf internal  $\text{CO}_2$  concentration is assumed, namely [75, p. 41]:

$$(5.60) \quad c_i \approx 0.67 c_a.$$

From this the non-stressed stomatal conductance (5.46) is computed. Next, to account for water stress, the stressed canopy conductance (5.50) instead of the unstressed is used in (5.44) to solve for the assimilation rate  $A$ . Using the resulting equation together with (5.55) and

(5.57) in (5.26) one finds after elimination of  $c_i$

$$(5.61) \quad J_{C,stress} = k \frac{g_0 c_a + r_d}{g_0 + k},$$

$$(5.62) \quad J_{E,stress} = J_E,$$

where as above the stressed stomatal conductance enters via the shorthand (5.53). Performing these calculations for each canopy layer, by entering  $J_{C,stress}$ ,  $J_{E,stress}$ , and  $r_d$  into (5.26) gives the assimilation rate for each layer in the canopy. Summing the values for all layers, then finally gives the canopy assimilation rate  $A_{stress}$  of plant productivity per leaf area that gives via (6.2) the carbon input to the land carbon cycle.

### 5.3.3 Implementation details

The canopy radiation and photosynthesis models are distributed across the files

`mo_bethy.f90`: Main module of the the canopy radiation model and the photosynthesis model. Main entry routine is `update_bethy()`.

`mo_bethy_photosy.f90`: Contains the photosynthesis model (main routine `photosyn()`).

`mo_bethy_constants.f90`: Contains constants and parameters.

PFT specific parameters  $V_{max,0}$ ,  $J_{max,0}$  (C3 plants) and  $V_{p,max,0}$ ,  $k_0$  (C4 plants) are read in from `lctlib.def` (see appendix C.3) under the keywords `CarboxRate` and `ETransport`, respectively. Parameters  $J_{max,0}$  and  $k_0$  are read in under the same keyword, which is a source of confusion, since not only their meaning is different (electron transport rate vs. PEP-case specificity), but also their units are different; in `lctlib.def`  $J_{max,0}$  must be given in [ $\mu\text{mol}(\text{CO}_2)/\text{m}^2\text{s}$ ] whereas  $k_0$  is expected in [ $\text{mmol}(\text{air})/\text{m}^2\text{s}$ ] (compare table 5.4).

The routine `update_bethy()` is called from the JSBACH main routine `jsbach_inter_1d()` each time step twice: First to compute the unstressed canopy conductance, and second to compute from the stressed canopy conductance the assimilation rate in the presence of water stress. The logical input parameter `waterLimitationFlag` is used to tell the routine which of the two modes (“unstressed” or “stressed”) is meant: At the first call `waterLimitationFlag` must have the value `.false.`, at the second call `.true.`. The water limitation flag is passed to the photosynthesis routine `photosyn()` where the actual computations take place. The photosynthesis routine is called separately for each canopy layer. All computations are done per  $\text{m}^2(\text{leaf})$  and thus must be multiplied with leaf area index to obtain values per  $\text{m}^2(\text{canopy})$ . There is no model implemented to compute canopy temperature, instead air temperature is used.

Tables 5.5 and 5.6 list the interfaces of the major routines `update_bethy()` and `photosyn()` together with the symbols used in this documentation.

Table 5.5: Interface of routine `update_bethy()`

name in code	symbol	intent	units	meaning
<code>waterLimitationFlag</code>	–	in	–	switch for the photosynthesis scheme: water stressed (true) or water unstressed (false) case
<code>canopy_conductance</code>	$g_C^{H_2O}$	out	$m/s$	Canopy conductance in absence of water stress (output only for <code>waterLimitationFlag=.false</code> ).
<code>canopy_conductance_limited</code>	$g_{C, stress}^{H_2O}$	in	$m/s$	canopy conductance taking water stress into account (used only for <code>waterLimitationFlag=.true</code> ).
Inputs for canopy radiation model				
<code>lai</code>	$\Lambda$	in	–	leaf area index
<code>par</code>	$R(0)+R_{\downarrow}(0)$	in	$\frac{W}{m^2(ground)}$	PAR: photosynthetically active radiation (400-700nm)
<code>frac_par_direct</code>	$\frac{R(0)}{R(0)+R_{\downarrow}(0)}$	in	–	fraction of direct radiation in PAR
<code>soil_albedo</code>	$\alpha_{soil, vis}$	in	–	albedo of soil below vegetation canopy
<code>useAlbedo</code>	–	in	–	flag for choice of albedo scheme to be used: interactive albedo (true) or ECHAM-scheme (false)
Inputs for photosynthesis model				
<code>cover_type</code>	–	in	–	land cover type - specifying the PFTs
<code>lctlib</code>	–	in	–	structure with PFT specific parameters from <code>lctlib.def</code>
<code>pressure</code>	$p$	in	$N/m^2$	atmospheric surface pressure
<code>canopy_temp</code>	$T$	in	$K$	temperature in vegetation canopy
<code>CO2_concentration_air</code>	$c_a$	in	$\frac{kg(CO_2)}{kg(air)}$	CO <sub>2</sub> concentration in lowest atmospheric layer(mass mixing ratio)
Other technical information				
<code>kidx</code>	–	in	–	number of grid points to be handled in this call
<code>domain</code>	–	in	–	structure containing domain information (for parallelization)
<code>mask</code>	–	in	–	logical mask indicating the presence/absence of vegetation
<code>bethy</code>	–	inout	–	structure holding diverse up-to-date information on canopy radiation and photosynthesis

Table 5.6: Interface of the routine `photosyn()` (the photosynthesis model).

name in code	symbol	intent	units	meaning
<code>waterLimitationFlag</code>	–	in	–	switch for water stressed (true) or unstressed (false) case
<code>C4flag</code>	–	in	–	switch for photosynthetic pathway: C4 (true) and C3 (false)
<code>GS</code>	$g_C^{H_2O}$ , $g_C^{H_2O}$ , $g_{C,stress}$	inout	$m/s$	stomatal conductance of water vapor: “out” for water stressed, and “in” for water unstressed case
<code>CI</code>	$c_i$	inout	$\frac{mol(CO_2)}{mol(air)}$	CO <sub>2</sub> concentration inside a leaf: “in” for unstressed, and “out” for stressed case
<code>PAR</code>	$(I(l_1) - I(l_2))/\Lambda$	in	$\frac{mol(photons)}{m^2(leaf)s}$	PAR absorbed in canopy layer per leaf area
<code>TC</code>	–	in	°C	canopy temperature
<code>P</code>	$p$	in	$N/m^2$	atmospheric surface pressure
<code>atm_co2_conc</code>	$c_a$	in	$\frac{mol(CO_2)}{mol(air)}$	CO <sub>2</sub> concentration in lowest atmospheric layer (volume mixing ratio)
<code>PAR_DOWN</code>	$R(0)+R_{\downarrow}(0)$	in	$\frac{mol(photons)}{m^2(ground)s}$	downward PAR, used for dark inhibition only
Output computed only for <code>waterLimitationFlag=.true.</code>				
<code>gross_assimilation</code>	$A_{stress}$	out	$\frac{mol(CO_2)}{m^2(leaf)s}$	gross assimilation (including photorespiration), computed only for water stressed case
<code>dark_respiration</code>	$r_d$	out	$\frac{mol(CO_2)}{m^2(leaf)s}$	dark respiration, computed only for water stressed case
<code>max_carbox_rate</code>	$V_{max}$	out	$\frac{\mu mol(CO_2)}{m^2s}$	maximum carboxylation rate
<code>max_e_transport_rate</code>	$J_{max}$	out	$\frac{mol(CO_2)}{m^2(leaf)s}$	Only C3 plants: maximum electron transport rate
<code>carbox_rate</code>	$J_C$	in	$\frac{\mu mol(CO_2)}{m^2s}$	actual carboxylation rate
<code>e_transport_rate</code>	$J_E$	in	$\frac{mol(CO_2)}{m^2(leaf)s}$	actual electron transport rate
Photosynthesis parameters				
<code>CarboxRate</code>	$V_{max,0}$ , $V_{p,max,0}$	in	$\frac{\mu mol(CO_2)}{m^2(leaf)s}$	maximum carboxylation rate at 25 degrees Celsius (see <code>lctlib</code> )
<code>ETransport</code>	$J_{max,0}$ , $k_0$	in	$\frac{\mu mol(CO_2)}{m^2(leaf)s}$ , $\frac{mmol(air)}{m^2(leaf)s}$	maximum rate of electron transport (C3) or PEPcase specificity (C4) at 25 degrees Celsius (see <code>lctlib</code> )
<code>NSCL</code>	$e^{-k_{noon}l}$	in	–	nitrogen scaling factor (see eq. (5.39))
Other technical information				
<code>kidx</code>	–	in	–	number of grid points in this call
<code>VegetationFlag</code>	–	in	–	mask for presence/absence of vegetation

## 5.4 Diagnostics

This section lists in table 5.7 some output variables that are useful to diagnose the results from the canopy radiation and photosynthesis model. Numerous other diagnostics exist (see the routine `bethy_init_memory()` in `mo_bethy.f90`), but are not written to output in the standard configuration of JSBACH. This can be easily changed in `bethy_init_memory()` by changing `lpost=.FALSE.` to `lpost=.TRUE.` for the respective variable.

Table 5.7: Some diagnostic variables from the canopy radiation and photosynthesis model. All these outputs are averages over the output interval. Note that  $c_i^{can}$  in this table is the gap corrected cover fraction and has nothing to do with leaf internal  $\text{CO}_2$  concentration (compare eq. (1.12)).

name in output	symbol	units	meaning
Output for the canopy on tile level ( <code>jsbach-stream</code> )			
<code>apar_acc</code>	$I = I_{tot}(\Lambda) - I_{tot}(0)$	$\frac{\text{mol}(\text{photons})}{\text{m}^2(\text{canopy})\text{s}}$	PAR absorbed in the canopy
<code>gross_assimilation_acc</code>	$A + r_d$	$\frac{\text{mol}(\text{CO}_2)}{\text{m}^2(\text{canopy})\text{s}}$	total carbon assimilation, including the amount needed for dark respiration
<code>net_assimilation_acc</code>	$A$	$\frac{\text{mol}(\text{CO}_2)}{\text{m}^2(\text{canopy})\text{s}}$	carbon assimilation taking into account the loss by dark respiration
<code>canopy_conductance</code>	$g_{C, stress}^{H_2O}$	$\text{m/s}$	canopy conductance
Output for the total grid box ( <code>land-stream</code> )			
<code>par_acc</code>	$R(0) + R_{\downarrow}(0)$	$\frac{\text{mol}(\text{photons})}{\text{m}^2(\text{box})\text{s}}$	incoming PAR
<code>apar_acc</code>	$(I_{tot}(\Lambda) - I_{tot}(0)) * veg_{max} * c_i^{can}$	$\frac{\text{mol}(\text{photons})}{\text{m}^2(\text{box})\text{s}}$	PAR absorbed in the canopy
<code>gross_assimilation</code>	$(A + r_d) * veg_{max} * c_i^{can}$	$\frac{\text{mol}(\text{CO}_2)}{\text{m}^2(\text{box})\text{s}}$	total carbon assimilation, including the amount needed for dark respiration
<code>net_assimilation</code>	$A * veg_{max} * c_i^{can}$	$\frac{\text{mol}(\text{CO}_2)}{\text{m}^2(\text{box})\text{s}}$	carbon assimilation taking into account the loss by dark respiration



# Chapter 6

## Natural carbon cycle

### Contents

---

6.1	Introduction . . . . .	85
6.2	From GPP to NPP . . . . .	86
6.3	Dynamics of vegetation carbon . . . . .	89
6.3.1	Distribution of NPP across vegetation pools . . . . .	89
6.3.2	Carbon losses of vegetation (grazing, litter production, crop harvest) . . . . .	91
6.4	Dynamics of litter and soil carbon in Yasso . . . . .	93
6.4.1	Dynamic equations for Yasso pools . . . . .	94
6.4.2	Carbon inputs from vegetation . . . . .	95
6.4.3	Decomposition fluxes of the litter pools . . . . .	96
6.4.4	Gains and losses of the humus pool . . . . .	97
6.4.5	Heterotrophic carbon losses to atmosphere . . . . .	97
6.5	Dynamics of litter and soil carbon in CBALANCE . . . . .	98
6.5.1	Dynamics of green litter . . . . .	98
6.5.2	Dynamics of wood litter . . . . .	99
6.5.3	Dynamics of the slow pool . . . . .	99
6.5.4	C-loss to atmosphere . . . . .	100
6.6	Diagnostics . . . . .	100
6.7	Implementation details . . . . .	100
6.7.1	Code structure . . . . .	100
6.7.2	Handling of CO <sub>2</sub> exchange with atmosphere . . . . .	102
6.7.3	Model parameters . . . . .	102

---

### 6.1 Introduction

This chapter describes the representation of the *natural* land carbon cycle in JSBACH, i.e. the processing of carbon since its uptake by photosynthesis from the atmosphere (chapter 5) via its temporal storage in vegetation and soils until it is released once more to the atmosphere by heterotrophic respiration. This chapter does not cover wildfires and windthrow that also affect the natural land carbon cycle – because of the importance of these processes for natural landcover change these aspects are described in the respective chapter 9. Also

anthropogenic interference with the land carbon cycle is postponed until chapter 10 describing how anthropogenic landcover change is modeled. The carbon aspects of those other processes not covered here all rely on the model for land carbon cycling described in the present chapter.

Whereas JSBACH contains a single representation of the carbon dynamics of living vegetation (for short: “vegetation carbon”), it contains two alternative submodels for carbon in plant residues and soils (for short: “soil carbon”).<sup>1</sup> CBALANCE is the original land carbon cycle model of JSBACH covering vegetation and soil carbon. CBALANCE was used until recently as standard JSBACH carbon cycle model. Since version 3.0 (the version described here) the CBALANCE component for soil carbon is replaced by the Yasso model as part of the standard configuration. Yasso was originally developed by Liski et al. [86], first implemented into JSBACH by Thum et al. [134] and later on made a standard component of JSBACH by Goll et al. [52]. For consistency with earlier versions, the full CBALANCE model is still shipped with the JSBACH 3 release and will thus be described here. But when using JSBACH with the soil component of CBALANCE instead of Yasso, one should be aware that this component isn’t any more maintained and accordingly is lacking quality assurance by the JSBACH development team.

Although mathematically the two models for soil carbon are quite similar (systems of first order differential equations), their development targets have been very different. In CBALANCE the pool structure for soil carbon is chosen mainly to represent the characteristic time scales of litter and soils to obtain a reasonably realistic seasonal cycle of atmospheric CO<sub>2</sub> and a realistic increase of atmospheric CO<sub>2</sub> during the 20<sup>th</sup> century. In contrast, Yasso was designed to obtain realistic heterotrophic respiration rates in different climates. To achieve this goal, a huge set of observed litter decomposition rates measured in “litter bag” experiments worldwide have been used to determine the parameters of the Yasso model. Accordingly, Yasso results in a much more realistic distribution of worldwide soil carbon than could be obtained with CBALANCE as demonstrated in [52]. As a consequence, the seasonal cycle of atmospheric CO<sub>2</sub> is with Yasso an emerging feature that is not any more tunable by turnover rates of soil carbon. Since turnover time for the very slowly decomposed soil carbon “humus pool” of Yasso (multi-centennial) is not well constrained by the field data underlying Yasso, it is still a suitable tuning parameter for 20th century rise of CO<sub>2</sub>, but this option has not yet been explored. Overall, the main reason to introduce Yasso as the new standard model for soil carbon is its strong foundation on observational data with good coverage across all climates.

The state variables of the CBALANCE and the Yasso carbon models are the carbon content of their different carbon pools. This pool structure is illustrated in figure 6.1. Carbon pools are defined for each tile<sup>2</sup> (see sec. 1.3.2) and are measured in units mol(C)/m<sup>2</sup>(canopy). To obtain e.g. a global total one must thus first scale the values according to Eq. (1.11) to values per m<sup>2</sup>(ground).

## 6.2 From GPP to NPP

The land biosphere persists and grows by utilizing the gain of organic carbon (sugar) from photosynthesis. The rate at which organic carbon is provided by photosynthesis is called *GPP* (gross primary productivity), e.g. measured in [mol(C)/m<sup>2</sup>(ground) s]. Only part of this gain is available for investment into growth. Other parts must as well be used to maintain basic functionalities like photosynthesis, nutrient and water transport, repairs, or defence. These

<sup>1</sup>Which of the submodels is used in simulation can be selected by the logical `with_yasso` namelist parameter.

<sup>2</sup>Except the anthropogenic pools (sec. 10.3.3) which are associated with a grid box as a whole.

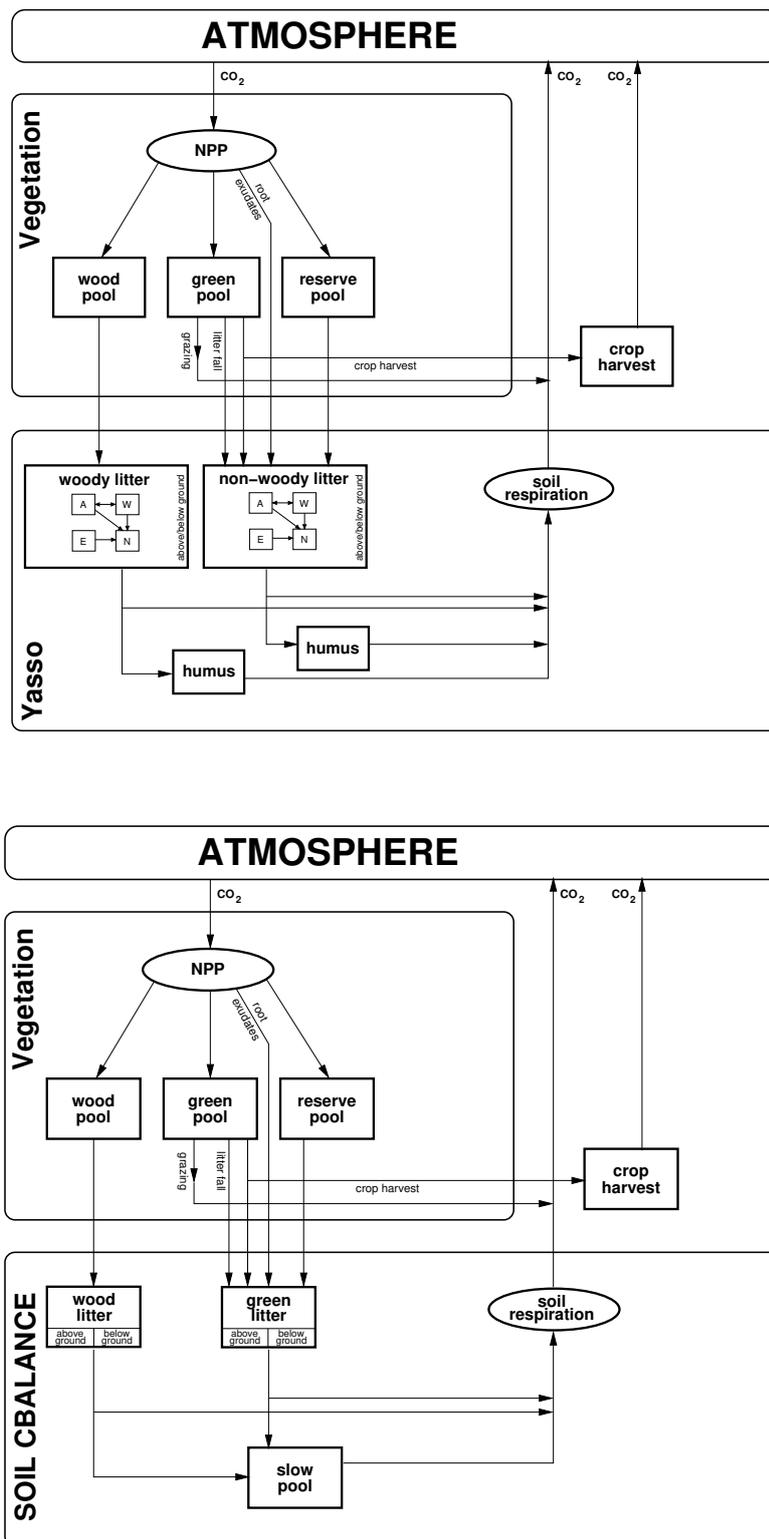


Figure 6.1: Structure of carbon pools using either the Yasso (top) or the CBALANCE (bottom) soil carbon models

investments are called collectively *maintenance respiration* and will be denoted by  $R_m$ . But also to perform growth photosynthesates are needed; this part of the plant carbon budget is called *growth respiration* (symbol  $R_g$ ). The rate at which sugars are actually available for growth, called *NPP* (net primary productivity), is therefore<sup>3</sup>

$$(6.1) \quad NPP = GPP - R_g - R_m.$$

To compute *NPP* first *GPP* is needed. This is obtained from the carbon assimilation  $A_{stress}$  determined in the photosynthesis component of JSBACH accounting for water stress (compare equ. (5.54)). There  $A_{stress}$  is computed as carbon gain *per leaf area*. Accordingly, whole canopy productivity *per ground area* is obtained by multiplication with the leaf area index  $\Lambda$  obtained from the phenology model (see equ. (3.1))<sup>4</sup>

$$(6.2) \quad GPP = \Lambda A_{stress}$$

Next  $R_g$  and  $R_m$  must be determined. Together, these two fluxes are called *autotrophic respiration*. Autotrophic respiration depends on many aspects of the plants, e.g. on their growth stage and different environmental stresses, but in particular on the size of plant organs (roots, stems, leaves, fruits, ..). Nevertheless, in the present version of JSBACH, the simplified *NPP*-scheme of the BETHY model [75, 76] is implemented, which is based on an upscaling of maintenance respiration of leaves, called *dark respiration*, to the whole canopy, assuming average investment conditions across all vegetation types. Dark respiration  $r_d$  is computed *per leaf area* as part of the Farquhar and Collatz models of photosynthesis (compare equ. (5.37) as well as (5.31) and (5.58)). Accordingly,<sup>5</sup>

$$(6.3) \quad R_d = \Lambda r_d$$

is the dark respiration *per ground area*. It is assumed that  $R_d$  is a fixed fraction  $f_{leaf}$  of the whole canopy maintenance respiration so that

$$(6.4) \quad R_m = \frac{R_d}{f_{leaf}}.$$

For computing *NPP* from (6.1) it remains to determine  $R_g$ . Let  $CC$  denote the total construction costs for one unit of allocated carbon. Since total carbon invested into growth is  $NPP + R_g$  but the finally allocated carbon is only  $NPP$  one obtains that  $CC = (NPP + R_g)/NPP$ . Hence, including the case of no growth (where  $NPP$  is negative) one finds

$$(6.5) \quad R_g = \begin{cases} (CC - 1)NPP & \text{for } NPP > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Inserting this into (6.1) and solving for *NPP* finally gives

$$(6.6) \quad NPP = (GPP - R_m) \begin{cases} CC^{-1} & \text{for } GPP > R_m \\ 1 & \text{otherwise.} \end{cases}$$

After additional corrections (see section 6.3.1), this *NPP* is distributed across the different vegetation carbon pools as described below and subtracted from the atmospheric  $\text{CO}_2$ .<sup>6</sup>

<sup>3</sup>What is called here *NPP* will be called below potential *NPP* (denoted by  $NPP^{pot}$ ) in contrast to actual *NPP* that accounts for structural allocation limits (e.g. maximum forest density expressed as carbon density, see section 6.3.1) and for lack of nutrients (see section 7.3).

<sup>4</sup>More precisely:  $A_{stress}$  is computed in JSBACH for each canopy layer separately. Accordingly, *GPP* is obtained by computing (6.2) separately for each canopy layer and final summation over all layers.

<sup>5</sup>As for *GPP* also  $R_d$  is obtained from the individual contributions of all canopy layers.

<sup>6</sup> $\text{CO}_2$ -exchange with the atmosphere happens every time step, carbon allocation at land only once a day. Accordingly, *NPP* is summed over a day and allocated only then. This poses a problem when structural vegetation limits for carbon allocation are hit, so that corrections to the  $\text{CO}_2$ -exchange with the atmosphere get necessary. See section 6.7.2 for more details.

## 6.3 Dynamics of vegetation carbon

Independent of the soil carbon model, the same three pools are used to describe the storage of organic carbon in the living plant:

**Green pool** ( $C_G$ ): Contains carbon of the “green” or living parts of plants (leaves, fine roots, vascular tissues), except carbon stored as reserve (see below).

**Wood pool** ( $C_W$ ): Contains the carbon of the woody parts of plants (stems, branches, roots).

**Reserve pool** ( $C_R$ ): Contains the carbon stored in sugars and starches that the plants keep as an energy reserve.

For reasons related to nitrogen cycling there is one additional carbon pool always present:

**Crop harvest pool** ( $C_{CH}$ ): Contains carbon from that part of crop harvest to be consumed by people or animals.

The following discussion concentrates on the three vegetation pools, while the dynamics of the crop harvest pool is presented later in section 6.3.2. The dynamics of the three vegetation pools is described by<sup>7</sup>

$$(6.7) \quad \frac{dC_G}{dt} = \eta_{Nlim} NPP_{\triangleright G} - F_{litter} - F_{grazing}$$

$$(6.8) \quad \frac{dC_W}{dt} = \eta_{Nlim} NPP_{\triangleright W} - \frac{C_W}{\tau_W}$$

$$(6.9) \quad \frac{dC_R}{dt} = NPP_{\triangleright R} - \frac{C_R}{\tau_R},$$

where  $NPP_{\triangleright G}$ ,  $NPP_{\triangleright W}$ , and  $NPP_{\triangleright R}$  are those parts of  $NPP$  that enter the respective pools, although reduced by  $\eta_{Nlim} \in [0, 1]$  which accounts for a possible limitation in nitrogen availability,<sup>8</sup>  $F_{litter}$  and  $F_{grazing}$  are carbon losses due to litter production and grazing, and  $\tau_W$  and  $\tau_R$  are the carbon turnover times of the respective pools.<sup>9</sup> The rest of this section describes how all these determinants of the dynamics are derived. These quantities are described separately in the subsections below. It should be noted that vegetation carbon is also lost by disturbances (fire, windthrow) and human use. These aspects of the dynamics of vegetation carbon are treated separately in sections 9.4 and chapter 10.

### 6.3.1 Distribution of NPP across vegetation pools

Vegetation grows when photosynthetic productivity is larger than respiration from metabolic needs, and shrinks otherwise. In the first case  $NPP$  is positive and the vegetation stores carbon (allocation), while in the second case  $NPP$  is negative and it loses carbon (deallocation). In this section it is described how this allocation and deallocations happens in JSBACH by calculating the contribution of  $NPP$  – being negative or positive – to the various vegetation pools.

<sup>7</sup>Numerically, equations (6.7) to (6.9) are solved by an Euler discretization using a time step of one day – this is sufficient for a stable numerics since the dynamics is overdamped. Moreover the Euler discretization has the advantage that carbon conservation is guaranteed to computational accuracy. But despite the 1 day time step, the resulting  $\text{CO}_2$  exchange with the atmosphere happens at the basic sub-diurnal time step of the overall JSBACH, meaning that  $\text{CO}_2$  exchange rates during a particular day are those that were computed for the day before. This time lag has no important consequences for the carbon cycle.

<sup>8</sup>See chapter 7 and in particular eq. (7.22). In the absence of nitrogen limitation  $\eta_{Nlim} = 1$ , which is especially the situation when JSBACH is run with nitrogen cycling switched off.

<sup>9</sup>Note that the same turnover time  $\tau_W$  is used in DYNVEG to describe the shrinkage of vegetation extent of woody types (see section 9.3).

Table 6.1: Typical values for allometric coefficients used to split NPP; partly adapted from [146]. For the actual values used in JSBACH please check the entries in the land cover type library file in appendix C.3.

	trees	shrubs	grasses	crops
$f_{NPP \triangleright G}$	0.50	0.50	0.70	0.70
$f_{NPP \triangleright R}$	0.15	0.15	0.25	0.25
$f_{NPP \triangleright W}$	0.30	0.30	0	0
$f_{NPP \triangleright exud}$	0.05	0.05	0.05	0.05

Since the calculation of  $NPP$  as obtained in the previous section from eq. (6.6) is independent of the amount of carbon already stored in the vegetation, this kind of  $NPP$  allocation does neither account for structural allocation limits (trees cannot grow arbitrarily large or thick) nor for lack of resources to survive starvation periods. The JSBACH carbon allocation scheme accounts for such situations in a very rough way by introducing a post hoc correction to the  $NPP$  calculated by eq. (6.6). This makes it necessary to distinguish between *potential* and *actual*  $NPP$ , the former (called  $NPP^{pot}$ ) given by eq. (6.6), the latter (called  $NPP^{act}$ ) being the corrected one. The term 'actual' is used here to indicate that  $NPP^{act}$  is the carbon flux from living vegetation that is actually exchanged with the atmosphere (see section 6.7.2). – Note that  $NPP^{act}$  may get further reduced by lack of nutrients (see sec. 7.3).

First the case of positive  $NPP^{pot}$  is considered. Here  $NPP^{pot}$  is distributed to the vegetation pools (Green, Wood, Reserve) and partially added as root exudates directly to soil carbon pools at fixed PFT-dependent fractions (for typical values of these allometric coefficients see table 6.1):

$$(6.10) \quad NPP_{\triangleright X} = f_{NPP \triangleright X} NPP^{pot}, \quad X \in \{G, W, R\}$$

$$(6.11) \quad F_{exud} = f_{NPP \triangleright exud} NPP^{pot},$$

where the allometric coefficients fulfill  $\sum_{X \in \{G, W, R, exud\}} f_{NPP \triangleright X} = 1$ , such that all of the  $NPP$  is distributed:

$$(6.12) \quad NPP = NPP_{\triangleright G} + NPP_{\triangleright W} + NPP_{\triangleright R} + F_{exud}.$$

Modeling growth in this way does not account for structural limits to carbon allocation arising e.g. from mechanical stability limits to tree height or from mutual spatial exclusion. Such limits are roughly accounted for in JSBACH by assuming

- ★ a PFT-dependent maximum carbon content  $C_W^{max}$  of the wood pool,
- ★ a maximum carbon content  $C_G^{max}$  of the green pool proportional to the current leaf area index  $\Lambda(t)$ , and
- ★ an optimum value  $C_R^{opt}$  of the reserve pool proportional to maximum leaf area index  $\Lambda_{max}$ .

More precisely,

$$(6.13) \quad C_G^{max}(t) = \gamma_G \frac{\Lambda(t)}{sla},$$

$$(6.14) \quad C_R^{opt} = \gamma_R \frac{\Lambda_{max}}{sla},$$

where  $sla$  is the specific leaf area, i.e. the leaf area per mol leaf carbon, and  $\gamma_G$  and  $\gamma_R$  are dimensionless parameters larger than one. By using  $sla$  in these formulas the limiting values

$C_G^{max}$  and  $C_R^{opt}$  are related to the leaf carbon content. Accordingly,  $\gamma_G$  has the meaning of maximum total carbon in living tissues relative to leaf carbon, and similarly  $\gamma_R$  specifies the optimum storage of reserve carbon in relation to the maximum carbon content in leaves. Note that  $C_R^{opt}$  is fully determined from fixed PFT specific parameters although in reality the capacity for storing carbon reserves may vary e.g. seasonally [95].<sup>10</sup>

These limits are invoked in the dynamics of carbon allocation as follows. If  $C_W + NPP_{\triangleright W} > C_W^{max}$  or  $C_R + NPP_{\triangleright R} > C_R^{opt}$  the remaining carbon is added to  $NPP_{\triangleright G}$ . If, using this modified  $NPP_{\triangleright G}$ ,  $C_G + NPP_{\triangleright G}$  would be larger than  $C_G^{max}$  the remaining carbon cannot be allocated. This possibility of too much productivity in view of structural limits makes it necessary to distinguish between *potential* and *actual*  $NPP$ , denoted as  $NPP^{pot}$  and  $NPP^{act}$ : Actual  $NPP$  is potential  $NPP$  reduced by the surplus carbon that cannot be stored:<sup>11</sup>

$$(6.15) \quad NPP^{act} = \begin{cases} NPP^{pot} & \text{for } C_G^{max} > C_G \\ NPP^{pot} - (C_G - C_G^{max})/\Delta T & \text{otherwise,} \end{cases}$$

where  $NPP^{pot}$  is the  $NPP$  from eq. (6.1) and  $\Delta T$  is the update time step of the carbon allocation model (= 1 day).

It remains to consider the situation when  $NPP^{pot}$  is negative (which may happen e.g. due to a cold spell at already well developed  $LAI$ ). In this case there can be no allocation to the wood and green pools, i.e.  $NPP_{\triangleright W} = NPP_{\triangleright G} = 0$  and the reserve pool has to carry the full burden of respiratory needs of the plants. In some cases (parameters are chosen to prevent this for most of the time) it may happen that the reserve pool cannot fully compensate for the  $NPP$  deficit. This is the above mentioned starvation case, where a post hoc correction of potential  $NPP$  is necessary. Hence for negative  $NPP^{pot}$  two cases must be considered for the calculation of actual  $NPP$ :

$$(6.16) \quad NPP^{act} = \begin{cases} NPP^{pot} & \text{for } C_R/\Delta T > |NPP^{pot}| \\ -C_R/\Delta T & \text{otherwise.} \end{cases}$$

In the first case, the reserve carbon can compensate for the negative  $NPP$ , while in the second case there is not enough reserve carbon for full compensation and actual  $NPP$  is set to the maximum compensation possible.<sup>12</sup> As a consequence, in both cases  $NPP_{\triangleright R} = NPP^{act}$ , which represents a deallocation of reserve carbon since  $NPP^{act}$  is negative.

### 6.3.2 Carbon losses of vegetation (grazing, litter production, crop harvest)

The green pool loses carbon in three ways, by grazing, litter production (compare equation (6.7)), and crop harvest, while the wood and reserve pool are assumed to lose carbon by litter production only.

<sup>10</sup>It would not make sense to use  $\Lambda(t)$  instead of  $\Lambda_{max}$  in (6.14) because storage carbon must be kept across winter or dry seasons to be available for initiating growth in the next growing season.

<sup>11</sup>The biologist Körner is repeatedly criticizing global vegetation modelers (see e.g. [44]) for wrongly assuming that carbon allocation is driven by photosynthesis. He claims that plant growth is not driven by allocation *supply* but allocation *demand* because the limited ability of plant organs to store the photosynthates leads upon an overproduction (supply > demand) to a breakdown of photosynthesis because the sugar transport away from the production sites stagnates [130, see section 13.7.2 by C. Körner]. This is mimicked in JSBACH by introducing the mentioned allocation limits. Such an approach is also consistent with the observation of “idle respiration” by Chambers et al. [19] who found extremely low carbon use efficiencies in tropical forests, wasting sugars by overly high autotrophic respiration.

<sup>12</sup>This additional post hoc modification of  $NPP$  is important for carbon conservation when running JSBACH as part of the MPI Earth System model since carbon allocation happens only once a day, but  $\text{CO}_2$ -exchange with atmosphere happens every time step. See section 6.7.2 for more details.

The loss of vegetation carbon by grazing of herbivores ( $F_{grazing}$ ) can be quite substantial, especially in grasslands.<sup>13</sup> Here it is assumed that the amount grazed is simply proportional to the available green carbon with a PFT-dependent constant of proportionality:

$$(6.17) \quad F_{grazing} = r_{graz} C_G.$$

The second way the green pool loses carbon is by litter production ( $F_{litter}$ ), i.e. shedding of leaves and other living tissues including sapwood and fine roots. It is clear that for individual plants litter production is related to the seasonal dynamics of the *LAI*: A reduction of *LAI* means loss of leaves, i.e. litter production. And since the amount of fine roots and sapwood changes parallel with *LAI*, the *LAI* is as well a good indicator for production of other than only leaf litter. But in a vegetation model as JSBACH the *LAI* represents not the *LAI* of an individual plant but an average *LAI* across whole landscapes. Hence, in large parts of the tropics, where the model *LAI* is almost constant, a litter production model based simply on the decrease of the *LAI* would fail. Accordingly, a second aspect of landscape-wide litter production must be incorporated, namely leaf aging (leaf longevity). This results in a continuous loss of vegetation carbon independent of seasonality. Both aspects are covered by the following expression for the litter production term:

$$(6.18) \quad F_{litter} = \frac{\gamma_G}{sla} \max(r_{shed}\Lambda(t), -\frac{d\Lambda}{dt}).$$

The first term in the maximum function represents a permanent minimum loss rate of vegetation carbon by leaf aging. It is made proportional to the *LAI* (symbol  $\Lambda$ ) because one must expect larger productivity and thus larger carbon turnover for vegetation with larger *LAI*;  $\Lambda(t)$  is computed by the phenology model (see chapter 3). The PFT-dependent constant  $r_{shed}$  is the inverse leaf longevity. The second term in the maximum function describes litter production upon a decrease in *LAI*. When the reduction in  $\Lambda$  (as measured by  $d\Lambda/dt$ ) is larger than the permanent litter loss, it is this *LAI* reduction that determines the litter loss rate. The “LAI-loss” is converted to carbon loss by division with specific leaf area  $sla$ , and the multiplication with the constant  $\gamma_G$  scales leaf carbon loss to the carbon loss of the green pool as a whole.

Finally, for crops, there is a third way how the green pool loses carbon, namely by crop harvest. Actually, this is modeled as litter production, only that for crops  $F_{litter}$  from eq. (6.18) is interpreted as harvest and that the fate of this harvest is different from the litter of other vegetation types: while for them the all the litter carbon is put into the soil and litter pools of the CBALANCE or Yasso models, for crops part of it is put with a time delay as CO<sub>2</sub> into the atmosphere. This part of the harvest is meant to represent consumption of crop products (in particular as food) by humans and livestock, while the remaining part of the harvest is considered to be organic waste. More precisely, the crop harvest entering food consumption is computed as

$$(6.19) \quad F_{food} = f_G^\perp f_{food} F_{litter},$$

<sup>13</sup>Several studies have investigated the fraction of NPP which is consumed by herbivores in grasslands, savannas, shrublands, and forests. For example, Conant [29] reports that grazing animals remove in grasslands annual NPP in the range of 30–50%, occasionally up to 80%. Other studies note that in this ecosystem, herbivorous consumption is in the range of about 25–50% of aboveground NPP [91, 92, 83] and 25% of belowground productivity [26, 68, 83]. Milchunas and Lauenroth [83] suggest based on their data that grazing intensity for grassland, shrub lands, and forest ecosystems is 44%, 55%, and 60% of aboveground NPP, respectively. In addition, Cebrian [18] compiled an extensive data set (> 200 published reports) on the percentage of annual NPP consumed by herbivores in grassland, forest, and shrub lands ecosystems. He found that in grassland ecosystems, 0–60% (mean ca. 30%) of annual NPP is consumed by herbivory, whereas in forest and shrub lands the consumption is 0–10% (mean ca. 3%). Other studies report that that in forest ecosystems, less than 10% of annual NPP is consumed by grazers [119, 123].

where  $f_{food}$  is the fraction of crop harvest (i.e. of  $F_{litter}$  from crops) entering crop products, and  $f_G^+$  is the above ground fraction of the carbon in the green pool – it is assumed that crop products are obtained only from above ground biomass. This “food” flux is not put into the atmosphere directly but only after being intermediately stored in a *crop harvest pool* denoted by  $C_{CH}$  whose dynamics is described by

$$(6.20) \quad \frac{dC_{CH}}{dt} = F_{food} - F_{CH \triangleright A} \quad \text{with} \quad F_{CH \triangleright A} = \frac{C_{CH}}{\tau_{CH}},$$

where  $F_{CH \triangleright A}$  is the “food” consumption flux to the atmosphere obtained from an exponential turnover with time constant  $\tau_{CH}$  of the order of one year. This delay is not so much introduced because consumption and subsequent CO<sub>2</sub> emissions are delayed, but to mimick in a simple way fertilizer application; see section 7.4.4 for details.

So far only losses from the green pool have been considered. But also the wood and reserve pool must produce litter because of aging. This is modeled in (6.8) and (6.9) simply by assuming a typical time scale for carbon loss. The turnover time of wood ( $\tau_W$ ) should be similar to the mean age of trees, while the turnover time of the reserve pool ( $\tau_R$ ) is that of sugars and starches in absence of usage by the plants themselves. This must be much smaller than  $\tau_W$  because of their environmental instability upon death of plants. Accordingly, the PFT-dependent value of  $\tau_W$  is chosen on the order of decades and  $\tau_R$  is taken for all PFTs as 1 year. Generally, because of its small pool size, the turnover of the reserve pool is not very important for the carbon cycle as a whole so that the exact choice of its turnover time is not critical. This is different for  $\tau_W$  that is an important control of land carbon turnover at decadal time scales.

## 6.4 Dynamics of litter and soil carbon in Yasso

The present implementation of the Yasso litter and soil decomposition model is based on the Yasso07 model [137] which is a further development of the original Yasso model [86]. Yasso07 distinguishes five carbon pools according to the chemical quality of organic matter: acid hydrolyzable, water soluble, ethanol soluble, neither hydrolyzable nor soluble, and strongly recalcitrant organic matter (humus). To account not only for litter from leaves and other non-woody plant material, upon implementation into JSBACH, Yasso07 was extended to include also woody litter [138], which meant a doubling of all carbon pools. In addition, for implementation in JSBACH it was necessary to distinguish between above- and below-ground litter, because wildfires depend only on the availability of above ground litter (see section 9.4) that therefore needs to be tracked. This means an additional doubling of all non-woody and woody litter pools, but not the humus pools. Accordingly, the Yasso implementation in JSBACH is based on 18 carbon pools, 9 for organic material originating from non-woody litter (from the green and reserve pools), and another 9 for woody organic material (originating from the wood pool). Figure 6.1 depicts this pool structure.

Technically, the case of non-woody litter is identical to the case of woody litter, except for additional parameters that correct the decomposition rates according to the characteristic diameter (“size”) of the woody litter debris. Therefore Yasso is implemented in JSBACH as a subroutine handling 9 *generic* carbon pools, and upon call it is specified whether the 9 woody or the 9 non-woody carbon pools are meant. To distinguish those 18 carbon pools the following notation is used:

Notation for Yasso pools	
a : acid hydrolyzable	$\perp$ : above-ground
w : water soluble	$\top$ : below-ground
e : ethanol soluble	
n : neither hydrolyzable nor soluble	$w$ : woody
h : humus	$\psi$ : non-woody
<p><u>Example:</u> The symbol</p> <div style="text-align: center; margin: 10px 0;"><math>\psi C_e^\perp</math></div> <p>denotes an ethanol soluble above ground pool for non-woody litter</p>	

Simplified notation: If the top left identifier for woodiness ( $w, \psi$ ) is omitted the expression holds for both types of pools. This simplified notation is frequently used in the following because the models for those two cases are structurally identical.

Vector notation: Above and below ground litter pools are condensed into vectors

$$\begin{aligned}\vec{C}^\perp &= (C_a^\perp, C_w^\perp, C_e^\perp, C_n^\perp)^T \\ \vec{C}^\top &= (C_a^\top, C_w^\top, C_e^\top, C_n^\top)^T.\end{aligned}$$

Remark: There is only a single *humus pool* for each group of 9 woody/non-woody pools ( ${}^w C_h$  and  ${}^\psi C_h$ ) so that the above/below ground index ( $\perp, \top$ ) is absent. The non-humus pools are called *litter pools*.

#### 6.4.1 Dynamic equations for Yasso pools

Using the notation introduced above, the fundamental equations describing the dynamics of the Yasso carbon pools are

$$(6.21) \quad \frac{d\vec{C}^\perp}{dt} = \eta_{Nlim} \mathbf{AK} \vec{C}^\perp + \vec{b}^\perp$$

$$(6.22) \quad \frac{d\vec{C}^\top}{dt} = \eta_{Nlim} \mathbf{AK} \vec{C}^\top + \vec{b}^\top$$

$$(6.23) \quad \frac{dC_h}{dt} = -k_h C_h + \eta_{Nlim} a_h + b_h.$$

Here, the  $b$ -terms denote carbon gains from external inputs (e.g. from litter loss of living vegetation),<sup>14</sup> while the first right hand side terms denote a combination of carbon losses arising from decomposition of the litter pools and mixing of these fluxes between the pools. More precisely,  $\mathbf{A}$  and  $\mathbf{K}$  are 4x4 matrices specified in eqs. (6.32) and (6.28). By applying the diagonal matrix  $\mathbf{K}$  to the carbon pools one obtains the decomposition fluxes from the pools that by the action of the matrix  $\mathbf{A}$  are either redistributed between the pools, or lost to the atmosphere. The factor  $\eta_{Nlim} \in [0, 1]$  reduces the litter decomposition if not enough nitrogen is available (see chapter 7 and in particular eq. (7.22)). The  $a_h$ -term describes

<sup>14</sup>In the standard configuration of Yasso  $b_h$  is zero because humus is not formed directly from dead vegetation parts, but only indirectly via decomposition of litter. This explains why in Fig. 6.1 for Yasso there is no direct flux between the vegetation pools and the humus pool.

those decomposition losses from the litter pools that enter the humus pool; in the presence of nitrogen limitation these are also reduced by  $\eta_{Nlim}$  in accordance with the litter decomposition fluxes. – This is described below in more detail, considering first the distribution of carbon inputs from vegetation, then the decomposition fluxes of the litter pools and the humus pool, and finally the decomposition flux to the atmosphere.

### 6.4.2 Carbon inputs from vegetation

The Yasso pools are fed by the carbon lost from vegetation, i.e. by the loss terms in eqs. (6.7)-(6.9). Considering first the input to the 9 Yasso pools processing non-woody litter, they receive their input from the losses of the green and reserve pool. Particular attention need the grazing losses  $F_{grazing}$  of the green pool: Since part of the organic carbon grazed is mineralized by the herbivores, only the carbon in faeces enters the non-woody litter pools. Accordingly, the carbon transferred from the green and reserve pools is

$$(6.24) \quad L_{G+R} = \frac{C_R}{\tau_R} + (F_{litter} - F_{food}) + f_{faeces} F_{grazing},$$

where  $f_{faeces}$  is the fraction not mineralized by the herbivores.<sup>15</sup> The term  $F_{food}$  (see eq. (6.19)) is non-zero only for crops, it must be subtracted here from the litter flux because this “food” flux is computed as part of it, but then added to the crop harvest pool (see eq. (6.20)).<sup>16</sup> – Besides litter, also the carbon from root exudates is an input to Yasso from the vegetation. Since root exudates are modeled in JSBACH as a part of  $NPP$  (see eq. (6.11)) they contribute to the soil carbon directly without entering the vegetation pools. Distributing these carbon fluxes across the different non-woody litter Yasso pools above and below ground gives

$$(6.25) \quad \begin{aligned} \vec{b}^\perp &= \vec{\eta} f_G^\perp L_{G+R} \\ \vec{b}^\top &= \vec{\eta} f_G^\top L_{G+R} + \vec{e}_w F_{exud} \\ b_h &= \eta_h L_{G+R}, \end{aligned}$$

where the four components  $\eta_i$ ,  $i \in \{a, w, e, n\}$  of the vector  $\vec{\eta}$  and a similar factor for the humus pool  $\eta_h$  (see eq. (6.33)) are empirically obtained distribution factors (see [137]) such that  $\eta_h + \sum_i \eta_i = 1$ ,<sup>17</sup> and the positive  $f$ -factors split between above and below ground i.e.  $f^\perp + f^\top = 1$ . Note that by the unit vector  $\vec{e}_w = (0, 1, 0, 0)^T$  in (6.25) the root exudate carbon is fed only into the below-ground water-soluble compartment because exudates are pure sugars.

Similarly, the litter flux  $C_W/\tau_W$  from the wood pool (see eq. (6.8)) must enter the wood litter pools of Yasso:

$$(6.26) \quad \begin{aligned} \vec{b}^\perp &= \vec{\eta} f_W^\perp \frac{C_W}{\tau_W} \\ \vec{b}^\top &= \vec{\eta} f_W^\top \frac{C_W}{\tau_W} \\ b_h &= \eta_h \frac{C_W}{\tau_W} \end{aligned}$$

<sup>15</sup>Approximately only 30% of the biomass consumed by herbivores stays in the ecosystem [30, 62], the remainder of the carbon is lost directly by oxidation to the atmosphere.

<sup>16</sup>By not distinguishing for  $L_{G+R}$  between parts to above and below ground litter pools introduces a slight conceptual inconsistency in relation to how the “food” flux is obtained: This food flux is obtained in (6.19) from the above ground fraction of the litter from the green pool. Subtraction of this food flux in (6.24) from the total litter flux and distributing this reduced flux to the above and below ground Yasso pools in (6.25) implicitly distributes the above ground food losses also to above and below ground litter, although the food flux was obtained only from above ground carbon.

<sup>17</sup>The presence of  $\eta_h$  is a technical extension of Yasso in JSBACH compared to the published model structure. Accordingly, in the standard configuration  $\eta_h = 0$ .

with the  $\eta$ -values specific to wood decomposition [138].

### 6.4.3 Decomposition fluxes of the litter pools

To describe decomposition it is convenient to introduce the decomposition fluxes

$$(6.27) \quad \vec{D}^\perp := \mathbf{K}\vec{C}^\perp \quad \text{and} \quad \vec{D}^\top := \mathbf{K}\vec{C}^\top$$

that appear in eqs. (6.21) and (6.22). Besides depending on the pool size, these decomposition fluxes are determined by the rate constants  $k_i$ ,  $i \in \{a, w, e, n\}$ , that make up the diagonal matrix

$$(6.28) \quad \mathbf{K} = \text{diag}(k_a, k_w, k_e, k_n).$$

The rate constants are computed as

$$(6.29) \quad k_i = \alpha_i k_{clim} h_s(d), \quad i \in \{a, w, e, n\}$$

with the following meaning of the individual factors:

$\alpha_i$ : This is for each pool type the empirically determined baseline rate constant.<sup>18</sup>

$k_{clim}$ : This factor modifies the baseline decomposition rate according to the ruling climate:

$$(6.30) \quad k_{clim} = e^{\beta_1 T_{air} + \beta_2 T_{air}^2} (1 - e^{-\gamma P}).$$

Here  $T_{air}$  and  $P$  are air temperature and precipitation rate, both take as average over several weeks to mimic the observational input data used to calibrate Yasso.<sup>19</sup>  $\beta_1$ ,  $\beta_2$ , and  $\gamma$  are empirical factors.

$h_s(d)$ : This factor accounts for the diameter  $d$  of woody litter. The presence of this factor makes the only structural difference between the processing of woody litter and non-woody litter, i.e. for the non-woody litter pools  $h_s(d) = 1$ , while for the wood litter pools

$$(6.31) \quad h_s(d) = \min((1 + \Phi_1 d + \Phi_2 d^2)^r, 1),$$

where  $d$  is the diameter of wood litter and  $\Phi_1$ ,  $\Phi_2$  and  $r$  are empirical constants. This correction factor is limited to 1, i.e. it can only slow down the decomposition rate.

To complete the description of the litter pool dynamics of eqs. (6.21) and (6.22) it remains to specify  $\mathbf{A}$ , the ‘‘mass flow matrix’’ [137]. This matrix is defined as

$$(6.32) \quad \mathbf{A} = \begin{pmatrix} -1 & A_{w \triangleright a} & A_{e \triangleright a} & A_{n \triangleright a} \\ A_{a \triangleright w} & -1 & A_{e \triangleright w} & A_{n \triangleright w} \\ A_{a \triangleright e} & A_{w \triangleright e} & -1 & A_{n \triangleright e} \\ A_{a \triangleright n} & A_{w \triangleright n} & A_{e \triangleright n} & -1 \end{pmatrix},$$

<sup>18</sup>In the JSBACH code the factors  $\alpha_i$  are represented by the variables `ref_decomp_rate_acid`, `ref_decomp_rate_water`, ... . But whereas in this documentation and in the literature on Yasso the  $\alpha_i$  are considered positive, the respective variables in the JSBACH code are defined to be negative. This must be kept in mind when comparing formulas from this documentation with the respective formulas in the code.

<sup>19</sup>For proper representation of causalities it would make more sense to relate the decomposition rates to temperature and moisture within the litter and soil layers where decomposition actually happens. But in the original derivation of the Yasso model parameters from observed litter decomposition rates (e.g. [137]) air temperature and precipitation are used because these are readily available proxies. Hence, to be consistent with the way Yasso parameters are derived, also in JSBACH air temperature and precipitation are used. In particular for precipitation, attempts to replace it by soil moisture were not convincing when implementing Yasso into JSBACH.

where all non-diagonal entries are empirically determined values with  $A_{i \triangleright j} \in [0, 1]$ . Would all non-diagonal entries be zero, the diagonal entries “-1” would cause a total loss of the decomposition flux  $\vec{D} = \mathbf{K}\vec{C}$  (compare (6.21) and (6.22)). But because of the non-diagonal entries, part of the decomposition flux is redistributed among the 4 litter pools. Accordingly, only the carbon not redistributed is lost as  $\text{CO}_2$  to the atmosphere (see section 6.4.5).

#### 6.4.4 Gains and losses of the humus pool

This section completes the description of the humus pool dynamics eq. (6.23). Besides a possible direct input  $b_h$  from living vegetation, the humus pool<sup>20</sup> receives carbon input  $a_h$  from the above- and below-ground litter pools as an empirical fraction  $\mu_h$  of the sum of the decomposition fluxes (6.27):

$$(6.33) \quad a_h = \mu_h \sum_{i \in \{e, w, a, n\}} (D_i^\perp + D_i^\top).$$

And the carbon loss rate of the humus pool is determined by the value of  $k_h$  in (6.23). It is given as

$$(6.34) \quad k_h = \alpha_h k_{clim},$$

with  $k_{clim}$  as defined by eq. (6.30) and the empirical constant  $\alpha_h$ .

#### 6.4.5 Heterotrophic carbon losses to atmosphere

The carbon exchange between land biosphere and atmosphere originating from photosynthesis and autotrophic respiration is given by actual *NPP* as discussed in section 6.3.1. To close the land carbon budget, it remains to determine the carbon losses from heterotrophic respiration. These include besides the respiration fluxes from the Yasso pools also the respiration flux  $(1 - f_{faeces})F_{grazing}$  respired by herbivores (compare (6.24)), and the losses from crop harvest via food consumption  $F_{CH \triangleright A}$  (compare (6.20)). To determine the carbon losses from the Yasso pools to the atmosphere, one has to subtract from all respiration fluxes the re-distribution fluxes between pools, i.e. one has to sum all right hand terms in Eqs. (6.21), (6.22), and (6.23) except the  $a$  and  $b$  input terms. After a bit of algebra one finds that the total heterotrophic carbon losses including grazing and food consumption are given by

$$(6.35) \quad F_{\triangleright A} = (1 - f_{faeces})F_{grazing} + F_{CH \triangleright A} + k_h C_h - \eta_{Nlim} \sum_{i, j \in \{a, w, e, n\}} (A_{i \triangleright j} + \mu_h \delta_{ij})(D_i^\perp + D_i^\top),$$

where  $A_{i \triangleright i} = -1$  and the signs of the different terms been chosen in accordance with the convention of losses to the atmosphere being positive. Since the decomposition of litter may be reduced because of a lack in nitrogen (compare eqs. (6.21) and (6.22)), also the accompanying  $\text{CO}_2$  emissions to the atmosphere are reduced by the nitrogen limitation factor  $\eta_{Nlim}$ . – Note that both, the 9 pools for non-woody litter, as well as the 9 pools for woody litter produce loss fluxes  $F_{\triangleright A}$  that together make up the carbon release to the atmosphere. For woody litter the grazing term is zero.

---

<sup>20</sup>Actually there are two humus pools, one for leaf litter the other for wood litter. But once more the description is generic here covering both cases. Therefore the description assumes a single humus pool.

## 6.5 Dynamics of litter and soil carbon in CBALANCE

In CBALANCE the litter losses from living vegetation are processed using five pools (compare Fig. 6.1):

**Wood litter pools** ( $C_{LW_a}, C_{LW_b}$ ): These 2 pools decompose the above-ground (LW<sub>a</sub>) and below-ground (LW<sub>b</sub>) litter from the woody parts of vegetation (stems, branches, roots).

**Green litter pools** ( $C_{LG_a}, C_{LG_b}$ ): These 2 pools decompose the above-ground (LG<sub>a</sub>) and below-ground (LG<sub>b</sub>) litter from the non-woody parts of vegetation (leaves, fine roots) having turnover times ranging from months to years.

**Slow pool** ( $C_S$ ): This pool contains the organic soil carbon mineralized at a slow rate (decades to centennia). It receives its input from the 4 litter pools.

The dynamics of these three classes of pools is discussed separately in the following.

### 6.5.1 Dynamics of green litter

Carbon inputs to the CBALANCE litter pools are the same as for Yasso (compare section 6.4.2), although distributed in another way. Denoting the carbon lost by the green and reserve pool (compare eqs. (6.7) and (6.9)) as<sup>21</sup>

$$(6.36) \quad L_{G+R} = \frac{C_R}{\tau_R} + F_{litter},$$

the dynamics of the green litter pools is:

$$(6.37) \quad \frac{dC_{LG_a}}{dt} = f_{LG_a}L_{G+R} - F_{food} + f_{faeces}F_{grazing} - \eta_{Nlim}r_{LG}C_{LG_a}$$

$$(6.38) \quad \frac{dC_{LG_b}}{dt} = f_{LG_b}L_{G+R} + F_{exudates} - \eta_{Nlim}r_{LG}C_{LG_b}.$$

Here  $r_{LG}$  is a decomposition rate depending on environmental conditions (see below) and the constants  $f_{LG_a}$  and  $f_{LG_b}$  denote fixed fractions for splitting the litter losses of the green and reserve pool into the above and below ground litter pools, i.e.  $f_{LG_a} + f_{LG_b} = 1$ . Since  $L_{G+R}$  includes also above ground crop harvest  $F_{food}$  that is put into the crop harvest pool (see eq. (6.20)) this crop harvest must be subtracted here.  $\eta_{Nlim} \in [0, 1]$  accounts for a possible nitrogen limitation of the decomposition fluxes (see chapter 7 and in particular eq. (7.22)). The grazing loss of the green pool enters the above ground green litter pool by animal faeces, but only as a fraction  $f_{faeces}$  of the total grazing loss (compare section 6.4.2). Root exudates ( $F_{exudates}$ ) are put directly from  $NPP$  to the below ground green litter pool (compare section 6.4.2).

The above- and below-ground green litter pools decompose organic carbon at the same rate according to a  $Q_{10}$ -model [108]:

$$(6.39) \quad r_{LG} = \hat{w}^\kappa \frac{Q_{10}^{(T_{soil} - T_{ref})/T_{10}}}{\tau_{LG}}.$$

with reference and normalization temperatures  $T_{ref} = 0^\circ C$  and  $T_{10} = 10^\circ C$ . Here  $T_{soil}$  is a soil temperature<sup>22</sup>, and  $\tau_{LG}$  a characteristic decay time (1-2 years).  $\hat{w}$  is a modified soil

<sup>21</sup>Note that for Yasso  $L_{G+R}$  had been defined differently in (6.24).

<sup>22</sup>In the code  $T_{soil}$  is taken as the temperature of the third soil layer (see the call of `update_cbalance_bethy()` in `mo_jsbach_interface`), so that decomposition is driven by a time averaged temperature; that exactly the third layer is taken has no deeper meaning.

humidity:

$$(6.40) \quad \hat{w} = \max(w_{min}, \frac{w_{rel} - w_c}{1 - w_c}),$$

where  $w_{rel}$  is the relative amount of the soil water in the root zone (see eq. (2.41)),  $w_c$  a critical filling value (usually 0.35), and  $w_{min}$  a lower limit value (usually 0.1). This formulation guarantees that the respiration rate (6.39) does not drop to zero, i.e. that even in very dry regions green litter is decomposed at a minimum rate. The exponent  $\kappa$  in (6.39) can in principle account for the nonlinear dependence of the soil respiration rates on soil humidity, as found in [108], although the JSBACH standard setup simply uses  $\kappa = 1$ .

### 6.5.2 Dynamics of wood litter

The pools processing the litter losses from the wood pool (compare (6.8)) follow the dynamics

$$(6.41) \quad \frac{dC_{LW_a}}{dt} = f_{LW_a} \frac{C_W}{\tau_W} - \eta_{Nlim} r_{LW} C_{LW_a}$$

$$(6.42) \quad \frac{dC_{LW_b}}{dt} = f_{LW_b} \frac{C_W}{\tau_W} - \eta_{Nlim} r_{LW} C_{LW_b}.$$

The coefficients  $f_{LW_a}$  and  $f_{LW_b}$  and the nitrogen reduction factor  $\eta_{Nlim}$  have meanings analogous to those found in (6.37) and (6.38). But in contrast to (6.39) the decomposition is assumed to be independent from soil moisture because it is unclear how to parameterize it – e.g. termites like it really dry!. Hence it is assumed that wood decomposition depends only on temperature:

$$(6.43) \quad r_{LW} = \frac{Q_{10}^{T_{soil}/10^\circ C}}{\tau_{LW}}.$$

The time constant  $\tau_{LW}$  is of the order of several decades.

### 6.5.3 Dynamics of the slow pool

The decomposition losses of the 4 litter pools are released at fractions  $f_{LW \triangleright A}$  (wood) and  $f_{LG \triangleright A}$  (green) to the atmosphere. The rest is further decomposed in the slow litter pool:

$$(6.44) \quad \frac{dC_S}{dt} = \eta_{Nlim} (1 - f_{LW \triangleright A}) r_{LW} (C_{LW_a} + C_{LW_b}) + \eta_{Nlim} (1 - f_{LG \triangleright A}) r_{LG} (C_{LG_a} + C_{LG_b}) - r_S C_S.$$

The decomposition of the slow pool is assumed to follow a  $Q_{10}$ -model of the same type as that for green litter (Eq. 6.39):

$$(6.45) \quad r_S = \hat{w}^\kappa \frac{Q_{10}^{T_{soil}/10^\circ C}}{\tau_S}.$$

The time constant  $\tau_S$  is of the order of 100 years.

### 6.5.4 C-loss to atmosphere

Collecting terms from the above equations for the pool dynamics the carbon lost to the atmosphere is (compare the analogous equation for Yasso (6.35)):

$$(6.46) \quad F_{\triangleright A} = (1 - f_{faeces})F_{grazing} + F_{CH\triangleright A} + r_S C_S + \\ + \eta_{Nlim} f_{LW\triangleright A} r_{LW} (C_{LW_a} + C_{LW_b}) + \eta_{Nlim} f_{LG\triangleright A} r_{LG} (C_{LG_a} + C_{LG_b}).$$

Indeed, in equilibrium this loss flux must on average be compensated by the land uptake of carbon given by NPP.

## 6.6 Diagnostics

Typically one is interested in the carbon storage or fluxes with respect to grid box area, but the computations in CBALANCE/Yasso are done with respect to canopy area (see section 1.3.2). In addition it has to be kept in mind that the  $i$ -th PFT (respectively each tile) covers only a fraction  $c_i$  (called `cover_fract` in the code) of the (uncorrected) vegetated area. This in mind, the carbon content per square meter grid box of any CBALANCE carbon pool  $C_X$  is obtained from

$$(6.47) \quad {}^{box}C_X = veg_{max} f_i^{corr} C_X,$$

where  $f_i^{corr}$  is given by (1.7) and  $veg_{max}$  by (1.1). Such “box values” are provided by the standard output of JSBACH. Analogously, box-fluxes are provided in the output.

Table 6.2 lists the most important output fields of CBALANCE – these are located in the veg-stream (see section D.6). The Yasso pools are output to the yasso stream (see section D.7). Additional diagnostics for the CO<sub>2</sub> exchange with the atmosphere can be found in the CO<sub>2</sub>-stream (see section D.9).

For a correct functioning of the carbon cycle in JSBACH it is very important that the carbon budget is conserved. This can be checked in simulations by switching the namelist parameter `test_Cconservation` to `.TRUE.` (see appendix C.2.11). Thereby an additional output field `jsbachCconserv` is produced. Up to numerical errors this field should be zero, otherwise somewhere in JSBACH the carbon balance is broken. This is testing the carbon conservation across the whole model, but there is also a more detailed test for analyzing carbon conservation of several processes separately. This test must be activated in the code by setting the switch `debug_Cconservation` in `mo_jsbach` to `.TRUE..` As a result a whole bunch of additional output fields for the different processes will be found in the output.

With CMIP6 new aggregated land carbon and nitrogen variables have been introduced [69]. Many of them are not in the standard output of JSBACH but a result of further CMOR-conform postprocessing. For convenience table 6.3 lists how the aggregated nitrogen variables are related to standard JSBACH variables.

## 6.7 Implementation details

### 6.7.1 Code structure

The CBALANCE and Yasso models are distributed across two modules: `mo_cbal_bethy` and `mo_cbal_pools`. The main entry routine is `mo_cbal_bethy::update_cbalance_bethy()`. This routine computes each time step  $NPP$  from `grossAssimilation` and `darkRespiration` by calling the function `NPP_rate_bethy()`, and sums  $NPP$ , soil temperature, soil humidity

Table 6.2: Standard carbon diagnostics of CBALANCE and Yasso models. Note that flux variables are positive for land uptake. The output fields are located in the veg-stream (see section D.6). Abbreviations: “ag” for above ground and “bg” for below ground.

name in output	symbol	meaning
<i>Variables written from CBALANCE and Yasso (veg stream):</i>		
box_GPP_yDayMean	$^{box}GPP$	potential <i>GPP</i> as used in (6.1)
box_NPP_yDayMean	$^{box}NPP^{pot}$	potential <i>NPP</i> obtained from (6.6)
box_NPP_act_yDayMean	$^{box}NPP^{act}$	<i>NPP</i> actually allocated into pools (after accounting for limits to allocation; see sec. 6.3)
box_NPP_flux_correction	-	<i>NPP</i> that could not be allocated due to upper limits of pool sizes (see section 6.3)
boxC_green	$^{box}C_G$	C in green pool
boxC_woods	$^{box}C_W$	C in wood pool
boxC_reserve	$^{box}C_R$	C in reserve pool
boxC_crop_harvest	$^{box}C_{CH}$	C in intermediate pool collecting crop harvest
box_Cpools_total	-	C from all land pools (natural and anthropogenic)
box_soil_respiration	-	sum of C respired from soil and litter pools (the non-grazing part in (6.46))
box_root_exudates	$^{box}F_{exudates}$	Root exudate carbon flux
box_Cflux_herbivory	$^{box}F_{grazing}$	C flux from grazing the green pool
box_litter_flux	$^{box}F_{litter} + \frac{^{box}C_W}{\tau_W} + \frac{^{box}C_R}{\tau_R}$	Litter flux from all three vegetation pools
box_Cflux_crop_harvest_2_atm	$^{box}F_{CH \triangleright A}$	C flux respired from consumption of crops
<i>Variables written only from CBALANCE (veg stream):</i>		
boxC_litter_wood	$^{box}C_{LW a} + ^{box}C_{LW b}$	C in litter wood pools
boxC_litter_green_ag	$^{box}C_{LG a}$	C in ag litter green pool
boxC_litter_green_bg	$^{box}C_{LG b}$	C in bg litter green pool
boxC_slow	$^{box}C_S$	C in slow soil pool
<i>Variables written only from Yasso (in the yasso stream):</i>		
boxYC_acid_ag1	$^{box, \psi}C_a^\perp$	C in ag acid-soluble non-woody litter
boxYC_water_ag1	$^{box, \psi}C_w^\perp$	C in ag water-soluble non-woody litter
boxYC_ethanol_ag1	$^{box, \psi}C_e^\perp$	C in ag ethanol-soluble non-woody litter
boxYC_nonsoluble_ag1	$^{box, \psi}C_n^\perp$	C in ag non-soluble non-woody litter
boxYC_acid_bg1	$^{box, \psi}C_a^\top$	C in bg acid-soluble non-woody litter
boxYC_water_bg1	$^{box, \psi}C_w^\top$	C in bg water-soluble non-woody litter
boxYC_ethanol_bg1	$^{box, \psi}C_e^\top$	C in bg ethanol-soluble non-woody litter
boxYC_nonsoluble_bg1	$^{box, \psi}C_n^\top$	C in bg non-soluble non-woody litter
boxYC_humus_1	$^{box, \psi}C_h$	C in humus fraction of non-woody litter
boxYC_acid_ag2	$^{box, w}C_a^\perp$	C in ag acid-soluble woody litter
boxYC_water_ag2	$^{box, w}C_w^\perp$	C in ag water-soluble woody litter
boxYC_ethanol_ag2	$^{box, w}C_e^\perp$	C in ag ethanol-soluble woody litter
boxYC_nonsoluble_ag2	$^{box, w}C_n^\perp$	C in ag non-soluble woody litter
boxYC_acid_bg2	$^{box, w}C_a^\top$	C in bg acid-soluble woody litter
boxYC_water_bg2	$^{box, w}C_w^\top$	C in bg water-soluble woody litter
boxYC_ethanol_bg2	$^{box, w}C_e^\top$	C in bg ethanol-soluble woody litter
boxYC_nonsoluble_bg2	$^{box, w}C_n^\top$	C in bg non-soluble woody litter
boxYC_humus_2	$^{box, w}C_h$	C in humus fraction of woody litter
<i>Carbon conservation (only if namelist switch test_Cconservation is set .TRUE.):</i>		
jsbachCconserv	-	When this field is non-zero (up to numerical accuracy) the carbon balance is broken

Table 6.3: Correspondence between some CMIP6 carbon variables and standard JSBACH variables. This list is confined to non-obvious cases.

CMIP6	JSBACH	comment
cVeg	$C_G + C_W + N_R$	
cLitter	$\sum_{i \in \{e,w,a,n\}} wC_i^+ + \sum_{i \in \{e,w,a,n\}} \psi C_i^+$	JSBACH above ground litter
cSoil	$\sum_{i \in \{e,w,a,n\}} wC_i^\top + \sum_{i \in \{e,w,a,n\}} \psi C_i^\top + wC_h + \psi C_h$	JSBACH below ground litter plus humus

and LAI to compute once a day a daily average; note that at the time they are used, these are averages over the past day. Using these averages, each first time step of a day the subroutine `mo_cbal_cpools::update_cpools()` is called to update the vegetation and soil carbon pools (which are the state variables of the model). It returns the carbon emissions from the land processes as well as the corrected carbon uptake (`NPP_act_yDayMean`) (see model description above).

Which of the two soil carbon models is used, is controlled by the namlist switch `with_yasso` (see appendix C.2.11). Depending on this switch, in the subroutine `update_cpools()` either the CBALANCE soil pools are updated, or the Yasso pools by calling `mo_cbal_cpools::yasso()`; this yasso update routine is called separately for woody and green carbon. Most of the CBALANCE soil carbon calculations are also performed when using Yasso, but the results are not used.<sup>23</sup>

### 6.7.2 Handling of CO<sub>2</sub> exchange with atmosphere

At each time step the CO<sub>2</sub> exchange with the atmosphere is computed. To this end the carbon losses  $F_{\triangleright A}$  to the atmosphere (see eqs. (6.35) and (6.46)) have to be set against the time step-wise proper  $NPP$  obtained from (6.6). In addition one has to account for the corrections made to  $NPP$  during allocation of vegetation carbon (see eqs. (6.15) and (6.16)). Therefore the net exchange with the atmosphere is computed as

$$(6.48) \quad F_{L \leftrightarrow A}(t) = F_{\triangleright A}(d-1) - \{NPP^{pot}(t) - [NPP^{pot}(d-1) - NPP^{act}(d-1)]\},$$

where  $NPP^{pot}(t)$  is  $NPP$  from (6.6) at time step  $t$ . Since actual  $NPP$  is calculated only once a day (as average over the day), the correction from the difference between actual and potential  $NPP$  can be taken into account only as average over a day, hence  $NPP^{pot}(d-1)$  is the average of  $NPP^{pot}(t)$  over the day before  $t$  (indicated as  $d-1$ ). Thereby, in (6.48) values from different times are mixed. This is permitted because the carbon pool dynamics is very slow (order of months to centuries) so that it is irrelevant whether to take those fluxes from today or yesterday. Not so for  $NPP(t)$  because this is significantly changing from day to night. It is important to note that with the  $NPP$  correction in (6.48) carbon is conserved not time stepwise, but with a delay of one day.

### 6.7.3 Model parameters

Model parameters are either hard-coded, or found in the land cover type library file (see appendix C.3). For suitable parameters values of the Yasso model see [139]. Table 6.4 shows where parameters are defined and what their names are in the code.

<sup>23</sup>When running JSBACH in the CBALONE configuration, `mo_cbal_cpools::update_cpools()` is called directly from the CBALONE driver, using  $NPP$  read from the forcing files.

Table 6.4: Parameters of the CBALANCE and Yasso models.

name in code	symbol	where defined	units	meaning
cCost	$CC$	code	-	construction costs (measured in carbon units) to grow one unit of carbon
f_aut_leaf	$f_{leaf}$	code	-	fraction of leaf respiration on total maintenance respiration
tau_Cpool_reserve	$\tau_R$	code	days	time constant of reserve pool
tau_Cpool_wood	$\tau_W$	lctlib <sup>1</sup>	days	time constant of wood pool
tau_Cpool_slow <sup>2</sup>	$\tau_S$	code	days	time constant of slow pool
tau_Cpool_crop_harvest	$\tau_{CH}$	code	days	time constant of crop harvest pool
tau_Cpool_litter_leaf <sup>2</sup>	$\tau_{LG}$	lctlib <sup>1</sup>	days	time constant of litter green pools
tau_Cpool_litter_wood <sup>2</sup>	$\tau_{LW}$	lctlib <sup>1</sup>	days	time constant of wood litter pool
frac_npp_2_woodPool	$f_{NPP \triangleright W}$	lctlib <sup>1</sup>	-	fraction of NPP to be put into wood pool
frac_npp_2_reservePool	$f_{NPP \triangleright R}$	lctlib <sup>1</sup>	-	fraction of NPP to be put into reserve pool
frac_npp_2_exudates	$f_{exud}$	lctlib <sup>1</sup>	-	fraction of NPP reserved for exudates
frac_green_aboveGround	$f_{LGa}, f_G^\perp$	code	-	fraction of litter loss put into above ground litter green pool(s). Note: $f_{LGb} = 1 - f_{LGa}$ and $f^\top = 1 - f^\perp$ .
frac_wood_aboveGround	$f_{LWa}, f_W^\perp$	code	-	fraction of litter loss put into above ground litter wood pool(s). Note: $f_{LWb} = 1 - f_{LWa}$ and $f^\top = 1 - f^\perp$ .
frac_C_litter_green2atmos <sup>2</sup>	$f_{LG \triangleright A}$	lctlib <sup>1</sup>	-	fraction of green litter decomposition flux entering the atmosphere directly
frac_C_litter_wood2atmos <sup>2</sup>	$f_{LW \triangleright A}$	code	-	fraction of woody litter decomposition flux entering the atmosphere directly
frac_C_crop_harvest	$f_{food}$	code	-	fraction of crop harvest ending as crop products
greenC2leafC	$\gamma_G$	code	-	ratio of maximum carbon in green pool as compared to leaf carbon
reserveC2leafC	$\gamma_R$	lctlib <sup>1</sup>	-	ratio of optimal carbon content of reserve pool as compared to leaf carbon
Max_C_content_woods	$C_W^{max}$	lctlib <sup>1</sup>	$\frac{mol(C)}{m^2(canopy)}$	maximum C content of wood pool
specific_leaf_area_C	$sla$	lctlib <sup>1</sup>	$\frac{m^2(leaf)}{mol(C)}$	specific leaf area
LeafLit_coef1, LeafLit_coef2, ..., LeafLit_coef5 <sup>3</sup>	$\eta_i$	lctlib <sup>1</sup>	-	fraction of green litter entering specific solubility compartments, $i \in \{a, w, e, n, h\}$

... continued next page

Table 6.4: continued

name in code	symbol	where defined	units	meaning
WoodLit_coef1, WoodLit_coef2, ..., WoodLit_coef5 <sup>3</sup>	$\eta_i$	lctlib <sup>1</sup>	-	fraction of woody litter entering specific solubility compartments, $i \in \{a, w, e, n, h\}$
temp_p1, temp_p2 <sup>3</sup>	$\beta_1, \beta_2$	code	$K^{-1}, K^{-2}$	temperature dependence of respiration
precip_p1 <sup>3</sup>	$\gamma$	code	$m^2/kg$	precipitation dependence of respiration
size_p1, size_p2 <sup>3</sup>	$\Phi_1, \Phi_2$	code	$cm^{-1}, cm^{-2}$	litter size dependence of respiration
size_p3 <sup>3</sup>	$r$	code	-	Exponent of litter size dependence of respiration
WoodLitterSize <sup>3</sup>	$\lambda$	lctlib <sup>1</sup>	$cm$	typical size of woody litter
ref_decomp_rate... <sup>3</sup>	$\alpha_i$	code	$year^{-1}$	baseline decomposition rates of Yasso pools: $i \in \{a, w, e, n, h\}$
A_2_W, A_2_E, ..., W_2_A, W_2_E, ..., N_2_A, ..., N_2_E <sup>3</sup>	$A_{i \triangleright j}$	code	-	fractions for redistribution between Yasso pools: $i, j \in \{a, w, e, n\}$ , $i \neq j$
AWEN_2_H <sup>3</sup>	$\mu_h$	code	-	fraction of decomposed litter carbon entering humus pool
Q10 <sup>2</sup>	$Q_{10}$	code	-	base for temperature dependence of heterotrophic respiration
kappa <sup>2</sup>	$\kappa$	code	-	exponent for soil moisture dependence of heterotrophic respiration
frac_green_2_herbivory	$r_{graz}$	lctlib <sup>1</sup>	$day^{-1}$	loss rate of green pool by herbivory
LAI_shed_constant	$r_{shed}$	lctlib <sup>1</sup>	$day^{-1}$	constant leaf shedding rate
alpha_min <sup>2</sup>	$w_{min}$	code	-	minimum soil moistness assumed in heterotrophic respiration
alpha_critical <sup>2</sup>	$w_c$	code	-	critical soil moistness assumed in heterotrophic respiration
frac_C_faeces2_LG	$f_{faeces}$	code	-	fraction of grazed carbon entering the ag litter green pool

<sup>1</sup>Value depends on the land cover type (lct). See appendix C.3.<sup>2</sup>Parameter only effective using the CBALANCE soil carbon model.<sup>3</sup>Parameter only effective using the Yasso soil carbon model.

# Chapter 7

## Natural nitrogen cycle

### Contents

---

7.1	Introduction . . . . .	105
7.2	Structure of the JSBACH nitrogen allocation model . . . . .	106
7.3	Representation of nitrogen limitation . . . . .	112
7.4	Mineral input and output fluxes . . . . .	116
7.4.1	Mineral nitrogen gain from humus decomposition . . . . .	116
7.4.2	Biological nitrogen fixation . . . . .	117
7.4.3	Nitrogen cycling from grazing . . . . .	117
7.4.4	Nitrogen gain from crop harvest (fertilizer application) . . . . .	117
7.4.5	Nitrogen losses by leaching . . . . .	118
7.4.6	N <sub>2</sub> O losses . . . . .	119
7.4.7	Nitrogen losses from denitrification . . . . .	120
7.5	Diagnostics . . . . .	121
7.6	Implementation details . . . . .	122
7.6.1	Code structure . . . . .	122
7.6.2	Model parameters . . . . .	122
7.6.3	Estimating nitrogen limitation from model parameters . . . . .	123

---

### 7.1 Introduction

To keep running, the land carbon cycle needs nutrients: To grow and to maintain their metabolism, plants need CO<sub>2</sub> and water, which are plenty, while other macronutrients, like nitrogen, phosphorus, or potassium may not be as readily available. Similarly, the decomposition of litter and soil organic material takes place due to the growth of microbes that exploit the chemical energy stored in organic compounds, which also requires the presence of nutrients. Hence, all these processes slow down the carbon cycle when not enough nutrients are available.

The version of JSBACH described here contains a simple representation of land nitrogen cycling [101, 50]; a version additionally including phosphorus cycling has been developed [50] but is not part of the standard JSBACH code. The implemented model for nitrogen cycling is not meant to describe nitrogen limitation of vegetation in general: Since it is assumed

that the carboxylation rate  $V_{max}$  of the Farquhar model for photosynthesis (compare 5.27) is independent of nutrient availability, the JSBACH nitrogen model mainly describes that part of nitrogen limitation potentially arising upon increasing atmospheric  $\text{CO}_2$  (see the discussion in [50]). Accordingly, the pre-industrial land carbon cycle is assumed to be not limited by nitrogen availability. This restriction of the JSBACH nitrogen model could potentially be relaxed by adding a model for the regulation of leaf nitrogen and its consequences for leaf growth (compare [147]). An evaluation against  $^{15}\text{N}$  data of the nitrogen cycling simulated by JSBACH has been presented in [53].

The present chapter covers only the model for the natural nitrogen cycle without accounting for disturbances or biogeographical shifts – such complications are described in the respective chapters 8.5 and 9.8 of this documentation. Also for the modifications of the land nitrogen cycle induced by land use change the reader is referred to the respective chapter 10.3.

## 7.2 Structure of the JSBACH nitrogen allocation model

The carbon in the JSBACH allocation scheme (chapter 6) refers to carbon of vegetation, soil, litter, and crop harvest bound in organic molecules. And except for the carbon in the reserve pool (sugar, starches), such molecules typically also contain nitrogen. Accordingly, to add the cycling of nitrogen to the carbon scheme, every carbon pool, except the reserve pool, has to be complemented by a partner pool for nitrogen that describes the nitrogen content of the respective organic molecules. In addition, for building organic molecules in all types of tissues and organs, there are forms of nitrogen flexibly available in plants that makes it necessary to introduce a another pool for such mobile nitrogen in plants that has no carbon counterpart, called 'plant mobile nitrogen'. And finally, the processes in the soils crucially depend on the availability of mineral nitrogen, i.e. nitrate and ammonia, so that another pool called 'soil mineral N pool' is needed. This pool structure is depicted in Fig. 7.1.

When using Yasso (section 6.4) instead of the CBALANCE (section 6.5) soil carbon model, two further simplifications are employed. First, instead of complementing each of the Yasso pools A, W, E, N (see Fig. 7.1) with a separate nitrogen partner pool, each quadruple of such pools (below/above ground, woody/non-woody litter) is complemented by only a single nitrogen partner pool. This simplification can be made because the measured ratio between carbon and nitrogen in those A-W-E-N pools is very similar.<sup>1</sup> And second, only a single pool is used to represent the nitrogen in the two humus pools. In this way, the 18 Yasso carbon pools are complemented by only 5 nitrogen pools (1 for humus, and 4 for above and below ground woody and and non-woody nitrogen). Thereby, the nitrogen pool structure is the same with and without Yasso (compare Fig. 7.1).

Fig. 7.1 also shows how nitrogen is cycled between the pools:

- Nitrogen enters the system either by nitrogen fixation or by nitrogen deposition and is stored as mineral nitrogen in the 'soil mineral nitrogen pool'.
- During growth, the vegetation pools receive their nitrogen from the soil mineral N pool and/or the plant mobile N pool.
- Upon litter fall, nitrogen from wood is transferred to the wood litter pools, but not completely: since the measured C:N ratio of wood is smaller than that of its litter (compare table 7.2) the surplus nitrogen is transferred to the soil mineral N pool.
- Upon litter fall, nitrogen from the green pool is only partly transferred to the green litter pool because plants in preparing for litter fall (leaf yellowing) retranslocate leaf

---

<sup>1</sup>Personal communication by Jari Liski.

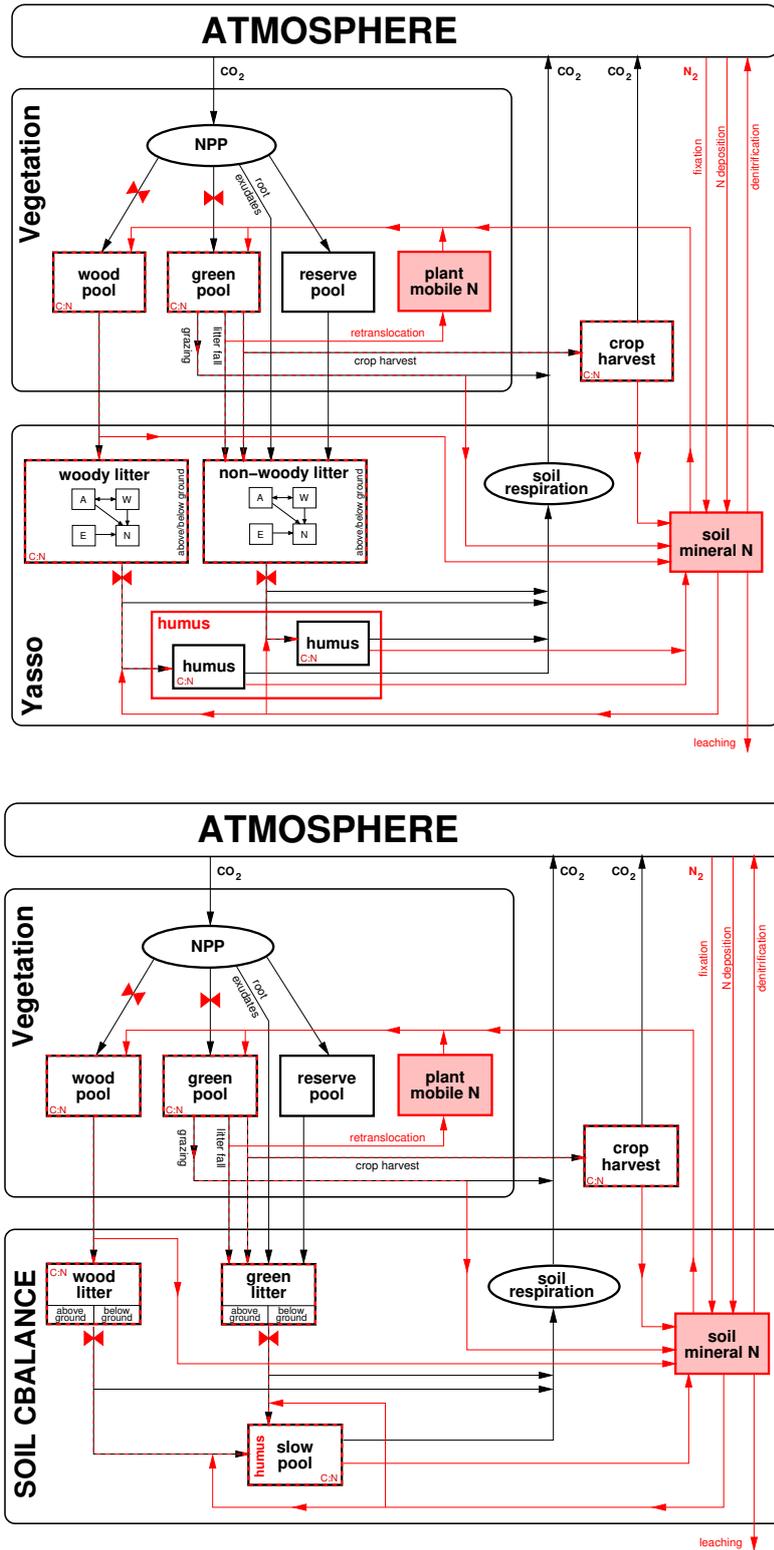


Figure 7.1: Structure of carbon and nitrogen pools for the Yasso (top) or CBALANCE (bottom) soil carbon models. Some carbon pools have a nitrogen counterpart (dashed red-black edge), while others not (black edge). The two pools with red filling have no carbon counterpart. The red 'bow-tie' sign means that these carbon fluxes are controlled by the availability of plant mobile and/or soil mineral nitrogen. The inset "C:N" means that the ratio between carbon and nitrogen content of the respective pools is fixed.

Table 7.1: Symbols used in the description of the JSBACH model for nitrogen cycling.

---



---

<b>Vegetation pools:</b>	
$N_G$	nitrogen content of green plant pool
$N_W$	nitrogen content of wood plant pool
$N_{pmob}$	nitrogen content of plant mobile pool
<b>Crop harvest:</b>	
$N_{CH}$	nitrogen content of crop harvest pool
<b>Litter and soil pools (Yasso and CBALANCE):</b>	
$wN^\perp$	nitrogen content of above ground woody litter
$wN^\top$	nitrogen content of below ground woody litter
$\psi N^\perp$	nitrogen content of above ground non-woody litter
$\psi N^\top$	nitrogen content of below ground non woody litter
$N_h$	nitrogen content of humus (Yasso) and slow (CBALANCE) pools
$N_{smin}$	nitrogen content of soil mineral N pool
<b>Nitrogen fluxes</b>	
$D_{plant}$	plant nitrogen demand
$D_{soil}$	soil nitrogen demand
$D_{total}$	total nitrogen demand from plants and soils
$G_{L\triangleright}$	total nitrogen flux from the decomposition of all litter pools
$G_{h\triangleright}$	total nitrogen flux from the decomposition of all humus pools
$G_{h\triangleright smin}$	nitrogen flux into soil mineral N pool from humus decomposition
$G_{fix}$	soil mineral nitrogen gain from microbial fixation
$G_{graz}$	soil mineral nitrogen gain from grazing
$G_{leach}$	soil mineral nitrogen loss from leaching
$G_{depo}$	soil mineral nitrogen gain from atmospheric deposition
$G_{\triangleright N_2O}$	total $N_2O$ emissions
$G_{L\triangleright N_2O}^{pot}$	potential $N_2O$ emissions related to litter decomposition
$G_{h\triangleright N_2O}$	$N_2O$ emissions related to humus decomposition
$G_{graz\triangleright N_2O}$	$N_2O$ emissions related to grazing
$G_{depo\triangleright N_2O}$	$N_2O$ emissions related to nitrogen deposition
$G_{fert}$	nitrogen from crop harvest re-applied as fertilizer
<b>Global parameters</b>	
$nc_G$	stoichiometric N:C ratio of green pool
$nc_W$	stoichiometric N:C ratio of wood pool
$nc_{LG}$	stoichiometric N:C ratio of litter flux from green pool
$nc_{LW}$	stoichiometric N:C ratio of litter pool(s) fed from wood pool
$nc_h$	stoichiometric N:C ratio of slow (CBALANCE) and humus (Yasso) pools
$f_{leach}$	soluble fraction of soil mineral N
$G_{fix,max}$	maximum rate of N fixation
$NPP_{fix,ref}$	reference NPP for N fixation
$f_{graz}^{N_2O}$	fraction of nitrogen in dung or urine from grazers volatilized to $N_2O$
$f_{denit}^{(N_2O)}$	fraction of ammonium in the denitrification pathway converted to $N_2O$
$f_{nitri}^{(N_2O)}$	fraction of ammonium in the nitrification pathway converted to $N_2O$
$f_{depo}^{N_2O}$	fraction of atmospherically deposited nitrogen volatilized as $N_2O$
$f_{nitri}^{(NH_4)}$	fraction of ammonium from humus decomposition ending in the nitrification pathway
<b>Other quantities</b>	
$NPP_{\triangleright X}^{dir}$	NPP flux to green or wood pools ( $X \in \{G, W\}$ ) allowed by available plant mobile N
$\tilde{r}_{LG}$	effective decomposition rate of all non-woody litter pools together
$\eta_{Nlim}$	reduction factor describing the strength of nitrogen limitation
$r_{denit}$	denitrification rate of soil mineral N pool

---



---

nitrogen to internal storages. This is modeled here by translocating part of the affected leaf nitrogen to the plant mobile nitrogen pool. This is consistent with the measured C:N ratios for leaves being much smaller than that of leaf litter (compare table 7.2).

- Noting that the C:N ratio for humus is (much) smaller than that of litter, the decomposition of litter to build humus may need additional nitrogen; this nitrogen is obtained from the soil mineral N pool. The formulation 'may need' is used here because during decomposition  $\text{CO}_2$  is lost to the atmosphere but no nitrogen (such losses are modeled separately, see next point) so that the C:N ratio in the transfer flux to humus may be reduced to a level such that no additional nitrogen is needed.
- Nitrogen is lost from the system by microbial nitrification and denitrification in gaseous form ( $\text{N}_2$ ,  $\text{N}_2\text{O}$ , ...), or by leaching to the soil water and river systems. The amount of these losses is determined but the fate of these losses is not further tracked.
- Nitrogen from harvested crop biomass is transferred with a time delay to the soil mineral N pool, mimicking fertilizer application.

For some of the pools a further simplification is made by assuming that the ratio between the content of the carbon and its nitrogen partner pool does not change in time, i.e. the respective C:N ratio is a model parameter. To have an idea of typical values for C:N ratios, table 7.2 lists measured ranges. It would be simplest to assume a fixed C:N ratio for all pairs of carbon/nitrogen pools, but the non-woody (Yasso) and green litter (CBALANCE) nitrogen pools are fed from a combination of pools with nitrogen (green pool) and without nitrogen (reserve pool) at varying relative amounts and receive also nitrogen-free exudate carbon (sugar), implying that the C:N ratio of this litter pool must vary in time. Hence, in this implementation, the C:N ratio is kept constant for all except the non-woody (Yasso) and green litter (CBALANCE) pools (Compare Fig. 7.1).

Table 7.2: Stoichiometric carbon to nitrogen ratios (C:N) from literature (cited after [101] with modifications from [143]) and typical values used in JSBACH 3.2. For the description of nitrogen cycling it is more convenient to use the reciprocal N:C values; the last column lists the respective symbols used here.

	literature	JSBACH 3.2	symbol
leaves/fine roots	16 - 70	35	$1/nc_G$
wood	50 - 250	150	$1/nc_W$
wood litter	212 - 1400	330	$1/nc_{LW}$
leaf/root litter	20 - 147	55	$1/nc_{LG}$
humus	8 - 30	10	$1/nc_h$

Accounting for all these simplifications, in the following the resulting equations describing the dynamics of nitrogen in JSBACH are displayed. The notation used here is analogous to that used in the carbon chapter (see in particular sections 6.3, 6.4, and 6.5). Since the structure of carbon pools is identical for CBALANCE and Yasso, a unified notation is employed, borrowed from the Yasso notation for carbon; in particular the symbols  $\perp$  and  $\top$  are used to distinguish above and below ground quantities. See table 7.1 for a comprehensive listing of the symbols used in the context of nitrogen cycling in addition to the carbon related symbols.

With these remarks the dynamics of the three plant nitrogen pools is described by

$$(7.1) \quad N_G = nc_G C_G$$

$$(7.2) \quad N_W = nc_W C_W$$

$$(7.3) \quad \frac{dN_{pmob}}{dt} = (nc_G - nc_{LG})F_{litter} - nc_G NPP_{\triangleright G}^{dir} - nc_W NPP_{\triangleright W}^{dir}.$$

Here the nitrogen content of the green and wood pools ( $N_G$ ,  $N_W$ ) is directly linked to the values of the carbon partner pools via the prescribed N:C ratios. The plant mobile N pool ( $N_{pmob}$ ) receives its nitrogen from the retranslocation of nitrogen upon litter fall; the difference  $nc_G - nc_{LG}$  describes that part of nitrogen in the litter flux  $F_{litter}$  from the green pool (see eq. (6.18)) being retranslocated<sup>2</sup> and the two loss terms provide the nitrogen needed for allocation to the green and wood pools (see eq. (7.11) below); this is why NPP shows up here (see below for the particular meaning of  $NPP^{dir}$ ).

The above and below ground woody nitrogen pools are linked by a fixed N:C ratio to their (group of) carbon partner pools:

$$(7.4) \quad wN^\perp = nc_{LW} \begin{cases} \sum_{i \in \{e, w, a, n\}} wC_i^\perp & \text{(Yasso)} \\ C_{LW a} & \text{(CBALANCE)} \end{cases}$$

$$(7.5) \quad wN^\top = nc_{LW} \begin{cases} \sum_{i \in \{e, w, a, n\}} wC_i^\top & \text{(Yasso)} \\ C_{LW b} & \text{(CBALANCE)} \end{cases}$$

Also for the humus the nitrogen is linked to carbon by a fixed N:C ratio:

$$(7.6) \quad N_h = nc_h \begin{cases} wC_h + \psi C_h & \text{(Yasso)} \\ C_S & \text{(CBALANCE)} \end{cases}$$

Note that the single nitrogen humus pool has for Yasso *two* carbon partner pools, namely the humus pools for woody and non-woody carbon ( $wC_h$ ,  $\psi C_h$ ).

The N:C ratio of non-woody (green) litter from the leaves and fine roots may change dynamically so that here a rate equation is needed: The rate by which the above and below ground nitrogen pools ( $\psi N^\perp$ ,  $\psi N^\top$ ) change is given as

$$(7.7) \quad \frac{d\psi N^\perp}{dt} = nc_{LG} f^\perp F_{litter} - G_{food} - \eta_{Nlim} \tilde{r}_{LG} \psi N^\perp + nc_G f_{faeces} F_{grazing}$$

$$(7.8) \quad \frac{d\psi N^\top}{dt} = nc_{LG} f^\top F_{litter} - \eta_{Nlim} \tilde{r}_{LG} \psi N^\top$$

These pools receive their nitrogen from litter fall (compare the retranslocation term in (7.3)), distributed to below and above ground pools as for carbon according to the fractions  $f^\perp = f_{LG a}$  and  $f^\top = f_{LG b}$  (compare eqs. (6.25) (Yasso) and (6.37), (6.38) (CBALANCE)). But for crops not all above ground littered nitrogen is put into the above ground nitrogen pool because part of it is interpreted as crop harvest put into a separate pool so that in (7.7) the nitrogen harvest flux  $G_{food}$  obtained from eq. (7.27) must be subtracted; note that  $G_{food}$  is

---

<sup>2</sup>Note that  $nc_{LG}$  is not the N:C ratio of the non-woody (green) litter *pool*, but of the litter *flux* entering this pool; all other  $nc$ -values refer to pools. Note also that to compute retranslocated nitrogen it is correct to use  $F_{litter}$  and not  $F_{litter} - F_{food}$  because the biomass is assumed to be harvested *after* retranslocation of nitrogen.

non-zero only for crops.<sup>3</sup> In addition, the above ground litter receives nitrogen from grazing, like the associated carbon pool (see eq. (6.7)). Here  $F_{grazing}$  is the carbon grazed obtained from eq. (6.17), and  $f_{faeces}$  is the fraction of the grazing flux not lost as CO<sub>2</sub> to the atmosphere but entering the litter via faeces; the associated nitrogen flux has the N:C ratio  $nc_G$  of the green pool.<sup>4</sup> The considered litter pools loose nitrogen at the same rate of decomposition  $\tilde{\tau}_{LG}$  as their respective carbon partner pools. Note that this is an *effective* decomposition rate, because it describes the *combined* decomposition of all above and below ground carbon partner pools as seen from eq. (7.20). If the system lacks nitrogen, the litter decomposition is reduced by the factor  $\eta_{Nlim} \in [0, 1]$  described in section 7.3 (see in particular eq. (7.22)).

Finally, the changes of the soil mineral N pool are described by<sup>5</sup>

$$(7.9) \quad \frac{dN_{smin}}{dt} = G_{W \triangleright smin} + G_{h \triangleright smin} + G_{fix} + (1 - f_{depo}^{N_2O})G_{depo} + (1 - f_{graz}^{N_2O})G_{graz} \\ + G_{fert} - G_{leach} - \eta_{Nlim}D_{total} - G_{denit}.$$

The different terms have the following meaning:

**$G_{W \triangleright smin}$ :** Nitrogen gain from wood shedding because of a larger N:C ratio of wood than for its litter:

$$(7.10) \quad G_{W \triangleright smin} = (nc_W - nc_{LW}) \frac{C_W}{\tau_W},$$

where  $C_W/\tau_W$  is the carbon decomposition flux from the wood pool (see eq. (6.8)).

**$G_{h \triangleright smin}$ :** Nitrogen gain from humus decomposition (see eq. (7.23)).

**$G_{fix}$ :** Nitrogen gain from microbial nitrogen fixation; see eq. (7.25).

**$G_{depo}$ :** Atmospheric nitrogen deposition. This is an external forcing (compare table C.1).

The fraction  $f_{depo}^{N_2O}$  showing up in the prefactor determines that part of the nitrogen deposition flux not entering the soil mineral N pool, but being emitted as N<sub>2</sub>O to the atmosphere (see eq. (7.34)).

**$G_{graz}$ :** Nitrogen gain via grazing of animals; see eq. (7.26). The fraction  $f_{graz}^{N_2O}$  showing up in the prefactor determines that part of nitrogen flux not entering the soil mineral N pool, but being emitted as N<sub>2</sub>O to the atmosphere (see section 7.4.6).

**$G_{fert}$ :** Nitrogen gain from harvested crop biomass applied as fertilizer (see section 7.4.4).

**$G_{leach}$ :** Nitrogen loss from leaching of mineral nitrogen via soil water fluxes; see eq. (7.29).

**$D_{total}$ :** By this flux the soil mineral N pool supplies nitrogen for plant growth and litter decomposition (see eqs. (7.21)). Here  $D_{total}$  is the total potential nitrogen demand described in section 7.3 (see in particular eq. (7.21)) and the pre-factor  $\eta_{Nlim}$  in eq. (7.9) is the reduction factor accounting for nitrogen limitation (see (7.22)). Depending on the situation, this 'demand' term may also be a gain term by which the soil mineral N pool may receive nitrogen from the litter pools.

**$G_{denit}$ :** This term represents the denitrification losses to the atmosphere, mainly N<sub>2</sub>, but not N<sub>2</sub>O that is handled separately (see section 7.4.7).

<sup>3</sup>Since the crop harvest  $F_{food}$  and thus also the associated nitrogen harvest  $G_{food}$  are obtained from the above ground green biomass, it is consistent to reduce the nitrogen litter flux to the non-woody litter pools for crops only for above ground litter as seen in eqs. (7.7) and (7.8). But for the Yasso model the related reduction of litter input to the carbon litter pools is done for both above and below ground carbon (see eq. (6.24) and associated footnote). This is a slight conceptual inconsistency appearing in relation to Yasso. There is no such inconsistency with the CBALANCE soil carbon model, because for this model crop harvest is correctly removed from the litter flux into only the above ground litter pool (see eqs. (6.37) and (6.38)).

<sup>4</sup>Part of the nitrogen of this grazing flux also enters the soil mineral N pool, and another part is lost as N<sub>2</sub>O to the atmosphere (see sections 7.4.3 and 7.4.6).

<sup>5</sup>In the following, the symbol  $G$  is used to denote nitrogen fluxes, in contrast to carbon fluxes that are denoted by  $F$ . An exception are the fluxes of the nitrogen demand denoted by  $D$ .

### 7.3 Representation of nitrogen limitation

There are four points in the carbon pool scheme where carbon fluxes may be limited by a lack of nitrogen (see the bow-tie signs in Fig. 7.1): in the allocation of carbon to the wood and green pools, i.e. in the NPP flux, and in the decomposition fluxes of woody and non-woody litter entering the humus pools. The equations for carbon cycling in chapter 6 were already written to account for such a nitrogen related reduction of carbon fluxes by including the *nitrogen limitation factor*  $\eta_{Nlim} \in [0, 1]$ :

- In the equations (6.7) and (6.8) for carbon allocation to the green and wood pools the respective NPP fluxes  $NPP_{\triangleright G}$  and  $NPP_{\triangleright W}$  are reduced by  $\eta_{Nlim}$ .
- In the Yasso equations (6.21) and (6.22), the decomposition fluxes<sup>6</sup>  $\vec{D}^\perp = \mathbf{K}\vec{C}^\perp$ ,  $\vec{D}^\top = \mathbf{K}\vec{C}^\top$  are reduced by  $\eta_{Nlim}$  for woody and non-woody litter. As a consequence also the transfer of decomposed litter to humus is reduced accordingly in eq. (6.23).
- In the CBALANCE equations for above and below ground green litter (6.37), (6.38) and wood litter (6.41), (6.42) the decomposition fluxes  $C_{LGa}/\tau_{LG}$ ,  $C_{LGb}/\tau_{LG}$  and  $C_{LWa}/\tau_{LW}$ ,  $C_{LWb}/\tau_{LW}$  are reduced by  $\eta_{Nlim}$  and accordingly also the transfer of litter carbon to the slow soil pool in eq. (6.44).

Technically, the main aim of the implemented model for nitrogen cycling is the determination of this reduction factor  $\eta_{Nlim}$ . It is the central quantity that determines the influence of the nitrogen cycle on carbon cycling in JSBACH. Hence, a main task of the following description of nitrogen cycling in JSBACH is to explain how the nitrogen limitation factor  $\eta_{Nlim}$  – that was introduced in the carbon chapter without further explanation – is calculated in JSBACH.

To represent the limitation of carbon cycling by nitrogen availability a demand-supply ansatz is chosen, i.e. the nitrogen demand from the vegetation and soil processes in absence of nitrogen limitation is compared with the nitrogen flexibly available from the mobile and soil mineral pools to reduce the carbon turnover accordingly. This is a first major design decision underlying the JSBACH model for nitrogen limitation. For the vegetation, the nitrogen demand is obtained from NPP calculated in the absence of nitrogen limitation, denoted in the following as *potential NPP*.<sup>7</sup> Hence, concerning the plants, nitrogen limitation is not modeled directly as reduced photosynthetic production, e.g. via an explicit model for Rubisco availability. Instead, the problem is tackled indirectly: If there is not enough nitrogen available to perform the allocation of potential NPP (supply less than demand), this is interpreted as a lack of Rubisco so that photosynthesis must happen at a reduced rate. This demand-supply ansatz is also applied to the soil and litter decomposition. Alternatively, the problem could be approached by explicitly modeling the soil microbial dynamics depending on nitrogen availability, but in view of today's very limited knowledge on this microbial dynamics this would introduce additional large uncertainties. Instead, the implicit representation of microbial activity by the CBALANCE ( $Q_{10}$ ) and Yasso models are kept unchanged, and the calculated decomposition rates are interpreted as potential turnover rates in absence of nitrogen limitation (demand) from which by comparison with the nitrogen available (supply) the actual soil respiration fluxes are calculated.

A second major design decision concerns the question on the relative use of nitrogen from

<sup>6</sup>Compare eqs. (6.27).

<sup>7</sup>The term *potential NPP* was already used in the carbon allocation chapter 6.3.1. There it meant the value of NPP in the absence of structural allocation limits, while the term *actual NPP* was used to denote NPP accounting for such allocation limits. Instead, in the present chapter, potential NPP is considered to already include such a structural limitation, i.e. it is the actual NPP from the carbon chapter. And the term 'actual NPP', re-introduced below, is used in the present chapter to distinguish NPP accounting for nitrogen limitation from NPP without (potential NPP). Accordingly, here actual NPP is considered to already include the reduction from structural allocation limits without being further mentioned.

the two pools with rapidly available nitrogen (plant mobile N, soil mineral N) to satisfy the plant and soil nitrogen demand. JSBACH has no separate model for plant nitrogen uptake from roots or their symbiotic interaction with mycorrhiza that in particular trees feed by sugars.<sup>8</sup> Instead, it is assumed that the plants have immediate access not only to their internal store of mobile N, but also to the store of soil mineral N, which is essentially the assumption that the processes to acquire nitrogen via roots are sufficiently fast to not limit plant access to nitrogen. Technically, it is assumed that the plants first use all the nitrogen from the plant mobile N pool before they make use of the content of the soil mineral N pool. It is therefore convenient to consider as “plant nitrogen demand” the nitrogen needed from the soil mineral N pool *after* plant mobile N has already been used up. Accordingly, to obtain this demand, first that part of potential NPP has to be determined for whose allocation the vegetation itself can provide sufficient nitrogen from the plant mobile N pool. This part will be called “direct” ( $NPP_{\triangleright X}^{dir}$ ), because it can be directly allocated without further use of soil mineral nitrogen. Assuming nitrogen limited allocation to the green and wood pools at equal proportions to the available plant mobile nitrogen, these direct parts of NPP are straightforwardly given by

$$(7.11) \quad NPP_{\triangleright X}^{dir} = \min \left( NPP_{\triangleright X}^{pot}, \frac{NPP_{\triangleright X}^{pot}}{nc_G NPP_{\triangleright G}^{pot} + nc_W NPP_{\triangleright W}^{pot}} \left[ \frac{dN_{pmob}}{dt} \right]^{max} \right)$$

for plant pools  $X \in \{G, W\}$ ,

where the allometric NPP-fractions  $NPP_{\triangleright X}^{pot}$  were introduced in eq. (6.10) (written there without upper index “*pot*”),  $nc_G$  and  $nc_W$  are the prescribed N:C ratios of the respective pools, and the derivative term is the maximum possible nitrogen flux that the plant mobile N pool can provide.<sup>9</sup> The minimum function is needed because at most the *potential* NPP flux must be supported by additional nitrogen. With (7.11) the nitrogen demand that additionally has to be satisfied by the soil mineral N pool to support plant growth is

$$(7.12) \quad D_{plant} = nc_G(NPP_{\triangleright G}^{pot} - NPP_{\triangleright G}^{dir}) + nc_W(NPP_{\triangleright W}^{pot} - NPP_{\triangleright W}^{dir}).$$

Note that the reserve pool is not showing up here because there is no associated nitrogen demand for the allocation of the respective biomolecules.<sup>10</sup>

Next the nitrogen demand by the microbial decomposition of litter has to be determined. To this end, first the general situation is discussed. The two carbon fluxes that may be nitrogen limited are those from the litter green and litter wood pools to the slow soil pool (CBALANCE) or the two humus pools (Yasso) (collectively called ‘humus’ in the following). This is depicted in Fig. 7.2: Part of the carbon decomposition flux  $F_{L\triangleright}$  from litter is emitted into the atmosphere ( $F_{\triangleright A}$ ) and the rest enters the humus pool ( $F_{\triangleright h} = F_{L\triangleright} - F_{\triangleright A}$ ). Since the N:C ratio of litter ( $nc_L$ ) is much smaller than that of humus ( $nc_h$ ), it needs additional nitrogen in the transfer of organic carbon from litter to humus that must (if necessary) be

<sup>8</sup>These exudate fluxes are the ‘carbon costs’ to acquire nitrogen. In JSBACH they are modeled as fraction of NPP entering directly the non-woody litter (see the exudate flux in eqs. (6.11), (6.25) and (6.38)).

<sup>9</sup>Numerically, this derivative term is  $N_{pmob}/\Delta t$  because the pool size  $N_{pmob}$  is the maximally available plant mobile nitrogen available during one time step of carbon and nitrogen calculations ( $\Delta t = 1$  day). Note that the amount of mobile nitrogen  $N_{pmob}$  is always small compared to the nitrogen bound in plant tissues so that a more sophisticated model for the limitation of mobile nitrogen fluxes is not needed.

<sup>10</sup>Indeed there are carbon costs to provide the nitrogen needed for growth and maintenance. These costs are assumed to be already roughly accounted for by the calculation of autotrophic respiration, since it is calculated from dark respiration (see eq. (6.3)) and this is proportional to the nitrogen availability: Dark respiration directly depends on  $V_{max}$  (see eq. (5.31)) that is known to be closely related to the Rubisco content of leaves, and therefore to their nitrogen content.

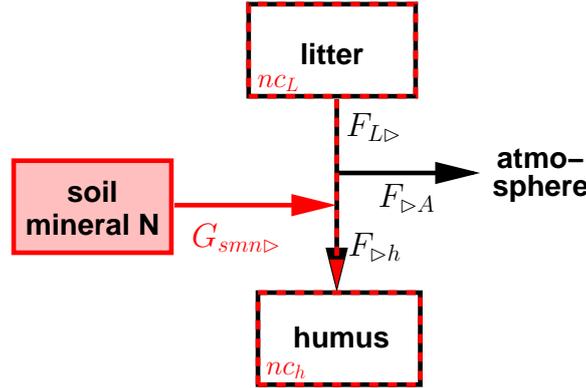


Figure 7.2: This figure depicts the situation to calculate the soil nitrogen demand. Carbon fluxes are shown in black and nitrogen fluxes in red.

supplied by the soil mineral N pool. From the figure it is seen that this additional nitrogen flux from the soil mineral N pool is given by

$$(7.13) \quad G_{smn\triangleright} = nc_h F_{\triangleright h} - nc_L F_{L\triangleright} \\ = (nc_h - nc_L) F_{L\triangleright} - nc_h F_{\triangleright A},$$

where, considering the first line, the first right hand side term is the nitrogen flux entering the humus pool as derived from the carbon flux into the humus pool  $F_{\triangleright h}$ , while the second right hand side term is the nitrogen flux from litter decomposition. As the second line shows, it is assumed that the nitrogen freed during  $\text{CO}_2$  emissions is available in the transfer of organic carbon from litter to humus (nitrogen losses to the atmosphere are handled separately, see section 7.4.6 below). Note also from the second line that  $G_{smn\triangleright}$  may be negative for sufficiently large carbon losses  $F_{\triangleright A}$  to the atmosphere. In this case, the soil mineral N pool is gaining carbon from litter decomposition. – In applying these considerations in the following to *potential* carbon fluxes,  $G_{smn\triangleright}$  is – if positive – the nitrogen demand to allow for these fluxes.

To determine this nitrogen demand, first the potential decomposition flux  $F_{\triangleright h}^{pot}$  into the humus pools is computed. To obtain it, one has to inspect for Yasso eqs. (6.23) and (6.33) while for CBALANCE the relevant equation is (6.44). Since there are four nitrogen pools associated with litter (woody/nonwoody, above/below ground) but only a single nitrogen pool for humus, one has essentially to apply Fig. 7.2 four times. But the fluxes to the single nitrogen humus pool must have the same prescribed N:C ratio  $nc_h$  of the humus so that in application of (7.14) they can be summed to give

$$(7.14) \quad F_{\triangleright h}^{pot} = \begin{cases} \mu_h \sum_{i \in \{e, w, a, n\}} ({}^w D_i^\perp + {}^w D_i^\top) + \mu_h \sum_{i \in \{e, w, a, n\}} ({}^\psi D_i^\perp + {}^\psi D_i^\top) & \text{(Yasso)} \\ (1 - f_{LW\triangleright A}) r_{LW} (C_{LW a} + C_{LW b}) + (1 - f_{LG\triangleright A}) r_{LG} (C_{LG a} + C_{LG b}) & \text{(CBALANCE)}. \end{cases}$$

This is the total potential carbon flux from all litter pools (woody/non-woody, above/below ground) to both humus pools.

Next the potential carbon fluxes leaving the four litter pools must be determined. Since woody and non-woody litter have different N:C ratios, one has to consider the associated fluxes separately in the application of eq. (7.13). To derive these fluxes one has to identify

the litter decomposition fluxes from eqs. (6.21), (6.22) with (6.27) (Yasso) and (6.37), (6.38), (6.41), (6.42) (CBALANCE). For the decomposition of woody litter one finds

$$(7.15) \quad wF_{L>}^{pot} = \begin{cases} \sum_{i \in \{e,w,a,n\}} (\mathbf{A}^w \mathbf{D}^\perp)_i + \sum_{i \in \{e,w,a,n\}} (\mathbf{A}^w \mathbf{D}^\top)_i & \text{(Yasso)} \\ r_{LW}(C_{LWa} + C_{LWb}) & \text{(CBALANCE)} \end{cases}$$

while for the non-woody litter

$$(7.16) \quad \psi F_{L>}^{pot} = \begin{cases} \sum_{i \in \{e,w,a,n\}} (\mathbf{A}^\psi \mathbf{D}^\perp)_i + \sum_{i \in \{e,w,a,n\}} (\mathbf{A}^\psi \mathbf{D}^\top)_i & \text{(Yasso)} \\ r_{LG}(C_{LGa} + C_{LGb}) & \text{(CBALANCE)} \end{cases}$$

Having determined the relevant fluxes, application of eq. (7.13) thus gives for the soil nitrogen demand

$$(7.17) \quad D_{soil} = nc_h F_{>h}^{pot} - nc_{LW} wF_{L>}^{pot} - nc_{LG} \psi F_{L>}^{pot},$$

where it has to be kept in mind that the N:C ratios  $nc_h$  and  $nc_{LW}$  of humus and woody litter are fixed, while that of non-woody (green) litter  $nc_{LG}$  is prognostic so that it has to be calculated from the carbon and nitrogen pool sizes by

$$(7.18) \quad nc_{LG} = \begin{cases} (\psi N^\perp + \psi N^\top) / \sum_{i \in \{e,w,a,n\}} (\psi C_i^\perp + \psi C_i^\top) & \text{(Yasso)} \\ (\psi N^\perp + \psi N^\top) / (C_{LGa} + C_{LGb}) & \text{(CBALANCE)}. \end{cases}$$

Actually, in the JSBACH code not the equations (7.17), (7.18) are used to calculate the soil nitrogen demand, but the equivalent equations

$$(7.19) \quad D_{soil} = nc_h F_{>h}^{pot} - nc_{LW} wF_{L>}^{pot} - \tilde{r}_{LG}(\psi N^\perp + \psi N^\top),$$

$$(7.20) \quad \tilde{r}_{LG} := \begin{cases} \psi F_{L>}^{pot} / \sum_{i \in \{e,w,a,n\}} (\psi C_i^\perp + \psi C_i^\top) & \text{(Yasso)} \\ \psi F_{L>}^{pot} / (C_{LGa} + C_{LGb}) & \text{(CBALANCE)}, \end{cases}$$

where  $\tilde{r}_{LG}$  is an *effective* litter decomposition rate describing the decomposition of all green litter pools at once.

Having determined the plant and soil nitrogen demand to satisfy *potential* carbon allocation, the effect of available nitrogen on *actual* carbon allocation can be determined. With eqs. (7.12) and (7.19) the total nitrogen demand is given as

$$(7.21) \quad D_{total} = D_{plant} + D_{soil} + G_{L>N_2O}^{pot},$$

where  $G_{L>N_2O}^{pot}$  represents the potential  $N_2O$  losses associated with the nitrification and denitrification during *litter* decomposition as described in section 7.4.6. These  $N_2O$  losses must be included in the determination of the strength of nitrogen limitation, because they are computed as a fraction of the litter decomposition flux that may be limited by a lack of nitrogen so that the  $N_2O$  flux must be reduced accordingly. Note that the  $N_2O$  emissions arising in connection with *humus* decomposition are not included in  $D_{total}$  because humus decomposes independently of the availability of soil mineral N.

Comparing the demand  $D_{total}$  for soil mineral N with the amount of actually available soil mineral N determines how strongly plant growth and litter decomposition are affected by lack of nitrogen. The strength of this limitation is expressed by the reduction factor

$$(7.22) \quad \eta_{Nlim} = \begin{cases} \frac{\left[ \frac{dN_{smin}}{dt} \right]^{max}}{D_{total}} & \text{for } \left[ \frac{dN_{smin}}{dt} \right]^{max} < D_{total} \\ 1 & \text{otherwise,} \end{cases}$$

where the term in square brackets is the largest possible flux from the soil mineral N pool.<sup>11</sup> Only if the demand for nitrogen is larger than available from the soil mineral N pool,  $\eta_{Nlim}$  has a value smaller than one, which is the sign of limited availability of nitrogen. Note that because the square bracket term is always non-negative, so is  $\eta_{Nlim}$ . The important step now for the calculation of the *actual* carbon and nitrogen fluxes is to use  $\eta_{Nlim}$  for the reduction of the *potential* fluxes. Luckily, the model for carbon and nitrogen cycling is linear<sup>12</sup> so that the actual fluxes accounting for limited nitrogen availability are obtained simply by multiplying the potential fluxes with  $\eta_{Nlim}$ . This explains why the reduction factor  $\eta_{Nlim}$  shows up in the equations for the allocation of plant carbon (6.7), (6.8), and also in the equations for the litter decomposition and associated carbon transfer from litter to humus, namely in equations (6.21), (6.22), (6.23) (Yasso) and (6.37), (6.38), (6.41), (6.42) and (6.44) (CBALANCE). And because of the reduction of litter decomposition, also the CO<sub>2</sub> flux to the atmosphere is reduced (see eqs. (6.35) (Yasso) and (6.46) (CBALANCE)).

## 7.4 Mineral input and output fluxes

This section describes the mineral input and output fluxes of the soil mineral N pool  $N_{smin}$  showing up in eq. (7.9). Not discussed are inputs prescribed externally (nitrogen deposition) and also not the term for nitrogen demand  $\eta_{Nlim}D_{total}$  that was already discussed in section 7.3 (see in particular eq. (7.21)).

### 7.4.1 Mineral nitrogen gain from humus decomposition

Decomposition of humus is the major source of mineral nitrogen for the growth of plants. The flux of nitrogen released from the humus pool is derived from the respective carbon fluxes by accounting for the N:C ratio  $nc_h$  of the humus pool by

$$(7.23) \quad G_{h\triangleright} = nc_h \begin{cases} k_h C_h & \text{(Yasso)} \\ r_s C_S & \text{(CBALANCE);} \end{cases}$$

compare eqs. (6.23) (Yasso) and (6.44) (CBALANCE). Since part of the N<sub>2</sub>O fluxes are derived directly from the decomposition fluxes (see section 7.4.6 below), not all of the nitrogen mineralized from the humus pool enters the soil mineral N pool. Hence, the mineral nitrogen flux from the humus pool to the soil mineral N pool is

$$(7.24) \quad G_{h\triangleright smin} = G_{h\triangleright} - G_{h\triangleright N_2O},$$

where the nitrogen loss flux  $G_{h\triangleright N_2O}$  is given by eq. (7.33).

<sup>11</sup>Numerically this maximum flux is taken as  $N_{smin}/\Delta t$  because the pool size  $N_{smin}$  is the maximum amount of nitrogen available during the particular time step of  $\Delta t = 1$  day used for the carbon and nitrogen cycling in JSBACH.

<sup>12</sup>Actually, plant allocation behaves non-linear when structural limits are hit as described in section 6.3.1. But even then, the error made by the equal reduction of all fluxes should be small.

### 7.4.2 Biological nitrogen fixation

In absence of anthropogenic perturbations, ecosystems gain nitrogen mostly from the fixation of atmospheric  $N_2$  by soil microbes [46]. Roughly, primary productivity is positively correlated with nitrogen fixation rates [25]. To mimic this behaviour, in JSBACH an empirical formula adopted from the Community Land Model (CLM) is used to derive the fixation flux  $G_{fix}$  from net primary productivity [133]. But NPP is not used directly: Assuming that the microbial activity is buffered against environmental variability by their persistent symbiotic relationship to plants,  $NPP$  is used only after performing an exponential smoothing of its time series.<sup>13</sup> Denoting this smoothed quantity as  $\overline{NPP}$ , fixation is calculated as

$$(7.25) \quad G_{fix} = G_{fix,max}(1 - e^{-\overline{NPP}/NPP_{fix,ref}}),$$

where the maximum fixation rate  $G_{fix,max}$  is used as a tuning parameter to give about 120 Mt fixated nitrogen per year, while the value  $NPP_{fix,ref}$  determines where the dependence on  $NPP$  turns into saturation. The reduction of nitrogen fixation for high  $NPP$  is meant to mimic limitation from other nutrients like e.g. phosphorus.

### 7.4.3 Nitrogen cycling from grazing

A substantial amount of plants is grazed by animals. This not only speeds up the cycling of carbon, but also that of nitrogen. The carbon flux from grazing  $F_{grazing}$  was described in section 6.3.2 (see in particular eq. (6.17)). There are two pathways for this carbon: a fraction  $f_{faeces}$  is transferred to the green litter pools, while the remaining fraction  $1 - f_{faeces}$  is emitted as  $CO_2$  to the atmosphere. The associated nitrogen flux enters partly the above ground non-woody (green) litter pool (see eq. (7.7)), but since the N:C ratio of the green pool is larger than that of the green litter pool, the surplus nitrogen must go either to the soil mineral N pool or leave the system in gaseous form. Of these, the flux to the soil mineral N pool is modeled as

$$(7.26) \quad G_{graz} = (nc_G - nc_{LG})f_{faeces}F_{grazing} + nc_G(1 - f_{graz}^{N_2O})(1 - f_{faeces})F_{grazing}.$$

Here, the first right hand side term is the surplus nitrogen associated with the transfer of biomass from faeces to above ground litter. The second term handles the nitrogen from that dung biomass whose carbon entered the atmosphere as  $CO_2$ : this is the fraction  $1 - f_{faeces}$  of the grazing flux (having N:C ratio  $nc_G$ ), but the fraction only from that part  $1 - f_{graz}^{N_2O}$  not volatilized as  $N_2O$  to the atmosphere (see eq. (7.34)).

### 7.4.4 Nitrogen gain from crop harvest (fertilizer application)

JSBACH doesn't use external data of fertilizer application to compensate for nitrogen losses from crop harvest.<sup>14</sup> Instead, the nitrogen from harvested crop biomass is fed with a time delay into the soil mineral N pool to mimic fertilizer application and other agricultural practices to enhance the soil nitrogen content (e.g. by green manure). Using external fertilizer data would only make sense if data would be available specifying with sufficiently high resolution where and in what growth phase of the respective crops the fertilizer is applied – otherwise, using low resolution data, the fertilizer might show effects at the wrong places or even no effect, because, applied at the wrong time, it is leached away. Therefore, in JSBACH

<sup>13</sup>The exponential smoothing is technically identical to the filtering of temperature in eqs. (3.9) to (3.11).

<sup>14</sup>Technically, the JSBACH code is prepared to handle input of fertilizer data. But this is an unofficial feature.

a different strategy is followed in connection with agricultural lands to prevent depletion of nitrogen from harvesting.

As for carbon (see eq. (6.20)), the nitrogen from harvested crop biomass is stored in a separate crop harvest pool called  $N_{CH}$ . This harvested nitrogen is obtained from the harvested carbon  $F_{food}$  by assuming that crop harvest happens after plant internal retranslocation of nitrogen before litter fall so that the harvested biomass has the N:C ratio of leaf litter from natural vegetation  $n_{CLG}$ .<sup>15</sup> Hence the associated nitrogen flux  $G_{food}$  is

$$(7.27) \quad G_{food} = n_{CLG}F_{food}.$$

Then the crop harvest pool for nitrogen develops as

$$(7.28) \quad \frac{dN_{CH}}{dt} = G_{food} - G_{fert} \quad \text{with} \quad G_{fert} = \frac{N_{CH}}{\tau_{CH}}.$$

Here the nitrogen loss flux  $G_{fert}$  is obtained with same turnover rate  $\tau_{CH}$  as for the carbon partner pool (see eq. (6.20)).<sup>16</sup> But while the carbon loss flux  $F_{CH \triangleright A}$  is released as CO<sub>2</sub> into the atmosphere, its nitrogen partner flux  $G_{fert}$  is released as mineral nitrogen into the soil mineral N pool to fertilize crop growth. Because  $\tau_{CH}$  is chosen to be of the order of one year, large parts of the harvested nitrogen stays available until the next growing season. If instead the nitrogen would have been released to the soil mineral N pool at the harvest event (which is determined by the crop phenology model: see section 3.2.7, eqs. (6.18) and (6.19)), most of the nitrogen would not any more be available to support growth during the next growing season because it has leached away.

#### 7.4.5 Nitrogen losses by leaching

Mineral nitrogen may be leached away into the river systems by soil water drainage. Assuming that the soluble fraction  $f_{leach}$  of soil mineral nitrogen is homogeneously distributed in the wet part of the soil, the leaching losses from the soil mineral N pool are calculated as

$$(7.29) \quad G_{leach} = f_{leach}N_{smin} \frac{R_d \Delta t}{h_{tot}}.$$

Here  $h_{tot}$  is the total soil moisture content (compare eq. (2.17)),  $R_d$  is the drainage flux (see eq. (2.43)), and  $\Delta t = 1$  day is the time step at which the carbon and nitrogen budgets are updated. Thereby  $R_d \Delta t / h_{tot}$  is the fraction of the stored soil water leached away during this time step, and this fraction, corrected by  $f_{leach}$ , is then taken in (7.29) to compute the fraction of soil mineral nitrogen leached away. Indeed one could argue that soil mineral nitrogen is not uniformly distributed across the soil, and in particular not across the wet fraction of the soil that is changing in time. But firstly, the mineral nitrogen stems from biological activity and this happens in the moist parts of the soils. And second, the pre-factor  $f_{leach}$  is used as the main tuning parameter to adjust the level of nitrogen limitation in JSBACH so that a more detailed description for leaching would have no effect because the tuning would anyway lever it out.

<sup>15</sup>Indeed one could argue that many crops are harvested before nitrogen has been retranslocated to prepare for leaf shedding. But since harvest happens for some crops before, for others after retranslocation there is a large degree of freedom in the choice of this global N:C ratio.

<sup>16</sup>Indeed, because the harvest dynamics is for carbon and nitrogen completely analogous, the carbon and nitrogen loss fluxes are related by  $G_{fert} = n_{CLG}F_{CH \triangleright A}$ . Hence to derive only  $G_{fert}$  one would not have needed to introduce the separate nitrogen pool  $N_{CH}$ . This pool is mainly introduced to allow for tracing the conservation of nitrogen in the model.

### 7.4.6 N<sub>2</sub>O losses

Part of the nitrogen mineralized during litter decomposition is lost in gaseous form, in particular as dinitrogen (N<sub>2</sub>) and in various oxidized forms of nitrogen. Of these, only N<sub>2</sub>O is separately considered in JSBACH because it is a potent greenhouse gas whose emissions may be of particular interest in applications of JSBACH.

Generally, N<sub>2</sub>O losses arise during nitrification and denitrification. Therefore, in principle, the origin of the nitrogen undergoing nitri- and denitrification is irrelevant. But for modeling the N<sub>2</sub>O losses, it is convenient to relate them directly to the processes from which the nitrogen originates because for particular processes quantitative estimates in form of emission factors are available. In the calculation of N<sub>2</sub>O losses, JSBACH distinguishes fluxes from the decomposition of litter and humus ( $G_{L \triangleright N_2O}$ ,  $G_{h \triangleright N_2O}$ ), grazing ( $G_{graz \triangleright N_2O}$ ), and nitrogen deposition ( $G_{depo \triangleright N_2O}$ ). Hence, the total N<sub>2</sub>O emissions are given as

$$(7.30) \quad G_{\triangleright N_2O} = G_{h \triangleright N_2O} + \eta_{Nlim} G_{L \triangleright N_2O}^{pot} + G_{graz \triangleright N_2O} + G_{depo \triangleright N_2O},$$

Note that the reduction factor  $\eta_{Nlim}$  in front of the N<sub>2</sub>O flux from litter decomposition is needed here because  $G_{L \triangleright N_2O}^{pot}$  represents *potential* emissions, calculated before accounting for a possible lack of nitrogen (compare eq. (7.21)). – How the individual emission fluxes are calculated is the content of the rest of this section.

#### N<sub>2</sub>O losses from soil decomposition

N<sub>2</sub>O losses arise from nitrification and denitrification. By nitrification ammonium (NH<sub>4</sub><sup>+</sup>) from the decomposition of humus is transformed into nitrate (NO<sub>3</sub><sup>-</sup>), and part of this, if not taken up by plants, is further transformed by denitrification into N<sub>2</sub>. In both of these processes, N<sub>2</sub>O emerges as a side product. Employing the “balloon” concept of Li et al. [85], one can assume that nitrification and denitrification happen simultaneously, but at different micro-sites in the soils. These two types of sites are thought to show either aerobic or anaerobic conditions, with nitrification happening under the former and denitrification under the latter conditions. These ideas in mind, one artificially separates the nitrogen flux from soil decomposition into a fraction  $f_{NH_4}$  yielding the ammonium content of the soil mineral N pool, while the rest  $1 - f_{NH_4}$  yields its nitrate content. On this basis, the fraction  $f_{decomp}^{N_2O}$  of the litter or humus decomposition flux emitted as N<sub>2</sub> can be written as

$$(7.31) \quad f_{decomp}^{N_2O} = \left( f_{NH_4} f_{nitri}^{(N_2O)} + w_{rel} (1 - f_{NH_4}) f_{denit}^{(N_2O)} \right) \begin{cases} 1 & \text{for } T \geq 38^\circ C \\ T/38^\circ C & \text{for } 0 < T < 38^\circ C \\ 0 & \text{for } T \leq 0^\circ C. \end{cases}$$

Here, by means of  $f_{NH_4}$ , the contributions to N<sub>2</sub>O emissions are split into separate parts for nitrification and denitrification. The formula is constructed such that N<sub>2</sub>O is most efficiently produced for temperatures beyond 38°C, and stops N<sub>2</sub>O production below 0°C. This temperature dependence is based on the observation by Sierra [127] that nitrification is most efficient at 38°C; for simplicity this parametrization is also taken for denitrification. And to account for denitrification to happen efficiently only under anaerobic conditions, the term for this pathway is multiplied by the relative soil moisture  $w_{rel}$  in the root zone (compare eq. (2.41)).

With help of  $f_{decomp}^{N_2O}$  the N<sub>2</sub>O fluxes from litter and humus decomposition are then obtained as

$$(7.32) \quad G_{h \triangleright N_2O} = f_{decomp}^{N_2O} G_{h \triangleright}$$

$$(7.33) \quad G_{L \triangleright N_2O}^{pot} = -f_{decomp}^{N_2O} \min(0, D_{soil})$$

These two formulas look a bit different, first because for humus the whole decomposition flux  $G_{h>}$  (given by eq. (7.23)) is mineralized, while for litter only part of it. And second, for litter the equation gives only the *potential*  $N_2O$  emissions, while for humus the actual ones. These complications for the case of litter come about as follows. The humus pool is fed with organic carbon from the litter pools, and this transfer flux also contains nitrogen. Hence only part of the litter decomposed is also mineralized, and only this mineralized part may lead to  $N_2O$  emissions. A *potential* mineral flux from litter exists when the soil nitrogen demand  $D_{soil}$  (given by eq. (7.17)) is negative, because only then no additional mineral nitrogen is needed from the soil mineral N pool to transfer all decomposed litter to the humus pool. This explains why in (7.33) the minimum function shows up; and the negative sign is needed because  $D_{soil}$  is by definition positive when additional mineral nitrogen is needed. This explains also why  $G_{L>N_2O}^{pot}$  is a *potential* flux: It is calculated from the nitrogen demand *before* the actual amount of available mineral nitrogen is known; therefore,  $G_{L>N_2O}^{pot}$  shows up in the calculation of the total nitrogen demand in eq. (7.21). The actual  $N_2O$  emissions from litter decomposition are obtained by reducing  $G_{L>N_2O}^{pot}$  according to the prevailing nitrogen availability as expressed by the reduction factor  $\eta_{Nlim}$  (see eq. (7.30)), and this reduced  $N_2O$  flux is implicitly accounted for in the mass balance of the soil mineral N pool (eq. (7.9)) in the term  $\eta_{Nlim}D_{total}$ .

### $N_2O$ losses from grazing

As already mentioned in section 7.4.3, part of the dung and urine of grazers is assumed to be mineralized fastly, and a small fraction of this fast mineralization flux is given to the atmosphere as  $N_2O$ . This is expressed in the the following formula for the calculation of  $N_2O$  losses from grazing:

$$(7.34) \quad G_{graz>N_2O} = nc_G f_{graz}^{N_2O} (1 - f_{faeces}) F_{grazing}.$$

Here  $nc_G F_{grazing}$  is the total nitrogen flux from grazing,  $1 - f_{faeces}$  is the fraction of nitrogen mineralized fastly, and  $f_{graz}^{N_2O}$  is the fraction of fastly mineralized nitrogen volatilized as  $N_2O$  to the atmosphere. The rest of the fastly mineralized nitrogen is added to the soil mineral N pool (compare eqs. (7.26) and (7.9)).  $f_{graz}^{N_2O}$  can also be understood as the emission factor for the conversion of dung to  $N_2O$ . According to the IPCC guidelines for greenhouse gas accounting, for 'sheep and other animals' it ranges between 0.003 to 0.03 [39, table 11.1].

### $N_2O$ losses from nitrogen deposition

Most of the atmospherically deposited nitrogen is put into the soil mineral N pool (see eq. (7.9)), but for a fraction  $f_{depo}^{N_2O}$  it is assumed that the deposited nitrogen is once more fastly lost by the formation of  $N_2O$ . This  $N_2O$  flux is calculated as

$$(7.35) \quad G_{depo>N_2O} = f_{depo}^{N_2O} G_{depo}.$$

According to the IPCC guidelines for greenhouse gas accounting,  $f_{depo}^{N_2O}$  is in the range 0.002-0.05 [39, table 11.3].

### 7.4.7 Nitrogen losses from denitrification

During denitrification the microbial reduction of oxidized forms of nitrogen ultimately leads to the emission of dinitrogen. In the model, this needs to be described as nitrogen losses of

the soil mineral N pool (compare eq. (7.9)). The formula used for this flux is

$$(7.36) \quad G_{denit} = w_{rel} r_{denit} N_{smin}.$$

Here denitrification losses are represented by a first order kinetics with loss rate  $r_{denit}$ . And since denitrification works effectively only under anaerobic conditions, the loss rate is reduced according to the relative amount of soil water in the root zone  $w_{rel}$  (compare eq. (2.41)).

## 7.5 Diagnostics

Similar to what has been said in the section on carbon diagnostics 6.6, also for nitrogen one is typically interested in storages and fluxes with respect to grid box area. These grid box values are calculated as for carbon (see eq. (6.47)) from the values per canopy area, which is the unit for the internal calculations (compare section 1.3.2).

Table 7.3 lists the most important output fields. They are found in the nitro-stream (see section D.4).

Table 7.3: Standard nitrogen diagnostics. Note that flux variables are positive for land uptake. The output fields are located in the veg-stream (see section D.6). Abbreviations: “ag” for above-ground and “bg” for below ground.

name in output	symbol	meaning
<i>Nitrogen pools:</i>		
boxN_green	$N_G$	nitrogen content of green pool
boxN_woods	$N_W$	nitrogen content of wood pool
boxN_mobile	$N_{pmob}$	nitrogen content of plant mobile pool
boxN_litter_green_ag	${}^wN^\perp$	nitrogen content of above ground non-woody litter pool
boxN_litter_green_bg	${}^wN^\top$	nitrogen content of below ground non-woody litter pool
boxN_litter_wood_ag	${}^wN^\perp$	nitrogen content of above ground wood litter pool
boxN_litter_wood_bg	${}^wN^\top$	nitrogen content of below ground wood litter pool
boxN_slow	$N_h$	nitrogen content of humus pool
boxN_sminN	$N_{smin}$	nitrogen content of soil mineral N pool
box_Npools_total	–	total nitrogen store of a grid box
<i>Nitrogen fluxes:</i>		
box_Nfix_to_sminN	$G_{fix}$	mineral nitrogen gain from microbial fixation
box_Ndep_to_sminN	$G_{depo}$	mineral nitrogen gain from atmospheric deposition
box_sminN_leach	$G_{leach}$	mineral nitrogen loss from leaching
box_N2O_total	$G_{\triangleright N_2O}$	total $N_2O$ emissions
<i>Nitrogen conservation (only if namelist switch test_Nconservation is set .TRUE.):</i>		
jsbachNconserv	–	When this field is non-zero (modulo numerical accuracy) the nitrogen balance is broken

As for carbon, also the mass conservation of nitrogen can be tested. This test is activated by setting the namelist parameter `test_Nconservation` in namelist `jsbach_ctl` to `.TRUE.` (see appendix C.2.11). Thereby an additional field `jsbachNconserv` will be found in the output whose values should be zero within numerical accuracy, otherwise the mass conservation

is broken. This is testing the nitrogen conservation across the whole model, but there is also a more detailed test for analyzing nitrogen conservation of particular processes separately. This test must be activated in the code by setting the switch `debug_Nconservation` in `mo_jsbach` to `.TRUE.`. As a result a whole bunch of additional output fields for the different processes will be found in the output.

With CMIP6 new aggregated land carbon and nitrogen variables have been introduced [69]. Many of them are not in the standard output of JSBACH but a result of further CMOR-conform postprocessing. For convenience table 7.4 lists how some of the aggregated nitrogen variables relate to standard JSBACH variables.

Table 7.4: Correspondence between some CMIP6 nitrogen variables and standard JSBACH variables. This list is confined to non-obvious cases.

CMIP6	JSBACH	comment
nVeg	$N_G + N_W + N_{pmob}$	
nLitter	$\psi N^{\perp} + w N^{\perp}$	JSBACH above ground litter
nSoil	$\psi N^{\top} + w N^{\top} + N_h$	JSBACH below ground litter plus humus
fNnetmin	$G_{h>smn} - \eta_{Nlim} D_{total}$	
fNup	$\eta_{Nlim} D_{plant}$	

## 7.6 Implementation details

### 7.6.1 Code structure

The code for the nitrogen cycle is located in the same modules as the code for the land carbon cycle, namely in `mo_cbal_bethy` and `mo_cbal_cools`. And since carbon and nitrogen cycle are so intricately connected, the calculations for carbon and nitrogen are done together in the same subroutines (compare section 6.7.1). The only exception concerns the mineral nitrogen fluxes related to an exchange with atmosphere and soil water which are calculated in the separate routine `mo_cbal_cools::N_process()`.

As described in section 7.2 and as also notified in Fig. 7.1, for a large number of pools a fixed C:N ratio is assumed. Accordingly, most of the equations (7.1) to (7.8) describing the JSBACH nitrogen dynamics are not differential equations, but link nitrogen pool sizes directly to carbon pool sizes by the prescribed C:N ratios. This is not how the model is implemented. Instead, the implemented code handles all pools dynamically, meaning that all exchange fluxes between pools are calculated explicitly to update the values of the nitrogen pools. This increases code complexity significantly but allows to test the implementation more rigorously by checking the resulting C:N ratios that should match the intended ones within numerical accuracy.<sup>17</sup>

### 7.6.2 Model parameters

The C:N ratios used in the nitrogen model are set in the namelist `cbal_parameters_ctl` (see C.5), while all other nitrogen related parameters (listed in table 7.5) are hard-coded in

<sup>17</sup>The model is equipped with a separate routine `mo_cbal_bethy::verify_CN_ratio()` to check repeatedly the accuracy to which the calculated C:N ratios match the intended C:N ratios. The tolerated inaccuracy is defined in the namelist `cbalance_ctl`. The program stops when the accuracy is less than tolerated. Otherwise, the nitrogen in the pools is corrected to match the intended values exactly in order to prevent numerical drifts.

`mo_cbal_cpools`.<sup>18</sup> Key parameters used for model tuning are the leaching efficiency  $f_{leach}$  and the base rate  $G_{fix,max}$  for nitrogen fixation.

Table 7.5: Key parameters used for the nitrogen model. For their meaning see table 7.1. Values for C:N ratios are listed in table 7.2. All the parameters listed are defined explicitly or implicitly in the code (module `mo_cbal_cpools`).

Parameter	name in code	units	typical value
$f_{leach}$	<code>sf</code>	–	0.2
$G_{fix,max}$	–	$\frac{mol(N)}{m^2 yr}$	0.31
$NPP_{fix,ref}$	–	$\frac{mol(C)}{m^2 yr}$	333
$f_{graz}^{N_2O}$	<code>N20_ef_grazing</code>	–	0.0
$f_{NH_4}$	<code>sminn_NH4_fraction</code>	–	0.4
$f_{denit}^{(N_2O)}$	<code>N20_rate_denitrification</code>	–	0.00125
$f_{nitri}^{(N_2O)}$	<code>N20_rate_nitrification</code>	–	0.001
$f_{depo}^{N_2O}$	<code>N20_ef_Ndepo</code>	–	0.01
$r_{denit}$	<code>dnp</code>	1/day	0.000275

### 7.6.3 Estimating nitrogen limitation from model parameters

Of particular importance for the emergence of nitrogen limitation are the carbon parameters that determine what fraction of the litter flux  $F_{L\triangleright}$  is directly emitted as  $CO_2$  instead of being transferred to the humus pool (compare Fig. 7.2). These fractions are of importance because by the emission of  $CO_2$  during litter decomposition the nitrogen of the decomposed organic carbon is left in the system, whereby the transfer flux from the litter to the humus pools is enriched in nitrogen (compare Fig. 7.2). Such an enrichment is indeed needed in the transfer of organic carbon from litter (having a low N:C ratio) to humus (having a much higher N:C ratio). Hence, these fractions for  $CO_2$  emissions control whether the litter decomposition fluxes already provide enough nitrogen to transfer all organic carbon from the litter to the humus pools, or whether additional nitrogen is needed from the soil mineral N pool to make this transfer possible (compare eq. (7.13)). Hence, the value of these fractions is decisive for the question whether carbon allocation can get limited from soil nitrogen demand or not. The consequence of this limitation would be in particular that in scenario simulations with increasing plant nitrogen demand, the carbon piles up in the litter pools because plants and soils compete for soil mineral nitrogen, so that the soil litter pools increasingly suffer from a lack of nitrogen for decomposition.

Whether soil decomposition tends to be nitrogen limited can be checked from the parameters by the following considerations. Denoting the considered fraction of  $CO_2$  emissions from litter decomposition as  $f_{\triangleright A}$ , it is related to the litter decomposition flux  $F_{L\triangleright}$  and the loss flux to the atmosphere  $F_{\triangleright A}$  by  $F_{\triangleright A} = f_{\triangleright A} F_{L\triangleright}$ . Entering this into eq. (7.13) gives  $G_{smin} = (nc_h(1 - f_{\triangleright A}) - nc_L) F_{L\triangleright}$ , where  $nc_L$  is the N:C ration of the considered litter pool. The condition for the litter flux to be sufficiently nitrogen-rich to transfer all litter carbon left after  $CO_2$  emissions into the humus pool without further need of nitrogen from the soil mineral N pool is that the nitrogen demand  $G_{smin}$  is non-positive. This is equivalent to the

<sup>18</sup>The nitrogen model has no vegetation specific parameters so that there is no nitrogen related parameter in the land cover type library file. The parameter `NitrogenScalingFlag` found there controls the photosynthesis in different canopy layers (see section 5.3.1.2) which is conceptually related to the nitrogen content of leaves but has otherwise nothing to do with the nitrogen calculations implemented in JSBACH.

condition

$$(7.37) \quad f_{\triangleright A} \geq 1 - \frac{nc_L}{nc_h}.$$

For CBALANCE  $f_{\triangleright A}$  is given for green and woody litter by the parameters  $f_{LG \triangleright A}$ ,  $f_{LW \triangleright A}$  (compare eq. (6.46)) while for YASSO these fractions are determined by a combination of Yasso parameters that can be read off from eq. (6.35) and (6.33): Considering a particular Yasso pool  $i \in \{a, w, e, n\}$ , the respective emission fraction is

$$(7.38) \quad f_{i \triangleright A} = \frac{F_{\triangleright A}}{F_{L \triangleright}} = \frac{\sum_{j \in \{a, w, e, n\}} (A_{i \triangleright j} + \mu_h \delta_{ij}) D_i}{a_h} = \frac{1}{\mu_h} \sum_{j \in \{a, w, e, n\}} (A_{i \triangleright j} + \mu_h \delta_{ij})$$

The parameters used in JSBACH 3.2 for the N:C ratios show that  $f_{\triangleright A}$  must be larger than 0.97 for woody litter and larger than 0.84 for non-woody litter.<sup>19</sup> For the Yasso pools a,w,e,n one finds 0.55,0.99,0.89,0.89, respectively, meaning that only the decomposition from the acid hydrolyzable pool is nitrogen limited, while the other pools produce a surplus of nitrogen. Since typically this a-pool receives the largest fraction of litter carbon,<sup>20</sup> its nitrogen demand is largest and it is not clear whether the nitrogen surplus from the other pools can compensate this demand. Hence, for Yasso the used parameter values are probably such that there is a slight nitrogen limitation from soil nitrogen demand. For CBALANCE in JSBACH 3.2 the fractions  $f_{\triangleright A}$  are 0.85 and 0.5 for green litter<sup>21</sup> from trees and grasses, respectively, and 0.2 from woody litter, so that only the decomposition of green litter from trees may not be limited by a lack of nitrogen. This probably compensates the nitrogen demand from the a-pool because of the related decomposition flux.

<sup>19</sup>The N:C ratio of non-woody litter is variable. This N:C ratio is smaller than  $nc_G$ , the N:C ratio of the influx from green litter. To calculate the threshold for  $f_{\triangleright A}$  this optimistic  $nc_G$  value was used.

<sup>20</sup>see the entries for `LeafLit_coef` and `WoodLit_coef` in the `lctlib`-file C.3.

<sup>21</sup>see the entries for `frac_C_litter_green2atmos` in the `lctlib`-file C.3 and `frac_C_litter_wood2atmos` in the JSBACH code.

# Chapter 8

## Natural disturbances of vegetation

### Contents

---

8.1	Introduction	125
8.2	Wind-throw	126
8.3	Vegetation fires	126
8.3.1	The simple JSBACH fire model	126
8.3.2	The SPITFIRE model	127
8.4	Carbon relocation from disturbances	128
8.4.1	Carbon relocation from wind-throw	128
8.4.2	Carbon relocation and emissions from vegetation fires	130
8.5	Nitrogen relocation from disturbances	133
8.5.1	Nitrogen relocation from wind-throw	133
8.5.2	Nitrogen relocation and emissions from vegetation fires	133
8.6	Diagnostics	134
8.7	Implementation details	134

---

### 8.1 Introduction

JSBACH accounts for two types of natural disturbances of vegetation: vegetation fires and wind-throw. These disturbances affect the land carbon: While by wind-throw carbon in vegetation is converted to litter, vegetation fires in addition produce immediate  $\text{CO}_2$  and  $\text{NO}_x$  emissions. Natural disturbances also affect the biogeography of vegetation: they modify the competition between species. Accordingly, natural disturbances are in particular an important element of DYNVEG, the biogeography component of JSBACH (see chapter 9).

The key quantity determined by the disturbance models is the vegetated area affected per time unit; for the two disturbance types the respective rates are called  $D_{fire}$  and  $D_{wind}$ . The advantage of formulating the strength of disturbances in terms of these area-related rates instead of e.g. formulating them in terms of the amount of carbon affected is that this rate can be used directly in DYNVEG (which is also formulated in terms of changes in area; see eq. (9.4)) and for deriving the associated carbon relocations.

## 8.2 Wind-throw

Wind-throw affects only woody types, i.e. trees and shrubs.<sup>1</sup> The disturbance rate for wind-throw is modeled as

$$(8.1) \quad D^{wind} = \begin{cases} \kappa \frac{v}{\bar{v}} v^2 & \text{if } v > q\bar{v} \\ 0 & \text{otherwise.} \end{cases}$$

Wind-throw is zero when the wind speed  $v$  is below a certain multiple  $q$  of the local longterm average maximum windspeed  $\bar{v}$ . Using such a local average windspeed as reference instead of a fixed prescribed windspeed accounts for adaptation of the local vegetation to the prevailing wind forces.  $\kappa$  is a calibration parameter. If wind-throw happens, the disturbance rate is set proportional to the acting wind power, which raises with the third power of the windspeed. On the other hand, because it is assumed that existing vegetation is adapted to the prevailing typical windspeeds, the formula is written in such a way that the disturbance rate is as well proportional to  $v/\bar{v}$ . The average maximum windspeed  $\bar{v}$  is computed as a running mean over a period of several years.<sup>2</sup>  $q$  and  $\kappa$  are chosen such that wind-throw leads to only a few severe storm damages per year in regions with high wind speeds (mainly extra-tropics). Since windspeed extremes depend on the resolution of both the atmospheric fields and the JSBACH model,  $\kappa$  and  $q$  must be chosen differently for the different atmospheric drivers, and need to be adapted e.g. to the resolution of ECHAM.<sup>3</sup>

## 8.3 Vegetation fires

In JSBACH two models for vegetation fires are implemented, a very simple parametrization [111], and the more detailed SPITFIRE model [81] from Thonike et al. [132] that is now part of the standard configuration. The advantage of the simple fire model is that it works without external data, while SPITFIRE model needs data for population density to derive anthropogenic ignitions.

### 8.3.1 The simple JSBACH fire model

It is assumed that wildfires happen only if (i) sufficient above ground plant litter  $L$  is available for combustion ( $L$  must be larger than a threshold value), and (ii) the litter is sufficiently dry to catch fire. Such a relationship between humidity of plant litter and flammability is well supported by observations [17]. In JSBACH, litter dryness is estimated from a running mean of relative air humidity over several weeks (denoted by  $\bar{h}$ ) of the lowest atmospheric level, and “sufficiently dry” then means that  $h$  must be lower than a certain threshold humidity  $h_0$ . From experiments in grasslands it is known that once ignited, the spread of wildfires increases with decreasing fuel moisture, but the uncertainty in the observational data is large [22]. Under these conditions the fire disturbance rate is assumed to increase linearly with decreasing humidity:

$$(8.2) \quad D_{fire}^{(i)} = \alpha_0^{(i)} + \frac{1}{\tau_{fire}^{(i)}} \begin{cases} \frac{h_0 - \bar{h}}{h_0} & \text{if } L > L_0/veg_{max} \text{ and } \bar{h} < h_0 \\ 0 & \text{otherwise.} \end{cases} \quad \text{for } i = w, g.$$

<sup>1</sup>“Woody” is meant here in the sense of DYNVEG; see the remarks on “competition by growth form” in section 9.2. Which PFTs are woody and which not is defined in the `lctlib`-file (see appendix C.3).

<sup>2</sup>Technically, this averaging happens in the “climate buffer” component of JSBACH, located in module `mo_climbuf`.

<sup>3</sup>To adapt to different drivers, wind-throw parameters can be modified via `namelist`; see table 8.3.

Here  $\alpha_0^{(w)}$  and  $\alpha_0^{(g)}$  are minimum fire disturbance rates for woody types and grasses (as distinguished in DYNVEG), and  $\tau_{fire}^{(w)}$ ,  $\tau_{fire}^{(g)}$  are the inverse of a characteristic frequency for the appearance of wildfires in woody types and grasses (with  $\tau_{fire}^{(w)} > \tau_{fire}^{(g)}$  so that the fire rate for woody types is smaller than for grasses).  $L$  is the total litter density in the vegetated part of a grid box, i.e. the litter of all PFTs together contributes to the wild fires. This is compared with the threshold value  $L_0/veg_{max}$ , where  $L_0$  is a parameter (see table 8.3) suitably adjusted to reveal a reasonable vegetation distribution from DYNVEG. The threshold is chosen inversely to the vegetated fraction  $veg_{max}$  of a grid box (see (1.1)) to reduce fires with increasing bare land in a grid box. Thereby it is assumed that the larger the bare land in a grid box, the less clumpy vegetation should be so that large scale fires get less probable.

The equation for the litter density  $L$  is different for the two soil carbon models used in JSBACH (see section 6) because litter is represented differently in the two models. For the standard soil carbon model Yasso:

$$(8.3) \quad L = \sum_i \sum_{j \in \{a,w,e,n\}} c_i^{can} ({}^w C_{j,i}^\perp + {}^g C_{j,i}^\perp),$$

and for CBALANCE:

$$(8.4) \quad L = \sum_i c_i^{can} (C_{LGa,i} + C_{LWa,i}),$$

Here  $c_i^{can}$  denotes the cover fraction in tile  $i$  corrected for canopy gaps (compare (1.10)). The litter pools are explained in sections 6.4 and 6.5, they are denoted here by an additional tile index  $i$ .

### 8.3.2 The SPITFIRE model

The second model for vegetation fires implemented in JSBACH is SPITFIRE [81, 132].<sup>4</sup> The following provides only a rather superficial description of the model, for more details the reader may consult the mentioned papers.

From the many equations SPITFIRE uses only three are considered here, because they give a rough idea how SPITFIRE works and make the essential links to the other parts of JSBACH, in particular to the fire disturbance rate  $D_{fire}^{(i)}$  needed by the dynamical vegetation (compare section 9.4). These three equations are:

$$(8.5) \quad r_{fire}^{(i)} = c_{fire}^{(i)} (n_{lightning} + n_{human}) FDI,$$

$$(8.6) \quad P_{mort}^{(i)} = \begin{cases} P_{crown}^{(i)} + P_{camb}^{(i)} - P_{crown}^{(i)} P_{camb}^{(i)} & \text{for woody types} \\ 1 & \text{for grass types} \end{cases}$$

$$(8.7) \quad D_{fire}^{(i)} = P_m^{(i)} r_{fire}^{(i)}.$$

The various terms have the following meaning (more explanations follow below):

- $r_{fire}^{(i)}$ : Rate at which PFT  $i$  is affected by fires in fraction of vegetated area per day.
- $c_{fire}^{(i)}$ : Fraction of area covered by PFT  $i$  burnt by a single fire under ideal fire conditions. This is calculated from the spread rate of fire (depending on PFT characteristics and prevailing windspeed) and fire danger index  $FDI$ .

<sup>4</sup>In the JSBACH source code, the SPITFIRE model goes under the name ‘‘Thonicke’’-fire model.

- FDI*: Fire Danger Index. This is a kind of probability for the ignition of a fire, calculated from environmental dryness, temperature and the availability of fuel in the form of above ground litter biomass.
- $n_{lightning}$ : Number of fire ignition events per day caused by lightnings.<sup>5</sup>
- $n_{human}$ : Number of fire ignition events per day caused by humans. This frequency is calculated from population density with a maximum at about 15 people/km<sup>2</sup> and a region specific propensity for human ignitions.<sup>6</sup>
- $P_{mort}^{(i)}$ : Probability that the fire is mortal for vegetation. The particular value depends on PFT, therefore the index ‘*i*’.
- $P_{crown}^{(i)}$ : Probability that the fire is mortal because tree crowns catch fire.
- $P_{camb}^{(i)}$ : Probability that the fire is mortal because the tree cambium is damaged by fire.
- $D_{fire}^{(i)}$ : Rate at which vegetated area of PFT *i* is lost. This is the quantity needed in equations (9.2) and (9.2) describing the natural changes in vegetation distribution.<sup>7</sup>

The first equation (8.5) gives the rate at which the different PFTs are affected by fires. But presence of fire does not mean that all vegetation being on fire dies. Therefore, to obtain the disturbance rate  $D_{fire}^{(i)}$  expressing the rate at which area covered by vegetation is lost, the spread rate  $r_{fire}$  is multiplied in the third equation (8.7) by the probability  $P_{mort}^{(i)}$  that the fire is mortal. Looking at the second equation (8.6), this probability is 1 for grasses (fires are always lethal). For woody types this probability is obtained from a combination of two probabilities, namely the probability  $P_{crown}^{(i)}$  to die from crown fires, and the probability  $P_{camb}^{(i)}$  to die from damage of the tree cambium – and since already one of these causes is considered lethal, from their sum the joint probability must be subtracted – you can’t die twice. Since in observational products the areas seen as “burnt” contain trees affected but not killed by fire (“survivors”), these products must be compared with  $r_{fire}^{(i)}$ , and not with  $D_{fire}^{(i)}$  for which these survivors are excluded by the factor  $P_{mort}^{(i)}$  in eq. (8.7).<sup>8</sup>

## 8.4 Carbon relocation from disturbances

Both wind-throw and vegetation fires have consequences for the land carbon. While by wind-throw carbon is transferred from living vegetation to the litter, wild fires lead in addition to direct CO<sub>2</sub> emissions from vegetation and litter. How this happens in JSBACH is described in this section along Fig. 8.1 depicting the carbon (and nitrogen) fluxes induced by fire and wind-throw.

### 8.4.1 Carbon relocation from wind-throw

The model for wind throw provides the rate  $D_{wind}$  at which vegetation is killed. Hence the vegetation carbon turns at exactly this rate into litter, i.e. the vegetation carbon pools change

<sup>5</sup>This frequency of lightnings is obtained as a daily interpolation between monthly climatological values read in for each grid box from a map (compare table C.1). The data for this map are obtained from observations; see [81] for the origin of these data.

<sup>6</sup>In JSBACH, the population data are read in as a sequence of maps, one for each decade (1850–2010) which during simulation are linearly interpolated at the start of each month, while the propensity data are provided as a single map (compare table C.1). For details in particular on the propensity data see [132].

<sup>7</sup>There is a slight notational inconsistency in relation to the usage of  $D_{fire}^{(i)}$  here and in the equations for the dynamic vegetation – see footnote 3 on page 141 for details.

<sup>8</sup>In the output  $r_{fire}^{(i)}$  is found as variable `box_burned_frac_diag_avg`, while  $D_{fire}^{(i)}$  is found as variable `box_burned_frac_avg` (both are averaged over the output period and converted to grid box fractions).

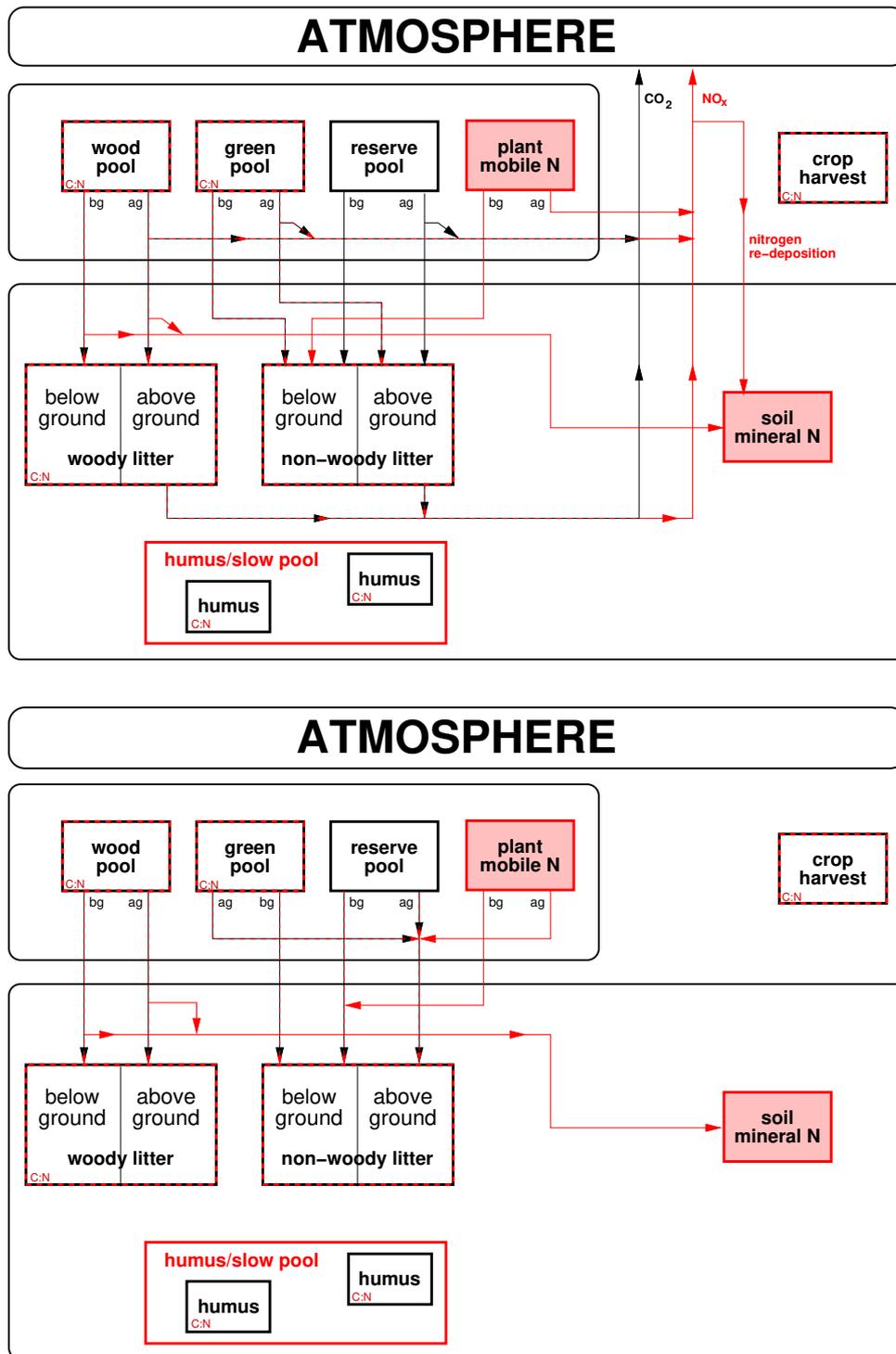


Figure 8.1: Carbon and nitrogen fluxes induced by vegetation fires (top) and windthrow (bottom). The pool structure is shown slightly simplified to cover that of Yasso and CBALANCE in one figure (compare Fig. 7.1). The figure distinguishes above ground (ag) and below ground (bg) pools because they are affected differently. The inset “C:N” marks pools having a fixed ratio between carbon and nitrogen content. The humus and crop harvest pools are neither affected by fire nor by wind throw. The system loses carbon and nitrogen to the atmosphere by fire but not by wind throw. Note that some links exist only for SPIFIRE but not for the simple JSBACH fire model: For the latter *all* burned above ground carbon and nitrogen is emitted instead of partly being transferred to the litter pools and there is also no nitrogen re-deposition.

according to<sup>9</sup>

$$(8.8) \quad \frac{dC_X}{dt} = -D_{wind}C_X \quad \text{for } X \in \{W, G, R\}.$$

The right hand side flux of carbon lost from the vegetation pools must be distributed to the litter pools. While the logic for this distribution is the same for the Yasso and CBALANCE soil carbon models, the formulas are different:

**Yasso:** Let as in section 6.4.2 denote  $\eta_i f^\perp$  and  $\eta_i f^\top$ ,  $i \in \{a, w, e, n\}$  the fractions of litter to be transferred into the respective above and below ground woody and non-woody litter pools. Then the carbon transfer to the group of woody and non-woody Yasso pools obeys

$$(8.9) \quad \frac{d^w C_i^\perp}{dt} = \eta_i f_W^\perp C_W D_{wind}$$

$$(8.10) \quad \frac{d^w C_i^\top}{dt} = \eta_i f_W^\top C_W D_{wind}$$

$$(8.11) \quad \frac{d^G C_i^\perp}{dt} = \eta_i f_G^\perp (C_G + C_R) D_{wind}$$

$$(8.12) \quad \frac{d^G C_i^\top}{dt} = \eta_i f_G^\top (C_G + C_R) D_{wind}.$$

**CBALANCE:** Let as in section 6.5 denote  $f_{LWa}$ ,  $f_{LWb}$ ,  $f_{LGa}$ , and  $f_{LGb}$  the fractions of woody and non-woody carbon to be transferred to the respective woody and non-woody above- and below-ground litter pools upon litter production. Then the litter pools change according to

$$(8.13) \quad \begin{aligned} \frac{dC_X}{dt} &= f_X C_W D_{wind} && \text{for } X \in \{LWa, LWb\}, \\ \frac{dC_X}{dt} &= f_X (C_G + C_R) D_{wind} && \text{for } X \in \{LGa, LGb\}. \end{aligned}$$

#### 8.4.2 Carbon relocation and emissions from vegetation fires

For fires the rate at which vegetated area is lost is  $D_{fire}$ . It is assumed that all vegetation on this area dies although only part of it is combusted completely. For the carbon relocation on this area it is thus assumed (compare top of Fig. (8.1)):

- Below ground vegetation carbon turns into below ground litter.
- Above ground vegetation carbon is partly combusted as  $\text{CO}_2$  and the rest turns into above ground litter.
- Above ground litter is partly combusted.

These assumptions lead to the following formulas for the relocation of carbon from vegetation fires.

First the carbon losses from the vegetation carbon pools that need to be relocated are identified. These are

$$(8.14) \quad \frac{dC_X}{dt} = -F_{X\triangleright} \quad \text{with} \quad F_{X\triangleright} = C_X D_{fire} \quad \text{for } X \in \{W, G, R\}.$$

Here the loss fluxes  $F_{X\triangleright}$  are proportional to the rate  $D_{fire}$  by which vegetated area is lost by fires. These fluxes split into three parts:

$$(8.15) \quad F_{X\triangleright} = \underbrace{f_X^\perp f_{X\triangleright A} F_{X\triangleright}}_{F_{X\triangleright A}^\perp} + \underbrace{f_X^\perp (1 - f_{X\triangleright A}) F_{X\triangleright}}_{F_{X\triangleright litter}^\perp} + \underbrace{(1 - f_X^\perp) F_{X\triangleright}}_{F_{X\triangleright litter}^\top}.$$

<sup>9</sup>Here and throughout this chapter, the time derivative is understood here to reflect only the changes from disturbances, surely other model components affect the pools simultaneously.

The first flux  $F_{X \triangleright A}^\perp$  denotes the carbon lost as  $\text{CO}_2$  to the atmosphere. This flux arises from the combustion of the fraction  $f_{X \triangleright A}$  of the above fraction  $f_X^\perp$ <sup>10</sup> of pool  $X$ . The second flux  $F_{X \triangleright \text{litter}}^\perp$  denotes the rest of the above ground carbon lost from pool  $X$  to the related above ground litter pool. And the third flux  $F_{X \triangleright \text{litter}}^\top$  denotes the carbon from the below ground fraction  $1 - f_X^\perp$  to the related below ground litter pool. How these fluxes contribute to the litter pools and fire emissions needs to be described separately for Yasso and CBALANCE:

**Yasso:** Considering first the below ground woody ( $w$ ) and non-woody ( $\psi$ ) litter pools, they change according to

$$(8.16) \quad \frac{d^w C_i^\top}{dt} = \eta_i F_{W \triangleright \text{litter}}^\top$$

$$(8.17) \quad \frac{d^\psi C_i^\top}{dt} = \eta_i (F_{G \triangleright \text{litter}}^\top + F_{R \triangleright \text{litter}}^\top),$$

where the index  $i \in \{a, w, e, n\}$  refers to the different solubility pools (compare top of Fig. 8.1). From the equations it is seen that these below ground pools only receive carbon coming from the respective vegetation pools (compare top of Fig. 8.1) and this carbon is distributed according to the Yasso distribution factors  $\eta_i$  to the different solubility pools.

While the below ground litter pools only receive carbon, part of the above ground litter pools is also combusted so that they also loose carbon. For Yasso the woody and non-woody above ground litter pools thus change according to

$$(8.18) \quad \frac{d^w C_i^\perp}{dt} = \eta_i F_{W \triangleright \text{litter}}^\perp - {}^w r_{\triangleright A} {}^w C_i^\perp$$

$$(8.19) \quad \frac{d^\psi C_i^\perp}{dt} = \eta_i (F_{G \triangleright \text{litter}}^\perp + F_{R \triangleright \text{litter}}^\perp) - {}^\psi r_{\triangleright A} {}^\psi C_i^\perp,$$

where the factors  ${}^w r_{\triangleright A}$  and  ${}^\psi r_{\triangleright A}$  are rates at which woody and non-woody litter is combusted, being differently computed for the simple JSBACH fire model and SPITFIRE (see below).

Collecting terms describing the carbon losses to the atmosphere from the vegetation and above ground litter pools, the  $\text{CO}_2$  fire emissions are given by

$$(8.20) \quad F_{\triangleright \text{atmos}}^{\text{fire}} = F_{W \triangleright A}^\perp + F_{G \triangleright A}^\perp + F_{R \triangleright A}^\perp + {}^w r_{\triangleright A} \sum_{i \in \{a, w, e, n\}} {}^w C_i^\perp + {}^\psi r_{\triangleright A} \sum_{i \in \{a, w, e, n\}} {}^\psi C_i^\perp$$

**CBALANCE:** By similar arguments, the analogous equations for the CBALANCE litter pools read as follows.

Below ground:

$$(8.21) \quad \frac{C_{LWb}}{dt} = F_{W \triangleright \text{litter}}^\top$$

$$(8.22) \quad \frac{C_{LGb}}{dt} = F_{G \triangleright \text{litter}}^\top + F_{R \triangleright \text{litter}}^\top.$$

Above ground:

$$(8.23) \quad \frac{dC_{LWa}}{dt} = F_{W \triangleright A}^\perp - {}^w r_{\triangleright A} C_{LWa}$$

$$(8.24) \quad \frac{dC_{LGa}}{dt} = F_{G \triangleright A}^\perp + F_{R \triangleright A}^\perp - {}^\psi r_{\triangleright A} C_{LGa}$$

<sup>10</sup>For the model-wide above ground fractions  $f_W^\perp$  and  $f_G^\perp$  see table 6.4. Because everywhere in JSBACH  $f_R^\perp = f_G^\perp$  only  $f_G^\perp$  exists as separate parameter in JSBACH.

And carbon loss to the atmosphere:

$$(8.25) \quad F_{\triangleright atm}^{fire} = F_{W\triangleright A}^{\perp} + F_{G\triangleright A}^{\perp} + F_{R\triangleright A}^{\perp} + {}^w r_{\triangleright A} C_{LWa} + {}^y r_{\triangleright A} C_{LGa}$$

This description of the Yasso and CBALANCE carbon relocation covers both fire models implemented in JSBACH. But they differ in their parameters. Table 8.1 lists how the parameter values are obtained for the two fire models. For the **JSBACH fire model** the fractions  $f_{G\triangleright A}$  and  $f_{R\triangleright A}$  are set to 1, meaning that all above ground carbon from the green and reserve pools affected by fires is combusted and emitted to the atmosphere. Instead,  $f_{W\triangleright A}$  is a fixed parameter less than one, so that only part of the above ground carbon from the wood pool is emitted as CO<sub>2</sub> to the atmosphere, while the rest turns into litter. From the table it also seen that for the JSBACH fire model the two rates  ${}^w r_{\triangleright A}$  and  ${}^y r_{\triangleright A}$  at which litter carbon is lost to the atmosphere is set to the rate  $D_{fire}$  at which vegetation cover is lost, meaning that for this simple fire model all above ground litter on the area under fire is completely combusted.

Table 8.1: Parameters used for the carbon and nitrogen relocation in the simple JSBACH fire model and SPITFIRE. Index  $i$  refers to PFT type. For more details see text.

symbol	JSBACH fire	SPITFIRE
$f_{G\triangleright A} = f_{R\triangleright A}$	1	$1 - \frac{P_{crown}^{(i)}}{P_{mort}^{(i)}}$
$f_{W\triangleright A}$	< 1	$f_W^{cmbst} \frac{P_{crown}^{(i)}}{P_{mort}^{(i)}}$
${}^w r_{\triangleright A}$	$D_{fire}^{(i)}$	$r_{fire} \sum_{n=1}^4 cc_n f_n^{size}$
${}^y r_{\triangleright A}$	$D_{fire}^{(i)}$	$r_{fire} cc_1 f_{active}$
$f_{redep}$	0	> 0

For **SPITFIRE** all parameters are complicated functions of environmental conditions and vegetation type. An important difference to the simple JSBACH fire model is the distinction seen in eq. (8.7) between the rate at which vegetated area catches fire ( $r_{fire}$ ) and the rate at which vegetated area is lost because of lethal damage of plants from fire ( $D_{fire}^{(i)}$ ). Assuming that the above ground green and reserve pools get combusted mainly when fire is such that it is damaging the cambium, the rate at which the pools are combusted can be written as

$$(P_{mort}^{(i)} - P_{camb}^{(i)})r_{fire} = \left(1 - \frac{P_{crown}^{(i)}}{P_{mort}^{(i)}}\right)D_{fire}^{(i)}, \text{ i.e. in view of e.g. } F_{G\triangleright A}^{\perp} = f_G^{\perp} f_{G\triangleright A} C_G D_{fire} \text{ (see}$$

eqs. (8.15) and (8.14)) the fractions  $f_{G\triangleright A}$  and  $f_{R\triangleright A}$  are identified as shown in table 8.1. By a similar argument one obtains the expression for  $f_{W\triangleright A}$  in table 8.1, except that for the woody vegetation parts it is assumed that they are mostly affected by crown fires and that in addition the thicker wood of trees is only partly burned so that an additional reduction factor  $f^{cmbst}$  obtained from assumptions on the distribution of thick and thin branches is applied. Concerning the rate  ${}^w r_{\triangleright A}$  at which above ground woody litter is combusted, also a distribution of thin and thick twigs, trunks, and stems is assumed – more precisely 4 size classes  $n = 1, 2, 3, 4$  of litter are assumed being present at fractions  $f_n^{size}$  with biomass from larger size classes being less completely combusted. This ‘combustion completeness’, expressed by the factor  $cc_n$ , is made in addition dependent on the moistness of the litter calculated from environmental conditions. For the rate  ${}^y r_{\triangleright A}$  of combustion of non-woody litter the combustion completeness of the smallest size class  $n = 1$  is applied, assuming in addition that only a

fixed fraction  $f_{active}$  can be burned, being less than 1 for woody types, and equal 1 for grass types.

For more details the reader is directed to the source code.

## 8.5 Nitrogen relocation from disturbances

Most JSBACH nitrogen pools are related to their carbon partner pools by a fixed N:C ratio. Accordingly, after accounting for disturbance induced changes in the carbon pools, for these nitrogen pools their new values follow from the carbon partner pools, whose changes were already described in the previous section. Hence, in the following, nitrogen relocation needs to be described only for nitrogen pools that are not related to carbon pools via a fixed N:C ratio (above and below ground non-woody nitrogen), and for nitrogen pools that have no carbon partners ('soil mineral N pool' and 'plant mobile pool'). Since the logic of nitrogen relocation follows that of carbon described in the previous section, the equations for nitrogen relocation are presented here without much explanation. In addition, a glance at Fig. 8.1 depicting the nitrogen relocations induced by disturbances may be helpful in following these explanations. – For the notation used in connection with nitrogen, see section 7.2, in particular Fig. 7.1, and table 7.1.

### 8.5.1 Nitrogen relocation from wind-throw

Analogous to the carbon equation (8.8), also the plant nitrogen pools change proportional to the rate  $D_{wind}$  of vegetation area lost. Hence the plant mobile nitrogen pool changes from wind-throw by

$$(8.26) \quad \frac{dN_{pmob}}{dt} = -D_{wind}N_{pmob}.$$

Upon wind-throw, the non-woody litter pools receive nitrogen from the green and plant mobile nitrogen pools, i.e.

$$(8.27) \quad \frac{d^{\psi}N^{\perp}}{dt} = f_G^{\perp}D_{wind}(N_G + N_{pmob})$$

$$(8.28) \quad \frac{d^{\psi}N^{\top}}{dt} = f_G^{\top}D_{wind}(N_G + N_{pmob})$$

Here the notation adapted to Yasso is used. The CBALANCE equations are analogous with  $f^{\perp}$  and  $f^{\top}$  replaced by  $f_{LGa}$  and  $f_{LGb}$ . The surplus mineral nitrogen appearing during the transfer of wind-throw induced woody litter to the wood litter pools because of a mismatch of N:C values is calculated as for usual wood litter production (compare eq. (7.10)) and added to the soil mineral N pool:

$$(8.29) \quad \frac{dN_{smin}}{dt} = (nc_W - nc_{LW})D_{wind}N_W.$$

### 8.5.2 Nitrogen relocation and emissions from vegetation fires

Upon vegetation fires, plant nitrogen is assumed to be lost at the same rate  $D_{fire}$  as carbon. Thereby the value of the plant mobile nitrogen pool changes by

$$(8.30) \quad \frac{dN_{pmob}}{dt} = -D_{fire}N_{pmob}.$$

The non-woody litter pools are updated according to

$$(8.31) \quad \frac{d^w N^\perp}{dt} = nc_G F_{G \triangleright L}^\perp - F_{L \triangleright A}^\perp \frac{N_L^\perp}{C_L^\perp}$$

$$(8.32) \quad \frac{d^w N^\top}{dt} = nc_G F_{G \triangleright L}^\top + f^\top D_{fire} N_{pmob},$$

meaning that only above ground litter is burned and that the below ground litter pool gains the nitrogen from the killed below ground tissues of the green pool and the below ground part of the plant mobile N pool. For the carbon fluxes  $F$  see eq. (8.15).

Part of the burned biomass is lost in gaseous form to the atmosphere. The gaseous loss of nitrogen is described consistently with the carbon loss flux (compare eqs. (8.20) and (8.25)), only that a term for the nitrogen losses of the plant mobile N pool needs to be added:

$$(8.33) \quad G_{\triangleright atm}^{fire, gross} = nc_W F_{W \triangleright A}^\perp + nc_G F_{G \triangleright A}^\perp + f^\perp D_{fire} N_{pmob} + w_{r \triangleright A} w N^\perp + w_{r \triangleright A}^w N^\perp$$

This is interpreted as the *gross* nitrogen emissions from vegetation fires. These need to be distinguished from the *net* emissions

$$(8.34) \quad G_{\triangleright atm}^{fire, net} = (1 - f_{redep}) G_{\triangleright atm}^{fire, gross},$$

being smaller than gross emissions by the factor  $1 - f_{redep}$ . This factor is meant to account for partial local re-deposition of mineral nitrogen emitted from vegetation fires. It accounts for the experience that in simulations with JSBACH in fire-prone regions land nitrogen gets heavily depleted when not accounting for such a re-deposition, resulting in a reduced photosynthetic productivity followed by a severe loss in vegetation cover (see [89]). Therefore in JSBACH, part of the calculated gross emissions is once more made available to the vegetation by adding it to the soil mineral N pool. This results in soil mineral N pool changes according to

$$(8.35) \quad \frac{dN_{smin}}{dt} = (nc_W - nc_{LW})(F_{W \triangleright litter}^\perp + F_{W \triangleright litter}^\top) + f_{redep} G_{\triangleright atm}^{fire, gross},$$

where the first right hand side term is the surplus nitrogen that cannot be transferred to the woody litter pool because of a mismatch of N:C ratios. Note that re-deposition is not implemented for the simple JSBACH fire model so that in this case  $f_{redep} = 0$ .

## 8.6 Diagnostics

Table 8.2 lists some useful output variables to diagnose the behaviour of the disturbances. Further diagnostics can be enabled by setting the `ldiag` (SPITFIRE only) and/or `lemissions` to `.TRUE.` in the `disturbance_ctl` namelist. The additional output produced will be written to a separate “disturb” stream. The latter option converts the carbon released to the atmosphere into amounts of different chemical tracers using emission factors read from file (table C.1).

## 8.7 Implementation details

The code for disturbances is found in the modules `mo_disturbance`, `mo_disturbance_jsbach` and `mo_disturbance_thonicke`. When JSBACH is run without DYNVEG, disturbances are computed via calling the routine `mo_disturbance::update_disturbance()` in the main

Table 8.2: Variables for diagnosing disturbances

name in output	symbol	units	stream	meaning
box_damaged_frac_avg	$D_{wind}^{(i)} veg_{max} c_i^{can}$	$\frac{m^2(windthrow)}{m^2(gridbox)}$	veg	Fractional area affected by windthrow averaged over output period
box_burned_frac_avg	$D_{fire}^{(i)} veg_{max} c_i^{can}$	$\frac{m^2(burned)}{m^2(gridbox)}$	veg	Fractional area affected by fires averaged over output period relevant for carbon and nitrogen relocation
box_burned_frac_diag_avg	$r_{fire}^{(i)} veg_{max} c_i^{can}$	$\frac{m^2(burned)}{m^2(gridbox)}$	veg	Fractional area affected by fires averaged over output period as seen from satellite (incl. survivors)
box_fire_CO2_flux_2_atmos	$veg_{max} \sum_i c_i^{can} F_{\triangleright atm, i}$	$\frac{kg(CO_2)}{m^2(gridbox)s}$	jsbach	CO <sub>2</sub> emissions from vegetation fires
fuel	$L$	$\frac{mol(C)}{m^2(canopy)}$	veg	Amount of fuel used to calculate $D_{fire}$
dist_nitrogen_2_atmos	$veg_{max} \sum_i c_i^{can} G_{\triangleright atm, i}^{fire, net}$	$\frac{mol(N)}{m^2(gridbox)day}$	veg	nitrogen emissions from vegetation fires

Table 8.3: Parameters of the disturbance models. For SPITFIRE only those few parameters are listed that were mentioned in the model description above. Default values are set in the source code modules. Abbreviations for locations are explained at the table foot.

variable name	name in code/file	units	default value	value defined in
<i>wind break:</i>				
$q$	wind_threshold	–	2.25	nml_wind, mo_di_jsb
$\kappa$	wind_damage_scale	s <sup>2</sup> /m <sup>2</sup> year	0.005	nml_wind, mo_di_jsb
<i>Fire model selection:</i>				
	fire_name		thonicke <sup>1</sup>	nml_dist, mo_dist
<i>simple JSBACH fire model:</i>				
$L_0$	fire_litter_threshold	mol(C)/m <sup>2</sup>	16.67	nml_fire_jsb, mo_di_jsb
$h_0$	fire_rel_hum_threshold	%	70	nml_fire_jsb, mo_di_jsb
$\alpha_0^{(w)}$	fire_minimum_woody	1/year	0.002	nml_fire_jsb, mo_di_jsb
$\alpha_0^{(g)}$	fire_minimum_grass	1/year	0.006	nml_fire_jsb, mo_di_jsb
$\tau_{fire}^{(w)}$	fire_tau_woody	years	6	nml_fire_jsb, mo_di_jsb
$\tau_{fire}^{(g)}$	fire_tau_grass	years	2	nml_fire_jsb, mo_di_jsb
<i>SPITFIRE:</i>				
$f_{redep}$	frac_N_2_atm	–	0.5	nml_fire_th, mo_di_th
$f_{active}$	frac_green_active	–	0.3	nml_fire_th, mo_di_th

<sup>1</sup>Valid values are: *none*, *jsbach* and *thonicke*. Abbreviations used for namelists: “nml\_wind”: windbreak\_jsbach\_ctl, “nml\_dist”: disturbance\_ctl, “nml\_fire\_jsb”: fire\_jsbach\_ctl, “nml\_fire\_th”: fire\_thonicke\_ctl. Abbreviations used for source code modules: “mo\_dist”: mo\_disturbance, “mo-dist”: mo\_disturbance, “mo\_di\_th”: mo\_disturbance\_thonicke.

jsbach routine `mo_jsbach_interface::jsbach_inter_1d()`. But when run with DYNVEG the disturbances are called within the DYNVEG routine `mo_dynveg::fpc_daily()` because disturbances are computed at a daily time step. Whatever configuration, to compute the disturbance rates  $D_{fire}$  and  $D_{wind}$ , the routine `mo_disturbance::disturbed_frac()` is called. This routine needs to be called separately for wind-throw and vegetation fires. Internally, for computing  $D_{wind}$ , this routine calls `mo_disturbance_jsbach::broken_woody_frac_jsbach()`. And to compute  $D_{fire}$  either the routine `mo_disturbance_jsbach::burned_frac_jsbach()` from the simple JSBACH fire model, or the routine `mo_disturbance_jsbach::burned_frac_thonicke()` from the SPIFIRE model ('thonicke') is called.

At highest code level the main routine for performing the carbon and nitrogen relocations from disturbances is `mo_disturbance_jsbach::relocate_disturbed_carbon()` which is called twice, once for wind-throw and then for fires. Internally this routine calls the carbon relocation routine for wind-throw `mo_cbal_cools::relocate_carbon_damage()` or `mo_cbal_cools::relocate_carbon_fire()` for fire which then call the appropriate carbon and nitrogen relocation routines depending on the particular soil carbon and fire model used.

Table 8.3 lists important parameters of the disturbance models. These parameters should be changed with care, since they are crucial for the behaviour of DYNVEG. Accordingly, the disturbance parameters are important tuning parameters for DYNVEG.

# Chapter 9

## Natural land cover change

### Contents

---

9.1	Introduction . . . . .	137
9.2	Leading principles . . . . .	138
9.3	Dynamics of natural vegetation shifts . . . . .	139
9.4	Disturbances . . . . .	140
9.5	Bioclimatic limits . . . . .	141
9.6	Dynamics of inhospitable land (deserts) . . . . .	142
9.7	Potential and actual natural vegetation cover . . . . .	143
9.8	Carbon and nitrogen relocation following land cover change . . . . .	144
9.8.1	Basic logic of carbon and nitrogen relocation . . . . .	144
9.8.2	Relocation of carbon and nitrogen from natural vegetation changes . . . . .	145
9.8.3	Handling of carbon and nitrogen affected by the desert dynamics . . . . .	147
9.9	Diagnostics . . . . .	147
9.10	Implementation details . . . . .	148

---

### 9.1 Introduction

In interaction with climate the worldwide distribution of vegetation and deserts may change. The dynamics of such geographical shifts is simulated in JSBACH by the DYNVEG component [15]. Details of the model have been described in [111]. From a technical point of view, DYNVEG updates once a year

- ★ the global maps of *potential* cover fractions  $c_i^{pot}$  of the various *natural* PFTs used in JSBACH (see eq. (9.8)), and
- ★ the fraction  $veg_{max}$  of the grid boxes covered by vegetation (compare eqs. (1.1) and (9.7)).

The resulting maps represent the distribution of *potential* vegetation. In absence of anthropogenic land cover change (i.e. land-use change, see section 10) the potential vegetation fractions are also the *actual* vegetation fractions, i.e. in this case the cover fractions  $c_i$  (compare (1.5)) are directly obtained from DYNVEG.

Besides climate, natural land cover is also controlled by disturbances (vegetation fires, windbreak). These induce carbon and nitrogen fluxes not only directly (fire emissions and

litter production), but also indirectly by shrinking or expansion of the PFT cover fractions that lead to a reshuffling of carbon between different carbon and nitrogen pools. Accordingly, the following not only describes the representation of natural biogeographical changes in JSBACH, but also the implementation of the associated changes in vegetation carbon and nitrogen. Conversely, biogeography is also affected by carbon and nitrogen because wild fires develop only if enough litter carbon is available to catch fire. Therefore the land carbon and nitrogen dynamics and the biogeography are intimately coupled in JSBACH and must be tuned jointly to achieve realistic model behavior.

The following description of DYNVEG is largely taken from [111]. The parameters of the DYNVEG model are listed in table 9.3 at the end of this chapter.

## 9.2 Leading principles

DYNVEG is based on a number of principles. Mostly, these principles are common to many DGVMs. Nevertheless, to make the assumptions underlying DYNVEG and its implementation in JSBACH more transparent, it seems worth to state these principles explicitly:

**Increase/reduction of vegetation cover:** When individual plants die because they come to age or because of other causes (fire, wind-throw, diseases, pests), the space left by them can be taken by other plants. This is e.g. the cause for gap formation and subsequent plant successions in forests [6]. For PFTs this means that their coverage is reduced by natural death or by disturbances (in DYNVEG: fires, wind-throw) and increases by migration into spaces opened in this way. These open spaces are called in the following *uncolonized land* and the different vegetation types compete to colonize the open spaces. There is a second type of increase/reduction of vegetation cover, namely when regions inhospitable to vegetation expand or shrink (e.g. deserts). Changes in vegetation cover arising from this type of dynamics are assumed to affect all vegetation types equally, so that no competition is assumed here.

**Competition by growth form:** Part of the competitive advantages/disadvantages of woody vegetation as compared to grasses can be explained by their growth form: In the absence of disturbances trees and shrubs dominate because they take all the light by overgrowing the grasses at the bottom. Therefore, in DYNVEG woody (trees and shrubs) and non-woody (grasses) PFTs are distinguished<sup>1</sup> On the other hand, because of the buildup of stems the growth of woody types is slow as compared to grasses so that after disturbances grasses can migrate faster into open spaces. Accordingly, grasses conquer uncolonized land first, and only later woody types appear. These structural advantages/disadvantages are represented in the model by (i) allowing grasses to access uncolonized land faster than woody types, and (ii) by slowing down the expansion of grasses when uncolonized land gets rare, whereas the expansion rate of woody types is kept unaffected by the availability of uncolonized land – this slowing down assures that at low disturbance rates woody types dominate.

**Competition by productivity:** Net primary productivity (NPP) is the amount of carbon photosynthesized by plants that is available for growth and reproduction (see section 6.2). Comparing the NPP of different vegetation types, a higher NPP can thus be considered to be an indicator for a competitive advantage if not other aspects like the growth form are dominating competition. Therefore, in DYNVEG, competition within the classes of woody and non-woody types is modeled such that it is controlled by NPP: PFTs with higher NPP migrate into uncolonized land faster than PFTs with lower NPP.

<sup>1</sup>This classification is specified in the lctlib-file (see C.3) under the keyword `WOODY_PFT`.

**Physiological constraints:** To account for physiological constraints, PFTs are endowed with *bioclimatic limits*, i.e. climatic ranges in which a particular type of vegetation is able to exist. Such limits prevent only the *establishment* of vegetation, but once a vegetation has established and temperatures fall outside the range of bioclimatic limits they do not prevent the further *existence* of the vegetation.

**Inhospitable land:** Vegetation can establish only when climate is such that at least in some years NPP is positive. This criterion determines the extent of cold and hot “deserts”, or conversely, the amount of land available for vegetation.

**Universal presence:** It is assumed that every PFT is potentially present always and everywhere. This means, that at the level of PFTs the dispersal of seeds is assumed to be much faster than all other processes relevant for their expansion (“seeds are everywhere”).

### 9.3 Dynamics of natural vegetation shifts

For calculating changes in biogeography, the grid box is split into two parts, one that is covered with vegetation, called ‘vegetated area’, and the rest, that was called ‘inhospitable land’ above. In this way, the calculation of biogeographical changes is split into two separate problems: The calculation of the distribution of PFTs within the vegetated area, and the calculation of the extent of the vegetated area. This section (and the next two sections) describe how DYNVEG tackles the first problem, how DYNVEG solves the second problem is described in section 9.6.

The basic variables of DYNVEG describing the distribution of PFTs within the vegetated area are:

- $w_i$  : fraction of vegetated area covered by the  $i$ -th woody PFT,
- $g_i$  : fraction of vegetated area covered by the  $i$ -th non-woody (i.e. grass) PFT,
- $u$  : fraction of vegetated area currently uncolonized (without vegetation) because of natural mortality and disturbances (fires, wind-throw).

These cover fractions make up the whole vegetated area of a grid box so that they sum up to one:

$$(9.1) \quad u + \sum_{i=1}^{N^{(w)}} w_i + \sum_{i=1}^{N^{(g)}} g_i = 1,$$

Here,  $N^{(w)}$  and  $N^{(g)}$  are the number of natural woody and grass PFTs. By this relation,  $u$  can be determined from  $w_i$  and  $g_i$ , so that only the latter are considered as independent variables. These cover fractions are different from the PFT cover fraction  $c_i$  introduced in section 1.3.2; how they are related is described in section 9.7.<sup>2</sup>

The dynamics of the cover fractions  $w_i$  and  $g_i$  is governed by the coupled differential equations

$$(9.2) \quad \frac{dw_i}{dt} = \theta(u) \frac{w_i}{\tau_i^{(w)}} \frac{\overline{NPP}_i^\alpha}{\sum_{n \in \mathcal{W}} w_n \overline{NPP}_n^\alpha} - \frac{w_i}{\gamma_i^{(w)}} - w_i D^{(w)}, \quad i \in \mathcal{W} = \{1, 2, \dots, N^{(w)}\}$$

$$(9.3) \quad \frac{dg_j}{dt} = \frac{u g_j}{\tau_j^{(g)}} \frac{\overline{NPP}_j}{\sum_{n \in \mathcal{G}} g_n \overline{NPP}_n} - \frac{g_j}{\gamma_j^{(g)}} - g_j D^{(g)}, \quad j \in \mathcal{G} = \{1, 2, \dots, N^{(g)}\}.$$

<sup>2</sup>In the DYNVEG code, found in module `mo_dynveg`, the array `act_fpc` is used to represent the cover fractions  $w_i$  and  $g_i$ , while `bare_fpc` is used to handle  $u$ . Instead, throughout the JSBACH code, the cover fractions  $c_i$  are called `cover_fract`.

These equations are integrated by iterating their Euler discretization at a daily time step. In both equations the first term on the right hand side describes establishment, the second term reduction of vegetation cover by natural mortality, and the last terms represent reduction of vegetation cover by disturbances (fire, wind-throw) (see below). The establishment and natural mortality have their own characteristic time scales  $\tau_i^{(w)}$ ,  $\tau_i^{(g)}$ , and  $\gamma_i^{(w)}$ ,  $\gamma_i^{(g)}$  for woody types and for grasses, although for lack of knowledge generally  $\tau_i = \gamma_i$  has been chosen. For woody types, the value  $\tau_i^{(w)}$  is assumed to be identical to the turnover time  $\tau_W$  for wood carbon (compare eq. (6.8)). Mortality is assumed to be proportional to the abundance of the particular vegetation type. The function  $\theta(u)$  is the stepfunction, which is 1 for  $u > 0$  and otherwise zero.

Establishment happens only on uncolonized land, i.e. in this model there is no immediate gain of area by a particular PFT at the cost of area covered by another PFT – in such a case the equations would contain explicit competition terms of the form  $w_i w_j$ ,  $g_i g_j$ , or  $w_i g_j$ . In DYNVEG competition between woody types and grasses arises only indirectly by their different dynamics with respect to uncolonized land: whereas grass types establish proportionally to the presence of uncolonized land (see the term  $u g_j$  in (9.3)), the establishment of woody types is assumed to be independent of the extent of uncolonized land, with the technically necessary exception that it is set to zero when all uncolonized land is already occupied (therefore the step function in the term  $\theta(u) w_i$  in (9.2)). Giving grasses a larger expansion rate than woody types ( $\tau_i^{(g)} \approx 1$  year,  $\tau_i^{(w)} \approx$  several decades) this approach gives grasses a competitive advantage when large areas of uncolonized land are available, whereas woody types outcompete grasses when uncolonized land is rare. Since the availability of uncolonized land increases with the strength of disturbances (see below), DYNVEG implicitly assumes that the relative presence of grasses and woody types is controlled by the level of disturbance, in accordance with ecological theory [10].

Considering the groups of woody and non-woody types, the intra-group competition mechanism is different from the mechanism of inter-group competition: Within a group the different PFTs compete by their longterm NPP average  $\overline{NPP}_i$  (average over several years). This competition is implemented by weighting the expansion terms in (9.2) and (9.3) by the relative NPP of the particular PFT compared to total NPP within the particular vegetation group. When a PFT falls outside its bioclimatic limits (see section 9.5) its ability to expand is zero. Therefore in this case the respective value of  $\overline{NPP}_i$  is set to zero. The exponent  $\alpha$  is an additional control parameter by which competition within the class of woody PFTs is further controlled. For  $\alpha > 1$  PFTs with high NPP are given an additional migratory advantage. Accordingly,  $\alpha$  controls the time scale of succession, which is for tree species of the order of centuries. For typical relative differences in  $\overline{NPP}_i$  this is assured by choosing  $\alpha \approx 1.5$ .

Finally, there are terms describing disturbance losses by wildfires and wind-throw. These are described in the following section.

## 9.4 Disturbances

DYNVEG accounts for two types of disturbances: vegetation fires and wind-throw. These processes determine the disturbance rates  $D^{(w)}$  and  $D^{(g)}$  in eqs. (9.2) and (9.3) (units:  $years^{-1}$ ) which accordingly consist of separate contributions from fires and wind-throw:

$$(9.4) \quad D^{(i)} = D_{fire}^{(i)} + \begin{cases} D_{wind} & \text{for } i = w \\ 0 & \text{otherwise} \end{cases} \quad \text{with } i \in \{w, g\}.$$

Whereas the rate of canopy area affected by fire is different for grasses and woody types, wind-throw affects only woody types because of negligible aerodynamic resistance of grasses. The disturbance rate for wind is given by (8.1), while that for fire is either (8.2) or (8.7), depending on the fire model used.<sup>3</sup>

The disturbances by wildfires introduce an additional coupling between grasses and woody types because in both the simple JSBACH fire model (section 8.3.1) and in the SPITFIRE model (section 8.3.2) the fire disturbance rate depends on the *combined* amount of litter from woody and grass types. Moreover, in both fire models, the emergence and/or intensity of fire disturbances depends on thresholds so that interactions between the two groups of vegetation are also nonlinear. For SPITFIRE this has been discussed in [82].

## 9.5 Bioclimatic limits

The global distribution of vegetation is not only determined by competition, but also by climate related physiological constraints: it may e.g. be too cold or too hot for a particular vegetation type to survive. Although the DYNVEG dynamics is formulated independently of a particular set of PFTs – except that it needs both, woody and grass PFTs – the formulation of these constraints depends heavily on the chosen set of PFTs. In DYNVEG these climatic constraints are accounted for by setting the establishment term in (9.2) and (9.3) to zero if one of the limits is not fulfilled; table 9.1 lists typical values of bioclimatic limits for our standard set of PFTs.

The temperature limits for trees are taken from the LPJ model ([129]), which can partly be justified by observations [131, 63]. But whereas DYNVEG uses only two PFTs for extra-tropical trees, LPJ distinguishes between needle-leaved and broad-leaved forest as well as temperate and boreal forest. Accordingly, in the DYNVEG standard setup the bioclimatic ranges of extra-tropical forests had to be chosen wider than in LPJ: they were chosen such that they cover the full bioclimatic range of the extra-tropical forest types of LPJ. Thereby, in DYNVEG all extra-tropical forests are subject to the same bioclimatic limits (with the exception of  $TC_{min}$ ; see table 9.1) so that the geographic pattern of evergreen and deciduous forests is mostly determined by competition and not by numerous bioclimatic limits as in LPJ.

Shrubs represent marginal vegetation, restricted in growth either by very dry and hot or very cold conditions. Accordingly, the bioclimatic limits of raingreen shrubs are chosen to exclude them from areas with a cold winter and frequent frost, so that they appear only in dry tropical and sub-tropical regions. In contrast, deciduous shrubs, representing woody tundra, are excluded by bioclimatic limits from regions with a warm climate.

Grasses may grow everywhere, but regionally grasslands are dominated either by C3 or C4 species. To obtain a distribution of C3 and C4 grasses that roughly fits the observations [38], temperature limits for the coldest month are chosen such that geographically there is a transition region between their distribution. Accordingly, only in this transition region the two groups of grasses compete with another according to the dynamics of DYNVEG.

The geography of the bioclimatic limits may shift in a simulation with a change in climate. Therefore, at the beginning of each simulated year it is analyzed whether bioclimatic limits are still met by the different PFTs. This is checked by comparing the limits with the respective

---

<sup>3</sup>Note that there is a notational inconsistency for the fire disturbance rates: While for the simple JSBACH fire model only two disturbance rates exist,  $D_{fire}^{(w)}$  for woody types and  $D_{fire}^{(g)}$  for grass types, for SPITFIRE each PFT may have a different disturbance rate  $D_{fire}^{(i)}$ , where now the upper index  $i$  denotes the number of the PFT. Accordingly, for SPITFIRE the PFT dependence is notationally not properly reflected in eqs. (9.2), (9.3), and (9.4).

Table 9.1: Bioclimatic limits of the PFTs used in the standard setup. When one of these limits is violated, in eqs. (9.2) and (9.3) the respective establishment term is set to zero. More precisely: If  $TC$  and  $TW$  are the (average) temperature of the coldest and warmest months, establishment is suppressed if either  $TC < TC_{min}$ ,  $TC > TC_{max}$ , or  $TW > TW_{max}$ . Similarly, establishment is suppressed when the value of previous year’s growing degree days  $GDD$  is smaller than  $GDD_{min}$ . An entry “–” means that no limit is applied. These values are defined in the land cover library file (see appendix C.3).

	tropical trees		extra-tropical trees		shrubs		grasses		name in lctlib
	evergr.	decid.	evergr.	decid.	raingr.	decid.	C3	C4	
$TC_{min}$	15.5	15.5	-32.5	–	2.0	–	–	10.0	BCLIMIT_MIN_COLD_mmTEMP
$TC_{max}$	–	–	18.5	18.5	–	-2.0	15.0	–	BCLIMIT_MAX_COLD_mmTEMP
$TW_{max}$	–	–	–	–	–	18.0	–	–	BCLIMIT_MAX_WARM_mmTEMP
$GDD_{min}$	–	–	400	400	900	350	–	–	BCLIMIT_MIN_GDD
$GDD_{base}$	5	5	5	5	5	5	5	5	GDD_BASE

$TC_{min}$ : minimum temperature of coldest month [ $^{\circ}\text{C}$ ]

$TC_{max}$ : maximum temperature of coldest month [ $^{\circ}\text{C}$ ]

$TW_{max}$ : maximum temperature of warmest month [ $^{\circ}\text{C}$ ]

$GDD_{min}$ : minimum growing degree days [ $^{\circ}\text{C}\cdot\text{days}$ ]

$GDD_{base}$ : base temperature for computing GDD [ $^{\circ}\text{C}$ ]. This has the same meaning as  $T_{alt}$  in eq. (3.4).

climate variables. But before comparison, these climate variables are time-averaged to smooth out short term climate variations by using an exponentially decreasing memory (e-folding time  $\sim 20$  years) that allows gradual adaptation to the actual climate.<sup>4</sup> This is done for all limits based on monthly averaged temperature, namely  $TC_{min}$ ,  $TC_{max}$ ,  $TW_{max}$  (compare table 9.1). In contrast, for  $GDD_{min}$  just the previous year’s value is used.

## 9.6 Dynamics of inhospitable land (deserts)

DYNVEG includes a simple model to determine the extent of inhospitable land, which can be cool deserts like in the arctic, or hot deserts as the Sahara. The extent of inhospitable land is defined as  $(1 - veg_{max})$ , with  $veg_{max}$  being the fraction of a grid box where vegetation can grow (‘vegetated area’, compare section 1.3), and is therefore another central element determining land cover in JSBACH.<sup>5</sup>

The model for inhospitable land is based on the idea that deserts develop when NPP is repeatedly so low that vegetation cannot grow an extended canopy. The fraction of a grid box with substantial vegetation cover at least once in year  $y$  is estimated as

$$(9.5) \quad f(y) = \sum_{i \in \mathcal{W}} w_i \left(1 - e^{-a(Q_i)^b}\right) + \frac{g+u}{g} \sum_{i \in \mathcal{G}} g_i \left(1 - e^{-a(Q_i)^b}\right)$$

where  $g = \sum_{k \in \mathcal{G}} g_k$  is the total grass fraction of vegetation. The quantity  $Q_i$  appearing in the exponents is a proxy for leaf biomass (details below). Formula (9.5) is constructed such that for large leaf biomass,  $f(y)$  approaches 1 (compare eq. (9.1)) and is otherwise smaller than 1.<sup>6</sup> Hence, leaf biomass is taken here as a measure for colonization success.

<sup>4</sup>The time-averaging with exponentially decreasing memory is done similar to (9.7).

<sup>5</sup>The calculation of  $veg_{max}$  is performed in the routine `desert_fraction` of module `mo_dynveg`. Output of this routine is the variable `desert_fpc`. On glacier-free land `desert_fpc` corresponds to  $(1 - veg\_ratio\_max)$ , i.e.  $(1 - veg_{max})$ .

<sup>6</sup>The prefactor  $(g+u)/g$  boosts the contribution from grasses. This boost accounts for the contributions from the uncolonized land. Alternatively, a factor  $(w+u)/w$  could have been applied to the term arising from woody vegetation, but it is applied here to the grass term because uncolonized land is easier colonized

More precisely,  $Q_i$  is calculated as

$$(9.6) \quad Q_i = sla_i \frac{C_{G,i}^{max}}{3}.$$

Here  $C_{G,i}^{max}$  is the maximum living biomass<sup>7</sup> found in the green pool of PFT  $i$  in the considered year  $y$  so that  $C_{G,i}^{max}/3$  is approximately the leaf biomass since the green pool includes besides leaf carbon also root and sapwood biomass (compare section 6.3). By multiplication with the specific leaf area  $sla_i$ , this leaf carbon is converted into a kind of leaf area index computed here, in contrast to the phenology of JSBACH (section 3), from the carbon content. Accordingly,  $Q_i$  has the meaning of a leaf area index and since it is computed from the carbon in living vegetation it is a measure for growth success. The parameter  $a$  is chosen such that for  $Q_i = 1$  the expression  $1 - \exp(-a)$  gives a substantial vegetation cover, e.g. 85%. The parameter  $b$  describes the steepness of the transition between vegetation and desert and has been chosen to give a realistic distribution of deserts.

Next  $f(y)$  is used to derive  $veg_{max}$ . Since one year of low vegetation growth doesn't make a desert, a delayed development is assumed: For year  $y$  we calculate  $veg_{max}$  by

$$(9.7) \quad veg_{max} = \frac{1}{\mathcal{N}} \sum_{y'=-\infty}^y f(y') e^{-\frac{y-y'}{\tau_{desert}}} \quad \text{with} \quad \mathcal{N} = \left(1 - \exp^{-\frac{1}{\tau_{desert}}}\right)^{-1},$$

where  $\tau_{desert}$  is the time scale for the development of inhospitable conditions and  $\mathcal{N}$  normalizes the exponentially weighted average.

## 9.7 Potential and actual natural vegetation cover

This section describes how the DYNVEG cover fractions  $w_i$ ,  $g_i$ , and  $u$  are translated into the JSBACH cover fractions  $c_i$  introduced in section 1.3.<sup>8</sup> Following the definition of [142], the term ‘‘potential vegetation’’ denotes the distribution of vegetation that one would see under the ruling climate conditions in a world without humans. This is exactly what can be obtained from DGVMs in general [10], and DYNVEG in particular: a map of potential natural vegetation, which - in the absence of land use - corresponds to the actual land cover.

As indicated in the above list of basic principles, it is assumed that the competitive advantage of grasses compared to woody types is their expansion speed. This principle is used here to derive potential natural vegetation cover from the DYNVEG cover fractions, by assuming that only grass types expand within one year into areas formerly disturbed by vegetation fires or wind-throw. Hence, in terms of JSBACH cover fractions, fraction  $u$  of disturbed area is fully counted as grasslands. Moreover, it is assumed that all grass types existing in the considered grid box expand proportionally to their presence. Combining these considerations with equations (9.1) and (1.6) gives when accounting for (1.5)

$$(9.8) \quad c_i^{pot} = \frac{1}{veg_{max}} \begin{cases} g_i \left(1 + \frac{u}{\sum_{j \in \mathcal{G}} g_j}\right) & \text{for } i \in \mathcal{G} \\ w_i & \text{for } i \in \mathcal{W}, \end{cases}$$

by grasses and its physical properties are closer to grasslands than to woodlands or forests. Compare also eq. (9.8).

<sup>7</sup> $C_{G,i}$  is measured in units  $[mol(C)/m^2(canopy)]$  so that more precisely  $C_{G,i}^{max}$  is the maximum living biomass density.

<sup>8</sup>In module `mo_dynveg` the array `act_fpc` is used to represent  $w_i$  and  $g_i$ , while `bare_fpc` represents  $u$ .  $c_i$  has the name `cover_fract` in the code.

where  $\mathcal{W}$  denotes the woody types, and  $\mathcal{G}$  the non-woody types (grasses), and  $veg_{max}$  is given by (9.7). In the absence of land use,  $c_i^{pot}$  is the actual land cover  $c_i$ .

The relationship of DYNVEG cover fractions to the actual cover fractions in the presence of land use and land-use change is described in chapter 10.

## 9.8 Carbon and nitrogen relocation following land cover change

Changes in the distribution of vegetation and inhospitable land (deserts) necessarily induce changes in the distribution of carbon and nitrogen. The purpose of the present section is to describe how this relocation of carbon and nitrogen is described in JSBACH. Technically, the relocation is a re-distribution of carbon between equal carbon and nitrogen pools at different tiles. Therefore, the particular structure of the pool scheme and even the meaning of the pools is irrelevant for the relocation. Accordingly, there is nothing specific for carbon or nitrogen in this respect, so that the following description applies - with the exception of green carbon (compare 9.8.3) - to both carbon and nitrogen and a unified notation is used, denoting all carbon and nitrogen pools by  $X$ .

The relocation of carbon and nitrogen is performed at the beginning of each simulation year after the translation of the internal cover fractions of DYNVEG ( $w_i, g_i, u$ ) into those of JSBACH ( $c_i$ ). It is performed separately for changes in the  $c_i$  and for changes of  $veg_{max}$ . Before describing these two elements of carbon relocation in subsections 9.8.2 and 9.8.3, first the general logic for carbon and nitrogen relocation is introduced. It applies not only to the natural land cover change addressed in this chapter, but will also be referred to when describing carbon and nitrogen relocation induced by anthropogenic land cover change in chapter 10.

### 9.8.1 Basic logic of carbon and nitrogen relocation

As a basis for understanding the formulas in the sections below, in this section the general logic of carbon and nitrogen relocation is described. The relocation must assure carbon and nitrogen conservation. This sounds trivial, but is complicated in application to JSBACH because JSBACH handles all carbon and nitrogen in form of carbon densities, i.e. as mass per square meter, and not as mass alone. Moreover, JSBACH distinguishes between densities per grid box area, per vegetation area, and per canopy area (see end of section 1.3.2). All this introduces complications that need to be considered.

Considering a particular grid box, let  $X_i^{new}$  and  $X_i^{box,new}$  denote the carbon or nitrogen *canopy* and *box* densities of a pool  $X$  at tile  $i$  after the relocation (compare (1.12)). The carbon or nitrogen *mass* of this pool in the considered grid box is then  $A X_i^{box,new}$ , where  $A$  is the area of the grid box. This can as well be expressed using canopy density  $X_i^{new}$  by

$$(9.9) \quad A X_i^{box,new} = A veg_{max}^{new} c_i^{new} f_i^{corr} X_i^{new}$$

when applying the definitions from section 1.3.2. Here,  $A veg_{max}$  is the area of the vegetated part of the grid box and  $c_i^{new}$  is the cover fraction of tile  $i$  with respect to vegetated area. Since it is assumed that all carbon and nitrogen is located only at canopy area and not in canopy gaps, the canopy gap correction factor  $f_i^{corr}$  defined by equation 1.7 is applied. – A similar equation holds for the carbon and nitrogen before relocation.

The carbon/nitrogen after relocation is the carbon/nitrogen remaining in the pool plus the carbon/nitrogen shuffled from other pools into it. The carbon/nitrogen remaining in the

pool  $X$  can be written as a fraction  $\alpha$  of the carbon/nitrogen  $A X_i^{box,old}$  that was in the pool before the change in vegetation distribution. In the simplest case

$$(9.10) \quad \alpha_i = 1,$$

i.e. all carbon/nitrogen of a tile  $i$  remains in it. In another case, the amount of carbon/nitrogen in a tile changes proportional to its shrinkage/expansion in area. Calling  $A_i^{old}$  and  $A_i^{new}$  the areas of tile  $i$  before and after its shrinkage, in this case

$$(9.11) \quad \alpha_i = \frac{A_i^{new}}{A_i^{old}} = \frac{veg_{max}^{new} c_i^{new}}{veg_{max}^{old} c_i^{old}}.$$

Calling  $A R_{\triangleright X,i}^{box}$  the carbon/nitrogen mass relocated from other pools to  $X_i^{new}$ , one thus can write

$$(9.12) \quad A X_i^{box,new} = \alpha_i A X_i^{box,old} + A R_{\triangleright X,i}^{box}.$$

Obviously  $A$  drops out. Applying the formulas for the box values one thus finds

$$(9.13) \quad X_i^{new} = \alpha_i X_i^{old} \frac{c_i^{old} veg_{max}^{old}}{c_i^{new} veg_{max}^{new}} + \frac{R_{\triangleright X,i}^{box}}{veg_{max}^{new} c_i^{new} f_i^{corr}}.$$

This equation is based on two assumptions, namely that carbon and nitrogen transfers are proportional to the area changes, and that carbon and nitrogen are shuffled only between pools of equal type (from one tile to another). With one exception in the context of the desert dynamics (compare 9.8.3), both assumptions are considered to be valid for carbon and nitrogen relocations from natural land cover changes discussed in this chapter. For anthropogenic land cover change both assumptions must be relaxed and equation (9.13) needs to be generalized (see section 10.3.2).

### 9.8.2 Relocation of carbon and nitrogen from natural vegetation changes

This section describes the relocation of carbon and nitrogen due to changes in the cover fractions  $c_i$  alone, i.e. when  $veg_{max}$  remains unchanged. Hence, in this section  $veg_{max}^{new}$  and  $veg_{max}^{old}$  of equation (9.13) are identical.

The relocation of carbon and nitrogen in vegetation and soils is handled slightly different so that two cases need to be considered:

- ★ For **vegetation carbon and nitrogen** it is assumed that while vegetation cover is shrinking the respective tiles keep all their carbon/nitrogen. On shrinking tiles the density of living carbon/nitrogen thus increases because the same carbon/nitrogen is condensed on a smaller area. On expanding tiles, it is assumed that the area gained is empty of vegetation carbon/nitrogen, resulting in a lowering of the carbon density in the particular tile upon expansion.
- ★ The **Carbon and nitrogen of litter and soils** is assumed to stay physically where it is (no matter whether the CBALANCE or Yasso scheme for litter and soil carbon/nitrogen is used) and thus must be relocated between tiles when their area changes. On shrinking tiles, the litter/soil density stays constant. Since no detailed information is available on the exact tile to tile relation for the redistribution of area, all affected litter and soil carbon/nitrogen is distributed from the shrinking tiles proportionally to the area gained by the expanding tiles. This approach is consistent with JSBACH's general assumption of the vegetation types being homogeneously mixed over the vegetated area.

Following these considerations, the total amount of **vegetation carbon/nitrogen** in a particular tile remains unchanged. Accordingly, the factor  $\alpha$  in (9.13) is with (9.10) equal 1. And since there is no addition of carbon/nitrogen from other pools, application of this equation gives

$$(9.14) \quad X_i^{new} = X_i^{old} \frac{c_i^{old}}{c_i^{new}}$$

where  $X_i$  denotes any vegetation pool  $X$  (green, wood, reserve, mobile) of tile  $i$ . The factor  $c_i^{old}/c_i^{new}$  is  $> 1$  for shrinking tiles and  $< 1$  for expanding tiles and thus accounts for the necessary increase/decrease of carbon/nitrogen density in both cases.<sup>9</sup>

For the **litter and soil carbon/nitrogen** the situation is a bit more complicated because of the necessary relocation of carbon/nitrogen between different tiles. Upon expansion all litter and soil carbon/nitrogen stays in the particular tile, i.e. in application of (9.13) one has with (9.10)  $\alpha_i = 1$ . But upon shrinkage carbon/nitrogen is lost proportional to the remaining area  $A_i^{new}$  of the tile. Hence in this case the reduction factor from (9.11) must be applied so that (9.13) gives<sup>10</sup>

$$(9.15) \quad X_i^{new} = \begin{cases} X_i^{old} \frac{c_i^{old}}{c_i^{new}} + R_{\triangleright X,i} & \text{if } c_i^{new} > c_i^{old} \\ X_i^{old} & \text{otherwise} \end{cases},$$

where the shortcut

$$(9.16) \quad R_{\triangleright X,i} = \frac{R_{\triangleright X,i}^{box}}{veg_{max} c_i^{new} f_j^{corr}}$$

has been used. Here  $X$  denotes any soil or litter pool and  $R_{\triangleright X,i}^{box}$  is the total carbon/nitrogen relocated into pool  $X$  on tile  $i$ .<sup>11</sup> To compute  $R_{\triangleright X,i}^{box}$  one first has to determine the total carbon/nitrogen that needs to be relocated. With  $\alpha_i = c_i^{new}/c_i^{old}$  from above being the fraction that remains in a tile,  $1 - \alpha_i$  is the fraction lost. Accordingly, for the considered pool type  $X$  the total carbon or nitrogen to be relocated is

$$(9.17) \quad R^{box} = \sum_{j \in shrink} \left(1 - \frac{c_j^{new}}{c_j^{old}}\right) X_j^{old} c_j^{old} f_j^{corr} veg_{max},$$

where the sum runs only over shrinking tiles because these are the source of carbon/nitrogen that needs to be redistributed. Next one has to determine how  $R^{box}$  has to be distributed to the expanding tiles. This is done proportional the area gained. Let  $A_j^{gain}$  denote the canopy area gained by an expanding tile  $j$ , it is given as  $A_j^{gain} = A veg_{max} f_j^{corr} (c_j^{new} - c_j^{old})$ . Accordingly, the fraction of total carbon/nitrogen that needs to be redistributed to a shrinking tile  $i$  is

$$(9.18) \quad \beta_i = \frac{A_i^{gain}}{\sum_{j \in expand} A_j^{gain}} = \frac{f_i^{corr} (c_i^{new} - c_i^{old})}{\sum_{j \in expand} f_j^{corr} (c_j^{new} - c_j^{old})}.$$

<sup>9</sup>To prevent division by zero, all calculations of cover fractions are made such that they don't fall below a minimum value. This minimum value is hard coded in module `mo_land_surface` as parameter `fract_small`. The value is chosen sufficiently large so that the range of representable numbers is not exceeded upon division, but sufficiently small (e.g.  $10^{-10}$ ) to assure that the amount of carbon or nitrogen stored in these minute tiles remains negligible.

<sup>10</sup>Alternatively one can argue for the case of shrinkage that the carbon/nitrogen density remains unchanged.

<sup>11</sup>Actually,  $R_{\triangleright X,i}^{box}$  is a carbon/nitrogen density and  $AR_{\triangleright X,i}^{box}$  is the associated carbon/nitrogen mass. But since the box area  $A$  drops out in all calculation it is reasonable to use the terms 'total carbon' and 'total nitrogen' also for densities.

Hence inserting

$$(9.19) \quad R_{\triangleright X,i}^{box} = \beta_i R^{box}$$

into (9.16) finally gives

$$(9.20) \quad R_{\triangleright X,i} = \left(1 - \frac{c_i^{old}}{c_i^{new}}\right) \frac{\sum_{j \in shrink} (c_j^{old} - c_j^{new}) f_j^{corr} X_j^{old}}{\sum_{j \in expand} (c_j^{new} - c_j^{old}) f_j^{corr}}$$

### 9.8.3 Handling of carbon and nitrogen affected by the desert dynamics

After a change in the extent of inhospitable land, i.e. after a change in  $veg_{max}$ , carbon and nitrogen are assumed to stay in their pool at the particular tile, i.e. no relocation between pools of different tiles is necessary. Desert dynamics thus does not change carbon and nitrogen masses in the different pools, only their densities change.

The desert dynamics changes  $veg_{max}$ , while all cover fractions  $c_i$  remain unchanged<sup>12</sup>. It is seen from eq. (9.13) that because of  $\alpha = 1$  (compare (9.10)) this means a rescaling of the carbon/nitrogen densities  $X_i$  of all pools according to

$$(9.21) \quad X_i^{new} = \frac{veg_{max}^{old}}{veg_{max}^{new}} X_i^{old}.$$

However, it needs to be taken into account that the green (living) pool cannot exceed the amount  $C_{G,i}^{max}$  as given by (6.13). The green carbon exceeding this limit, namely

$$(9.22) \quad \Delta C_{G,i} = \max(0, C_{G,i}^{new} - C_{G,i}^{max}),$$

is not kept in the green pool but added to the leaf litter. Hence, in such a case  $C_{G,i}^{new}$  is set to  $C_{G,i}^{max}$ . Concerning the fate of  $\Delta C_{G,i}$ : If the CBALANCE soil scheme is used  $\Delta C_{G,i}$  is added to  $C_{LGA}$ , while if the Yasso scheme is used,  $\Delta C_{G,i}$  is distributed to the four above ground green Yasso pools according to the  $\eta_i$  coefficients (see eq. (6.25)).<sup>13</sup> The nitrogen of the green pool is handled accordingly to keep the N:C ratio of the green pools constant.

## 9.9 Diagnostics

Table 9.2 lists some useful output variables of DYNVEG.

Table 9.2: Useful output variables from DYNVEG

name in output	symbol	units	stream	meaning
cover_fract_pot	$c_i^{pot}$	–	jsbach	potential cover fraction of natural PFTs
cover_fract	$c_i$	–	jsbach	actual cover fractions
veg_ratio_max	$veg_{max}$	–	jsbach	fraction of grid box covered with vegetation
bio_exist	–	–	veg	logical indicating PFTs participating in the dynamics of eq. (9.2) and (9.3).

<sup>12</sup>This is because the  $c_i$  denote cover fractions of the vegetated area and not of grid area.

<sup>13</sup>Theoretically carbon could also be put into the green humus pool, but the respective distribution coefficient  $\eta_h$  is 0 for all PFTs (compare the entries under the keyword `LeafLit_coef` in the `lctlib`-file displayed in appendix C.3).

## 9.10 Implementation details

The code of the dynamic part of the DYNVEG model is found in `mo_dynveg.f90`, bioclimatic limits and the long term averaging of NPP are performed in `mo_climbuf.f90`, and carbon and nitrogen relocation induced by vegetation shifts is handled in `mo_cbal_cpools.f90`.

The dynamic equations of DYNVEG (see sections 9.3 and 9.6) are implemented as an Euler discretization that are evaluated once a day. A more sophisticated discretization scheme is not necessary, because the dynamics is rather slow compared to the daily Euler time step. In contrast, the transformation of the DYNVEG cover fractions  $w, g, u$  into the basic JSBACH cover fractions  $c_i$  and  $veg_{max}$ , i.e. the evaluation of (9.8), is performed only once a year.

The parameters of DYNVEG are listed in table 9.3. If parameters are not PFT-specific, they are hard-coded. Nevertheless, some of these hard-coded parameters can be overwritten by namelist entries (see table 9.3).

Since DYNVEG contains century long time scales, running the global vegetation distribution to equilibrium is computationally quite demanding. For this purpose an acceleration mechanism is implemented that speeds up the dynamics by the factor `accelerate_dynveg` from the namelist `dynveg_ctl`. This simply enforces a reduction of all time constants so that the dynamics speeds up.

Table 9.3: Parameters of DYNVEG

variable name	name in code/file	units	typical value	defined in
<i>dynamics:</i>				
$\tau_i^{(w)} = \gamma_i^{(w)}$	TAU_CPOOL_WOODS	years	12-60	lctlib
$\tau_i^{(g)} = \gamma_i^{(g)}$	TAU_CPOOL_WOODS <sup>1</sup>	years	1	lctlib
<i>bioclimatic limits:</i>				
	sum_npp_min	kg(C)/m <sup>2</sup>	10 <sup>-12</sup>	mo_dynveg
	tree_fpc_max	–	1	mo_dynveg
	grass_mortality	1/day	0.01/365	mo_dynveg
	npp_nonlinearity	–	1.5	mo_dynveg
	desert_extend	–	0.65	mo_dynveg
	desert_margin	–	2	mo_dynveg
	tau_desert	years	50	mo_dynveg
	life_to_estab	years	20	mo_dynveg
<i>dynamics of inhospitable land:</i>				
$a/3$	desert_extend	–	0.65	mo_dynveg
$b$	desert_margin	–	2	mo_dynveg
$\tau_{desert}$	tau_desert	years	50	mo_dynveg
$sla$	specificLeafArea_C	m <sup>2</sup> (leaf)/mol(C)	0.11-0.45	lctlib

<sup>1</sup>Despite the name TAU\_CPOOL\_WOODS, this variable also contains the values for grass types. LCT-library file (see appendix C.3)

# Chapter 10

## Anthropogenic land cover change

### Contents

---

10.1	Introduction . . . . .	150
10.2	Change in vegetation distribution . . . . .	151
10.2.1	The transition model: land-use change as a sequence of transition matrices . . . . .	151
10.2.1.1	Mathematical formulation of the problem . . . . .	153
10.2.1.2	Application of the priority rule . . . . .	155
10.2.1.3	Daily land cover transitions . . . . .	157
10.2.1.4	PFT transitions . . . . .	158
10.2.1.5	Assuring non-zero cover fractions . . . . .	160
10.2.1.6	Interplay between land use transitions and natural vegetation dynamics . . . . .	161
10.2.2	The maps model: land-use change as a sequence of land cover maps . . . . .	164
10.2.2.1	Representation of the maps model as (net) transitions . . . . .	165
10.2.2.2	Net transitions from gross transitions . . . . .	166
10.3	Carbon and nitrogen relocation . . . . .	167
10.3.1	Exclusively active or passive relocations . . . . .	168
10.3.2	The ‘litter pool LUC scheme’: Relocation to litter pools . . . . .	172
10.3.3	The ‘anthro pool LUC scheme’: Relocation to product pools . . . . .	173
10.3.4	Harvest . . . . .	176
10.3.4.1	Fate of harvested carbon and nitrogen in the ‘litter pool LUC scheme’	177
10.3.4.2	Fate of harvested carbon and nitrogen in the ‘anthro pool LUC scheme’	178
10.4	Diagnostics . . . . .	180
10.5	Implementation details . . . . .	180
10.5.1	Code structure . . . . .	180
10.5.2	Control of model setup by namelist switches . . . . .	181
10.5.3	Model parameters . . . . .	181

---

## 10.1 Introduction

Changes in land cover have biogeophysical and biogeochemical consequences. To the former belong changes in albedo and surface roughness, for example, due to de- and reforestation. The latter are related primarily to the associated carbon and nutrient cycling, so that carbon and nutrients are taken up from the environment and released from the vegetation to the atmosphere and the soil compartments. Accordingly, the task of a model for anthropogenic land cover change is twofold:

1. To change the geographic distribution of PFTs, i.e. to change the cover fractions  $c_i$  (eqn. (1.5)), and
2. to perform the associated relocations of carbon and nutrients between the different pools and to the atmosphere.

In JSBACH two different models for anthropogenic land cover change are implemented. They differ by the type of input data used: The first model (“maps model”) is based on a sequence of global maps of land cover, one for each year. In the presence of more than two PFTs in a grid box, in such an approach one is missing the information which PFT has to “pay” for the extension of another PFT. Such an information is provided by the *New Hampshire Land-Use Harmonization (LUH) project*<sup>1</sup> which was first developed for CMIP in preparation of the 5<sup>th</sup> IPCC Assessment Report to have a common data format for land-use change forcing in the different scenarios and has since been updated for CMIP6. This protocol is the basis for the second land-use change model implemented into JSBACH (“transition model”). This second model uses as input a sequence of maps of so called land use transitions [65]. The difference between the two approaches gets most obvious when considering a situation where the land use transitions are such that before and after the transition the area of each PFT is identical. An example could be the simultaneous conversion of forests to pasture, pasture to croplands, and croplands to forest within a grid box. Such a cyclic conversion produces relocation of carbon and nutrients between different pools, even though the considered PFT distribution is identical before and after the transition. Accordingly, both land-use change models give the same changes in biophysical properties upon land-use change, but they differ in the biogeochemical consequences.

Every anthropogenic land cover change, i.e. land-use change (LUC), is associated with a loss of biomass. JSBACH implements two different ways of treating its carbon and nitrogen content. In the first approach (“litter pool LUC scheme”), being for long the only scheme for land-use change implemented in JSBACH, the carbon and nitrogen from the areas influenced by land-use changes is partly released to the atmosphere and partly added to the litter pools of the expanding PFTs. The other approach implements additional anthropogenic carbon pools according to the so called Grand Slam Protocol [90]. Here in particular the vegetation carbon affected by the land-use change is only partly left as litter in place, most of it is transferred into three specific anthropogenic carbon pools whose turnover times are determined by human activities: a pool representing fast human consumption, a pool for short-lived products (such as paper), and a pool long-lived products (houses, furniture etc.). This way to treat carbon affected by land-use change is called “anthro pool LUC scheme” in the following.

Besides *land-use change*, also *land use* has biogeophysical and biogeochemical consequences. Some of the biophysical effects from land use are part of the JSBACH implementations for calculating albedo (section 4.1) and surface roughness (section 4.2) where croplands

---

<sup>1</sup>For the New Hampshire Land-Use Harmonization project see <http://luh.umd.edu>.

and pastures enter with characteristics different from forests and grasslands. Other biophysical effects, like the consequences for water holding capacity of soils or albedo differences due to different management practices, are not represented in JSBACH. Biochemically important is harvest. JSBACH accounts explicitly for the carbon and nitrogen fluxes from harvesting woody vegetation (section 10.3.4). Harvest of crops is not part of this harvest model, but implicit to the carbon and nitrogen relocations following changes in crop LAI (see eqn. (6.18) and the section on the crop phenology 3.2.7).

The present description of the JSBACH implementation of anthropogenic land cover change is complicated by the many possible model configurations that arise from the alternative submodels, namely the two models representing land-use change (“maps” and “transitions”), the two ways to handle the carbon and nitrogen affected by land-use change (“litter pool LUC scheme” and “anthro pool LUC scheme”), plus the two alternative models representing carbon (and the related nitrogen) in litter and soil (“CBALANCE” and “Yasso”). Accordingly, anthropogenic land cover change can be run in 8 different variants that need to be described here. To cope with this diversity, the present chapter is divided into two main parts along the distinction made at the beginning of this introduction: in 10.2 changes in biogeography are thematized with separate subsections for the transitions and maps models, while the carbon and nitrogen handling is tackled in the subsequent section 10.3. To describe the carbon and nitrogen handling, advantage is taken of the possibility to represent the maps model by transitions. The remaining model alternatives are addressed in further subsections of the carbon and nitrogen part, where also land use in the form of harvest is discussed.

## 10.2 Change in vegetation distribution induced by land-use change

This section describes how the two models for land-use change implemented in JSBACH modify the cover fractions (1.5).

### 10.2.1 The transition model: land-use change as a sequence of transition matrices

The following description of the “transition model” of land-use change largely follows [111]. The model is driven by input maps provided by the New Hampshire Land-Use Harmonization (LUH) project[65, 66]. These maps provide fractions of area converted from one land cover class to another (“transitions”) from one year to the next.

The Harmonized Protocol is not based on PFTs, but uses a classification characterizing vegetation cover only with respect to land use: the first version of the LUH data “LUH1” used two classes for natural vegetation (*primary* and *secondary*), and two classes of agricultural land cover: *crops* and *pastures*.<sup>2</sup> By the difference between primary and secondary natural vegetation the LUH data provides information on land use of natural vegetation: whereas lands with secondary natural vegetation show traces of current or past land use (e.g. in the form of wood harvest), primary natural vegetation has never been touched by man.<sup>3</sup>

---

<sup>2</sup>The second version “LUH2” is much more complex and further distinguishes forested and non-forested primary and secondary lands, managed pasture, rangeland, urban land, as well as five different cropland types. Since the transition model was developed using the LUH1 data, the LUH2 data is aggregated into the four LUH1 categories in a preprocessing step before used in JSBACH.

<sup>3</sup>Accordingly, in this protocol transitions *to* primary lands do not exist.

In JSBACH this difference between primary and secondary lands is irrelevant,<sup>4</sup> so that the two classes of primary and secondary vegetation are lumped together into one large class of *natural* vegetation. This is done in a preprocessing step so that the actual data used from the (reduced) Harmonized Protocol to drive JSBACH contain only transitions between natural vegetation (N), crops (C), and pasture (P).<sup>5</sup>

The major challenge in implementing the Harmonized Protocol is to derive the much more detailed information on land use transitions between PFTs from the land use transitions between the three broad land cover classes. To achieve this higher degree of detail additional rules are required. The selection of such rules is left to the modeler using the Harmonized Protocol. JSBACH employs the following set of *land distribution rules* to derive the PFT transitions:

1. *Pasture rule*: Pastures expand first into grasslands before other natural vegetation is touched. For croplands there is no such priority. On abandonment of agricultural lands these rules are reversed.
2. *Pathway rule*: This rule concerns the relative extent of plants with C3 and C4 photosynthetic pathways: On conversions the ratio between C3 and C4 types is kept fixed among grasslands, among crops, as well as among pastures.
3. *Rule of equal relative gains and losses*: After accounting for the two rules above: following the expansion of crops and pastures, all natural PFTs undergo the same relative loss. Similarly, on back-conversion of agricultural lands to natural, the relative gain of natural PFTs with respect to the area maximally available for their extension is equal for all PFTs.

The rationale for the pasture rule comes from the higher effort necessary to chop down trees to gain agricultural lands than making use of grasslands: The pasture rule plays only a role in mixed tree/grass environments, because only there farmers have a choice. In such mixed environments forest sites have typically richer soils so that they are more suitable for crop growth; this may justify the effort of deforestation. In contrast, for grazing of livestock the advantage of a higher productivity at forest sites is marginal, so that it is easier to use grass sites directly.<sup>6</sup> This latter remark explains at least partly also the pathway rule: Most probably the effort for conversion between pastures and grasslands is kept low so that grasses are left in place without sowing new grass types. For crops the situation may be different, but without incorporating additional information for another distribution of gains and losses this rule is still a pragmatic choice. The same holds true for the rule of equal relative gains and losses: Because information on a possible preferential selection of one PFT over another during land conversion is missing, there is not much alternative to this rule.

---

<sup>4</sup>Including this difference makes only sense if there is a process implemented for which this difference is relevant, e.g. forestry could be such a process but currently this is not simulated by JSBACH.

<sup>5</sup>The matrix for transitions of the reduced set of land cover classes N, C and P is only incompletely determined by the transition values for the full set of primary, secondary, and crop and pasture land cover classes – in addition, the initial distribution of primary and secondary lands in each grid box is needed. Since in JSBACH the difference between primary and secondary lands is ignored, it would be unnatural to read in this information. For this reason the (reduced) transition matrix for N, C and P is determined by preprocessing the original data.

<sup>6</sup>The consequences of the pasture rule as compared to a rule without such a preferential allocation of pastures on grasslands has been discussed in [111].

### 10.2.1.1 Mathematical formulation of the problem

Let  $T$  denote the (reduced) transition matrix of the Harmonized Protocol, i.e. an element  $T_{i \triangleright j}$  denotes the fraction of area of land cover class  $i$  converted within one year to land cover class  $j$ ; the land cover classes  $i, j$  used here are natural land (N), crops (C), and pastures (P). Originally, the Harmonized Protocol refers to fractions of area in a grid box, but since box area and vegetated area are linearly related (see (1.5)), the transition matrix can as well be applied directly to the fractions of vegetated area. This is how the transition matrix is used in the following.<sup>7</sup>

Let  $c_N$ ,  $c_C$ , and  $c_P$  denote the fractions of the vegetated part of a grid box covered by vegetation from classes N, C, and P. Then the (reduced) Harmonized Protocol describes the annual transition in land cover by providing the matrix  $T$  for computing every year new cover fractions

$$(10.1) \quad \begin{pmatrix} c'_N \\ c'_C \\ c'_P \end{pmatrix} = \begin{pmatrix} T_{N \triangleright N} & T_{C \triangleright N} & T_{P \triangleright N} \\ T_{N \triangleright C} & T_{C \triangleright C} & T_{P \triangleright C} \\ T_{N \triangleright P} & T_{C \triangleright P} & T_{P \triangleright P} \end{pmatrix} \begin{pmatrix} c_N \\ c_C \\ c_P \end{pmatrix},$$

where  $0 \leq T_{i \triangleright j} \leq 1$ .<sup>8</sup> Since N, C, and P make up the whole vegetation in a grid box the land use transitions only re-partition the vegetated part of the grid box into these three cover types leaving the total extent of vegetation in the grid box unchanged so that

$$(10.2) \quad c'_N + c'_C + c'_P = c_N + c_C + c_P.$$

This relation implies the normalization<sup>9</sup>

$$(10.3) \quad \sum_{k \in \{N, C, P\}} T_{i \triangleright k} = 1 \quad \text{for } i \in \{N, C, P\},$$

i.e. the land of a particular class  $i$  can be distributed only once. Note that the diagonal elements  $T_{i \triangleright i}$  denote that fraction of a class that remains unconverted.

---

<sup>7</sup>Hidden here is a general problem pertaining to the usage of data from the Harmonized protocol in Earth system simulations. Hurtt et al. [66] derived the transition information by using a potential vegetation derived for fixed climate, representative for the years 1987-88. Hence, the assumptions on climate are inconsistent to scenario simulations with Earth system models. Concerning JSBACH, in scenario simulations the extent of vegetation in a grid box, i.e.  $veg_{max}$ , may change in time as described in the previous chapter (see eq. (9.7)). Therefore, applying the transition matrices of the Harmonized Protocol as described above to the cover fractions instead of grid box area thus means that implicitly it is assumed that shrinking/expanding vegetation cover also leads to larger/smaller areas converted upon land-use change as compared to the Harmonized Protocol. Since generally an inconsistency between the Harmonized Protocol and Earth System simulations cannot be avoided, this assumption seems to be as plausible as assuming that climate change has no influence on the transitions. Moreover, this implicit assumption has even some plausibility since human activities are closely related to the availability of natural resources. Finally, this difference must be small under moderate climate change as e.g. in simulations for historical climate. Only for strong climate change choosing one or the other assumption may be of relevance.

<sup>8</sup> $T_{i \triangleright j} \leq 1$  is actually not immediately obvious: Today large areas that have been deforested hundreds of years ago are afforested. Formally, this deforestation and afforestation may together be considered as a single land use transition, and obviously, in such a transition e.g.  $\Delta_{P \triangleright N}$  may be much larger than  $c_P$ . Here, obviously, the time step of the land use protocol is of importance: It is plausible that within a year farmers e.g. do not deforest land for growing crops, and afforest the same land in the same year. Hence, a time step of one year, as used in the LUH data, is consistent with the assumption  $T_{i \triangleright j} \leq 1$ . And this is also reflected in the LUH data as provided by the New Hampshire group.

<sup>9</sup>Proof:  $\sum_k c'_k = \sum_k \sum_i T_{i \triangleright k} c_i = \sum_i c_i \sum_k T_{i \triangleright k}$ . But  $\sum_k c'_k$  must be equal  $\sum_i c_i$ , independently of the values of the  $c_i$ . Therefore  $\sum_k T_{i \triangleright k} = 1$ .

The obvious aim is to translate eq. (10.1) into an analogous equation for PFTs. The associated transition matrix will be called  $t$  and its elements  $t_{i \triangleright j}$  denote the fraction of area of PFT  $i$  converted within one year to PFT  $j$ . Because of the tiling of grid boxes in JSBACH (see section 1.3), the set of PFTs allowed in a grid box is restricted to a subset of all possible PFTs. Therefore, in the equation analogous to eq. (10.1) not all PFTs, but only those allowed in a grid box have to show up. Hence, calling  $c_i$  the fraction of a grid box covered by the  $i$ -th PFT in a grid box the aim is to translate (10.1) into

$$(10.4) \quad \begin{pmatrix} c'_1 \\ c'_2 \\ \vdots \\ c'_K \end{pmatrix} = \begin{pmatrix} t_{1 \triangleright 1} & t_{2 \triangleright 1} & \dots & t_{K \triangleright 1} \\ t_{1 \triangleright 2} & t_{2 \triangleright 2} & \dots & t_{K \triangleright 2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{1 \triangleright K} & t_{2 \triangleright K} & \dots & t_{K \triangleright K} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{pmatrix},$$

where  $K$  is the number of PFTs in the particular grid box considered. Obviously, equations analogous to (10.2) and (10.3) hold for PFT related cover fractions and elements of the transition matrix  $t$ .

To obtain the transition matrix  $t$  from the transition matrix  $T$  according to the above mentioned rules, a complicated extremization problem has to be solved: According to the pasture rule the transitions from pastures to grasslands have to be maximized, and the back transitions minimized. Here natural vegetation is implicitly divided into grasslands and other natural vegetation; these types of vegetation cover will be abbreviated by “G” and “F”, respectively, in the following, where “F” stands for “Forest” well knowing that natural non-grassland vegetation can also be different from forests, like shrublands or tundra. With this notation the associated cover fractions obey

$$(10.5) \quad c_N = c_G + c_F.$$

It turns out to be convenient to solve first a reduced transition problem between the cover classes G, F, P, and C, before continuing with the full problem for all PFTs. Accordingly, first the (extended) 4x4 transition matrix  $T$  describing the dynamics between these four land cover classes has to be determined:<sup>10</sup>

$$(10.6) \quad \begin{pmatrix} c'_G \\ c'_F \\ c'_C \\ c'_P \end{pmatrix} = \begin{pmatrix} T_{G \triangleright G} & T_{F \triangleright G} & T_{C \triangleright G} & T_{P \triangleright G} \\ T_{G \triangleright F} & T_{F \triangleright F} & T_{C \triangleright F} & T_{P \triangleright F} \\ T_{G \triangleright C} & T_{F \triangleright C} & T_{C \triangleright C} & T_{P \triangleright C} \\ T_{G \triangleright P} & T_{F \triangleright P} & T_{C \triangleright P} & T_{P \triangleright P} \end{pmatrix} \begin{pmatrix} c_G \\ c_F \\ c_C \\ c_P \end{pmatrix}.$$

Because the four elements of the lower right 2x2 submatrix do not involve transitions to or from F or G they are identical to the respective entries in (10.1) and are thus directly known from the (reduced) Harmonized Protocol. Moreover

$$(10.7) \quad T_{F \triangleright G} = T_{G \triangleright F} = 0$$

because in our implementation on land-use change humans do not induce direct transitions between natural land cover classes.<sup>11</sup> The other matrix elements are unknowns of the problem, as well as the cover fractions  $c_G$ ,  $c_F$ ,  $c'_G$ ,  $c'_F$ , although of these the first two are related by (10.5) and the second two obey analogously

$$(10.8) \quad c'_N = c'_G + c'_F.$$

<sup>10</sup>Here and in the following the same symbol  $T$  is used for the (reduced) 3x3 and the (extended) 4x4 transition matrix, because several elements are identical and the meaning is always recognizable from the context.

<sup>11</sup>This is actually different from the original Harmonized Protocol, where transitions from primary to secondary vegetation are induced by harvest. This harvest is treated in JSBACH independently.

### 10.2.1.2 Application of the priority rule

In this section the 4x4 matrix for the extended land use transitions (10.6) is derived from the 3x3 matrix for the reduced land use transitions (10.1) by applying the pasture rule. In the following the term “F-lands” is used for land cover of type “F”.

Technically, the pasture rule means that

**Agricultural expansions** proceeds in the following order of conversions:

1.  $G \rightarrow P$ : For pastures first grasslands are used.
2.  $F \rightarrow P$ : Only if grasslands are exhausted, pastures are established on F-lands.
3.  $G + F \rightarrow C$ : For croplands the remaining grass- and F-lands are used at equal proportion.

**Shrinkage of agricultural lands** is for simplicity chosen to proceed in reverse order:

1.  $C \rightarrow F + G$ : First croplands are given back to natural grass- and F-lands according to their relative availability.
2.  $P \rightarrow F$ : Next, pastures are given back to F-lands.
3.  $P \rightarrow G$ : Finally, if F-lands have reached their maximum possible extent, pastures make room for grasslands.

These rules guarantee the priority for the establishment of pastures on former grasslands, even under conditions of back-conversion of agricultural lands. Note that when converting agricultural lands back to natural vegetation, one has to account for the limited extent of natural vegetation in a grid box. This maximum extent of natural vegetation is given by the distribution of potential natural vegetation (see section 9.7).

Using the above conversion order, the extended land use transitions are derived as follows. Basic to this derivation are the (directed) conversion fractions

$$(10.9) \quad \Delta_{i \triangleright j} = T_{i \triangleright j} c_i, \quad i, j \in \{N, F, G, C, P\}, \quad i \neq j,$$

that are fully determined by the current vegetation distribution  $c_N, c_C, c_P$ , and the land use transitions between  $N, C, P$  from the Harmonized Protocol. The aim is to determine the matrix elements for transitions from the natural vegetation types  $G$  and  $F$  to the agricultural types  $C$  and  $P$ , all other matrix elements are known anyway (like  $T_{C \triangleright P}$ ) or follow from them via (10.3).

First the expansion of agricultural lands is considered, in particular (rule 1 and rule 2) the expansion of pastures. Formally, this expansion is characterized by  $\Delta_{N \triangleright P} > 0$ . The problem is to distribute the conversion fraction  $\Delta_{N \triangleright P}$  to the separate conversion fractions from grasslands to pastures, and F-lands to pastures. We have

$$(10.10) \quad \Delta_{N \triangleright P} = \Delta_{G \triangleright P} + \Delta_{F \triangleright P}.$$

According to the sequential ordering of transitions described above, first the transition  $G \triangleright P$  has to be considered, i.e. as much as possible of  $\Delta_{N \triangleright P}$  has to be transferred to  $\Delta_{G \triangleright P}$ . Obviously,  $\Delta_{G \triangleright P}$  can neither get larger than the currently available grasslands  $c_G$ , nor larger than the total amount  $\Delta_{N \triangleright P}$  to be transferred. Hence,

$$(10.11) \quad \Delta_{G \triangleright P} = \min(c_G, \Delta_{N \triangleright P}),$$

so that using the definition (10.9) one finds the first of the desired matrix elements as

$$(10.12) \quad T_{G \triangleright P} = \min\left(1, \frac{\Delta_{N \triangleright P}}{c_G}\right) = \min\left(1, \frac{T_{N \triangleright P} c_N}{c_G}\right).$$

If insufficient grassland is available to allocate  $\Delta_{N \triangleright P}$  fully, then, according to rule 2., for the remainder F-lands are used: From (10.10) and (10.11) one finds

$$(10.13) \quad T_{F \triangleright P} = \min\left(1, \frac{\Delta_{N \triangleright P} - \Delta_{G \triangleright P}}{c_F}\right) = \min\left(1, \frac{T_{N \triangleright P} c_N - T_{G \triangleright P} c_G}{c_F}\right).$$

To handle rule 3 – the conversion of natural lands to croplands – one has to account for the grass- and F-lands that according to rules 1 and 2 were already converted to pastures. These remaining cover fractions  $\tilde{c}_G$  and  $\tilde{c}_F$  of grass- and F-lands are given as

$$(10.14) \quad \begin{aligned} \tilde{c}_G &= (1 - T_{G \triangleright P}) c_G \\ \tilde{c}_F &= (1 - T_{F \triangleright P}) c_F \end{aligned}$$

so that the distribution of natural lands to crops can equally be written as

$$(10.15) \quad T_{N \triangleright C} c_N = T_{G \triangleright C} c_G + T_{F \triangleright C} c_F$$

or as

$$(10.16) \quad T_{N \triangleright C} c_N = \tilde{T}_{G \triangleright C} \tilde{c}_G + \tilde{T}_{F \triangleright C} \tilde{c}_F$$

where the newly introduced transitions elements with a tilde describe the transitions with respect to the cover fractions already reduced by establishment of pastures. By rule 3 the establishment of crops happens at equal proportion with respect to these reduced cover fractions so that

$$(10.17) \quad \tilde{T}_{G \triangleright C} = \tilde{T}_{F \triangleright C} = \frac{T_{N \triangleright C} c_N}{\tilde{c}_G + \tilde{c}_F},$$

where the last equality is a consequence of the first equality when using (10.16). Since the first and second right hand terms in (10.15) and (10.16) must match separately one finally finds

$$(10.18) \quad \begin{aligned} T_{G \triangleright C} &= T_{N \triangleright C} (1 - T_{G \triangleright P}) \frac{c_N}{c_N - \Delta_{G \triangleright P} - \Delta_{F \triangleright P}} \\ T_{F \triangleright C} &= T_{N \triangleright C} (1 - T_{F \triangleright P}) \frac{c_N}{c_N - \Delta_{G \triangleright P} - \Delta_{F \triangleright P}} \end{aligned}$$

Coming now to the shrinkage of agricultural lands, the respective rule 1 has to be considered first. Here  $\Delta_{C \triangleright N} > 0$ . Giving croplands back at equal proportion to the relative availability of grass- and F-lands means that there is a common factor  $a$  for the transitions from croplands back to natural vegetation obeying

$$(10.19) \quad \begin{aligned} T_{C \triangleright G} c_C &= a(c_G^{pot} - c_G) \\ T_{C \triangleright F} c_C &= a(c_F^{pot} - c_F) \end{aligned}$$

Here  $c_G^{pot}$  and  $c_F^{pot}$  are the potential (maximum) natural extent of grass- and F-lands in the sense of section 9.7. The factor  $a$  is limited in size, because grasslands and F-lands cannot grow larger than their potential extent. A necessary condition for this is<sup>12</sup>

$$(10.20) \quad \begin{aligned} T_{C \triangleright G} &\leq \frac{c_G^{pot} - c_G}{c_C} \\ T_{C \triangleright F} &\leq \frac{c_F^{pot} - c_F}{c_C} \end{aligned}$$

---

<sup>12</sup>This condition would also be sufficient, if not at the same time also pastures could be given back to natural lands. Because this latter process is considered independently from back-conversion of croplands, the algorithms presented here may lead to more grass or F-lands than potentially available. In the implemented algorithm this is checked at the end, and the problematic transitions are then suppressed (see section 10.2.1.5).

so that by comparison with eqs. (10.19)

$$(10.21) \quad a \leq \frac{1}{c_C}.$$

On the other hand, using the obvious relation

$$(10.22) \quad T_{C \triangleright N} = T_{C \triangleright G} + T_{C \triangleright F}$$

together with eqs. (10.19) one can solve for  $a$ . Combining the resulting expression for  $a$  with condition (10.21) gives for the desired transition elements

$$(10.23) \quad \begin{aligned} T_{C \triangleright G} &= \min \left( \frac{1}{c_C}, \frac{T_{C \triangleright N}}{c_N^{pot} - c_N} \right) (c_G^{pot} - c_G) \\ T_{C \triangleright F} &= \min \left( \frac{1}{c_C}, \frac{T_{C \triangleright N}}{c_N^{pot} - c_N} \right) (c_F^{pot} - c_F) \end{aligned}$$

Applying now rules 2 and 3 for the shrinkage of pastures it is straightforward to derive the following equations for the remaining two unknown matrix elements when noting  $T_{P \triangleright N} = T_{P \triangleright F} + T_{P \triangleright G}$ :

$$(10.24) \quad \begin{aligned} T_{P \triangleright F} &= \min \left( T_{P \triangleright N}, \frac{c_F^{pot} - (c_F + T_{C \triangleright F} c_C)}{c_P} \right) \\ T_{P \triangleright G} &= \min \left( T_{P \triangleright N} - T_{P \triangleright F}, \frac{c_G^{pot} - (c_G + T_{C \triangleright G} c_C)}{c_P} \right). \end{aligned}$$

### 10.2.1.3 Daily land cover transitions

The discussion so far dealt with yearly land use transitions. But updating land cover only once a year may introduce shocks into the system. Therefore, the yearly land use transitions  $T_{i \triangleright j}$  are interpolated to daily land use transitions  $T_{i \triangleright j}^{(d)}$ , where  $d$  denotes the number of a day in a year.

The idea is to interpolate the cover fractions linearly between the cover fractions  $c_i$  and  $c'_i$  of (10.6). Denoting the number of days in the current year by  $N_d$  this can be achieved by adding each day the same part

$$(10.25) \quad \delta_{i \triangleright j} = \Delta_{i \triangleright j} / N_d, \quad i \neq j,$$

of the conversion fractions (10.9). With (10.25) the daily cover fractions are given by

$$(10.26) \quad c_i^{(d)} = c_i + d \sum_{j \in \{F, P, G, C\}} (\delta_{j \triangleright i} - \delta_{i \triangleright j}), \quad i \in \{F, P, G, C\}.$$

Noting that  $c_i^0 = c_i$  is the cover fraction of class  $i$  at the last time step of the previous year, and  $c_i^{(N_d)} = c'_i$  the cover fraction at the last time step in the current year, this can be rewritten as

$$(10.27) \quad c_i^{(d)} = c_i + \frac{d}{N_d} (c'_i - c_i), \quad i \in \{F, G, P, C\},$$

so that with these definitions the cover fractions increase or decrease linearly throughout the year. Daily land use transitions  $T_{i \triangleright j}^{(d)}$  for day  $d$  can be defined in analogy to (10.9) by

$$(10.28) \quad \delta_{i \triangleright j} = T_{i \triangleright j}^{(d)} c_i^{(d)}, \quad i \neq j,$$

where  $d < N_d$ . By inserting (10.27), (10.25) and (10.9) the desired daily land use transitions are found as

$$(10.29) \quad T_{i>j}^{(d)} = T_{i>j} \frac{c_i}{N_d c_i^{(d)}} = \frac{T_{i>j} c_i}{N_d c_i + d(c_i' - c_i)}, \quad i \neq j.$$

One can show<sup>13</sup> that with these definitions

$$(10.30) \quad 0 \leq \sum_{j, j \neq i} T_{i>j}^{(d)} \leq 1$$

so that by conservation of area (compare (10.3)) the diagonal term can safely be defined as

$$(10.31) \quad T_{i>i}^{(d)} = 1 - \sum_{j, j \neq i} T_{i>j}^{(d)}.$$

Thereby the matrix relationship (10.4) holds also on daily basis<sup>14</sup>:

$$(10.32) \quad \vec{c}^{(d+1)} = T^{(d+1)} \vec{c}^{(d)}.$$

#### 10.2.1.4 PFT transitions

In the previous sections it has been described how in JSBACH the (reduced) land use transitions of the Harmonized Protocol (10.1) are converted into the (extended) yearly transitions (10.6), and these finally into the daily transitions (10.29). The next task is to derive from the latter the PFT transitions (10.4). For brevity the superscript “(d)”, indicating the dependence of the transition matrix on the particular day in the year, is dropped in the following.

So far for the derivation of the (extended) transition matrix only the pasture rule has been used. For the derivation of the PFT transitions now the other two land distribution rules are invoked, i.e. the *pathway rule* and the *rule of equal relative gains and losses* (see above). On closer inspection, both rules are of similar structure: They demand that on land conversions (i) the relative loss of all PFTs in a shrinking land cover class is equal, and (ii) the relative gain of all PFTs in an expanding land cover class is also equal, although in the latter case “relative” means for the pathway rule relative to the current extent ( $c_i$ ), whereas for the rule of equal gains and losses it means relative to the area currently available ( $c_i^{pot} - c_i$ ).

To make this more precise, consider two land cover classes  $A$  and  $B$  from  $\{F, G, P, C\}$ . Each of these classes consists of a number of PFTs, which will be denoted by  $A_i$  and  $B_j$ , respectively, where the indices  $i, j$  number the PFTs in the particular class. Then the two rules are expressed by

$$(10.33) \quad \begin{aligned} \frac{\Delta_{A_i>B}}{c_{A_i}} &= \frac{\Delta_{A_j>B}}{c_{A_j}}, \quad i \neq j, \text{ “equal relative loss”} \\ \frac{\Delta_{A>B_i}}{x_{B_i}} &= \frac{\Delta_{A>B_j}}{x_{B_j}}, \quad i \neq j, \text{ “equal relative gain”,} \end{aligned}$$

where, depending on the applied rule,  $x$  has the following meaning:

$$(10.34) \quad x_{B_i} = \begin{cases} c_{B_i} & \text{when applying rule 2} \\ c_{B_i}^{pot} - c_{B_i} & \text{when applying rule 3,} \end{cases}$$

<sup>13</sup>  $\sum_{j, j \neq i} T_{i>j}^{(d)} = \frac{c_i}{N_d c_i^{(d)}} \sum_{j, j \neq i} T_{i>j} = \frac{c_i}{N_d c_i^{(d)}} (1 - T_{i>i}) = \frac{c_i}{N_d c_i + d(c_i' - c_i)} (1 - T_{i>i}) \leq 1.$

<sup>14</sup>  $c_i^{(d+1)} = c_i^{(d)} + \sum_{j, j \neq i} (T_{j>i}^{(d)} c_j^{(d)} - T_{i>j}^{(d)} c_i^{(d)}) = c_i^{(d)} T_{i>i}^{(d)} + \sum_{j, j \neq i} T_{j>i}^{(d)} c_j^{(d)} = \sum_j T_{j>i}^{(d)} c_j^{(d)}, \quad i, j \in \{F, G, P, C\}.$

and  $c_{B_i}^{pot}$  is the maximally possible (“potential”) cover fraction of PFT  $B_i$ . Here e.g.  $\Delta_{A_i \triangleright B}$  is the fraction of the vegetated part of the grid box that is transferred from PFT  $A_i$  into the class of PFTs  $B$ . These rules are fulfilled when choosing the desired relationship between the elements of the (extended) transition matrix  $T$  and the matrix of PFT transitions  $t$  as

$$(10.35) \quad t_{A_i \triangleright B_j} = T_{A \triangleright B} \frac{x_{B_j}}{x_B}, \quad \forall A_i \in A, \forall B_j \in B.$$

That thereby eqs. (10.33) are indeed fulfilled is seen from

$$(10.36) \quad \begin{aligned} \frac{\Delta_{A_i \triangleright B}}{c_{A_i}} &= \frac{\sum_{j=1}^{N_B} t_{A_i \triangleright B_j} c_{A_i}}{c_{A_i}} \stackrel{(10.35)}{=} \sum_{j=1}^{N_B} T_{A \triangleright B} \frac{x_{B_j}}{x_B} = T_{A \triangleright B}, \\ \frac{\Delta_{A \triangleright B_i}}{x_{B_i}} &= \frac{\sum_{j=1}^{N_A} t_{A_j \triangleright B_i} c_{A_j}}{x_{B_i}} \stackrel{(10.35)}{=} \frac{T_{A \triangleright B}}{x_B} \frac{x_{B_i} \sum_{j=1}^{N_A} c_{A_j}}{x_{B_i}} = T_{A \triangleright B} \frac{c_A}{x_B}, \end{aligned}$$

because the right hand sides are independent of  $i$  ( $N_A, N_B$  denote the number of PFTs in the respective classes). Moreover, the following lines show that by (10.35) also the ratio between two shrinking PFTs as well as between two expanding PFTs is kept fixed, as was demanded in particular by the pathway rule:

$$(10.37) \quad \begin{aligned} \frac{c'_{B_i}}{c'_{B_j}} &= \frac{c_{B_i} + \Delta_{B_i \triangleright A} - \Delta_{A \triangleright B_i}}{c_{B_j} + \Delta_{B_j \triangleright A} - \Delta_{A \triangleright B_j}} = \frac{c_{B_i}}{c_{B_j}} \frac{1 + \frac{\Delta_{B_i \triangleright A}}{c_{B_i}} - \frac{\Delta_{A \triangleright B_i}}{c_{B_i}}}{1 + \frac{\Delta_{B_j \triangleright A}}{c_{B_j}} - \frac{\Delta_{A \triangleright B_j}}{c_{B_j}}} = \frac{c_{B_i}}{c_{B_j}}, \\ \frac{c'_{A_i}}{c'_{A_j}} &= \frac{c_{A_i} + \Delta_{B \triangleright A_i} - \Delta_{A_i \triangleright B}}{c_{A_j} + \Delta_{B \triangleright A_j} - \Delta_{A_j \triangleright B}} = \frac{c_{A_i}}{c_{A_j}} \frac{1 + \frac{\Delta_{B \triangleright A_i}}{c_{A_i}} - \frac{\Delta_{A_i \triangleright B}}{c_{A_i}}}{1 + \frac{\Delta_{B \triangleright A_j}}{c_{A_j}} - \frac{\Delta_{A_j \triangleright B}}{c_{A_j}}} = \frac{c_{A_i}}{c_{A_j}}, \end{aligned}$$

in these equations the last equality is obtained by using eqs. (10.36).

Equation (10.35) gives the relation between land class transitions  $T$  and PFT-transitions  $t$  for all cases with enough space available for the expanding PFTs. But for back-conversion of agricultural lands to natural vegetation it may happen that one or more natural PFTs cannot be expanded any more, because they reach their potential extent. In such a case, the transitions prescribed by the Harmonized Protocol turn out to be inconsistent with the distribution of natural vegetation in the model. This may happen for various reasons: One possibility is that during simulations after times of agricultural expansion climate is changing such that, when JSBACH is run with DYNVEG, the potential extent of natural vegetation shrinks to values smaller than at times when agricultural expansion started. This type of problem may also arise because the New Hampshire group has derived the transition matrices using their own distribution of natural vegetation that they derived from simulations of global biomass with the MIAMI model [66]. A third possibility for the emergence of this problem may be related to the three land use distribution rules introduced here to implement the transitions, in some cases they may be too restrictive for the back-conversion of agricultural land. The latter possibility could in principle be partly remedied, by distributing the surplus area to the other PFTs using once more the same formula (10.35), but for a smaller number of PFTs. However, this would make the algorithms even more complicated, furthermore, under realistic scenarios it would affect only small areas on the globe, and finally such an approach would not remedy a possible general inconsistency between the prescribed transitions and the natural vegetation in the model. Therefore, a more simple strategy is used: We reduce the

values of the transition matrix such that not more area of agricultural land is back-converted than available by potential natural vegetation. Hence, for back-conversions of agricultural lands, eq. (10.35) has to be completed by the additional restriction

$$(10.38) \quad t_{A_i \triangleright B_j} c_{A_i} \leq c_{B_j}^{pot} - c_{B_j}$$

so that (10.35) then reads

$$(10.39) \quad t_{A_i \triangleright B_j} = \min \left( \frac{c_{B_j}^{pot} - c_{B_j}}{c_{A_i}}, T_{A \triangleright B} \frac{x_{B_j}}{x_B} \right), \quad \forall A_i \in A, \forall B_j \in B.$$

It should be noted that (10.38) is only a necessary, but not a sufficient condition, because it refers only to one of several parallel transitions increasing the area of PFT  $B_i$ . Therefore, after all transition elements have been computed, in a final check transitions that would impair the potential vegetation extent are suppressed.

Finally, the diagonal elements of the transfer matrix follow from the conservation of area, namely from an equation analogous to (10.3):

$$(10.40) \quad \sum_{j=1}^K t_{i \triangleright j} = 1, \quad i = 1, 2, \dots, K,$$

which gives for the diagonal elements

$$(10.41) \quad t_{i \triangleright i} = 1 - \sum_{j=1, j \neq i}^K t_{i \triangleright j}, \quad i = 1, 2, \dots, K.$$

With this last equation all elements of the PFT transition matrix  $t$  can be determined from the information provided by the Harmonized Protocol so that by (10.4) the new cover fractions can be computed.

#### 10.2.1.5 Assuring non-zero cover fractions in the land-use transition model

Generally in JSBACH, the cover fractions of the tiles in a gridbox must be non-zero to prevent divisions by zero when e.g. reshuffling the carbon during land-use changes. Therefore, the transition matrix elements as computed by (10.4) must be adjusted such that the new cover fractions do not get smaller than a certain very small value  $c_{min}$  (compare footnote in section 1.3.2). In addition, for back-conversion of agricultural land to natural vegetation this conversion is not allowed to lead to more natural vegetation than is potentially available. This latter point is not fully assured by the algorithms introduced above in all situations (see the discussion there) and needs to be prevented at all means. And this is both assured as follows.

After computing the new cover fractions  $c'_i$  from the tile transition matrix  $t$  via (10.4), it is checked whether all  $c'_i$  are larger than  $c_{min}$  and whether for all natural vegetation types the potential vegetation cover  $c_i^{pot}$  remains larger or equal  $c'_i$ . Assume that for the  $m$ -th cover fraction one of those conditions is violated. In this case the transition protocol is modified such that the cover fraction at the  $m$ -th tile remains unchanged, i.e. such that  $c'_m = c_m$ . This is achieved by setting all land use transitions to and from the  $m$ -th tile to zero, and the  $m$ -th diagonal element to 1. By this operation also the gain and losses of other tiles are affected, so that because of the conservation of total area under the transitions (compare (10.3)) the diagonal elements have to be readjusted such that once more each column sums up to 1; by

this procedure the transitions between tiles with index different from  $m$  remain unchanged. Overall, the following change in the tile transition matrix is performed:

$$(10.42) \quad \begin{pmatrix} t_{1 \triangleright 1} & \dots & t_{m \triangleright 1} & \dots & t_{K \triangleright 1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ t_{1 \triangleright m} & \dots & t_{m \triangleright m} & \dots & t_{K \triangleright m} \\ \vdots & \dots & \vdots & \ddots & \vdots \\ t_{1 \triangleright K} & \dots & t_{m \triangleright K} & \dots & t_{K \triangleright K} \end{pmatrix} \rightarrow \begin{pmatrix} t'_{1 \triangleright 1} & \dots & 0 & \dots & t_{K \triangleright 1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & \dots & \vdots & \ddots & \vdots \\ t_{1 \triangleright K} & \dots & 0 & \dots & t'_{K \triangleright K} \end{pmatrix}$$

with the new diagonal elements<sup>15</sup>

$$(10.43) \quad t'_{i \triangleright i} = t_{i \triangleright i} + t_{i \triangleright m}, \quad i \neq m.$$

After this replacement the new cover fractions are recomputed using the modified tile transition matrix. Some of these new cover fractions may still violate the above conditions – or may even violate them because of these changes. In this case the whole procedure is repeated for the respective tiles. It needs to be repeated at most  $K$  times, because then the tile transition matrix is identical to the unit matrix, and, accordingly, the cover fractions of all tiles remain unchanged under the action of the modified (unit) transition matrix. Note that the modification (10.42), (10.43) can also be done for several tiles in one step, whereby several rows and columns need to be simultaneously modified.

### 10.2.1.6 Interplay between land use transitions and natural vegetation dynamics

The transition model described in the previous sections simulates anthropogenic land cover change in the absence of natural change in vegetation distribution, while the DYNVEG model described in chapter 9 simulates natural vegetation dynamics in the absence of land-use change. The present section describes how natural vegetation dynamics is married with anthropogenic land cover change in JSBACH.

Such a component combining these two processes is necessary because concurrently with anthropogenic land cover change the composition of the natural vegetation may be altered by a climate change or other environmental factors (e.g. CO<sub>2</sub>). For example, a forest may expand after pastures were introduced. How would farmers respond to the growing forest? One reaction could be to stay with the pastures at the same locations and cut down periodically all emerging trees. Another reaction could be to shift the pastures gradually to adjacent natural grasslands. Which option is chosen may depend on many socio-economic factors like the existing infrastructure (location of settlements and roads), and on the existing agrotechnology. But modeling such processes explicitly is beyond the scope of JSBACH, so that a more simple approach is needed.

Obviously, such reactions of farmers to changes in the surrounding natural vegetation must be called anthropogenic land cover change. Therefore, the question arises whether such a type of land-use change is already accounted for in the Harmonized Protocol. This protocol reflects two types of land-use changes, namely all changes affecting the net extent of croplands and pastures, and rotational land cover changes driven by certain agricultural practices like slash-and-burn. In contrast, the type of land-use change in question concerns neither an expansion or shrinkage of agricultural activities, nor is it the result of a particular land use practice. Instead, it is a directed long-term adaptation of agriculture to changes in climate

<sup>15</sup>In the code this formula is not used; instead the diagonal elements are newly recomputed from the other elements in the respective column of the tile transition matrix using eq. (10.41).

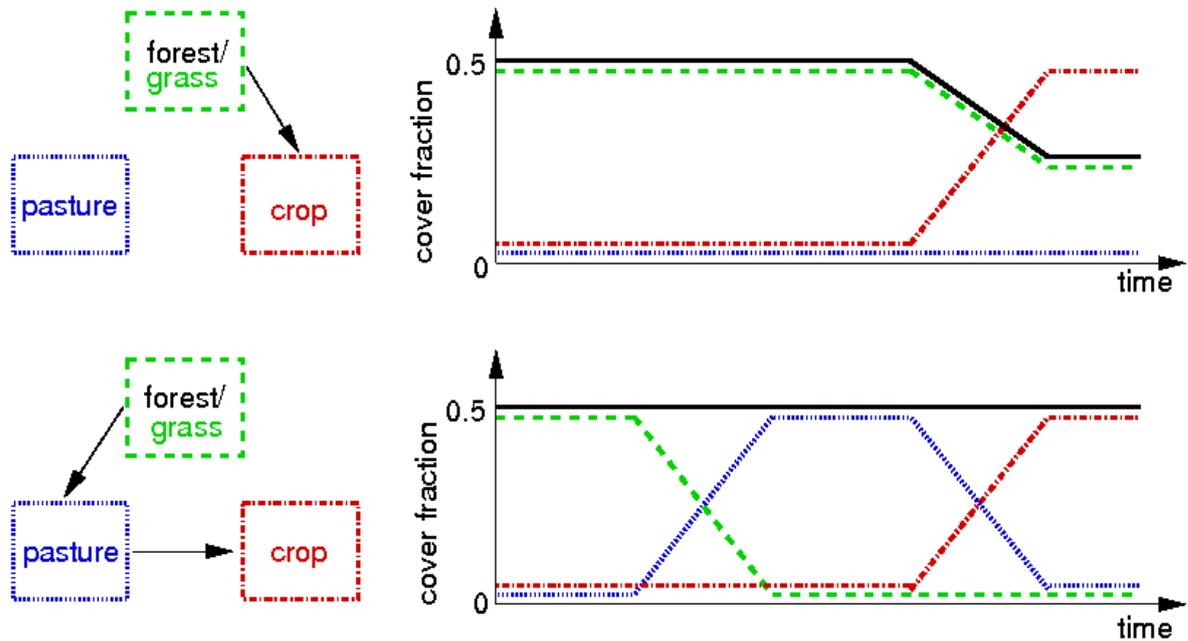


Figure 10.1: This graph demonstrates that due to the pasture rule cover fractions of the natural PFTs (forests, grasses) depend on the history of land use transitions (see text). *Black line*: forest fraction. *Dashed green line*: grassland. *Dashed-dotted red line*: croplands. *Dotted blue line*: pastures.

conditions by a very local translocation of croplands and pastures. Accordingly, we conclude that this additional type of anthropogenic land cover change is not included in the transitions described by the Harmonized Protocol and must therefore be represented independently in JSBACH.

How could a simple representation of this additional land-use change look like? The most direct approach would be to distribute pastures and croplands proportionally across the natural land cover types. But this would be completely inconsistent with the preferential establishment of pastures on grasslands (“pasture rule”, see section 10.2.1.2). Alternatively one could distribute the known extent of pastures and croplands according to the pasture rule to the potential natural vegetation (instead of distributing it to the *remaining* natural vegetation as in the previous section). But this would lead to very large deviations from the vegetation distribution obtained by applying the pasture rule sequentially year by year during land use transitions because this sequential application introduces a dependence of the actual land cover on the full land use history that led to this land cover. This is illustrated in Fig. 10.1: The upper panel shows a single land use transition from potential natural vegetation to cropland, which according to the rules set up in section 10.2.1.2 is established at equal proportion on grass- and F-lands (i.e. forest and shrublands). In the lower panel the land use transition to cropland is done via an intermediate step where first F-lands are converted to pasture which in a second step is fully converted to croplands. In the first step the pasture rule has to be applied so that the pastures are fully established on grasslands. Although at the end in both panels the extent of pastures and croplands is identical, the different land use histories give a different distribution for grass- and F-lands. Hence, to describe land cover change due to changes in potential land cover, the actual state of land cover obtained during the historical development of land use cannot be ignored.

The simple model described in the remainder of this section accounts for this. It is based

on the following ideas: (i) Because of the socio-economic aspects, the reaction of farmers to changing potential vegetation is conservative, allowing only small changes from year to year. (ii) On the other hand, since a land use adapted to the ruling climatic situation is advantageous for the farmers, one can assume that over several generations the land use distribution will be directed towards a distribution obtained from the pasture rule.

Formally the task is to translate the known distributions of potential vegetation given by (9.8) and the known land use as obtained by (10.4) into actual land cover  $c_i$  for all PFTs. To derive this land cover it is most convenient to consider first only the larger classes of vegetation F,G,C, and P (see section 10.2.1.1) before the land cover for all PFTs is derived.

According to the above considerations, changes in climatic conditions and thus in potential vegetation lead to moderate shifts into the direction of a vegetation distribution following the pasture rule. This “target cover fraction”, called  $c^{inst}$  in the following, is obtained by assuming that the pasture rule is applied to the potential vegetation distribution as if all anthropogenic land cover change during history had happened *in one instant* by a single land use transition. Following the sequence of transitions for agricultural expansion described in 10.2.1.2 the first step in deriving these target fractions is to allocate pastures with priority on grasslands. Two cases have to be distinguished: In the first case there is enough grassland for the allocation of pastures ( $c_P \leq c_G^{pot}$ ), in the second case also F-lands need to be converted to pastures. This gives

$$(10.44) \quad \left. \begin{aligned} \tilde{c}_G^{inst} &= c_G^{pot} - c_P \\ \tilde{c}_F^{inst} &= c_F^{pot} \end{aligned} \right\} \quad \text{for } c_P \leq c_G^{pot}$$

$$\left. \begin{aligned} \tilde{c}_G^{inst} &= 0 \\ \tilde{c}_F^{inst} &= c_F^{pot} - (c_P - c_G^{pot}) \end{aligned} \right\} \quad \text{otherwise}$$

From these intermediate values, indicated by a tilde over the symbols, in a second step the final target land cover fractions are obtained by distributing croplands proportionally to the remaining F-lands and grasslands. This gives for the F-lands

$$(10.45) \quad c_F^{inst} = \begin{cases} \tilde{c}_F^{inst} - c_C \frac{\tilde{c}_F^{inst}}{\tilde{c}_G^{inst} + \tilde{c}_F^{inst}} & \text{for } c_P \leq c_G^{pot} \\ \tilde{c}_F^{inst} - c_C & \text{otherwise.} \end{cases}$$

Using in accordance with (9.8) the relation  $c_F^{pot} + c_G^{pot} = 1$  one obtains

$$(10.46) \quad c_F^{inst} = \begin{cases} (1 - c_P - c_C) \frac{c_F^{pot}}{1 - c_P} & \text{for } c_P \leq c_G^{pot} \\ 1 - c_P - c_C & \text{otherwise.} \end{cases}$$

The target cover fraction for grasslands  $c_G^{inst}$  follows from  $c_F^{inst}$  by noting that  $c_F^{inst} + c_G^{inst} + c_P + c_C = 1$ .

Next the size of changes in land cover is determined that is induced solely by changes in potential vegetation. To this end a second, artificial target cover fraction for F-lands  $\bar{c}_F^{inst}$  is computed by entering into (10.46) *last year's* potential vegetation but still using as for  $c_F^{inst}$  *this year's* fractions of croplands and pastures  $c_P$  and  $c_C$ . The difference between these two target cover fractions  $\delta_F^{inst} = \bar{c}_F^{inst} - c_F^{inst}$  indicates how strongly natural land cover may have changed since last year at given land use. Changes arising from shifts in potential vegetation arise on top of the changes induced by land use. To derive the new land cover arising from both processes one thus needs in addition the cover fractions of natural vegetation that arose last year due to land use transitions alone; these cover fractions, denoted by  $\bar{c}_i$ ,  $i \in \{F, G\}$ ,

result from the application of the Harmonized Protocol as described in the previous section. Assuming as discussed above that farmers follow changes in potential land cover only if it is advantageous for them, this year's new distribution of F-lands accounting for both processes can then be obtained by

$$(10.47) \quad c_F = \begin{cases} \bar{c}_F + \delta_F^{inst} & \text{for } \text{mon}(c_F^{inst}, \bar{c}_F^{inst}, \bar{c}_F) \\ c_F^{inst} & \text{for } \text{mon}(c_F^{inst}, \bar{c}_F, \bar{c}_F^{inst}) \\ \bar{c}_F & \text{for } \text{mon}(\bar{c}_F, c_F^{inst}, \bar{c}_F^{inst}) \end{cases}$$

where  $\text{mon}(a, b, c)$  means that  $a, b, c$  are ordered monotonously, i.e.  $a \leq b \leq c$  or  $a \geq b \geq c$ . The cases in (10.47) have the following meaning: In the first case  $\bar{c}_F$  is farther away from  $c_F^{inst}$  than  $\bar{c}_F^{inst}$  so that a correction by  $\delta_F$  brings the cover fraction closer to the target  $c_F^{inst}$ . In the second case an adjustment by the full size of  $\delta_F$  would overshoot the target, so that the correction is made only until the target fraction is obtained. In the last case accounting for the correction  $\delta_F$  would drive the cover fraction away from the target so that the cover fraction is left unchanged from last to this year. The associated change in the fraction of grasslands can be obtained analogously, but it is simpler to compute it from the conservation of area by

$$(10.48) \quad c_G = 1 - c_F - c_C - c_P.$$

Finally, after  $c_F$  and  $c_G$  have been determined, the new cover fractions of the natural PFTs are calculated by partitioning  $c_F$  and  $c_G$  into the different PFTs according to their relative abundance  $c_i^{pot}$ :

$$(10.49) \quad \begin{aligned} c_i &= c_i^{pot} c_F / c_F^{pot}, \quad i \in \mathcal{W} \\ c_j &= c_j^{pot} c_G / c_G^{pot}, \quad j \in \mathcal{G}, \end{aligned}$$

where, as in chapter 9,  $\mathcal{W}$  denotes the woody types, and  $\mathcal{G}$  the non-woody types (grasses). This simple scaling is justified by noting that the competition mechanisms implemented in DYNVEG operate on much smaller scales than the extent of the grid boxes ( $\sim 10^4 \text{ km}^2$ ) so that the relative abundance of natural PFTs must be considered to be rather uniform within a grid box. Note that equations (10.49) are evaluated only once a year, namely during the first time step so that the cover fractions  $c_G$  and  $c_F$  are those from the last time step of the year before.

Overall, to account for climate induced changes in natural vegetation the scheme for land cover change described above largely retains the unproportional translation of  $c_F^{pot}, c_G^{pot}$  into  $c_F, c_G$  implied by the pasture rule, but allows in addition for variations in  $c_F, c_G$  according to natural vegetation dynamics in the presence of managed land. Moreover, the scheme accounts for the history dependencies in  $c_F, c_G$  arising from the sequence of land use transitions, but disperses them with time according to variations in the natural vegetation. In this way,  $c_F$  and  $c_G$  would converge on the long term to  $c_F^{inst}, c_G^{inst}$ , after land use transitions had ceased.

## 10.2.2 The maps model: land-use change as a sequence of land cover maps

In this ‘‘maps model’’ land-use change is represented as a sequence of maps describing not only the anthropogenically modified surfaces (croplands, pastures), but the whole distribution of land cover. Hence, by this approach also the changes in land cover of natural vegetation is prescribed implying that the DYNVEG model for natural land cover change described in chapter 9 cannot be used together with this maps model for land-use change. Accordingly,

it would be more appropriate not to call this a model for land *use* change, but a model for land *cover* change.

The geographic change of land cover is implemented as external forcing. More precisely, a sequence of land cover maps is fed into the model.<sup>16</sup> Each year, first of January, a new map representing the new year's land cover is read in. To prevent the atmosphere model from carbon emission shocks that may arise from abrupt changes in land cover, the vegetation distribution is linearly interpolated between the recent and last year's map.<sup>17</sup> More precisely, let  $c_i^{last}$  and  $c_i^{pres}$  be the vegetation fractions of the  $i$ -th PFT from the last and present year, then the current day's vegetation fractions  $c_i$  are recomputed at the beginning of each day from<sup>18</sup>

$$(10.50) \quad c_i = c_i^{last} + (c_i^{pres} - c_i^{last}) \frac{d}{N_d},$$

where  $N_d$  is number of days in the considered year, and  $d$  the number of the particular day in the year. In this way the new land cover distribution  $c_i^{pres}$  is fully reached at the last day of the "present" year.

### 10.2.2.1 Representation of the maps model as (net) transitions

Formally one can represent the maps model by transitions. This is currently not implemented in JSBACH but this possibility could be used in future versions to unify the representation of both approaches. This representation by transitions is described in the following mainly to simplify the description of carbon and nitrogen relocation in section 10.3.2 below.

While for the transition model it is known from the daily PFT transition matrix  $t_{i \rightarrow j}$  from which tile land is transferred to what other tile from day to day (see section 10.2.1.4), this needs to be specified by an additional assumption in the maps model to determine the carbon and nitrogen transfers between different tiles. Here, JSBACH assumes that the transfer happens such that the land from a shrinking tile is distributed to all expanding tiles relative to the expansion of the expanding tiles. More precisely one can define this for the maps model by an equivalent daily PFT transition matrix as follows. First non-diagonal elements are defined as

$$(10.51) \quad t_{i \rightarrow j} = \begin{cases} \frac{(c_j^{new} - c_j^{old}) f_j^{corr}}{\sum_{k \in expand} (c_k^{new} - c_k^{old}) f_k^{corr}} \left(1 - \frac{c_i^{new}}{c_i^{old}}\right) & \text{for } \begin{cases} i \in shrink \\ j \in expand \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i \neq j$$

where  $c_j^{new}$  is the cover fraction of tile  $j$  for the new day, while  $c_j^{old}$  is yesterday's cover fraction as computed day by day from (10.50). The notation  $i \in shrink$  and  $j \in expand$  means that the indices  $i, j$  refer to tiles that shrink or expand upon land cover change. The correction factors  $f_j^{corr}$  for clumping (see (1.7)) are included in the weighting of the land transfers since carbon and nitrogen are predominantly located under closed parts of canopies. With (10.51) area is only transferred from shrinking to expanding tiles, all other non-diagonal elements of the transition matrix are thus set to zero. To assure that the important normalization

<sup>16</sup>The construction of a suitable sequence of forcing maps for the land cover change during the last millennium has been described in [106].

<sup>17</sup>Technically, this part of the land-use change model updates the variable `cover_fract` of JSBACH.

<sup>18</sup>To prevent storage of the field of land cover fractions  $c_i^{last}$ , the actual implementation differs a bit: instead of Eq. (10.50) the equivalent iterative formula  $c_i = c_i + (c_i^{pres} - c_i)/(1 + N_d - d)$  is used.

condition (10.40) expressing conservation of land area is fulfilled, the diagonal elements of the transition matrix must have the values

$$(10.52) \quad t_{i \triangleright i} = 1 - \sum_{k=1, k \neq i}^K t_{i \triangleright k}, \quad i = 1, 2, \dots, K,$$

where  $K$  is the number of tiles. Separating the sum into parts for expanding and shrinking tiles, one sees from this formula when applying (10.51) that the diagonal elements are given by<sup>19</sup>

$$(10.53) \quad t_{i \triangleright i} = \begin{cases} c_i^{new}/c_i^{old} & \text{for } i \in \textit{shrink} \\ 1 & \text{for } i \in \textit{expand}. \end{cases}$$

Note that one could also define other transition matrices to represent the maps model for land-use change. But the representation given here is unique in the sense that it is the only representation as *net* transitions. And since the sequence of maps provides no information about the back-and-forth transitions within a year, this is a kind of minimal representation. That the representation introduced here indeed represents net transitions can be seen from the fact that area is transferred by (10.51) only from shrinking to expanding tiles, i.e. if  $t_{i \triangleright j} \neq 0$  then  $t_{j \triangleright i} = 0$  for  $i \neq j$ . This excludes circular transitions between three or more tiles. This difference between *net* and *gross* transitions gets relevant only when carbon or other matter flows induced by land-use change are considered (see [144]).

### 10.2.2.2 Net transitions from gross transitions

As discussed in the previous section, the maps model represents *net* transitions, meaning that there are only transitions from shrinking to expanding tiles, but no simultaneous transitions in the opposite direction and also no circular transitions between three or more tiles. As a consequence, the land-use change emissions are least for such net transitions. The transitions provided by the Harmonized Protocol are typically not minimal in this sense, but include e.g. circular transitions representing shifting cultivation. Calling these transitions provided by the Harmonized Protocol as *gross* transitions, the purpose of the present section is to show how net transitions can be obtained from gross transitions. This is of interest to derive in simulations a lower bound for land use emissions as done e.g. in [144].<sup>20</sup>

Since the maps model represents net transitions, a straight forward solution to obtain net transitions would be to derive from a given initial distribution of PFTs by repeated application of the PFT transition matrix a sequence of PFT landcover maps (see (10.4)) and use them in JSBACH. Here a more direct way implemented in JSBACH is described. As described in section 10.2.1.1 in JSBACH the PFT transitions are obtained from the landcover class transitions provided by the Harmonized Protocol, where the landcover classes are natural (N), crops (C), and pastures (P). To obtain net transitions it is therefore most convenient to derive the net transitions from the landcover class transitions *before* they are converted to PFT transitions. Technically this means to obtain from the matrix  $T$  in eq. (10.1) that represents the gross transitions from the harmonized protocol, a new matrix  $T^{net}$  that represents the associated net transitions and use this instead of  $T$  to derive the PFT transitions. And how

<sup>19</sup>That indeed eqs. (10.51) and (10.53) describe the distribution of area from shrinking to expanding tiles as intended can be checked by explicitly calculating  $c_i^{new}$  for  $i \in \textit{expand}$  from the old cover fractions by applying the transition matrix and noting that  $\sum_{k \in \textit{expand}} (c_k^{new} - c_k^{old}) f_k^{corr} = \sum_{k \in \textit{shrink}} (c_k^{new} - c_k^{old}) f_k^{corr}$ .

<sup>20</sup>For this purpose the use of net transitions calculated from gross transitions can be switched on in JSBACH via namelist switch `lcc_forcing_type=NET_TRANSITIONS` (see appendix C.2.11).

to do this is clear from the previous section, where it was explained how to derive from a sequence of maps the associated transition matrices representing net transitions.

Hence, the recipe is: given the cover fractions  $(c_N, c_C, c_P)^T$  of the previous distribution of landcover classes, calculate from eq. (10.1) the new cover fractions  $(c'_N, c'_C, c'_P)^T$  from the gross transitions. Then the non-diagonal matrix elements for net transitions are obtained in analogy to (10.51) from<sup>21</sup>

$$(10.54) \quad T_{i>j}^{net} = \begin{cases} \frac{c'_j - c_j}{\sum_{k \in \text{expand}} (c'_k - c_k)} \left(1 - \frac{c'_i}{c_i}\right) & \text{for } \begin{cases} i \in \text{shrink}(N, P, C) \\ j \in \text{expand}(N, P, C) \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i \neq j,$$

and the diagonal elements are in analogy to eq. (10.53)

$$(10.55) \quad T_{i>i}^{net} = \begin{cases} c'_i/c_i & \text{for } i \in \text{shrink}(N, P, C) \\ 1 & \text{for } i \in \text{expand}(N, P, C). \end{cases}$$

### 10.3 Carbon and nitrogen relocation induced by land use and land-use change

For the relocation of carbon and nitrogen induced by land-use change two alternative schemes are implemented in JSBACH, the *litter pool LUC scheme* and the *anthro pool LUC scheme*. In addition JSBACH contains a scheme to handle land use in the form of harvest from woody vegetation. To describe how in particular the schemes for land-use change work, it is useful to distinguish between *active* and *passive* carbon and nitrogen transfers induced by land-use change. *Active* transfers are a consequence of chopping down vegetation, implying that carbon and nitrogen are transferred from the pools representing the living vegetation to *different* types of pools in the *same* tile taking up the dead material, which can be the litter pools in the litter pool LUC scheme, the product pools of the anthro pool LUC scheme, or upon deforestation fires the atmosphere. In contrast, *passive* carbon and nitrogen transfers appear for technical reasons to deal with the shrinkage and expansion of tiles upon land-use change: Such passive transfers induce carbon and nitrogen relocation between the *same* type of pools at *different* tiles to assure that carbon and nitrogen stay “in place” despite the fact that land use has changed in that place. Alternatively, one could describe the land carbon and nitrogen dynamics also without passive transfers meaning that carbon and nitrogen would stay in their pools. But this has the disadvantage that upon shrinkage of a tile towards zero the carbon and nitrogen density of the litter and soil pools that do not lose carbon or nitrogen upon land-use change would have to increase towards infinity to conserve total carbon and nitrogen mass. Therefore, it is numerically more stable to perform such passive carbon and nitrogen transfers.<sup>22</sup>

<sup>21</sup>The calculation of net transitions is implemented in `mo_cbal_landcover_change::derive_net_transitions()` by equations that are obtained from (10.54) when using  $1 = c_N + c_C + c_P = c'_N + c'_C + c'_P$  (compare eq. (10.2)).

<sup>22</sup>Active and passive carbon/nitrogen transfers have without mentioning already been introduced in section 9.8.1: The first term of equation (9.13) describes the passive transfers, while the second term represents the active transfers.

Which pools participate in active and/or passive transfers is obtained from their meaning:

- Pools representing *living vegetation* (G,R,W,pmob)<sup>23</sup> are subject exclusively to active transfers (only losses upon land-use change), since it makes no sense to transfer e.g. carbon from the stems of trees to grasses.
- Pools representing *plant litter* and *soil mineral N* are subject to passive redistribution, but also participate in active transfers because they receive carbon and nitrogen from the active transfers of the living vegetation.
- All *other pools*, in particular the Yasso humus pools, the slow soil pool of CBALANCE and the crop harvest pool neither lose nor gain carbon or nitrogen from active transfers and are thus subject only to passive transfers.

Fig. 10.2 is depicting the relocation logic of the litter and anthro pool LUC schemes. In the *litter pool LUC scheme* a certain fraction of the carbon and nitrogen affected by land-use change is immediately emitted to the atmosphere, while the rest is added to the below and above ground litter pools. An exception is the transfer of nitrogen from the wood pool: because the wood litter pool has a lower N:C ratio than the wood pool, the surplus nitrogen is relocated to the soil mineral N pool. In the *anthro pool LUC scheme* the affected below ground vegetation carbon and nitrogen is handled exactly as in the litter pool LUC scheme, while the affected above ground carbon and nitrogen is handled differently: nitrogen is lost directly to the atmosphere and carbon is put into three separate 'product pools' as specified by the Grand Slam Protocol [90] from which the carbon is released with predefined turnover times to the atmosphere (see section 10.3.3 for more details).

Parallel to these two schemes for land-use change, there are two schemes for handling harvest from woody vegetation. Technically, the amount of harvest is prescribed from external data. Figure 10.3 shows the relocation structure of harvest in these two schemes, called *litter pool harvest scheme* and *anthro pool harvest scheme*. It is important to note that the crop harvest model presented in section 6.3.2 (which is related to the crop harvest pool in fig.10.3) is completely independent from the two harvest schemes described in the present chapter, that consider exclusively harvest from woody vegetation.

As already mentioned in the introduction to this chapter, the description of the implementation of land-use change is complicated by the many different possible configurations. To simplify the presentation of the associated relocation of carbon, it is taken advantage of the possibility to represent the "maps model" of land-use change (section 10.2.2) as a special case of the "transition model" of land-use change (section 10.2.1). Employing this trick, first in subsection 10.3.1 the relocation of pools participating exclusively in active or passive transfers is described because these are identical in both the litter and anthro pool LUC schemes. The specifics of the schemes show up only in relation to pools participating simultaneously in active and passive transfers (litter pools and the soil mineral N pool). These specifics are described in the subsequent subsections 10.3.2 and 10.3.3 addressing each scheme separately. A final issue is the induction of carbon and nitrogen relocation by land use in the form of harvest; this is the topic of the last subsection 10.3.4.

### 10.3.1 Exclusively active or passive relocations

Part of the carbon and nitrogen relocation happens in the same way for the litter and anthro pool LUC schemes. This concerns the pools representing living vegetation that participate only in active transfers, and the pools that have above been referred to as 'other' participat-

<sup>23</sup>These are the green (G), wood (W), reserve carbon (C), and plant mobile N (pmob) pools; compare Fig.7.1.

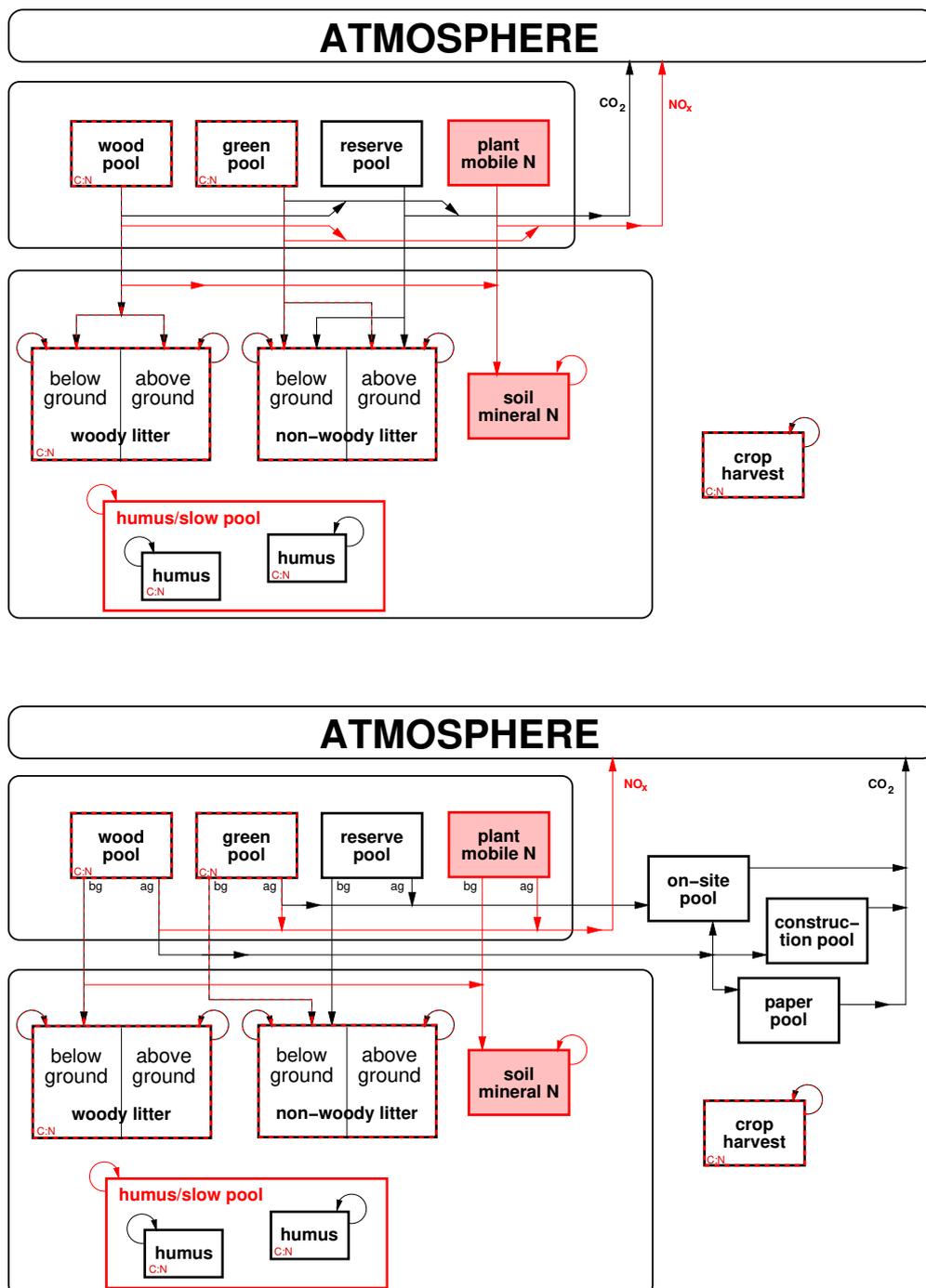


Figure 10.2: Carbon (black) and nitrogen (red) relocation induced by **land-use change** in the 'litter pool LUC scheme' (top) and 'anthropo pool LUC scheme' (bottom). Arrows denote *active* transfers between different pools at the same tile, while the recirculation arrows at pool corners indicate *passive* transfers between pools of the same type on different tiles. For the vegetation pools, in the 'anthropo pool LUC scheme' the fate of the active biomass transfers is different for the above ground (ag) and below ground (bg) fractions. – The diagram shows a simplified structure for soil and litter pools to jointly depict the situation for the CBALANCE and Yasso soil models; for more details on these soil models see fig. 7.1.

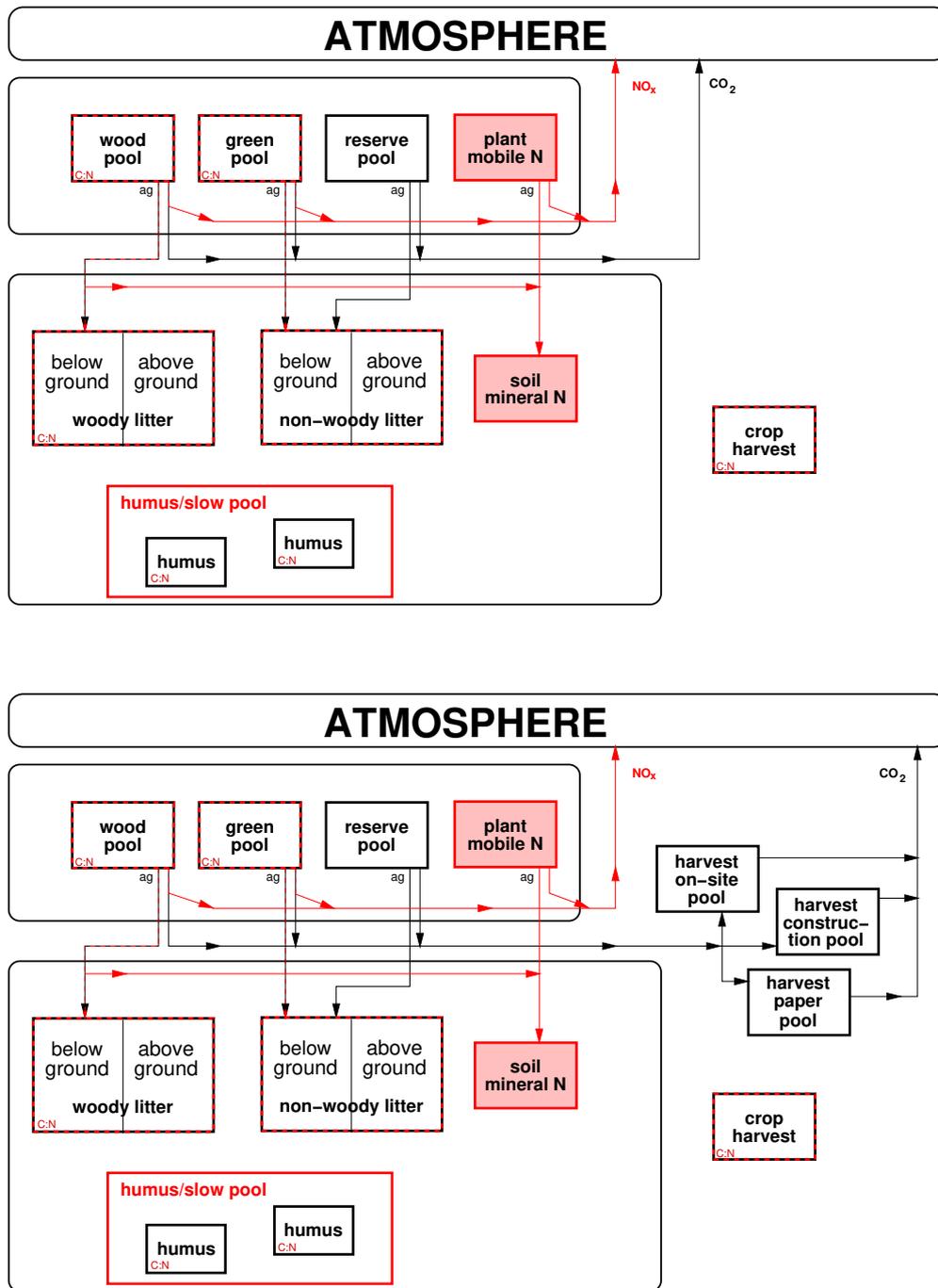


Figure 10.3: Carbon (black) and nitrogen (red) relocation induced by **land use** in the form of harvest from woody vegetation in the 'litter pool scheme' (top) and 'anthro pool scheme' (bottom). Only the above ground (ag) part of vegetation is harvested, of which in the anthro pool scheme a 'slash fraction' is relocated to litter pools and the soil mineral N pool while the rest enters the atmosphere directly (nitrogen) or indirectly via special anthropogenic harvest pools (carbon). In the litter pool scheme the fate of harvest is very similar, except that also the 'rest' of carbon enters the atmosphere directly. Note that in neither model the carbon of the slow or humus pools is affected. – The diagram shows a simplified structure for soil and litter pools to jointly depict the situation for the CBALANCE or Yasso soil models; for more details on these models see fig. 7.1.

ing only in passive transfers. The present section addresses relocations only for such pools undergoing exclusively passive or active changes. In addition this section introduces the basic logic of passive and active relocation that prepares for handling the mixed cases in the next subsections that deal with the specifics of the different relocation schemes implemented in JSBACH.

In section 10.2.2.1 it was shown how the maps model for land-use change can be represented by transitions (equations (10.51) and (10.52)). Therefore, the following description of carbon relocation induced by land-use change is presented in the framework of land use transitions only, although the equations implemented for the maps model in JSBACH don't use the transition formulation but implement the carbon relocation directly. But the equations implemented follow immediately from the equations presented in the present section when explicitly putting in the transition matrix for the maps model that was derived in section 10.2.2.1.

First the change in vegetation pools is discussed. These participate only in active changes. Assuming spatially homogeneous distributions of carbon and nitrogen within a tile, upon land-use change the four pools representing carbon and nitrogen of the living plants loose their contents according to the extent of converted land. Hence, the active losses of the carbon pools at tile  $i$  are (in units per box area)

$$(10.56) \quad \begin{aligned} F_{X\triangleright}^i &= \sum_{j, i \neq j} t_{i\triangleright j} c_i^{old} f_i^{corr} C_{X,i}^{old} \\ &= (1 - t_{i\triangleright i}) c_i^{old} f_i^{corr} C_{X,i}^{old}, \quad \text{for } X \in \{G, R, W\}, \end{aligned}$$

where equation (10.41) has been used. An analogous equation holds for the associated nitrogen losses<sup>24</sup>:

$$(10.57) \quad G_{X\triangleright}^i = (1 - t_{i\triangleright i}) c_i^{old} f_i^{corr} N_{X,i}^{old}, \quad \text{for } X \in \{G, W, pmob\}.$$

With the total carbon  $C_{X,i}^{box}$  in a pool of type  $X$  in tile  $i$  changing upon land-use change according to

$$(10.58) \quad C_{X,i}^{box,new} = C_i^{old} - veg_{max} F_{X\triangleright}^i, \quad \text{for } X \in \{G, R, W\}$$

and with an analogous equation for the update of nitrogen pools, one obtains by using the general relation (1.12) for box carbon and nitrogen concentrated under canopies<sup>25</sup>

$$(10.59) \quad X_i^{new} = t_{i\triangleright i} \frac{c_i^{old}}{c_i^{new}} X_i^{old}, \quad \text{for carbon and nitrogen pools } X \text{ of type } G, R, W, pmob.$$

Next the changes in those pools above referred to as other pools are described. These undergo exclusively passive changes. Once more a homogeneous distribution of carbon and nitrogen within a tile is assumed. Hence, the carbon and nitrogen transfer between similar pools at different tiles (passive transfer) is proportional to the transferred area. Following the logic from section 10.2 that part of cover fraction to be transferred from tile  $i$  to tile  $j$  is given as  $\Delta_{i\triangleright j} = t_{i\triangleright j} c_i^{old}$ , so that in units of box area the associated canopy area transferred is  $A veg_{max} f_i \Delta_{i\triangleright j}$ . Hence, assuming proportionality to the area transferred, the

<sup>24</sup>As in the nitrogen chapter 7, nitrogen fluxes are denoted by  $G$  while for carbon fluxes the symbol  $F$  is used.

<sup>25</sup>Equation (10.59) is implemented in the JSBACH code quite clumsily as  $X_i^{new} = (c_i^{old} veg_{max} X_i^{old} - F_{X\triangleright}^{box,i}) / c_i^{new} veg_{max}$ .

amount of carbon or nitrogen transferred is  $A \text{veg}_{max} f_i^{corr} \Delta_{i>j} X_i^{old}$ , or, written as transferred carbon/nitrogen box density

$$(10.60) \quad X_{i>j}^{box} = \text{veg}_{max} f_i^{corr} t_{i>j} c_i^{old} X_i^{old}, \quad X \in \text{'other' carbon and nitrogen pools.}$$

The pools not only loose, but also receive carbon or nitrogen by these passive transfers. Therefore

$$(10.61) \quad X_i^{box,new} = X_i^{box,old} + \sum_{j=1, j \neq i}^K X_{j>i}^{box} - \sum_{j=1, j \neq i}^K X_{i>j}^{box}, \quad X \in \text{'other' pools.}$$

Entering (10.60) and invoking (10.41) one finds after a bit of algebra that the new carbon and nitrogen densities can be computed from

$$(10.62) \quad X_i^{new} = \frac{\sum_{j=1}^K t_{j>i} c_j^{old} f_j^{corr} X_j^{old}}{f_i^{corr} c_i^{new}}. \quad X \in \text{'other' pools.}$$

### 10.3.2 The ‘litter pool LUC scheme’: Relocation to litter pools

While the carbon and nitrogen relocation described in the previous subsection applies to both LUC schemes, the present subsection is specific to the litter pool LUC scheme. This specificity concerns the carbon and nitrogen relocation of the litter pools and the soil mineral N pool, that in contrast to the relocations considered so far participate *simultaneously* in passive and active relocations. And this specificity concerns also the carbon and nitrogen emissions induced by land-use change.

As depicted in the top diagram of fig. 10.2, in the litter pool LUC scheme the total carbon and nitrogen lost upon land-use change by the vegetation pools is partly released directly to the atmosphere, while the rest is distributed to the litter pools, and in case of nitrogen also to the soil mineral N pool. Hence, these pools participate in the active transfers but undergo simultaneously also passive transfers. To describe how the carbon and nitrogen in these pools change one can start from the passive transfers described by (10.62) and add a term for the carbon or nitrogen to be transferred from the vegetation pools. This gives for carbon

$$(10.63) \quad C_{X,i}^{new} = \frac{\sum_{j=1}^K t_{j>i} c_j^{old} f_j^{corr} C_{X,j}^{old}}{f_i^{corr} c_i^{new}} + \frac{1}{f_i^{corr} c_i^{new}} \begin{cases} f_G^{\perp, \top} [(1 - f_{G>A}) F_{G>}^i + (1 - f_{R>A}) F_{R>}^i], & X \in \text{non-woody litter pools} \\ f_W^{\perp, \top} (1 - f_{W>A}) F_{W>}^i, & X \in \text{woody litter pools.} \end{cases}$$

Here  $F_{G>}^i$ ,  $F_{R>}^i$ , and  $F_{W>}^i$  are the active carbon losses from the vegetation pools given by eq. (10.56), and the parameters  $f_{X>A}$ ,  $X \in \{G, R, W\}$ , determine how much carbon is put directly from the respective vegetation pools to the atmosphere upon land-use change. The parameters  $f_X^{\perp}$  and  $f_X^{\top}$ ,  $X \in \{G, W\}$ , (jointly denoted here as  $f_X^{\perp, \top}$ ) determine the splitting for putting litter carbon from the green/reserve and wood pools into above ( $\perp$ ) and below ( $\top$ ) ground pools: these are the same factors used for the natural carbon cycle when using CBALANCE (see section 6.5) or Yasso (see section 6.4.2).<sup>26</sup>

<sup>26</sup>For CBALANCE  $f_X^{\perp, \top}$  can be  $f_{LGA}$ ,  $f_{LGB}$ ,  $f_{LWA}$ , or  $f_{LWB}$ , while for Yasso  $f_X^{\perp, \top}$  is the combined value  $\eta_i f^{\perp}$  or  $\eta_i f^{\top}$ , where  $\eta_i$  determines how carbon is distributed to the different Yasso pools. See also table 6.4.

For the nitrogen litter pools the analogous equations read

$$(10.64) \quad N_{X,i}^{new} = \frac{\sum_{j=1}^K t_{j>i} c_j^{old} f_j^{corr} N_{X,j}^{old}}{f_i^{corr} c_i^{new}} + \frac{1}{f_i^{corr} c_i^{new}} \begin{cases} f_G^{\perp,\top} (1 - f_{G>A}) G_{G>}^i, & X \in \text{non-woody litter pools} \\ f_W^{\perp,\top} (1 - f_{W>A}) \frac{nc_{LW}}{nc_W} G_{W>}^i, & X \in \text{woody litter pools,} \end{cases}$$

where in the case of woody litter the quotient  $nc_{LW}/nc_W$  accounts for the different N:C ratios of wood and woody litter (compare table 7.2), meaning that not all nitrogen to be relocated from the wood pool can be put into the woody litter pool. For the soil mineral N pool a separate equation is needed because (i) it needs to take up the surplus nitrogen from the wood pool, and (ii) there is only a single pool for above and below ground. Hence, the nitrogen relocation of the soil mineral N pool is described by

$$(10.65) \quad N_{smin,i}^{new} = \frac{\sum_{j=1}^K t_{j>i} c_j^{old} f_j^{corr} N_{smin,j}^{old}}{f_i^{corr} c_i^{new}} + (1 - f_{pmob>A}) G_{pmob>}^i + (1 - f_{W>A}) (1 - \frac{nc_{LW}}{nc_W}) G_{W>}^i.$$

Here  $G_{pmob>}^i$  and  $G_{W>}^i$  are the *nitrogen* loss fluxes from the plant mobile and the wood pool obtained from eq. (10.57)<sup>27</sup> and  $f_{pmob>A}$  is an additional parameter for the fraction of nitrogen emitted to the atmosphere from the plant mobile N pool upon land-use change.

It remains to determine the direct emissions to the atmosphere. Such direct emissions are obtained for the litter pool LUC scheme from that part of the dead carbon and nitrogen produced by land-use change that is not already accounted for in the active part of eqs. (10.63) and (10.65), giving for the carbon emissions

$$(10.66) \quad F_{>A}^{box} = veg_{max} \sum_{i=1}^K (f_{G>A} F_{G>}^i + f_{R>A} F_{R>}^i + f_{W>A} F_{W>}^i),$$

and a similar equation with  $F_{X>}^i$  replaced by  $G_{X>}^i$  holds for the nitrogen emissions  $G_{>A}^{box}$ .

### 10.3.3 The ‘anthro pool LUC scheme’: Relocation to product pools

The ‘anthro pool LUC scheme’ implements the ‘Grand Slam Protocol’ [90] for handling carbon affected by land-use change. The idea is that the above ground carbon from the areas converted during land-use change is either used fast (e.g. as fuelwood) or later in the form of products (paper, furniture, houses etc.). For this purpose the carbon is put into three additional pools (‘anthropogenic pools’) from which with a certain delay the carbon is released as CO<sub>2</sub> to the atmosphere. In this way the carbon becomes ‘invisible’ for the rest of JSBACH since it ‘waits’ to be released to the atmosphere. In contrast to the litter pool LUC scheme, in this approach the release of the deforested above ground carbon is independent from environmental conditions. Moreover, the carbon stored in those anthropogenic pools is not

<sup>27</sup>In eq. (10.65) the ratio of N:C-values is used, while in eq. (7.10) their difference is used. The reason is that in the former equation the nitrogen surplus flux is derived from a nitrogen flux ( $G_{W>}^i$ ), while in the latter it is derived from a carbon flux ( $C_W/\tau_W$ ). Noting that the carbon and nitrogen fluxes from the wood pool are related by  $nc_W$  explains why the formulas look different.

artificially increasing wildfires, which happens when using the litter pool LUC scheme where all deforested carbon enters the litter pools that are crucial for the size of wildfires (compare section 9.4). Furthermore, the introduction of anthropogenic pools allows more detailed diagnostics since it separates the carbon affected by land-use change into separately traceable carbon pools. But note that only above ground carbon is used to fill the anthropogenic pools, while the below ground carbon affected by land-use changes is transferred to the litter pools exactly as in the litter pool LUC scheme. Note also that the Grand Slam Protocol deals only with carbon, there are no anthropogenic pools for nitrogen. Therefore, in JSBACH the nitrogen of the biomass flux from the above ground vegetation to the anthropogenic pools is assumed to be lost directly to the atmosphere, while the nitrogen of the belowground biomass losses are put into the below ground nitrogen litter pools.

As already mentioned, the anthro pool LUC scheme introduces three additional pools for each *grid box* (and not for each PFT (or tile)!). Accordingly, these pools collect the conversion carbon from all the different vegetation types in a grid box.<sup>28</sup> An immediate consequence is that these pools are not subject to passive carbon transfers because passive transfers are relocations between pools at different tiles. Note also that in contrast to the other carbon pools in JSBACH, carbon stored in these anthropogenic pools is measured in moles carbon *per vegetated area* and not per canopy area (compare section 1.3.2). The three anthropogenic pools are (see Fig. 10.2):

- $C_{onSite}$ : This denotes a pool containing carbon locally used soon after the land-use change e.g. as fuelwood. It collects the above ground vegetation carbon affected by land-use change from the Green ( $C_G$ ) and Reserve ( $C_R$ ) pools, and partially also from the Wood pool ( $C_w$ ).
- $C_{paper}$ : This denotes a pool containing wood carbon used for intermediate-lived products like paper. It collects a PFT-dependent fraction of the carbon from the wood pool ( $C_w$ ) upon land cover changes.
- $C_{constr}$ : This denotes a pool containing wood carbon used for long-lived products like furniture and construction works. It collects a PFT-dependent fraction of the carbon from the Wood pool ( $C_w$ ) upon land cover changes.

The dynamics of the anthropogenic pools is described by<sup>29</sup>

$$(10.67) \quad \frac{dC_X}{dt} = F_{\triangleright X} - \frac{C_X}{\tau_X}, \quad X \in \{onSite, paper, constr\},$$

where  $\tau_X$  is a pool-specific turnover time<sup>30</sup> and  $F_{\triangleright X}$  represents the carbon fluxes entering the anthropogenic pool  $X$ . Following the logic of Fig. 10.2 these input fluxes are obtained

<sup>28</sup>Another three pools collect carbon from harvesting woody vegetation; see section 10.3.4.2.

<sup>29</sup>Numerically, the dynamics of the anthropogenic pools is iterated by a simple Euler scheme at one day time step as all other carbon and nitrogen pools.

<sup>30</sup>The characteristic decay times are input via namelist `cbal_parameters` as characteristic time  $\hat{\tau}_X$  until the pools have reduced from 100% to 10%. To use these values in (10.67) they have to be converted into the characteristic times  $\tau_X$  until the pools have reduced to  $1/e$ . Thereby the two times are related by  $0.1 = \exp(-\hat{\tau}_X/\tau_X)$ , i.e.  $\tau_X = -1/\ln(0.1^{1/\hat{\tau}_X}) \approx 1/(1 - 0.1^{1/\hat{\tau}_X})$ . This conversion takes place in the code.

from

$$(10.68) \quad F_{\triangleright onSite} = f_W^\perp \sum_{i=1}^K f_{W\triangleright onSite}^i F_{W\triangleright}^i + f_G^\perp \sum_{i=1}^K (F_{G\triangleright}^i + F_{R\triangleright}^i)$$

$$(10.69) \quad F_{\triangleright paper} = f_W^\perp \sum_{i=1}^K f_{W\triangleright paper}^i F_{W\triangleright}^i,$$

$$(10.70) \quad F_{\triangleright constr} = f_W^\perp \sum_{i=1}^K f_{W\triangleright constr}^i F_{W\triangleright}^i,$$

where the  $F_{X\triangleright}^i$ ,  $X \in \{W, G, R\}$  are the carbon loss fluxes of PFT  $i$  from the vegetation pools as given by eq. (10.56) and the parameters  $f_X^\perp$  assure that only the fraction of above carbon is taken from the vegetation pools. The parameters  $f_{W\triangleright X}^i$ ,  $X \in \{onSite, paper, constr\}$  control how the above ground wood carbon affected by land-use change is distributed to the anthro pools  $X$ . More precisely,  $f_{W\triangleright X}^i$  denotes the fraction of the affected wood carbon from PFT  $i$ . These fractions, defined in the land cover type library file (see C.3), sum to 1, meaning that all affected carbon is put into the anthropogenic pools and nothing is left on ground. Making those fractions dependent only on the type of PFT is a simplification compared to the original Grand Slam Protocol [90] where the  $f_{W\triangleright X}^i$  depend also on the geographic location and vary with time.

The below ground vegetation is assumed to die upon land-use change and left in place untouched by man. Accordingly, it is not collected into the anthro pools but transferred as in the litter pool LUC scheme to the respective below ground litter pools. In analogy to eq. (10.63) for carbon the litter pools are thus updated according to

$$(10.71) \quad X_i^{new} = \frac{\sum_{j=1}^K t_{j\triangleright i} c_j^{old} f_j^{corr} X_j^{old}}{f_i^{corr} c_i^{new}} + \frac{1}{f_i^{corr} c_i^{new}} \begin{cases} f_G^\top (F_{G\triangleright}^i + F_{R\triangleright}^i) & \text{for } X \in \text{non-woody litter pools} \\ f_W^\top F_{W\triangleright}^i & \text{for } X \in \text{woody litter pools,} \end{cases}$$

and a similar equation holds for the nitrogen litter pools. Concerning the soil mineral N pool similar considerations as for the litter pool LUC scheme lead to (compare eq. (10.65))

$$(10.72) \quad N_{smin,i}^{new} = \frac{\sum_{j=1}^K t_{j\triangleright i} c_j^{old} f_j^{corr} N_{smin,j}^{old}}{f_i^{corr} c_i^{new}} + f_G^\top G_{pmob\triangleright}^i + f_G^\top \left(1 - \frac{n_{CLW}}{n_{CW}}\right) G_{W\triangleright}^i,$$

meaning that besides passive relocations the soil mineral N pool participates in active nitrogen transfers by receiving the by land-use change affected below ground fraction of the plant mobile nitrogen pool and the surplus nitrogen from the transfer of below ground wood nitrogen.

It remains to describe the carbon and nitrogen emissions that arise from the anthro pool LUC scheme (compare fig. 10.2). The carbon losses to the atmosphere can be read off directly from (10.67) as

$$(10.73) \quad F_{\triangleright A}^{box} = veg_{max} \left( \frac{C_{onSite}}{\tau_{onSite}} + \frac{C_{paper}}{\tau_{paper}} + \frac{C_{constr}}{\tau_{constr}} \right).$$

The prefactor  $veg_{max}$  is needed here to convert from emissions per vegetated area to emissions per box area.

For nitrogen there are no anthropogenic pools so that all above ground nitrogen affected by land-use change is emitted directly to the atmosphere. Hence, for a whole grid box this emission flux is given as the sum of the nitrogen fluxes  $G_{X\triangleright}$  from eq. (10.57), but only from the above ground parts  $f_X^\perp$  of the vegetation pools:

$$(10.74) \quad G_{\triangleright A}^{box} = veg_{max} \sum_{i=1}^K [f_G^\perp (G_{G\triangleright}^i + G_{pmob\triangleright}^i) + f_W^\perp G_{W\triangleright}^i].$$

### 10.3.4 Carbon and nitrogen relocation from harvest

This section describes land use in the form of harvest, where ‘harvest’ is understood here as harvest exclusively from woody vegetation, which is in particular harvest from managed forests. Accordingly, the harvest considered here applies only to the shrub and tree PFTs of JSBACH. Harvest from crops was described in sections 6.3.2 and 7.4.4.

The implementation of wood harvest in JSBACH is based on the data provided by the New Hampshire Land-Use Harmonization project [66], using those data condensed into one value per gridbox and year. By distributing this annual harvest evenly across a year, one easily obtains the mass of carbon harvested per day and area from the vegetated part of a grid box, called  $\overline{H}^C$  in the following.<sup>31</sup> Here the upper index ‘C’ indicates that this is harvested carbon (in contrast to nitrogen for which an upper index ‘N’ will be used) and the overline is used here and in the following to indicate values that refer to the vegetated part of a grid box, i.e. these are values per m<sup>2</sup> vegetated area (compare section 1.3.2). In JSBACH harvest is taken exclusively from the above ground pools of the vegetation (compare fig. 10.3). In addition, although only the harvest of carbon is specified, it goes along with the removal of nitrogen that also needs to be described in the following.

Given the harvest rate  $\overline{H}^C$ , one needs to determine how much of this harvest is taken from the different PFTs, i.e. from the carbon pools at the different tiles. The above ground carbon in tile  $i$  per m<sup>2</sup> of the vegetated part of a grid box is given by

$$(10.75) \quad \overline{C}_{X,i}^{ag} = c_i f_i^{corr} f_X^\perp C_{X,i} \quad X \in \{G, R, W\}$$

where the factors  $f_X^\perp$  denote the fraction of above ground carbon in the respective pools. Accordingly, the total above ground carbon of the woody vegetation is

$$(10.76) \quad \overline{C}^{ag} = \sum_{i \in woody} (\overline{C}_{G,i}^{ag} + \overline{C}_{R,i}^{ag} + \overline{C}_{W,i}^{ag}).$$

The carbon harvest  $\overline{H}_{X,i}^C$  from pool  $X$  in tile  $i$  is then obtained by distributing the total box harvest across the tiles  $i$  with woody vegetation proportional to their above ground carbon, i.e.

$$(10.77) \quad \overline{H}_{X,i}^C = \overline{H}^C \frac{\overline{C}_{X,i}^{ag}}{\overline{C}^{ag}}.$$

In addition, the prescribed total harvest must not be larger than the available above ground carbon of the woody vegetation in a grid box; if it is larger the harvest is reduced accordingly.

<sup>31</sup>The New Hampshire project provides harvest data also as *fraction of forest area* harvested per year. Since the area of forest is model dependent, so will be the total harvest in simulations. Therefore, JSBACH uses only the mass based harvest data.

Harvest not only contains carbon but in particular also nitrogen. For the green and wood pools, the harvested nitrogen can be obtained by applying the known N:C ratios:

$$(10.78) \quad \overline{H}_{X,i}^N = nc_X \overline{H}_{X,i}^C \quad X \in \{G, W\}.$$

To obtain also the harvest from the plant mobile N pool (which has no carbon partner pool) it is assumed that its relative nitrogen loss equals the relative carbon loss of the green pool, i.e.

$$(10.79) \quad \overline{H}_{pmob,i}^N = N_{pmob,i} \frac{\overline{H}_{G,i}^C}{C_{G,i}}.$$

Upon removal of this carbon and nitrogen from the vegetation, the associated pools have to be recomputed. Accounting for the fact that those pool values refer to carbon per canopy area, carbon is removed from these pools at the rescaled rates

$$(10.80) \quad \frac{d}{dt} C_{X,i} = -\frac{\overline{H}_{X,i}^C}{c_i f_i^{corr}}, \quad X \in \{G, R, W\}$$

and a similar equation applies for the nitrogen pools:

$$(10.81) \quad \frac{d}{dt} N_{X,i} = -\frac{\overline{H}_{X,i}^N}{c_i f_i^{corr}}, \quad X \in \{G, W, pmob\}.$$

The fate of the harvested carbon and nitrogen is different depending on the selected carbon pool scheme for land-use change, namely the ‘litter pool LUC scheme’ or the ‘anthro pool LUC scheme’. This is described separately in the following two subsections.

#### 10.3.4.1 Fate of harvested carbon and nitrogen in the ‘litter pool LUC scheme’

Using the litter pool LUC scheme (see fig. 10.3, top), it is assumed that a fraction  $f_{H \triangleright A}$  of the harvest is emitted immediately to the atmosphere (this could e.g. be firewood that is used within weeks). Accordingly, the carbon flux to the atmosphere from a gridbox per box area is

$$(10.82) \quad F_{H \triangleright A}^{box} = veg_{max} f_{H \triangleright A} \sum_{i \in woody} (\overline{H}_{G,i}^C + \overline{H}_{R,i}^C + \overline{H}_{W,i}^C),$$

where the factor  $veg_{max}$  is needed to convert from vegetated area to box area. The analogous equation for the nitrogen losses is

$$(10.83) \quad G_{H \triangleright A}^{box} = veg_{max} \sum_{i \in woody} \left[ f_{H \triangleright A} (\overline{H}_{G,i}^N + \overline{H}_{W,i}^N) + f_{pmob \triangleright A}^{harv} \overline{H}_{pmob,i}^N \right],$$

with a separate factor  $f_{pmob \triangleright A}^{harv}$  for the fraction of nitrogen from the plant mobile N pool emitted to the atmosphere upon harvest.

The remaining harvested carbon and nitrogen is assumed to be used by humans within months. To mimic this, this carbon is simply put into the below ground litter pools that have the appropriate life time. Hence, for CBALANCE, the below ground carbon litter pools increase due to harvest at rates

$$(10.84) \quad \frac{d}{dt} C_{LGb}^i = (1 - f_{H \triangleright A}) \frac{\overline{H}_{G,i}^C + \overline{H}_{R,i}^C}{c_i f_i^{corr}}$$

$$(10.85) \quad \frac{d}{dt} C_{LWb}^i = (1 - f_{H \triangleright A}) \frac{\overline{H}_{W,i}^C}{c_i f_i^{corr}}.$$

while for Yasso the below ground litter pools increase as

$$(10.86) \quad \frac{d}{dt} {}^w \vec{C}_i^\top = \vec{\eta}(1 - f_{H \triangleright A}) \frac{\overline{H}_{G,i}^C + \overline{H}_{R,i}^C}{c_i f_i^{corr}}$$

$$(10.87) \quad \frac{d}{dt} {}^w \vec{C}_i^\top = \vec{\eta}(1 - f_{H \triangleright A}) \frac{\overline{H}_{W,i}^C}{c_i f_i^{corr}}.$$

The  $\vec{\eta}$  values are the same as in Eq. (6.25), describing how litter is distributed to the various Yasso pools.

For nitrogen there is no need not distinguish between CBALANCE and Yasso since the nitrogen pool model is the same (compare fig. 7.1). Hence, the below ground nitrogen litter pools change in both cases according to

$$(10.88) \quad \frac{d}{dt} {}^w \vec{N}_i^\top = (1 - f_{H \triangleright A}) \frac{\overline{H}_{G,i}^N + \overline{H}_{R,i}^N}{c_i f_i^{corr}}$$

$$(10.89) \quad \frac{d}{dt} {}^w \vec{N}_i^\top = \frac{nc_{LW}}{nc_W} (1 - f_{H \triangleright A}) \frac{\overline{H}_{W,i}^N}{c_i f_i^{corr}},$$

where for the transfer of nitrogen from the wood to the wood litter pools it is accounted for the different N:C ratios. The resulting surplus nitrogen is put into the soil mineral N pool together with the harvest from the plant mineral pool:

$$(10.90) \quad \frac{d}{dt} N_{min}^i = (1 - f_{pmob \triangleright A}^{harv}) \frac{\overline{H}_{pmob,i}^N}{c_i f_i^{corr}} + (1 - \frac{nc_{LW}}{nc_W})(1 - f_{H \triangleright A}) \frac{\overline{H}_{W,i}^N}{c_i f_i^{corr}}.$$

### 10.3.4.2 Fate of harvested carbon and nitrogen in the ‘anthro pool LUC scheme’

The ‘anthro pool LUC scheme’ described in section 10.3.3 has its own way of handling the harvest flux (10.77): To track the harvested carbon it introduces three additional pools for each grid box (compare top of Fig. 10.3):

$C_{onSite}^{harv}$ : This pool collects harvest that is left ‘on site’ for local usage e.g. as fuelwood.

$C_{paper}^{harv}$ : This pool collects harvested above ground carbon representing carbon stored in intermediate-lived products like paper.

$C_{constr}^{harv}$ : This pool collects harvested above ground carbon representing carbon stored in long-lived products like furniture and construction works.

Note that as for the other pools of the anthro pool LUC scheme also for these additional three pools the carbon content is measured in carbon per square meter vegetated area. And note also that the three additional harvest pools have no counterpart for nitrogen.

The dynamics of the three harvest pools is described by

$$(10.91) \quad \frac{d}{dt} C_{onSite}^{harv} = (1 - f_{slash}) \sum_{i \in woody} f_{\triangleright onSite}^{harv,i} \left( \overline{H}_{G,i}^C + \overline{H}_{R,i}^C + \overline{H}_{W,i}^C \right) - \frac{C_{onSite}^{harv}}{\tau_{onSite}}$$

$$(10.92) \quad \frac{d}{dt} C_{paper}^{harv} = (1 - f_{slash}) \sum_{i \in woody} f_{\triangleright paper}^{harv,i} \left( \overline{H}_{G,i}^C + \overline{H}_{R,i}^C + \overline{H}_{W,i}^C \right) - \frac{C_{paper}^{harv}}{\tau_{paper}}$$

$$(10.93) \quad \frac{d}{dt} C_{constr}^{harv} = (1 - f_{slash}) \sum_{i \in woody} f_{\triangleright constr}^{harv,i} \left( \overline{H}_{G,i}^C + \overline{H}_{R,i}^C + \overline{H}_{W,i}^C \right) - \frac{C_{constr}^{harv}}{\tau_{constr}}$$

These pools respire carbon with the same time constants  $\tau_{paper}$ ,  $\tau_{constr}$ , and  $\tau_{onSite}$  as the anthropogenic pools collecting carbon from land-use change (compare eq. (10.67)). The pools are filled with the above ground carbon harvest  $\overline{H}_{X,i}^C$  (given by eq. (10.77)) from the three vegetation carbon pools  $X \in \{G, R, W\}$  of all woody PFTs. The harvest is distributed across the harvest pool according to PFT-specific harvest fractions  $f_{>paper}^{harv,i}$  and  $f_{>constr}^{harv,i}$ , with  $f_{>onSite}^{harv,i} = 1 - f_{>paper}^{harv,i} + f_{>constr}^{harv,i}$ , defined in the land cover type library file (see C.3). The parameter  $f_{slash}$  determines the fraction of harvest remaining as residues unused in place, so that  $1 - f_{slash}$  is the fraction used by man. This additional parameter for 'slash' is needed because the LUH harvest data in connection with CMIP<sup>32</sup> contain a fraction of 30% that is not used but simply left in place.<sup>33</sup>

This “slash fraction” is not added to the above ground litter but to the below ground pools to prevent that wildfires are increased by harvest. Hence, the relocation of the slash fraction to the litter pools is described for the CBALANCE litter pools in tile  $i$  by

$$(10.94) \quad \frac{d}{dt} C_{LGb}^i = f_{slash} \frac{\overline{H}_{G,i}^C + \overline{H}_{R,i}^C}{c_i f_i^{corr}}$$

$$(10.95) \quad \frac{d}{dt} C_{LWb}^i = f_{slash} \frac{\overline{H}_{W,i}^C}{c_i f_i^{corr}}$$

while for Yasso the below ground litter pools increase as

$$(10.96) \quad \frac{d}{dt} \psi \vec{C}_i^{\tau} = \vec{\eta} f_{slash} \frac{\overline{H}_{G,i}^C + \overline{H}_{R,i}^C}{c_i f_i^{corr}}$$

$$(10.97) \quad \frac{d}{dt} w \vec{C}_i^{\tau} = \vec{\eta} f_{slash} \frac{\overline{H}_{W,i}^C}{c_i f_i^{corr}}$$

These equations are identical to those for the litter pool harvest scheme described above (see the equations from (10.84) on), except that  $f_{slash}$  replaces the fraction  $1 - f_{H>A}$  of harvest not put directly into the atmosphere.

With this replacement of  $1 - f_{H>A}$  by  $f_{slash}$ , equations (10.88) to (10.90) from the previous section describing the litter pool harvest scheme also describe the update of the belowground nitrogen pools for the anthro pool harvest scheme described here.

Finally, the carbon emissions to the atmosphere can be read off from eqs. (10.92) to (10.91):

$$(10.98) \quad F_{H>A}^{box} = veg_{max} \left( \frac{C_{paper}^{harv}}{\tau_{paper}} + \frac{C_{constr}^{harv}}{\tau_{constr}} + \frac{C_{onSite}^{harv}}{\tau_{onSite}} \right)$$

where the  $veg_{max}$  is needed to convert from flux per vegetated area to box area. And the nitrogen fluxes accompanying the carbon fluxes into the harvest pools that are not further tracked but emitted to the atmosphere are given from

$$(10.99) \quad G_{>A}^{harv} = (1 - f_{slash}) \sum_{i \in woody} \left( \overline{H}_{G,i}^N + \overline{H}_{W,i}^N + \overline{H}_{pmob,i}^N \right),$$

where the harvested nitrogen  $\overline{H}_{X,i}^N$  was obtained for the green and wood components of vegetation in eq. (10.78) and for the plant mobile N pool in eq. (10.79).

<sup>32</sup>See <http://luh.umd.edu>.

<sup>33</sup>Personal communication by Julia Pongratz, May 2016.

## 10.4 Diagnostics

The land-use change and harvest models provide the diagnostics summarized in table 10.1.

Table 10.1: Diagnostic output of the land-use change model.

name in code	symbol	meaning
LCC_flux_box_C2atmos	$F_{\triangleright A}^{box}$	Carbon flux emitted directly to atmosphere from land-use change.
LCC_flux_box_N2atmos	$G_{\triangleright A}^{box}$	Nitrogen flux emitted directly to atmosphere from land-use change.
boxC_flux_onSite_2_atmos_LCC	$veg_{max} \frac{C_{onSite}}{\tau_{onSite}}$	Carbon flux from onSite pool to atmosphere.
boxC_flux_paper_2_atmos_LCC	$veg_{max} \frac{C_{paper}}{\tau_{paper}}$	Carbon flux from paper pool to atmosphere.
boxC_flux_construction_2_atmos_LCC	$veg_{max} \frac{C_{constr}}{\tau_{constr}}$	Carbon flux from construction pool to atmosphere.
boxC_flux_onsite_2_atmos_harvest	$veg_{max} \frac{C_{onSite}^{harv}}{\tau_{onSite}}$	Carbon flux from onSite pool for harvest to atmosphere.
boxC_flux_paper_2_atmos_harvest	$veg_{max} \frac{C_{paper}^{harv}}{\tau_{paper}}$	Carbon flux from paper pool for harvest to atmosphere.
boxC_flux_construction_2_atmos_harvest	$veg_{max} \frac{C_{constr}^{harv}}{\tau_{constr}}$	Carbon flux from construction pool for harvest to atmosphere.
box_flux_harvest	$veg_{max} \bar{H}$	Total harvest from woody vegetation for full grid box (may be different from prescribed harvest).
box_flux_harvest_2atmos	$\bar{F}_{H \triangleright A}$	Harvest respired as CO <sub>2</sub> to the atmosphere.
boxC_onSite_LCC	$veg_{max} C_{onSite}$	Carbon content of onSite pool.
boxC_paper_LCC	$veg_{max} C_{paper}$	Carbon content of paper pool.
boxC_constr_LCC	$veg_{max} C_{constr}$	Carbon content of construction pool.
boxC_onsite_harvest_LCC	$veg_{max} C_{onSite}^{harv}$	Carbon content of onSite pool for harvest.
boxC_paper_harvest_LCC	$veg_{max} C_{paper}^{harv}$	Carbon content of paper pool for harvest.
boxC_construction_harvest_LCC	$veg_{max} C_{constr}^{harv}$	Carbon content of construction pool for harvest.

## 10.5 Implementation details

### 10.5.1 Code structure

The code for landcover and land-use change is located mainly in the module `mo_cba1_landcover_change`. This covers the code for initialization (`init_landcover_change()`) as well as the main routines `do_landuse_transitions()` and `do_landcover_change()` called from the JSBACH main routine `jsbach_inter_1d()` implementing the two models for land-use change, the ‘transition model’ (section 10.2.1) and the ‘maps model’ (section 10.2.2). The data for these models is read in via the routines `read_landuse_transitions()` and `read_landcover_fractions()`, while harvest data are read in by `read_harvest()`. The main routines for the two implemented models of land-use change not only perform the change in the cover

fractions, but also call the routines for carbon and nitrogen relocations induced by land-use change, namely, depending on the particular model, the routines `C_relocation_from_LUtransitions()` and `relocate_CarbonAndNitrogen()`, and the routine for relocation of carbon and nitrogen by harvest `C_relocation_from_harvest()`; these relocation routines are found in the module `mo_cbal_cpools`. If the ‘anthro pool LUC scheme’ is used, the former two routines call `C_loss_and_update_anthro_pools()` from the same module to transfer carbon to the product pools (sections 10.3.3 and 10.3.4.2).

### 10.5.2 Control of model setup by namelist switches

The variants to run the land use and land-use change components are controlled by the following switches from the `jsbach_ctl` namelist:

`lcc_forcing_type`: Switches land-use transitions on or off and thereby selects the maps or transition model.

`lcc_scheme`: Chooses the ‘litter pool’ or ‘anthro pool’ LCC scheme. Note that each of these choices goes along with its own model for handling the harvested carbon (see section 10.3.4). For the possible values of these namelist switches see in the appendix section C.2.11, or for CBALONE runs C.2.18.

### 10.5.3 Model parameters

Those parts of the ‘maps’ and ‘transitions’ models, describing how anthropogenic land cover change affects vegetation cover, are free of any parameters. But the associated models for carbon and nitrogen relocation are controlled by a number of parameters. These are either hard-coded, or found in the land cover type library file (see appendix C.3). Table 10.2 shows where parameters are defined and what their names are in the namelist or land cover type library file.

Table 10.2: Parameters of the models for land use and land-use change

name	symbol	where defined	meaning
<i>Parameters of ‘litter pool LUC scheme’:</i>			
<code>frac_green_2_atmos</code>	$f_{G \triangleright A}$	namelist <sup>1</sup>	Fraction of litter from green pool emitted directly to the atmosphere
<code>frac_reserve_2_atmos</code>	$f_{R \triangleright A}$	namelist <sup>1</sup>	Fraction of litter from reserve pool emitted directly to the atmosphere
<code>frac_wood_2_atmos</code>	$f_{W \triangleright A}$	namelist <sup>1</sup>	Fraction of litter from wood pool emitted directly to the atmosphere
<code>frac_mobile_2_atmos</code>	$f_{pmob \triangleright A}$	namelist <sup>1</sup>	Fraction of litter from plant mobile N pool emitted directly to the atmosphere
<code>frac_harvest_2_atmos</code>	$f_{H \triangleright A}$	namelist <sup>1</sup>	Fraction of total harvest emitted directly to the atmosphere
<code>frac_mo_harv_2_atmos</code>	$f_{pmob \triangleright A}^{harv}$	namelist <sup>1</sup>	Fraction of harvest from plant mobile N pool emitted directly to the atmosphere in the ‘litter pool LUC scheme’
<i>Parameters of ‘anthro pool LUC scheme’:</i>			
<code>frac_lcc_C_2_paper</code>	$f_{W \triangleright paper}^i$	lctlib <sup>2</sup>	fraction of woody litter put into paper pool
<code>frac_lcc_C_2_construction</code>	$f_{W \triangleright constr}^i$	lctlib <sup>2</sup>	fraction of woody litter put into construction pool

*table continued on next page*

Table 10.2: Parameters of the models for land use and land-use change — continued

name in code	symbol	where defined	meaning
<code>frac_harv_C_2_paper</code>	$f_{\triangleright paper}^{harv,i}$	lctlib <sup>2</sup>	fraction of non-slash wood harvest put into paper pool
<code>frac_harv_C_2_construction</code>	$f_{\triangleright constr}^{harv,i}$	lctlib <sup>2</sup>	fraction of non-slash wood harvest put into construction pool
<code>frac_harvest_2_slash</code>	$f_{slash}$	namelist <sup>1</sup>	fraction of total harvest left unused as litter
<code>tau_onSite</code>	$\tau_{onSite}$	namelist <sup>1</sup>	decay time [days] of onSite and onSite-harvest pool (annual time scale)
<code>tau_paper</code>	$\tau_{paper}$	namelist <sup>1</sup>	decay time [days] of paper and paper-harvest pools (decadal time scale)
<code>tau_construction</code>	$\tau_{construction}$	namelist <sup>1</sup>	decay time [days] of construction and construction-harvest pools (centennial time scale)

<sup>1</sup> namelist `cbal_parameters` (see appendix C.2.4).<sup>2</sup> land cover type library file (see appendix C.3).

# APPENDICES

# Appendix A

## Compiling and Running

### Contents

---

<a href="#">A.1 Overview</a>	184
<a href="#">A.2 JSBACH Compilation</a>	185
<a href="#">A.3 Running JSBACH</a>	186
<a href="#">A.4 Running CBALONE</a>	187
<a href="#">A.5 Running HD standalone</a>	188

---

### A.1 Overview

JSBACH3 is part of the Max Planck Earth System model MPI-ESM and is generally distributed together with the Earth System model setup. MPI-ESM comprises the atmospheric general circulation model ECHAM6, the ocean circulation model MPIOM, the land surface model JSBACH3 and the ocean bio-geochemistry model HAMOCC, as well as the coupling library oasis3mct and the runtime infrastructure mkexp. Table [A.1](#) lists the possible run modes of JSBACH. As full model, JSBACH can be run either as land component of ECHAM, or as an independent DGVM, externally driven by observational or simulated climate data (“JSBACH standalone”). The two other modes “CBALONE” and “HD standalone” are designed to run certain sub-components separately.

Model compilation is based on the GNU autotools `configure` and `GNU make`. The formerly used standard compiling environment SCE that had first been developed within the PRISM project in 2005 is no longer supported.

There are two methods to run JSBACH. The ECHAM community generally uses the `mkexp` infrastructure to run ECHAM6 (including the land surface scheme JSBACH3). The infrastructure has been developed by Karl-Hermann Wieners from MPI-M. A documentation is available in

*[mpiesm/util/mkexp/doc/mkexp.pdf](#)* .

The `mkexp` infrastructure also allows to run MPI-ESM and will be used for the institute’s CMIP6 simulations. However, for the time being, `mkexp` does not support JSBACH standalone nor CBALONE experiments.

As an alternative, the ‘cosmos’ runtime environment, also known as SRE or IMDI-environment, can be used. It was first developed within the PRISM project and was used

Table A.1: The different run modes of JSBACH. Each such mode has its own executable.

mode	description
ECHAM/JSBACH	JSBACH runs as component of ECHAM. Used for AMIP and MPI-ESM simulations.
JSBACH standalone	JSBACH runs driven by external forcing data. Simulations can not only be done globally (see appendix G), but also regionally or at arbitrarily scattered grid boxes (“site level” simulations; see appendix H).
CBALONE	For running separately only those JSBACH components with long memory (carbon and nitrogen cycle, natural and anthropogenic landcover change), driven by data from a previous JSBACH simulation (see appendices A.4 and G).
HD standalone	For separately running the Hydrological Discharge model (section 2.8) driven by precipitation data (see appendix A.5).

for the CMIP5 simulations. This runtime environment supports simulations with JSBACH standalone, with ECHAM6 and within the coupled MPI-ESM setup.

In this chapter we will give brief instructions on how to compile and run JSBACH within the different configurations.

## A.2 JSBACH Compilation

A compile script is provided to compile the different model configurations with JSBACH on the DKRZ supercomputer mistral and on MPI-M Linux PCs:

```
mpiesm/landveg-compile.ksh
```

Here, *mpiesm* is the root directory of this distribution (e.g. *~/mpiesm-landveg*). The compile script begins with a definition section, that needs to be adapted by the user.

```
#-----
# setup definition
#
vers=N01          # model version (tag used in the executable name)
cplmod=mpiesm-s  # coupled model name  mpiesm-s    (jsbach3,cbalance,hd)
                  #                    mpiesm-s4   (jsbach4)
                  #                    mpiesm-as    (echam6)
                  #                    mpiesm-as4   (echam6 with jsbach4)
                  #                    mpiesm-asob (mpiesm-1.2)
compiler=nag      # compiler to be used (nag/intel)
configure=yes     # yes: run configure, e.g. if dependencies changed
automake=no       # yes: run automake (only with configure=yes)
make_argument="-j 4" # number of processes for make, 'clean' or 'distclean'
#-----
```

The variable `vers` defines a model version tag, that will be attached to the executable name. It will be referred to by the run script to identify the correct executable. Variable `cplmod` defines the coupled (or uncoupled) model configuration, i.e. `mpiesm-s` for JSBACH3 standalone simulations, CBALONE or HD standalone, `mpiesm-as` for ECHAM6 (including JSBACH3) and `mpiesm-asob` for the complete Earth system model including ECHAM6, the ocean model MPIOM and the ocean bio-geochemistry model HAMOCC. It is also possible to compile JSBACH4 in a standalone setup (`mpiesm-s4`) or within ECHAM6 (`mpiesm-as4`). However, this documentation focuses on JSBACH3.

Two different compilers are supported (variable `compiler`). We recommend to use the NAG compiler ('nag') for testing and debugging and the intel compiler for production runs.

With variable `configure` you define whether or not you want to run configure. Choose 'yes' the first time you do the compilation of a specific model version, or if make dependencies changed (e.g. if you added new USE statements to the source code). With `configure=no` running configure is omitted, which makes the compilation considerably faster. The variable `automake` switches on and off automake. Automake is needed if new modules have been added to the code. If you run automake, you also need to run configure.

It is possible to specify different make arguments with variable `make_argument`. Option `-j` defines the number of processes that make will use for compilation. To delete all binaries created in previous tries you can use option 'clean' or 'distclean'. This is especially useful in the MPI-ESM setup, where the compiled object and module files are placed in the source code directories.

To start the compilation just execute the compile script:

```
cd mpiesm
./landveg-compile.ksh
```

Again, *mpiesm* represents the root directory of this distribution. The compile script will generate a build directory for the binaries and executables

```
mpiesm/build-cplmod-vers
```

with subdirectory `bin` for the executables. Besides, a short script listing all modules loaded for compilation (`modules_vers`) is placed into the build directory. It is referred to by the run script to make sure the same modules are loaded at compile and run time.

### A.3 Running JSBACH

This section describes how to run JSBACH either as part of ECHAM or in standalone mode (compare table A.1).

To run JSBACH3 using the mkexp infrastructure see the documentation

```
mpiesm/util/mkexp/doc/mkexp.pdf.
```

Please note that the mkexp infrastructure only supports simulations of ECHAM6 or MPI-ESM (including JSBACH3 as land component) but no JSBACH3 standalone simulations. In this section, we give instructions on running JSBACH using the cosmos scripting.

We do not provide ready-to-use run scripts, but a set of tools to generate the scripts for

a specific model configuration on a specific machine. The running environment embraces model execution as well as output data post-processing. Additional scripts are available for visualization and for archiving on the DKRZ tape archive.

To generate the run scripts, change to the tools directory and run `Create_TASKS.frm`

```
cd mpiesm/util/running/tools
./Create_TASKS.frm cplmod --id expid [--node node].
```

Depending on the command line parameters, scripts will be generated for the coupled model configuration `cplmod` (compare section A.2) on the computer `node`. Use 'linux-x64' as node name for MPI-M linux PCs. No node name needs to be given for simulations on mistral. Parameter `expid` defines an experiment ID of your choice. It is independent of the model tag `vers` in the executable names. This allows running several experiments (with different experiment IDs) using the same executables.

The first call of `./Create_TASKS.frm` with a specific experiment ID triggers the generation of a setup file:

```
mpiesm/util/running/setup/setup_cplmod_expid.
```

This file contains all configurable parameters for the simulation. You need to go through the file and edit the parameters according to the design of your experiment. In particular, the parameters `atmvers` or `srfvers` (the executable tags) have to be consistent with the version `vers` defined in the compile script. Besides, for simulations on mistral, you need to set a DKRZ account for computing resources `account` (e.g. 'mj0060'). Once you are done, call `./Create_TASKS.frm` again using the same command line parameters to create the run scripts.

Table A.2: Possible runtime tasks.

<code>expid.run</code>	model execution
<code>expid.post</code>	output data post-processing (E.1)

The scripts are placed in directory

```
mpiesm/experiments/expid/scripts.
```

To start the simulation, submit the run script either interactively (`./expid.run`) or using the queuing system of the executing machine (e.g. `sbatch` or `qsub`). The post-processing script will be submitted automatically.

## A.4 Running CBALONE

A run script for the CBALONE model is available in

```
mpiesm/contrib/runCbalone.
```

The script generates the CBALONE namelists and processes the input forcing data from a previously conducted JSBACH simulation, if necessary. There is no automatic restart cycling, the model runs in one go through the whole simulation period. A restart file is generated at the end of the simulation, which can be read in via `read_cpools`, `read_npools` and `read_fpc` to prolong the experiment.

The CBALONE model is not parallelized, it only runs on one processor. Nevertheless, it

is adopted to the queuing system of the DKRZ supercomputer 'mistral'. Running the model on a linux PC is generally more efficient.

The upper section of the script lists configurable parameters. Most of them are self-explanatory, and comments give the possible choices. The following list gives additional help:

**jsbach\_file**: defines the jsbach initial file. It should be the same file as used in the experiment providing the forcing data.

**cyear1, cyear2**: in equilibration runs, the model is generally forced with repeating climate data. Parameter **cyear1** is the first and **cyear2** the last year of this forcing period.

**syear1, syear2**: the first and the last year of the simulation. For a 1500-year spinup simulation choose e.g. **syear1=1** and **syear2=1500**. The simulation years are used in the file names for land cover maps or transitions, thus for a cbalance run with realistic land cover change, the simulation years should be defined e.g. as **syear1=1850** and **syear2=2000**.

**read\_cpools**: read initial values for the carbon pools from an external file. The corresponding file name is defined by parameter **cpool\_file**. The cpool file can serve as a restart file to continue a CBALANCE experiment (compare section C.6).

**read\_fpc**: read initial values for the FPCs of the dynamic vegetation. As the **cpools\_file**, the **fpc\_file** can be used as a restart file.

**input\_scaling**: Input data scaling is interesting in transient CBALANCE simulations under changing CO<sub>2</sub> conditions. Forcing data needs to be available only for two time periods, at the beginning and at the end of the experiment. The data of the first period is used repetitively. In runs with input scaling, the forcing data is scaled with the climatological difference of the two forcing periods (**climdiff\_file**) and the development of the atmospheric CO<sub>2</sub> (**scaling\_file**).

**ryear1, ryear2**: reference years for the input data scaling. Parameter **ryear1** corresponds to the first climatology of the forcing data, **ryear2** to the second climatology.

**get\_forcing\_data**: If true, forcing data for the CBALANCE simulation is generated from an existing JSBACH simulation. Three additional parameters are needed to define where the data of the forcing experiment resides. The period of the forcing data is defined by the climate year variables **cyear1** and **cyear2**.

## A.5 Running HD standalone

A run script for the standalone hydrological discharge model is available in

*mpiesm/contrib/runHDalone.*

The script generates the namelists needed to run the HD model. There is no automatic restart cycling, the model runs in one go through the whole simulation period. However, a restart file (**hdrestart.res.srv**) is generated at the end of the simulation and can be used to prolong the experiment.

The HD model is not parallelized, it uses one processor only.

## Appendix B

# Preprocessor Directives

Only few preprocessor directives appear within the JSBACH source code. Following the coding rules, physical switches should be rather controlled by namelist parameters to avoid extra compilation and to keep the code clear. Some technical preprocessor directives remain:

Table B.1: JSBACH preprocessor directives

STANDALONE	build the standalone JSBACH or CBALONE models
_CCDAS	build model for the Carbon Cycle Data Assimilation System CC-DAS
ECHAM_FRACTIONAL	build echam using a fractional land sea mask
NOMPI	compile without using the message passing interface MPI library
_OPENMP	compile for parallel runs using OpenMP
INPUT_IN_ECHAM	the independent reading module <code>mo_input</code> uses ECHAM infrastructure
INPUT_IN_ICON	the independent reading module <code>mo_input</code> uses ICON infrastructure
NO_FLUSH_AVAIL	to be used with input reading module if flush function is not available
JSBACH4	simulation with JSBACH4 (and not with JSBACH3)
__NO_JSBACH_HD__	simulation without hydrological discharge model HD
HAVE_F2003	<code>mo_input</code> uses features relying on Fortran 2003 specific code
USE_MPI	used for parallel applications without ECHAM infrastructure using <code>mo_input</code>
__ASYNC_GATHER	use MPI asynchronous gathering routines on some machines
__ASYNC_GATHER_ANY	use MPI asynchronous gathering routines on some machines
__NAG__, __PGI__, etc.	compiler specific settings
__SX__, __XT3__, etc.	hardware specific settings

Using the compile script `landveg-compile.ksh` (section A.2) preprocessing directives are set automatically to meet the model configuration, the execution host and the compiler.

# Appendix C

## Input Files

### Contents

---

C.1	Overview	191
C.2	The JSBACH namelists file <code>namelist.jsbach</code>	193
C.2.1	Namelist <code>albedo_ctl</code>	194
C.2.2	Namelist <code>bethy_ctl</code>	194
C.2.3	Namelist <code>cbalance_ctl</code>	194
C.2.4	Namelist <code>cbal_parameters_ctl</code>	195
C.2.5	Namelist <code>climbuf_ctl</code>	196
C.2.6	Namelist <code>disturbance_ctl</code>	196
C.2.7	Namelist <code>dynveg_ctl</code>	197
C.2.8	Namelist <code>fire_jsbach_ctl</code>	197
C.2.9	Namelist <code>fire_thonicke_ctl</code>	198
C.2.10	Namelist <code>hydrology_ctl</code>	199
C.2.11	Namelist <code>jsbach_ctl</code>	199
C.2.12	Namelist <code>soil_ctl</code>	201
C.2.13	Namelist <code>windbreak_jsbach_ctl</code>	202
C.2.14	Namelist <code>jsbalone_ctl</code>	202
C.2.15	Namelist <code>jsbalone_parctl</code>	203
C.2.16	Namelist <code>forcing_ctl</code>	203
C.2.17	Namelist <code>jsbgrid_ctl</code>	207
C.2.18	Namelist <code>cbalone_ctl</code>	207
C.2.19	Namelist <code>hdalone_ctl</code>	209
C.3	The land cover type library File <code>lctlib.def</code>	209
C.4	Initial data file <code>jsbach.nc</code>	213
C.4.1	Typical JSBACH setups	213
C.4.2	The initial file variables	214
C.4.3	Grid Information	216
C.4.4	Land Sea Mask	217
C.4.5	Orography	217
C.4.6	Vegetation Cover	217
C.4.7	Soil Conditions	218

C.4.8 Albedo . . . . . 218

C.5 Boundary and Initial Data for the Hydrological Discharge (HD) model . . . . . **219**

C.5.1 HD Boundary Conditions: `hdpara.nc` . . . . . 219

C.5.2 HD Initial Conditions: `hdstart.nc` . . . . . 219

C.5.3 The HD remapping matrix: `rmp_hd.nc` . . . . . 220

C.6 Initial data for the Carbon Pools: `Cpools.nc` . . . . . **220**

C.7 Initial data for the Nitrogen Pools: `Npools.nc` . . . . . **222**

C.8 Initial data for foliage projective cover (FPC): `fpc.nc` . . . . . **223**

C.9 Initial data for climate buffer variables: `climbuf.nc` . . . . . **224**

C.10 Forcing data for JSBACH standalone runs: `climate.nc` . . . . . **225**

C.11 Forcing data for CBALONE: `exp_date.jsbach.yDay_Mean.nc` . . . . . **226**

## C.1 Overview

Table C.1 lists the input files for the different JSBACH configurations. The standalone version of JSBACH reads atmospheric input data to drive the simulation. However, these input files are not necessary in the ECHAM configuration, where JSBACH receives the atmospheric input directly from the atmosphere model.

Table C.1: JSBACH input files. Parentheses indicate optional input.

File Name in the Archive	File Name at Runtime	Model Configuration		
		ECHAM	JSBACH	CBALONE
<b>Configuration Files</b>				
<code>namelist.jsbach</code> (here doc.)	<code>namelist.jsbach</code>	x	x	x
<code>lctlib.nlctnlet.def</code>	<code>lctlib.def</code>	x	x	x
<b>Initial Data</b>				
<code>jsbach.grid.nsoillayers.refyear.nc</code>	<code>jsbach.nc</code>	x	x	x
<code>hdpara.nc</code>	<code>hdpara.nc</code>	(x)	(x)	(x)
<code>hdstart.nc</code>	<code>hdstart.nc</code>	(x)	(x)	(x)
<code>rmp_hd.nc</code>	<code>rmp_hd.nc</code>	(x)	(x)	(x)
<code>Cpools.grid.refyear.nc</code>	<code>Cpools.nc</code>	(x)	(x)	(x)
<code>Cpools.grid.refyear.nc</code>	<code>Npools.nc</code>	(x)	(x)	(x)
<code>Cpools.grid.refyear.nc</code>	<code>fpc.nc</code>	(x)	(x)	(x)
<code>Cpools.grid.refyear.nc</code>	<code>climbuf.nc</code>	(x)	(x)	(x)
<b>Forcing Data</b>				
<code>Climate.grid.year.nc</code>	<code>climate.year.nc</code>		x	
<code>exp_date.jsbach.yDay_Mean.nc</code>	<code>exp_date.jsbach.yDay_Mean.nc</code>			x
<b>With Land Use Change</b>				
<code>ludir/cover_fract.grid.year.nc</code>	<code>cover_fract.year.nc</code>	with <code>lcc_forcing_type=maps</code>		
<code>ludir/LUH_transitions.grid.year.nc</code>	<code>landuseTransitions.year.nc</code>	with <code>lcc_forcing_type=transitions</code>		
<code>ludir/LUH_harvest.grid.year.nc</code>	<code>landuseHarvest.year.nc</code>	with <code>lcc_forcing_type=transitions</code>		
<b>With Fire</b>				
<code>firedir/LISOTD_HRMC_V2.2.grid.nc</code>	<code>lightning.nc</code>	with <code>fire_name=thonicke</code>		
<code>firedir/population_density_HYDE.grid.nc</code>	<code>population_density.nc</code>	with <code>fire_name=thonicke</code>		
<code>firedir/a_nd.grid.nc</code>	<code>a_nd.file.nc</code>	with <code>fire_name=thonicke</code>		
<code>emission_factors.nc</code>	<code>emission_factors.nc</code>	with <code>fire_emissions=true</code>		
<b>With Nitrogen</b>				
<code>grid.ndepo_CMIP_NCAR_CCMI-1-0_gr_year.nc</code>	<code>Ndepo.nc</code>	with <code>nitrogen=true</code>		

*table continued on next page*

Table C.1: JSBACH input files — continued

File Name in the Archive	File Name at Runtime	Model Configuration	
		ECHAM	JSBACH CBALONE
With Greenhouse Gas Forcing			
<code>greenhouse_scenario.nc</code>	<code>greenhouse_gases.nc</code>	read_ghg=true	
Restart Data			
<code>restart_exp_jsbach_prevdate.nc</code>	<code>restart_exp_jsbach.nc</code>	x	x
<code>restart_exp_veg_prevdate.nc</code>	<code>restart_exp_veg.nc</code>	x	x
<code>restart_exp_forcing_prevdate.nc</code>	<code>restart_exp_forcing.nc</code>		x
<code>restart_exp_driving_prevdate.nc</code>	<code>restart_exp_driving.nc</code>		x
<code>restart_exp_surf_prevdate.nc</code>	<code>restart_exp_surf.nc</code>	x	
<code>restart_exp_accw_prevdate.nc</code>	<code>restart_exp_accw.nc</code>	x	
<code>restart_exp_yasso_prevdate.nc</code>	<code>restart_exp_yasso.nc</code>	(x)	(x)
<code>restart_exp_nitrogen_prevdate.nc</code>	<code>restart_exp_nitrogen.nc</code>	(x)	(x)
<code>restart_exp_disturb_prevdate.nc</code>	<code>restart_exp_disturb.nc</code>	(x)	(x)
<code>hdstart_exp_prevdate.nc</code>	<code>hdstart.nc</code>	(x)	(x)
<code>restart_Cbalone_syear_cyear.nc</code>	<code>Cpools.nc, Npools.nc, fpc.nc, climbuf.nc</code>		x

The file names in the first column of table C.1 correspond to the default JSBACH configuration. The variable *nlct* represents the number of land cover types defined in the land cover type library file, *grid* is the acronym of the model grid, *nsoil* gives the number of soil layers and *refyear* defines a reference year. The files containing the forcing data for CBALONE runs have the ID *exp* in their names, which is the ID of the experiment the data stems from. Besides, *date* (yyyymm) gives the time period of the forcing data. The restart files are specified with the experiment ID *exp* and with the date the restart file was written *prevdate* (yyyymmdd) or *syear\_cyear*, respectively. Here variable *syear* indicates the simulation year, *cyear* the corresponding year of the climate forcing data.

The file names the model actually expects at run time are listed in the second column of table C.1. Most of these names are configurable by namelist parameters (compare section C.2). The file names listed here correspond to default settings.

Most input files reside in common pool directories, available on all MPI-M and DKRZ machines, namely

`/pool/data/JSBACH/input/rev/res/.`

Input is provided here in several resolutions *res* for all commonly used experimental setups. Revision numbers *rev* assure, that the input data is also available for older source code versions. Read `/pool/data/JSBACH/pool_data_versions.txt` for details.

At the beginning of an experiment, the input data is gathered from the pool to the experiment's input directory

`data/experiments/expid/input/jsbach.`

The variable *data* is configurable in the setup file (compare section A.3).

The configuration files `namelist.jsbach` and `lctlib.def` do not reside in the pool directories. As they are written in ASCII format they profit from our version control system (subversion, svn). The `lctlib` file can be found in

`mpiesm/util/running/adjunct_files/jsbach`

whereas the namelists file is generated as a 'here' document by the runscript.

In the following sections each of the JSBACH input files is described in detail in the order given in the above table (C.1).

## C.2 The JSBACH namelists file `namelist.jsbach`

In the standard setup, the JSBACH namelist file `namelist.jsbach` is generated by the run script. This assures that the namelists are up-to-date and the JSBACH configuration matches the configurations of the other Earth System model components, if run in a coupled configuration.

The JSBACH namelist file includes several independent Fortran namelists:

- `jsbach_ctl` defines the basic settings of a JSBACH simulation. The namelist includes parameters to switch on or off JSBACH modules, and controls IO.
- `albedo_ctl` defines parameters that are used in the albedo scheme (sec. 4.1)
- `bethy_ctl` controls the photosynthesis (BETHY) module (chapter 5)
- `cbalance_ctl` controls the carbon and nitrogen modules (chapters 6 and 7)
- `cbal_parameters_ctl` defines parameters of the carbon and nitrogen modules (chapters 6 and 7)
- `climbuf_ctl` defines parameters for the calculation of multi-year averages of climate variables used in the disturbance and DYNVEG modules (chapters 8 and 9).
- `disturbance_ctl` controls the disturbance modules, i.e. fire and windthrow calculations (chapter 8)
- `dynveg_ctl` controls the dynamic vegetation (chapter 9)
- `fire_jsbach_ctl` defines parameters for the 'jsbach' fire scheme (sec. 8.3.1)
- `fire_thonicke_ctl` defines parameters for the 'thonicke' fire scheme (SPITFIRE) (sec. 8.3.2)
- `hydrology_ctl` controls the hydrological discharge (HD) module (section 2.8)
- `soil_ctl` defines parameters used in the soil module (secs. 2.2, 2.3, and 2.4)
- `windbreak_jsbach_ctl` defines parameters for the 'jsbach' windthrow scheme (sec. 8.2)

The following namelists are used only in JSBACH standalone runs:

- `jsbalone_ctl` controls the flow of the JSBACH standalone experiment. In a combined run ECHAM/JSBACH these parameters are defined in the ECHAM namelist `runctl`.
- `jsbalone_parctl` corresponds to the ECHAM namelist `parctl`. It defines parameters for parallelization.
- `forcing_ctl` defines the type and frequency of the atmospheric forcing
- `jsbgrid_ctl` includes parameters to specify the model grid and parallelization issues

The CBALANCE model again has a special namelist:

- `cbalone_ctl` specifies a CBALONE experiment. It corresponds to `runctl` of combined simulations ECHAM/JSBACH or to `jsbalone_ctl` of JSBACH standalone experiments.

It is possible to run the hydrology discharge (HD) model (section 2.8) as standalone model. These standalone simulations are controlled by the namelist `hdalone_ctl`:

- `hdalone_ctl` is a namelist needed for standalone HD model simulations

The tables in the following subsections list the parameters of the different JSBACH namelists. Each parameter is listed in alphabetical order and is briefly described. Besides, the Fortran type and the default values are given.

### C.2.1 Namelist `albedo_ctl`

The namelist for the albedo scheme is read in by the routine `config_albedo` of module `mo_land_surface`. It is used only if the albedo scheme (section 4.1) is switched on, i.e. `use_albedo=.TRUE.` in namelist `jsbach_ctl` (compare table C.12).

Table C.2: Namelist `albedo_ctl`

Parameter	Type	Description	Default
<code>albedo_age_weight</code>	real	0: ECHAM5 scheme for snow albedo; 1: snow age scheme; between 0 and 1: snow albedo linearly weighted between values obtained from ECHAM5 and snow age scheme	0.5
<code>use_albedocanopy</code>	logical	<code>.TRUE.</code> : read maps of canopy albedo ( <code>albedo_veg_nir</code> and <code>albedo_veg_vis</code> from <code>jsbach.nc</code> ); <code>.FALSE.</code> : use PFT specific albedo values from <code>lctlib.def</code>	<code>.FALSE.</code>
<code>use_albedosoil</code>	logical	<code>.TRUE.</code> : calculate soil albedo depending on soil carbon and litter. <b>Note:</b> this option should not be used with the standard <code>jsbach</code> initial file!	<code>.FALSE.</code>
<code>use_albedosoilconst</code>	logical	<code>.TRUE.</code> : base albedo of the soil (without soil carbon and leaf litter) is set to a global constant; <code>.FALSE.</code> : base albedo of the soil is read from <code>jsbach</code> initial file. Only used with <code>use_albedosoil=.TRUE.</code>	<code>.FALSE.</code>
<code>use_litter</code>	logical	<code>.TRUE.</code> : soil albedo depends on leaf litter. Only used with <code>use_albedosoil=.TRUE.</code>	<code>.TRUE.</code>
<code>use_soc</code>	character	<code>linear</code> : soil albedo linearly depends on soil carbon; <code>log</code> : logarithmic dependence of soil albedo on soil carbon. Only used with <code>use_albedosoil=.TRUE.</code>	<code>'linear'</code>

### C.2.2 Namelist `bethy_ctl`

The namelist `bethy_ctl` controls the BETHY module for photosynthesis. It is used only if `use_bethy=.TRUE.` in namelist `jsbach_ctl` (compare table C.12). The namelist is read in routine `config_bethy` of `mo_bethy`.

Table C.3: Namelist `bethy_ctl`

Parameter	Type	Description	Default
<code>n canopy</code>	integer	number of canopy layers	3

### C.2.3 Namelist `cbalance_ctl`

The `cbalance` module handling the carbon and nitrogen pools is controlled by namelist `cbalance_ctl`. The namelist is read in routine `init_cbalance_bethy` in `mo_cbal_bethy`.

Table C.4: Namelist `cbalance_ctl`

Parameter	Type	Description	Default
<code>cn_tolerance</code>	real	tolerance limit for C/N ratio inconsistencies	$10^{-14}$
<code>cpools_file_name</code>	character	name of the file containing initial data for the carbon pools. Only used if <code>read_c pools=.TRUE.</code>	<code>'Cpools.nc'</code>

*table continued on next page*

Table C.4: `cbalance_ctl` — continued

Parameter	Type	Description	Default
<code>diag_humus_fluxes</code>	logical	activate diagnostic output of carbon fluxes into and from yasso humus pools needed for equilibration	<code>.FALSE.</code>
<code>init_npools</code>	logical	initialize nitrogen pools from carbon pools with respect to the C/N ratio.	<code>.FALSE.</code>
<code>ndepo_file_name</code>	character	name of the file containing nitrogen deposition data. Only used if <code>with_nitrogen=.TRUE.</code> in <code>jsbach_ctl</code> and <code>read_cpools=.TRUE.</code>	<code>'Ndepo.nc'</code>
<code>npool_file_name</code>	character	name of the file containing initial data for the nitrogen pools. Only used if <code>with_nitrogen=.TRUE.</code> in <code>jsbach_ctl</code> and <code>read_npools=.TRUE.</code>	<code>'Npools.nc'</code>
<code>read_cpools</code>	logical	initialize carbon pools with data from an external file.	<code>.FALSE.</code>
<code>read_ndepo</code>	logical	read nitrogen deposition data from an external file. Only used if <code>with_nitrogen=.TRUE.</code> in <code>jsbach_ctl</code>	<code>.FALSE.</code>
<code>read_npools</code>	logical	initialize nitrogen pools with data from an external file. Only used if <code>with_nitrogen=.TRUE.</code> in <code>jsbach_ctl</code>	<code>.FALSE.</code>

#### C.2.4 Namelist `cbal_parameters_ctl`

Several parameters needed for carbon cycle calculations are defined in namelist `cbal_parameters_ctl`. The namelist is read in routine `config_cbal_parameters` of module `mo_cbal_parameters`.

Table C.5: Namelist `cbal_parameters_ctl`

Parameter	Type	Description	Default
<code>cn_green</code>	real	carbon-to-nitrogen ratio of the green pools	35.
<code>cn_litter_green</code>	real	carbon-to-nitrogen ratio of the litter flux from leaves and fine roots	55.
<code>cn_litter_wood</code>	real	carbon-to-nitrogen ratio of woody litter pools	330.
<code>cn_slow</code>	real	carbon-to-nitrogen ratio of the slow soil pools	10.
<code>cn_woods</code>	real	carbon-to-nitrogen ratio of the wood pools	150.
<code>frac_green_2_atmos</code>	real	fraction of carbon and nitrogen from the green pools released into the atmosphere with anthropogenic land-cover change; only used with <code>lcc_scheme=1</code> in namelist <code>jsbach_ctl</code>	0.8
<code>frac_harvest_2_atmos</code>	real	fraction of harvested carbon and nitrogen immediately released into the atmosphere; only used with <code>lcc_scheme=1</code> in namelist <code>jsbach_ctl</code>	0.2
<code>frac_harvest_2_slash</code>	real	fraction of carbon and nitrogen harvested from woody pfts that is left unused and transferred to litter; only used with <code>lcc_scheme=2</code> in namelist <code>jsbach_ctl</code>	0.3
<code>frac_mobile_2_atmos</code>	real	fraction of nitrogen from the plant mobile N pool released into the atmosphere with anthropogenic landcover change; only used with <code>lcc_scheme=1</code> in namelist <code>jsbach_ctl</code>	0.8
<code>frac_mo_harv_2_atmos</code>	real	fraction of nitrogen harvested from the mobile N pool that is released into the atmosphere; only used with <code>lcc_scheme=1</code> in namelist <code>jsbach_ctl</code>	0.2

*table continued on next page*

Table C.5: `cbal_parameters_ctl` — continued

Parameter	Type	Description	Default
<code>frac_reserve_2_atmos</code>	real	fraction of carbon from the reserve pool released into the atmosphere with anthropogenic landcover change; only used with <code>lcc_scheme=1</code> in namelist <code>jsbach_ctl</code>	0.8
<code>frac_wood_2_atmos</code>	real	fraction of carbon and nitrogen from the wood pools released into the atmosphere with anthropogenic landcover change; only used with <code>lcc_scheme=1</code> in namelist <code>jsbach_ctl</code>	0.8
<code>tau_construction</code>	real	decay time from 100% to 10% of the construction pool (century time scale) in [days]; only used with anthropogenic pools (i.e. <code>lcc_scheme=2</code> in namelist <code>jsbach_ctl</code> )	100.*365.
<code>tau_onsite</code>	real	decay time from 100% to 10% for the onSite pool (centennial time scale) in [days] of the ‘anthro pools LUC scheme’ (i.e. <code>lcc_scheme=2</code> in namelist <code>jsbach_ctl</code> )	1.*365.
<code>tau_paper</code>	real	decay time from 100% to 10% for the paper pool (decadal time scale) in [days] of the ‘anthro pools LUC scheme’ (i.e. <code>lcc_scheme=2</code> in namelist <code>jsbach_ctl</code> )	10.*365.

### C.2.5 Namelist `climbuf_ctl`

The climate buffer provides climate variables as multi-annual running means, minimums or maximums. It is controlled by namelist `climbuf_ctl`. The namelist is read in routine `config_climbuf` (`mo_climbuf`).

Table C.6: Namelist `climbuf_ctl`

Parameter	Type	Description	Default
<code>climbuf_file_name</code>	character	name of the climate buffer file. Only used if <code>read_climbuf=.TRUE.</code>	<code>'climbuf.nc'</code>
<code>init_running_means</code>	logical	initialize the calculation of long term climate variables. (Should be <code>.TRUE.</code> at the beginning of the second year of an initialized experiment.)	<code>.FALSE.</code>
<code>read_climbuf</code>	logical	read climate buffer data from an external file.	<code>.FALSE.</code>

### C.2.6 Namelist `disturbance_ctl`

Fire and windthrow calculations are controlled by namelist `disturbance_ctl`. The namelist is read in routine `config_disturbance` (`mo_disturbance`). It is used only, if the disturbance module is switched on by setting `use_disturbance=.TRUE.` in namelist `jsbach_ctl` (compare table C.12).

Table C.7: Namelist `disturbance_ctl`

Parameter	Type	Description	Default
<code>fire_frac_wood_2_atmos</code>	real	fraction of carbon from the wood pool emitted to the atmosphere by fire	0.2
<code>fire_name</code>	character	fire scheme: <code>'none'</code> , <code>'jsbach'</code> or <code>'thonicke'</code>	<code>'jsbach'</code>
<code>fuel_name</code>	character	carbon relocation scheme for fire: <code>'none'</code> , <code>'jsbach'</code> or <code>'thonicke'</code> . Not all combinations of <code>fire_name</code> and <code>fuel_name</code> are valid.	<code>''</code>
<code>lburn_pasture</code>	logical	allow fire algorithm to burn pastures	<code>.FALSE.</code>
<code>ldiag</code>	logical	switch on/off additional diagnostic output; only available with <code>fire_name='thonicke'</code>	<code>.FALSE.</code>

*table continued on next page*

Table C.7: `disturbance_ctl` — continued

Parameter	Type	Description	Default
<code>lemissions</code>	logical	calculate diagnostic trace gas emissions	<code>.FALSE.</code>
<code>max_lat_tropic</code>	real	latitude separating tropical from extratropical land cover types; only with <code>lemissions</code>	23.5
<code>min_dist_woody</code>	real	minimum disturbance of woody types in fraction per year	0.002 with <code>fire_name='thonicke'</code> , otherwise 0.
<code>min_pasture</code>	real	minimum pasture fraction to calculate pasture emissions of the grid box; only with <code>lemissions</code>	0.5
<code>min_tropical_forest</code>	real	minimum tropical forest fraction to calculate tropical forest emissions of the grid box; only with <code>lemissions</code>	0.6
<code>windbreak_name</code>	character	windthrow scheme: <code>'none'</code> or <code>'jsbach'</code>	<code>'jsbach'</code>

### C.2.7 Namelist `dynveg_ctl`

The dynamic vegetation is controlled by `dynveg_ctl`. The namelist is read in `config_dynveg` (`mo_dynveg`). It is used only, if the dynamic vegetation is switched on by setting `use_dynveg=.TRUE.` in namelist `jsbach_ctl` (compare table C.12).

Table C.8: Namelist `dynveg_ctl`

Parameter	Type	Description	Default
<code>accelerate_dynveg</code>	real	factor to accelerate vegetation dynamics. Default: no acceleration	1.
<code>dynveg_all</code>	logical	activate competition between woody types and grasses (not recommended)	<code>.FALSE.</code>
<code>dynveg_feedback</code>	logical	switch on/off the feedback of the dynamic vegetation on the JSBACH physics. (Cover fractions are kept constant, while fire and windthrow still influence the carbon cycle.)	<code>.TRUE.</code>
<code>fpc_file_name</code>	character	name of an external vegetation data file. Only used if <code>read_fpc=.TRUE.</code>	<code>'fpc.nc'</code>
<code>read_fpc</code>	logical	read initial cover fractions from an external file; the file name is defined with parameter <code>fpc_file_name</code>	<code>.FALSE.</code>

### C.2.8 Namelist `fire_jsbach_ctl`

The standard JSBACH fire algorithm is controlled by namelist `fire_jsbach_ctl`. The namelist is read in routine `config_fire_jsbach` (`mo_disturbance_jsbach`). It is used only, if the disturbance scheme is activated by setting `use_disturbance=.TRUE.` in namelist `jsbach_ctl` and `fire_name='jsbach'` in namelist `disturbance_ctl` (compare tables C.12 and C.9).

Table C.9: Namelist `fire_jsbach_ctl`

Parameter	Type	Description	Default
<code>fire_litter_threshold</code>	real	minimum amount of litter needed for fire [ $mol(C)/m^2(gridbox)$ ]	16.67
<code>fire_minimum_grass</code>	real	minimum fraction of <code>act_fpc</code> of grass PFTs to be burned each year	0.006

*table continued on next page*

Table C.9: `fire_jsbach_ctl` — continued

Parameter	Type	Description	Default
<code>fire_minimum_woody</code>	real	minimum fraction of <code>act_fpc</code> of woody PFTs to be burned each year	0.002
<code>fire_rel_hum_threshold</code>	real	maximum relative humidity for fire [%]	70.
<code>fire_tau_grass</code>	real	return period of fire for grass PFT [year] assuming 0% relative humidity [year]	2.
<code>fire_tau_woody</code>	real	return period of fire for woody PFT [year] assuming 0% relative humidity [year]	6.

### C.2.9 Namelist `fire_thonicke_ctl`

The thonicke fire algorithm, i.e. the SPITFIRE model (see sec. 8.3.2), is controlled by namelist `fire_thonicke_ctl`. The namelist is read in by routine `config_fire_thonicke` of module `mo_disturbance_thonicke`. It is used only, if the disturbance scheme is activated by setting `use_disturbance=.TRUE.` in namelist `jsbach_ctl` and `fire_name='thonicke'` in namelist `disturbance_ctl` (compare tables C.12 and C.10).

Table C.10: Namelist `fire_thonicke_ctl`

Parameter	Type	Description	Default
<code>a_nd</code>	real	Propensity of humans to ignite a fire. Used only if no map of regional values available	0.251
<code>const_frac_fire</code>	real	set burned fraction to a constant annual value (-1: no)	-1.
<code>const_ignitions</code>	real	set number of ignitions to a constant annual value (-1: no)	-1.
<code>frac_green_active</code>	real	Fraction of leaves of woody types acting as fuel	0.3
<code>frac_N_2_atm</code>	real	Fraction of burned Nitrogen released to the atmosphere (the other part goes into the <code>sminN</code> pool)	0.5
<code>human_para</code>	real	tuning parameter for human ignitions	0.4
<code>ign_para</code>	real	tuning parameter for total ignitions	0.22
<code>lduration_popd</code>	logical	calculate fire duration depending on population density	.TRUE.
<code>flaming2total</code>	real	factor to calculate total burning time from the duration of the flaming phase	5.0
<code>llight_ground</code>	logical	lightening density input data only comprises cloud to ground flashes	.FALSE.
<code>lmortalityEQ1</code>	logical	set post fire mortality of plants to 1	.FALSE.
<code>lwind_speed_limit</code>	logical	use upper limit for wind speed in calculation of fire spread	.TRUE.
<code>moisture_scaling</code>	real	factor to calculate fuel moisture from the fuel area to volume ratio <code>surfarea2vol</code> .	30000.
<code>read_fuel_frac</code>	logical	read initial fractions of fuel from file	.FALSE.
<code>surfarea2vol</code>	real(3)	surface area to volume ratio for 1hr 10hr and 100hr fuels	(/66.0, 3.58, 0.98/)
<code>wind_max</code>	real	maximum effective windspeed [ft/min] in fire spread calculation; only used if <code>lwind_speed_limit=.TRUE.</code>	150.
<code>wind_slope</code>	real	determines the reduction of high windspeeds; only used if <code>lwind_speed_limit= .TRUE.</code>	1.5

### C.2.10 Namelist hydrology\_ctl

The hydrological discharge (HD) module is controlled by namelist `hydrology_ctl`. The HD module is active, if `with_hd=.TRUE.` in jsbach namelist `jsbach_ctl`. The hydrology namelist is read in routine `config_hydrology` (`mo_hydrology`).

Table C.11: Namelist `hydrology_ctl`

Parameter	Type	Description	Default
<code>diag_water_budget</code>	logical	switches on/off additional water budget diagnostics	<code>.FALSE.</code>
<code>fblog1</code>	real	latitude of first grid box for outflow diagnostics (with <code>nhd_diag=99</code> )	0.
<code>fblog2</code>	real	latitude of second grid box for outflow diagnostics (with <code>nhd_diag=99</code> )	0.
<code>fllog1</code>	real	longitude of first grid box for outflow diagnostics (with <code>nhd_diag=99</code> )	0.
<code>fllog2</code>	real	longitude of second grid box for outflow diagnostics (with <code>nhd_diag=99</code> )	0.
<code>gethd</code>	special	time interval for getting data from the HD model	1,'days','off',0
<code>lbase</code>	logical	switches on/off baseflow calculation	<code>.TRUE.</code>
<code>ldebughd</code>	logical	switches on/off additional output for debugging	<code>.FALSE.</code>
<code>lhd_highres</code>	logical	switches on/off outflow diagnostic on HD model grid (0.5 deg.)	<code>.FALSE.</code>
<code>lhd_rout</code>	logical	switch for routing scheme; true: via direction arrays; false: via index arrays	<code>.FALSE.</code>
<code>locean</code>	logical	closure of water budget for ocean coupling	<code>.TRUE.</code>
<code>nhd_diag</code>	integer	region number for outflow diagnostic (in former versions <code>isolog</code> ): 0: none, 1: Bothnian Bay/Sea, 2: Torneaelven, 4: St.Lawrence, 5: Paraguay, 6: Oder, 7: Elbe, 8: Oranje, 9: Amudarya, 10: Lena, 99: two user defined grid boxes defined by the longitude and latitudes of <code>fblog1</code> , <code>fllog1</code> , <code>fblog2</code> and <code>fllog2</code>	0
<code>puthd</code>	special	time interval for transferring data to the HD model	1,'days','off',0

### C.2.11 Namelist jsbach\_ctl

The namelist `jsbach_ctl` includes the basic parameters for a JSBACH simulation. It is needed to switch on or off the different physical modules as e.g. the dynamic vegetation or the albedo scheme. Besides, it controls file names and other IO-options. The namelist is read in routine `jsbach_config` of module `mo_jsbach`.

Table C.12: Namelist `jsbach_ctl`

Parameter	Type	Description	Default
<code>coupling</code>	character	Type of coupling: <code>implicit</code>	<code>'implicit'</code>
<code>debug</code>	logical	additional output for debugging	<code>.FALSE.</code>
<code>debug_Cconservation</code>	logical	additional debugging output to solve problems with carbon conservation	<code>.FALSE.</code>
<code>debug_Nconservation</code>	logical	additional debugging output to solve problems with nitrogen conservation	<code>.FALSE.</code>
<code>file_type</code>	integer	output format: 1: grib, 2: netcdf, 4: netcdf2, 6: netcdf4	1
<code>file_ztype</code>	integer	output compression type: 0: none, 1: szip (for grib), 2: zip (for netcdf4)	0
<code>grid_file</code>	character	input file containing grid information	<code>'jsbach.nc'</code>

*table continued on next page*

Table C.12: `jsbach_ctl` — continued

Parameter	Type	Description	Default
<code>input_verbose</code>	character	debug message level for <code>mo_input</code> . See <code>mo_input</code> documentation for details	<code>''</code>
<code>interface_test</code>	logical	<code>.TRUE.:</code> generates bit-identical results of an ECHAM/JSBACH run with <code>write_interface_vars</code> and a JSBACH standalone run with <code>read_interface_vars</code> .	<code>.FALSE.</code>
<code>lcc_forcing_type</code>	character	chooses the scheme for anthropogenic landcover changes: <code>NONE:</code> no anthropogenic landcover change; <code>MAPS:</code> use the ‘maps model’, i.e. a sequence of maps of landcover fractions is read in; <code>TRANSITIONS:</code> use the ‘transitions model’, i.e. a sequence of maps of land use transitions is read in; <code>NET_TRANSITIONS:</code> as <code>TRANSITIONS</code> , however only net transitions are taken into account.	<code>'NONE'</code>
<code>lcc_scheme</code>	integer	scheme for handling anthropogenically induced land carbon and nitrogen fluxes: 1: ‘litter pool LUC scheme’ (see section 10.3.2), 2: ‘anthro pool LUC scheme’ (see section 10.3.3); these options are in effect only when <code>lcc_forcing_type</code> $\neq$ <code>NONE</code> .	1
<code>lctlib_file</code>	character	name of the land cover library file	<code>'lctlib.def'</code>
<code>lpost_echam</code>	logical	if <code>.TRUE.</code> , write jsbach output variables, even if they are part of the echam output	<code>.FALSE.</code>
<code>ls3m_type</code>	integer	switch defining whether additional LS3-MIP output is generated; 0: no LS3-MIP output, 1: write LS3-MIP output	0
<code>lss</code>	character	land surface sceme: <code>ECHAM</code>	<code>'ECHAM'</code>
<code>missing_value</code>	real	missing value for the output (ocean values)	<code>NF_FILL_REAL</code>
<code>nfiles</code>	integer	number of tiles defined on each grid box; mandatory to define in namelist, as default value is not valid.	-1
<code>out_state</code>	logical	write the jsbach output stream	<code>.TRUE.</code>
<code>pheno_scheme</code>	character	phenology scheme: <code>LOGROP:</code> JSBACH phenology scheme by C. H. Reick used e.g. in CMIP5; <code>KNORR:</code> phenology scheme by W. Knorr used in CCDAS	<code>'LOGROP'</code>
<code>read_cover_fract</code>	logical	read cover fractions from the JSBACH initial file rather than from restart file	<code>.FALSE.</code>
<code>read_interface_vars</code>	logical	read stepwise interface variables from a netcdf file called <code>interface_variables.nc</code> ; only if <code>standalone = .TRUE.</code>	<code>.FALSE.</code>
<code>soil_file</code>	character	file containing initial data of soil properties	<code>'jsbach.nc'</code>
<code>standalone</code>	logical	Type of model run; <code>.TRUE.:</code> standalone JSBACH run; <code>.FALSE.:</code> JSBACH driven by an atmosphere model	<code>.TRUE.</code>
<code>surf_file</code>	character	file containing initial data of the land surface	<code>'jsbach.nc'</code>
<code>test_Cconservation</code>	logical	switches on/off carbon conservation test	<code>.FALSE.</code>
<code>test_Nconservation</code>	logical	switches on/off nitrogen conservation test	<code>.FALSE.</code>
<code>test_stream</code>	logical	additional stream for model testing	<code>.FALSE.</code>
<code>use_albedo</code>	logical	switches between the ECHAM5 albedo scheme (off) and the dynamic JSBACH albedo scheme (on) (see section 4.1.1)	<code>.FALSE.</code>
<code>use_bethy</code>	logical	switches on/off the BETHY model (photosynthesis, respiration)	<code>.FALSE.</code>

*table continued on next page*

Table C.12: `jsbach_ctl` — continued

Parameter	Type	Description	Default
<code>use_disturbance</code>	logical	switches on/off the disturbance module (independent of the dynamic vegetation)	<code>.FALSE.</code>
<code>use_dynveg</code>	logical	switches on/off the dynamic vegetation module	<code>.FALSE.</code>
<code>use_phenology</code>	logical	switches on/off the phenology module to calculate the LAI	<code>.FALSE.</code>
<code>use_roughness_lai</code>	logical	calculate roughness length depending on LAI	<code>.FALSE.</code>
<code>use_roughness_oro</code>	logical	calculate roughness length including sub-grid scale topographie	<code>.TRUE.</code>
<code>veg_at_1200</code>	logical	<code>.TRUE.:</code> write veg stream at 12:00 each day; <code>.FALSE.:</code> write veg stream at the same time steps as the other streams	<code>.TRUE.</code>
<code>veg_file</code>	character	file containing initial data for the vegetation	<code>'jsbach.nc'</code>
<code>with_hd</code>	logical	<code>.TRUE.:</code> use hydrological discharge (HD) model	<code>.FALSE.</code>
<code>with_nitrogen</code>	logical	calculate the nitrogen cycle	<code>.FALSE.</code>
<code>with_yasso</code>	logical	<code>.TRUE.:</code> YASSO is used for litter and soil carbon decomposition. <code>.FALSE.:</code> CBALANCE is used for litter and soil carbon decomposition	<code>.FALSE.</code>
<code>write_interface_vars</code>	logical	write out all input variables going through the JSBACH interface	<code>.FALSE.</code>

### C.2.12 Namelist `soil_ctl`

The configurable parameters to control the soil physics are defined in namelist `soil_ctl`. The namelist is read in `config_soil` in module `mo_soil`.

Table C.13: Namelist `soil_ctl`

Parameter	Type	Description	Default
<code>crit_snow_depth</code>	real	critical snow depth for correction of surface temperature for melting [m]	$5.85036 \times 10^{-03}$
<code>hcond_scheme</code>	character	scheme for dry soil heat conductivity ( <code>none</code> or <code>johansen</code> )	<code>'JOHANSEN'</code>
<code>heat_cap_map</code>	logical	<code>TRUE:</code> read heat capacity from initial file; <code>FALSE:</code> calculate it from fao soil types	<code>.FALSE.</code>
<code>lbsoil</code>	logical	separate handling of bare soil moisture for bare soil evaporation in multi-layer soil hydrology scheme (only with <code>nsoil &gt; 1</code> )	<code>.FALSE.</code>
<code>ldiag</code>	logical	switch on/off extended water balance diagnostics	<code>.TRUE.</code>
<code>ldynorg</code>	logical	calculate organic layer parameters dynamically; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.TRUE.</code>
<code>ldynsnow</code>	logical	calculate snow parameters dynamically; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.FALSE.</code>
<code>lfreeze</code>	logical	consider freezing and thawing in thermal soil calculations; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.TRUE.</code>
<code>lheatcap</code>	logical	calculate soil heat capacity dynamically; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.TRUE.</code>
<code>lheatcond</code>	logical	calculate soil heat conductivity dynamically; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.TRUE.</code>
<code>lorganic</code>	logical	consider organic layers in soil thermal calculations; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.FALSE.</code>
<code>lsnow</code>	logical	consider snow in soil thermal calculations; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.TRUE.</code>

*table continued on next page*

Table C.13: `soil_ctl` — continued

Parameter	Type	Description	Default
<code>lsupercool</code>	logical	allow for supercooled soil water; only relevant if <code>with_permafrost=.TRUE.</code>	<code>.FALSE.</code>
<code>moist_crit_fract</code>	real	critical value of soil moisture above which transpiration is not affected by the soil moisture stress; expressed as fraction of the maximum soil moisture content	0.75
<code>moist_max_limit</code>	real	upper limit for maximum soil moisture content: If positive, <code>max_moisture</code> from initial file is cut off at this value.	-1.
<code>moist_wilt_fract</code>	real	soil moisture content at permanent wilting point, expressed as fraction of maximum soil moisture content	0.35
<code>nsnow</code>	integer	number of snow layers; if <code>with_permafrost=.TRUE.</code> at least 3 layers are needed	5
<code>nsoil</code>	integer	number of soil layers; 1: bucket scheme for soil hydrology, five layers for soil temperature; >1: multi-layer soil scheme; if <code>with_permafrost=.TRUE.</code> the multi-layer soil scheme needs to be used.	1
<code>skin_res_max</code>	real	maximum water content of the skin reservoir of bare soil [m]	$2. \times 10^{-4}$
<code>with_permafrost</code>	logical	activate the permafrost scheme	<code>.FALSE.</code>

### C.2.13 Namelist `windbreak_jsbach_ctl`

The standard JSBACH windthrow algorithm is controlled by namelist `windbreak_jsbach_ctl`. The namelist is read in routine `config_windbreak_jsbach` (`mo_disturbance_jsbach`). It is used only, if the disturbance scheme is activated by setting `use_disturbance = .TRUE.` in namelist `jsbach_ctl` and `windbreak_name = 'jsbach'` in namelist `disturbance_ctl` (compare tables C.12 and C.9).

Table C.14: Namelist `windbreak_jsbach_ctl`

Parameter	Type	Description	Default
<code>wind.damage_scale</code>	real	scaling factor for windthrow. The default value corresponds to runs with ECHAM in T63 resolution.	0.005
<code>wind.threshold</code>	real	factor by which the maximum wind speed must be larger than the climatological maximum wind speed to allow any windthrow	2.25

### C.2.14 Namelist `jsbalone_ctl`

The basic flow of a standalone JSBACH simulations is controlled by namelist `jsbalone_ctl`. The namelist corresponds to the ECHAM namelist `runctl` and contains several parameters that are set in `runctl` in ECHAM/JSBACH simulations. Namelist `jsbalone_ctl` is read in routine `jsbach_config` of module `mo_jsbach`.

Table C.15: Namelist `jsbalone_ctl`

Parameter	Type	Description	Default
<code>cecc</code>	real	eccentricity of the earth's orbit. Default: today's value	0.016715
<code>clonp</code>	real	longitude of perihelion [degree]. Default: today's value	282.7
<code>cobld</code>	real	obliquity in degrees. Default: today's value	23.441
<code>default_output</code>	logical	switch off default output for monthly mean output	<code>.TRUE.</code>

*table continued on next page*

Table C.15: jsbalone\_ctl — continued

Parameter	Type	Description	Default
delta_time	integer	model time step length [s]	0
dt_resume	type	restart date of the experiment: (yr, mo, dy, hr, min, sec)	0,0,0,0,0,0
dt_start	type	start date of the experiment: (yr, mo, dy, hr, min, sec)	0,0,0,0,0,0
dt_stop	type	final date of the experiment: (yr, mo, dy, hr, min, sec)	0,0,0,0,0,0
l_orbvsop87	logical	Type of orbit computation: <code>.TRUE.:</code> vsop87; <code>.FALSE.:</code> PCMDI (AMIP)	<code>.TRUE.</code>
lresume	logical	<code>.TRUE.:</code> start from restart files; <code>.FALSE.:</code> start from initialization	<code>.FALSE.</code>
ltimer	logical	switches on/off timer to check model performance	<code>.FALSE.</code>
lyr_perp	logical	orbital parameters of a perpetual year; only with <code>l_orbvsop87=.TRUE.</code>	<code>.FALSE.</code>
no_cycles	integer	number of restart cycles within the run	1
no_days	integer	number of days to run the model for	-1
no_steps	integer	number of time steps to run the model for	-1
out_expname	character	experiment name (used in output file names)	'xxxxxx'
putdata	type	period of writing output. Default: Output is written at the first time step of each day.	1, 'days', 'first', 0
putrerun	type	period of restart writing. Default: write a restart file at the last time step of each month.	1, 'months', 'last', 0
trigfiles	type	period of output file generation. Default: A new output file is generated at the first time step of each month	1, 'months', 'first', 0
yr_perp	integer	perpetual year for orbital parameters; only with <code>l_orbvsop87=.TRUE.</code>	-99999

### C.2.15 Namelist jsbalone\_parctl

Namelist `jsbalone_parctl` corresponds to the ECHAM namelist `parctl` and defines the parallelization. The namelist is read in the JSBACH main program `jsbach_driver.f90`.

Table C.16: Namelist jsbalone\_parctl

Parameter	Type	Description	Default
nproca	integer	number of MPI processes for JSBACH	1
nprocb	integer	value has to be 1	1
nprocio	integer	number of processors for I/O (not yet fully implemented in the current version)	0
npedim	integer	working dimension for blocks on each domain. Default: each domain is processed in one call	-1

### C.2.16 Namelist forcing\_ctl

The atmospheric forcing for a standalone JSBACH simulation is controlled by namelist `forcing_ctl`. The namelist defines file names, the forcing frequency and the variables used as forcing arrays. Also included are the Earth's orbital parameters. The namelist is read in `init_forcing` (module `mo_jsbalone_forcing`).

Table C.17: Namelist forcing\_ctl

Parameter	Type	Description	Default
coef_ril_t	real	weighting factor for richardson numbers needed to calculate drag coefficient at time step t	0.25
coef_ril_tm1	real	weighting factor for richardson numbers needed to calculate drag coefficient at time step t-1	0.5

*table continued on next page*

Table C.17: forcing\_ctl — continued

Parameter	Type	Description	Default
coef_ril_tp1	real	weighting factor for richardson numbers needed to calculate drag coefficient at time step t+1	0.25
forcing_co2_const_co2	real	global value of the CO <sub>2</sub> -concentration in unit <forcing_co2_unit>. Only needed if forcing_co2_frequ='CONST'.	3.65*10 <sup>-4</sup>
forcing_co2_file	character	File name base of the netCDF file with variable CO <sub>2</sub> giving the CO <sub>2</sub> -concentration in unit <forcing_co2_unit>. The complete file name is <forcing_co2_file>_startyear.nc. Not used if forcing_co2_frequ='CONST'	'climate'
forcing_co2_frequ	character	frequency of CO <sub>2</sub> forcing data ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). A constant value (keyword 'CONST') needs to be defined by namelist parameter forcing_co2_const_co2.	'DAILY'
forcing_co2_interpol	logical	interpolation between CO <sub>2</sub> forcing steps required (only with forcing_co2_frequ=SUBDAILY)	.FALSE.
forcing_co2_steps	character	number of CO <sub>2</sub> forcing steps per day (only with forcing_co2_frequ=SUBDAILY)	0
forcing_co2_unit	character	unit of CO <sub>2</sub> concentration in forcing_co2_file: 'mol_per_mol': input data in ppmV CO <sub>2</sub> , 'mol_per_mol': input data in mol(CO <sub>2</sub> )/mol(dry air), 'kg_per_kg': input data in kg(CO <sub>2</sub> )/kg(dry air)	
forcing_lw_const_cloud	real	constant value globally used for cloud cover [%]. Only needed when forcing_lw_frequ='CONST' and forcing_lw_type='CLOUD'	HUGE(0.0)
forcing_lw_const_longwave	real	constant value globally used for downward longwave radiation [W/m <sup>2</sup> ]. Only needed when forcing_lw_frequ='CONST' and forcing_lw_type='MEAN_RAD'	HUGE(0.0)
forcing_lw_file	character	File name base of the netCDF file with longwave radiation data. The complete file name is <forcing_lw_file>_startyear.nc. With forcing_lw_type='CLOUD' it must contain the variable cloud: cloud cover [%]. With forcing_lw_type='MEAN_RAD' it needs the variable longwave: downward longwave radiation [W/m <sup>2</sup> ]. The file is not needed if forcing_lw_frequ='CONST'.	'climate'
forcing_lw_frequ	character	frequency of longwave forcing data ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). A constant value (keyword 'CONST') needs to be defined by namelist parameter forcing_lw_const_longwave or forcing_lw_const_cloud.	'DAILY'
forcing_lw_interpol	logical	interpolation between longwave forcing steps required (only with forcing_lw_frequ=SUBDAILY)	.FALSE.
forcing_lw_steps	integer	number of longwave forcing steps per day (only with forcing_lw_frequ=SUBDAILY)	0
forcing_lw_type	character	type of forcing data for longwave forcing. 'CLOUD': fractional cloud cover used as input data [%]; 'MEAN_RAD': mean daily or monthly downward longwave radiation [W/m <sup>2</sup> ]	'MEAN_RAD'
forcing_precip_const_precip	real	constant value globally used for precipitation. The unit is defined by keyword forcing_precip_in_mm_per_day. The value is only needed if forcing_precip_frequ='CONST'.	HUGE(0.0)

table continued on next page

Table C.17: forcing\_ctl — continued

Parameter	Type	Description	Default
forcing_precip_file	character	File name base of the netCDF file containing precipitation forcing data with the variables <code>precip</code> : rain plus snow and <code>fwet</code> : number of wet days per month (needed with monthly input, only). The unit of <code>precip</code> is defined by keyword <code>forcing_precip_in_mm_per_day</code> . The complete file name is <code>&lt;forcing_precip_file&gt;_startyear.nc</code> . The file is not needed if <code>forcing_precip_frequ='CONST'</code> .	'climate'
forcing_precip_frequ	character	frequency of precipitation forcing data ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). A constant value (keyword 'CONST') needs to be defined by namelist parameter <code>forcing_precip_const_precip</code>	'DAILY'
forcing_precip_in_mm_per_day	logical	unit of the input precipitation data. <code>.TRUE.:</code> mm/day; <code>.FALSE.:</code> kg m <sup>-2</sup> s <sup>-1</sup>	.TRUE.
forcing_precip_in_terpol	logical	interpolation between precipitation forcing steps required (only with <code>forcing_precip_frequ=SUBDAILY</code> )	.FALSE.
forcing_precip_steps	integer	number of precipitation forcing steps per day (only with <code>forcing_precip_frequ=SUBDAILY</code> )	0
forcing_qair_const_rh	real	constant value globally used for relative humidity of the atmosphere [%]. The value is only needed if <code>forcing_qair_frequ='CONST'</code> .	100.
forcing_qair_file	character	File name base of the netcdf file with atmospheric humidity forcing data with variables <code>rel_humidity</code> or <code>qair</code> , depending on <code>forcing_qair_type</code> . The complete file name is <code>&lt;forcing_qair_file&gt;_startyear.nc</code> . The file is not needed with <code>forcing_qair_frequ='CONST'</code> .	'climate'
forcing_qair_frequ	character	frequency of atmospheric humidity forcing data ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). A constant value (keyword 'CONST') needs to be defined by namelist parameter <code>forcing_qair_const_rh</code>	'DAILY'
forcing_qair_in_terpol	logical	interpolation between qair forcing steps required (only with <code>forcing_qair_frequ=SUBDAILY</code> )	.FALSE.
forcing_qair_steps	integer	number of qair forcing steps per day (only with <code>forcing_qair_frequ=SUBDAILY</code> )	0
forcing_qair_type	character	type of forcing data used for atmospheric humidity: 'RH': relative humidity [%]; 'QAIR': mean daily or monthly specific humidity [kg/kg]; NONE: specific humidity is calculated from air temperature	'NONE'
forcing_sw_const_cloud	real	constant value globally used for cloud cover [%]. Only needed when <code>forcing_sw_frequ='CONST'</code> and <code>forcing_sw_type='CLOUD'</code> .	HUGE(0.0)
forcing_sw_const_shortwave	real	constant value globally used for downward shortwave radiation [W/m <sup>2</sup> ]. Only needed when <code>forcing_sw_frequ='CONST'</code> and <code>forcing_sw_type='MEAN_RAD'</code> .	HUGE(0.0)
forcing_sw_file	character	File name base of the netCDF file with shortwave radiation data. The complete file name is <code>&lt;forcing_sw_file&gt;_startyear.nc</code> . With <code>forcing_sw_type='CLOUD'</code> it must contain the variable <code>cloud</code> : cloud cover [%]. With <code>forcing_sw_type='MEAN_RAD'</code> it needs the variable <code>shortwave</code> : downward shortwave radiation [W/m <sup>2</sup> ]. The file is not needed with <code>forcing_sw_frequ='CONST'</code> .	'climate'

table continued on next page

Table C.17: forcing\_ctl — continued

Parameter	Type	Description	Default
forcing_sw_frequ	character	frequency of shortwave forcing data ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). A constant value (keyword 'CONST') needs to be defined by namelist parameter forcing_sw_const_shortwave or forcing_sw_const_cloud.	'DAILY'
forcing_sw_interpol	logical	interpolation between shortwave radiation forcing steps required (only with forcing_sw_frequ=SUBDAILY)	.FALSE.
forcing_sw_scheme	character	scheme to generate the components (PAR, NIR) of shortwave forcing	'NONE'
forcing_sw_steps	integer	number of shortwave radiation forcing steps per day (only with forcing_sw_frequ=SUBDAILY)	0
forcing_sw_type	character	type of input data used for forcing by shortwave radiation. 'CLOUD': relative cloud cover [%]; 'MEAN_RAD': mean daily or monthly downward shortwave radiation [W/m <sup>2</sup> ]	'MEAN_RAD'
forcing_table_sw_pot_const	real	constant value globally used for potential shortwave radiation [W/m <sup>2</sup> ]. Only needed when FORCING_SW_TYPE='MEAN_RAD' and FORCING_TABLE_SW_POT_FREQU='CONST'.	HUGE(0.0)
forcing_table_sw_pot_file	character	File name base of the netCDF file with variable mpot giving a table of potential (i.e. clear-sky) shortwave radiation data [W/m <sup>2</sup> ]. The complete file name is <forcing_table_sw_pot_file>_startyear.nc. The file is needed only if FORCING_SW_TYPE='MEAN_RAD'.	'climate'
forcing_table_sw_pot_frequ	character	frequency of potential shortwave radiation ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). A constant value (keyword 'CONST') needs to be defined by namelist parameter forcing_table_sw_pot_const. Only needed for forcing_sw_type='MEAN_RAD'.	'DAILY'
forcing_table_sw_pot_interpol	logical	interpolation between forcing steps of potential shortwave radiation required (only with forcing_table_sw_pot_frequ=SUBDAILY)	.FALSE.
forcing_table_sw_pot_steps	integer	number of forcing steps of potential shortwave radiation per day (only with forcing_table_sw_pot_frequ=SUBDAILY)	0
forcing_temp_const_tmax	real	constant value globally used for maximum daily temperature [°C]. Only needed if forcing_temp_frequ='CONST'.	HUGE(0.0)
forcing_temp_const_tmin	real	constant value globally used for minimum daily temperature [°C]. Only needed if forcing_temp_frequ='CONST'.	HUGE(0.0)
forcing_temp_file	character	File name base of the netCDF file containing temperature forcing data with the variables tmin: minimum daily temperature [°C] and tmax: maximum daily temperature [°C]. The complete file name is <forcing_temp_file>_startyear.nc. The file is not needed if forcing_temp_frequ='CONST'.	'climate'
forcing_temp_frequ	character	frequency of temperature forcing data ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). Constant values (keyword 'CONST') need to be defined by namelist parameter forcing_temp_const_tmin and forcing_temp_const_tmax	'DAILY'

table continued on next page

Table C.17: forcing\_ctl — continued

Parameter	Type	Description	Default
forcing_temp_in terpol	logical	interpolation between temperature forcing steps required (only with forcing_temp_frequ=SUBDAILY)	.FALSE.
forcing_temp_ steps	integer	number of temperature forcing steps per day (only with forcing_temp_frequ=SUBDAILY)	0
forcing_wind_ const_wspeerd	real	constant value globally used for windspeed [m/s]. Only needed when forcing_wind_frequ='CONST'.	HUGE(0.0)
forcing_wind_file	character	File name base of the netCDF file with variable wspeerd giving the wind speed [m/s]. The complete file name is <forcing_wind_file>.startyear.nc. The file is not needed if forcing_wind_frequ='CONST'.	'climate'
forcing_wind_ frequ	character	frequency of wind speed forcing data ('SUBDAILY', 'DAILY', 'MONTHLY' or 'CONST'). A constant value (keyword 'CONST') needs to be defined by namelist parameter forcing_wind_const_wspeerd	'DAILY'
forcing_wind_in terpol	logical	interpolation between wind forcing steps required (only with forcing_wind_frequ=SUBDAILY)	.FALSE.
forcing_wind_step	integer	number of wind forcing steps per day (only with forcing_wind_frequ=SUBDAILY)	0
height_humidity	real	reference height of specific humidity data [m]; -1. for echam lowest layer values)	-1.
height_wind	real	reference height of wind speed forcing data [m]; -1. for echam lowest layer values)	-1.

### C.2.17 Namelist jsbgrid\_ctl

In standalone configuration JSBACH can be run on a subset of the grid the atmospheric data is defined on. This grid can be defined using the namelist `jsbgrid_ctl`. The namelist is read in `init_grid` (`mo_jsbach_grid`).

Table C.18: Namelist jsbgrid\_ctl

Parameter	Type	Description	Default
lon_east	real	Eastern boundary of region [degree]	360.
lon_west	real	Western boundary of region [degree]	0.
lat_north	real	Northern boundary of region [degree]	90.
lat_south	real	Southern boundary of region [degree]	-90.
set_lon_zero	logical	Parameter for JSBACH site level version: ignor longitude of initial data and assume local time forcing.	.FALSE.
time_offset	real	adjust local time according to timezone. Only needed with set_lon_zero=.TRUE.	0.

### C.2.18 Namelist cbalance\_ctl

The standalone version of the JSBACH carbon cycle model CBALANCE is controlled by namelist `cbalance_ctl`. The namelist is read in the CBALANCE main routine `cbalance_driver.f90`. It corresponds to the JSBACH namelists `jsbach_ctl` and `jsbalance_ctl`.

Table C.19: Namelist cbalance\_ctl

Parameter	Type	Description	Default
climate_yearend	integer	last year of forcing climate data	-9999
climate_yearstart	integer	first year of forcing climate data	-9999

*table continued on next page*

Table C.19: cbalance\_ctl — continued

Parameter	Type	Description	Default
climatology_diff_file	character	climatological differences of forcing variables that depend on CO <sub>2</sub> . Only needed with <code>input_scaling=.TRUE.</code>	'climdiff.nc'
co2_file	character	CO <sub>2</sub> development file. Only needed with <code>input_scaling=.TRUE.</code>	'co2.nc'
cpool_file_name	character	name of the netcdf file containing initial data of the carbon pools	'Cpools.nc'
debug	logical	additional output for debugging	.FALSE.
debug_cconservation	logical	additional debugging output to solve problems with carbon conservation	.FALSE.
debug_nconservation	logical	additional debugging output to solve problems with nitrogen conservation	.FALSE.
driver_data_path	character	path to the forcing data	''
experiment	character	name of the experiment providing forcing data	''
init_npools	logical	initialize nitrogen pools consistent to carbon pools	.FALSE.
input_scaling	logical	switch on/off CO <sub>2</sub> -scaling	.FALSE.
input_scaling_ndep	logical	switch on/off scaling for nitrogen deposition	.FALSE.
input_verbose	character	debug message level for <code>mo_input</code> . See <code>mo_input</code> documentation for details	''
lcc_forcing_type	character	chooses the scheme for anthropogenic landcover changes: <code>none</code> : no anthropogenic landcover change; <code>maps</code> : use the 'maps model', i.e. a sequence of maps of landcover fractions is read in; <code>transitions</code> : use the 'transitions model', i.e. a sequence of maps of land use transitions is read in	'NONE'
lcc_scheme	integer	scheme for handling anthropogenically induced land carbon fluxes: 1: "litter pool LUC scheme" (see section 10.3.2), 2: "anthro pool LUC scheme" (see section 10.3.3); these options are in effect only when <code>lcc_forcing_type≠NONE</code> .	1
nday_from_forcing	logical	use number of days per month from forcing data (rather than from simulated calendar year)	.TRUE.
ndepo_file_name	character	file name with nitrogen deposition. Only needed with <code>read_ndepo=.TRUE.</code>	'Ndepo.nc'
ndeposition_diff_file	character	climatological differences of forcing variables that depend on nitrogen deposition. Only needed with <code>input_scaling=.TRUE.</code>	'ndepdiff.nc'
npool_file_name	character	file name for initial nitrogen pools. Only needed with <code>read_npools=.TRUE.</code>	'Npools.nc'
out_interval	character	output data interval: 'YEAR', 'MONTH' or 'DAY'	'YEAR'
read_cpools	logical	initialize carbon pools from external file	.FALSE.
read_ndepo	logical	read nitrogen deposition from an external file	.FALSE.
read_npools	logical	initialize nitrogen pools from external file	.FALSE.

*table continued on next page*

Table C.19: cbalance\_ctl — continued

Parameter	Type	Description	Default
ref_year_past	integer	first reference years for input scaling. Only needed with <code>input_scaling=.TRUE.</code> or <code>input_scaling_ndep=.TRUE.</code>	-9999
ref_year_recent	integer	second reference years for input scaling. Only needed with <code>input_scaling=.TRUE.</code> or <code>input_scaling_ndep=.TRUE.</code>	-9999
run_year_first	integer	first year of the CBALANCE experiment	-9999
run_year_last	integer	last year of the CBALANCE experiment	-9999
specificname	character	forcing data identifier: filenames are assumed to be <i>experiment_yyyyymm.specificname.nc</i>	''
use_disturbance	logical	switch on/off disturbance calculations	.FALSE.
use_dynveg	logical	switch on/off vegetation dynamics	.FALSE.
with_nitrogen	logical	switch on/off nitrogen cycle	.FALSE.
with_yasso	logical	.TRUE.: YASSO soil carbon model is used	.FALSE.

### C.2.19 Namelist hdalone\_ctl

The standalone version of the hydrology (HD) model is controlled by namelist `hdalone_ctl`. The namelist is read in routine `config_hd` of `hd_driver.f90`.

Table C.20: Namelist hdalone\_ctl

Parameter	Type	Description	Default
delta_time	real	model time step length in seconds	86400.
do_remapping	logical	type of interpolation from input (atmospheric) grid to HD grid; .FALSE.: no interpolation, i.e. input data grid equals HD grid .TRUE.: remapping in routine <code>hd_remap</code>	.TRUE.
drainage_file	character	netcdf file with input drainage data	'drainage.nc'
forcing_freq	integer	data frequency of the forcing data (STEPWISE/DAILY)	STEPWISE
iout	integer	averaging period of some HD model output: 1: 30 days, 2: 10 days, 3: 7 days, 5: monthly	5
iswrit	integer	restart time step (0: no restart file)	0
nstep	integer	number of time steps within the run	365
out_datapath	character	path to where the output data shall be written	'./'
out_expname	character	experiment name	'hd'
runoff_file	character	netcdf file with input runoff data	'runoff.nc'
ufakru	real	unit factor for runoff and drainage input data	1.
year1	integer	initial year of the run	1900

## C.3 The land cover type library File lctlib.def

JSBACH is based on Plant Functional Types (PFTs). The diversity of species is reduced to some key plant types with similar physical and biogeochemical properties (compare 1.3.1). The number and kind of PFTs considered within a model simulation is defined in the land cover type library file `lctlib.def`. It lists the PFT dependent parameters such as roughness length, maximum leaf area index, canopy albedo, and others.

The name of the land cover type library file can be modified by using the keyword `lctlib_file` of the JSBACH namelist `jsbach_ctl`.

The standard land cover type library file specifies 21 different PFTs, however not all PFTs might actually be used together in an experiment. For CMIP5 and CMIP6 simulations only 13 out of the 21 PFTs have been used. Below, as an example, the CMIP6 landcover library file. Data columns for PFTs that had not been used within the experiments are skipped for clearness. The hash sign (`#`) at the beginning of a line indicates comment lines that are ignored by the model.

```

#-----
# LANDCOVER LIBRARY
#
# $HeadURL$
# $Revision$
#-----
#
# Contains additional specifications for each landcover type
#
# The data refer to the following landcover classification:
#
# 1: Glacier
# 2: Tropical evergreen trees
# 3: Tropical deciduous trees
# 4: Extra-tropical evergreen trees
# 5: Extra-tropical deciduous trees
# 6: Temperate broadleaf evergreen trees
# 7: Temperate broadleaf deciduous trees
# 8: Coniferous evergreen trees
# 9: Coniferous deciduous trees
# 10: Raingreen shrubs
# 11: Deciduous shrubs
# 12: C3 grass
# 13: C4 grass
# 14: Pasture
# 15: C3 pasture
# 16: C4 pasture
# 17: Tundra
# 18: Swamp
# 19: Crops
# 20: C3 crop
# 21: C4 crop
#-----

NLCT 21    # number of landcover types

#-----
#
#LctNumber: Landcover type index numbers (not used in the model)
LctNumber      1      2      3      4      5     10     11     12     13     15     16     20     21

#LandcoverClass (bare soil: 0; glacier: 1; lake: 2; natural forest: 3; natural grassland: 4
#                other natural vegetation: 5; crops: 6; pastures: 7)
LandcoverClass 1      3      3      3      3      5      5      4      4      7      7      6      6

#PhenologyType: Phenology type (none: 0; evergreen: 1; summergreen: 2; raingreen: 3; grass: 4; crop: 5)
PhenologyType  0      3      3      1      2      3      2      4      4      4      4      5      5

#NitrogenScalingFlag: Is nitrogen scaling needed? (no: 0; yes: 1)
NitrogenScalingFlag  0      1      1      1      1      1      1      0      0      0      0      0      0

#C4flag: Photosynthetic pathway (C3: 0; C4: 1)
C4flag          0      0      0      0      0      0      0      0      1      0      1      0      1

#CarboxRate: Maximum carboxylation rate at 25 Celsius [1.E-6 * Mol(CO2)/m^2/s] (s. Kattge et al. 2009)
CarboxRate      0.0  39.0  31.0  44.0  66.0  61.7  54.0  78.2  8.0  78.2  8.0  100.7  39.0

#ETransport: Maximum electron transport rate at 25 Celsius [1.E-6 * Mol/m^2/s] (Jmax=1.9*Vmax for C3 plants)
ETransport      0.0  74.1  58.9  83.6  125.4  117.2  102.6  148.6  140.0  148.6  140.0  191.3  700.0

#VegHeight: Vegetation height [m]; for non-vegetation value is arbitrary
VegHeight       0.0  30.0  30.0  15.0  15.0  1.0  1.0  0.3  0.3  0.3  0.3  0.3  0.3

#VegRoughness: Vegetation roughness length [m]
#               for non-vegetation value is arbitrary but should be larger than zero to avoid division by zero
VegRoughness    0.005  2.0  1.0  1.0  1.0  0.5  0.5  0.05  0.05  0.05  0.05  0.05  0.05

#MinVegRoughness: Vegetation roughness length at LAI=1 [m]
#                 for non-vegetation value is arbitrary but should be larger than zero to avoid division by zero
MinVegRoughness 0.005  2.0  1.0  1.0  1.0  0.3  0.3  0.005  0.005  0.005  0.005  0.005  0.005

#MaxVegRoughness: Vegetation roughness length at LAI=LAI_max [m]
#                 for non-vegetation value is arbitrary but should be larger than zero to avoid division by zero
MaxVegRoughness 0.010  5.0  3.0  3.0  3.0  1.0  1.0  0.1  0.1  0.1  0.1  0.1  0.10

```

```

#Frac_npp_2_woodPool: Maximum fraction of NPP used for buildup of woody parts
Frac_npp_2_woodPool      0.0  0.3  0.3  0.3  0.3  0.3  0.3  0.0  0.0  0.0  0.0  0.0  0.0

#Frac_npp_2_reservePool: Maximum fraction of NPP put into the storage pool
Frac_npp_2_reservePool  0.0  0.05  0.1  0.05  0.1  0.1  0.1  0.15  0.15  0.15  0.15  0.2  0.2

#Frac_npp_2_exudates: Maximum fraction of NPP put into the root exudates
Frac_npp_2_exudates     0.0  0.05  0.05  0.05  0.05  0.05  0.05  0.05  0.05  0.05  0.05  0.05  0.05

#Frac_green_2_herbivory: Maximum fraction of green cpool put into grazing
Frac_green_2_herbivory  .0.000822.000822.000137.000822 .00164.000822 .00164 .00164 .00164.000822.000822

#Frac_C_litter_green2atmos: Fraction of Carbon from heterotrophic respiration that is emitted to the atmosphere.
# (The rest enters slow pool.)
Frac_C_litter_green2atmos 1.0  0.85  0.85  0.85  0.85  0.85  0.85  0.5  0.5  0.5  0.5  0.5  0.5

#Tau_Cpool_litter_leaf: Life time of leaf litter [days]
Tau_Cpool_litter_leaf     1.0  870.0  590.0  900.0  750.0  660.0  660.0  660.0  820.0  660.0  820.0  660.0  660.0

#Tau_Cpool_litter_wood: Life time of woody litter [days]
Tau_Cpool_litter_wood     10950. 10950. 10950. 10950. 10950. 10950. 10950. 10950. 10950. 10950. 10950. 10950. 10950.

#Tau_Cpool_woods: Time scale of the Cpool_woods and vegetation dynamics [years]
Tau_Cpool_woods          1.0  30.0  30.0  60.0  60.0  12.0  24.0  1.0  1.0  1.0  1.0  1.0  1.0

#LAI_shed_constant: Time in which leaves are constantly shedded [days-1]
LAI_shed_constant        0.000342.000456 .0 .0 .00183 .0 .00548 .00548 .00548 .00548 .00274 .00274

#Max_C_content_woods: Maximum carbon content in woody parts [mol(C)/m^2]
# (values from IPCC-TAR, S. 192)
Max_C_content_woods      0.02997.252997.25 2397.8 1798.3 582.8 416.2 0.0 0.0 0.0 0.0 0.0 0.0

#reserveC2leafC: Factor by which the maximum carbon content of the reserve pool is larger than MAXLAI /SLA
reserveC2leafC           2.0  2.0  2.0  2.0  2.0  2.0  2.0  4.0  4.0  4.0  4.0  4.0  4.0

#frac_harv_C_2_paper: Fraction of wood going into the anthropogenically controlled paper woody pool on harvest
frac_harv_C_2_paper      0.4  0.04  0.04  0.24  0.24  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

#frac_harv_C_2_construction: Fraction of wood going into the anthropogenically controlled construction woody pool
frac_harv_C_2_construction 0.1  0.06  0.06  0.36  0.36  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

#frac_lcc_C_2_paper: Fraction of carbon that going the anthropogenically controlled paper pool on converting
# natural PFTs to agricultural PFTs
frac_lcc_C_2_paper       0.4  0.4030  0.4030  0.2985  0.2985  0.2  0.2  0.0  0.0  0.0  0.0  0.0  0.0

#frac_lcc_C_2_construction: Fraction of carbon going into the anthropogenically controlled construction pool on
# converting natural PFTs to agricultural PFTs
frac_lcc_C_2_construction 0.0  0.0  0.0  0.1045  0.1045  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

#MaxLAI: Maximum LAI used in the LogoP scheme [m2/m2]
MaxLAI                   0.0  7.0  7.0  5.0  5.0  2.0  2.0  3.0  3.0  3.0  3.0  4.0  4.0

#StemArea: Area of stem and branches [m2/m2] (this is an optical parameter for the snow masking by forests)
StemArea                 0.0  1.5  1.5  1.5  1.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

#specificLeafArea_C: Carbon content per leaf area in [m2(leaf)/mol(Carbon)]
specificLeafArea_C       0.0  0.264  0.376  0.110  0.304  0.184  0.307  0.451  0.451  0.451  0.451  0.451  0.451

#ClumpinessFactor: Factor for vegetation clumpiness: veg_ratio=veg_ratio_max*(1-exp(-LAI_max/ClumpinessFactor))
ClumpinessFactor         1.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  2.0  3.0  3.0

#AlbedoCanopyVIS: Albedo of the canopy in the visible range
AlbedoCanopyVIS          0  0.03  0.04  0.04  0.05  0.05  0.05  0.08  0.08  0.08  0.08  0.08  0.08

#AlbedoCanopyNIR: Albedo of the canopy in the near infrared range
AlbedoCanopyNIR          0  0.22  0.23  0.23  0.26  0.25  0.28  0.33  0.33  0.33  0.33  0.33  0.33

#AlbedoLitterVIS: Albedo of litter in the visible range
AlbedoLitterVIS          0.0  0.09  0.10  0.10  0.11  0.11  0.11  0.34  0.34  0.34  0.34  0.34  0.34

#AlbedoLitterNIR: Albedo of litter in the near infrared range
AlbedoLitterNIR          0.0  0.16  0.17  0.16  0.19  0.19  0.22  0.34  0.34  0.34  0.34  0.34  0.34

#AlbedoSnowVisMin: Minimum snow albedo in the visible range
AlbedoSnowVisMin         0.78  0.52  0.52  0.52  0.52  0.52  0.52  0.52  0.52  0.52  0.52  0.52  0.52

#AlbedoSnowVisMax: Maximum snow albedo in the visible range
AlbedoSnowVisMax         0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9  0.9

#AlbedoSnowNirMin: Minimum snow albedo in the NIR range
AlbedoSnowNirMin         0.4  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3

#AlbedoSnowNirMax: Maximum snow albedo in the NIR range
AlbedoSnowNirMax         0.75  0.65  0.65  0.65  0.65  0.65  0.65  0.65  0.65  0.65  0.65  0.65  0.65

#Albedo of Snow: only used with the old echam albedo scheme (USE_ALBEDO=.FALSE.)
AlbedoSnowMin            0.75  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4  0.4
AlbedoSnowMax            0.85  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8

```



```

#WoodLitterSize:
WoodLitterSize          4    4    4    4    4    4    4    4    4    4    4    4    4    4
#
# Parameters for Spitfire fire algorithm (Thonicke et al. 2010)
#
#moist_extinction: moisture of extinction used to determine rate of spread in spitfire units: [-]
moist_extinction        0    0.2  0.3  0.3  0.3  0.3  0.3  0.3  0.2  0.3  0.2  0.3  0.2  0.2
#fuel_dens: bulk fuel density [kg/m3]
fuel_dens                0    25   25   20   22   5    5    4    2    4    2    2    2
#f parameter to compute the flame length
flame_length_f          0.0000 0.1487 0.0610 0.1000 0.3710 0.0940 0.0940 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
# crown length parameters
crown_length            0.00   0.33  0.10  0.33  0.33  0.80  0.80  0.00  0.00  0.00  0.00  0.00  0.00
# bark parameter 1
bark_par1               0.0000 0.0301 0.1085 0.0367 0.0347 0.1085 0.0347 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
# bark parameter 2
bark_par2               0.0000 0.0281 0.2120 0.0592 0.1086 0.2120 0.1086 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
# RCK
rck                     0.00   1.00  1.00  1.00  1.00  1.00  1.00  0.00  0.00  0.00  0.00  0.00  0.00
# probability for mortality
mort_prob               0.00   3.00  3.00  3.75  3.00  3.00  3.00  0.00  0.00  0.00  0.00  0.00  0.00
#
# Parameters for fire emissions
#
#tropical_flag: is the pft explicitly tropical?
tropical_flag           0    1    1    0    0    0    0    0    0    0    0    0    0
#deciduous_flag: is the pft deciduous (separate tropical evergreen from tropical deciduous which both use raingreen
# phenology
deciduous_flag          0    0    1    0    1    0    1    1    1    1    1    1    1

```

Note on the photosynthesis parameter  $E_{Transport}$ :

For C3-plants it is the value  $V_{max}$  [ $\mu\text{mol}(\text{CO}_2)/\text{m}^2\text{s}$ ] from eq. (5.27). For C4-plants, however, it gives the PEPcase  $\text{CO}_2$  specificity  $k$  of eq. (5.55) in  $\text{milli mol}(\text{CO}_2)/\text{m}^2\text{s}$ .

## C.4 Initial data file jsbach.nc

The JSBACH initial file contains initial conditions for several quantities, such as initial soil wetness or initial surface temperature. Besides, it includes boundary conditions as the land sea mask or the water holding capacity of the soil. The file is read at the very beginning of a JSBACH simulation, some of the boundary conditions are re-read after each restart. Not all variables in `jsbach.nc` are needed to run JSBACH in the standard configuration, some are only used if certain JSBACH modules are switched on or off.

The name of the initial file is configurable in namelist `jsbach_ctl`. With keywords `grid_file`, `soil_file`, `surf_file` and `veg_file` it is possible to specify four separate initial files for different JSBACH modules (compare sec. C.2.11). In the standard setup all initial and boundary data is available from one file named `jsbach.nc`.

### C.4.1 Typical JSBACH setups

The `jsbach` initial file very much depends on the experiment you plan to perform. A standard initial file does not exist. The number of different PFTs that can co-exist on each grid box (i.e. the number of tiles) has to be defined trading physical accuracy by computing time and data storage capacities. Detailed information on the tiling concept is given in section 1.3.2. A simulation with dynamic vegetation needs at least eight tiles for the natural PFTs considered in `dynveg` (compare chapter 9). For runs with land-use transitions as specified for CMIP5 three additional tiles for C3-pastures, C4-pastures and crops need to be defined. Table C.21 lists JSBACH configurations that had been used in recent studies.

Table C.21: JSBACH configurations of recent projects

	Tile Configuration	Project	References
4 tiles	1: dominant woody PFT	Millennium	[71]
	2: dominant grass PFT		[105]
	3: second woody or second grass PFT		[110]
	4: crops		
6 tiles	1: dominant deciduous tree PFT	LUCID	[104]
	2: dominant evergreen tree PFT		
	3: dominant shrub PFT or tundra		
	4: C3 grass and pasture		
	5: C4 grass		
	6: crops		
8 tiles	1: tropical broadleaf evergreen		[15]
	2: tropical broadleaf deciduous		[98]
	3: extra-tropical evergreen		[3]
	4: extra-tropical deciduous		
	5: raingreen shrubs		
	6: deciduous shrubs		
	7: C3 grass		
	8: C4 grass		
11 tiles	1: tropical broadleaf evergreen	CMIP5	[49]
	2: tropical broadleaf deciduous	CMIP6	
	3: extra-tropical evergreen	Holocene	
	4: extra-tropical deciduous	PalMod	
	5: raingreen shrubs	CRESCENDO	
	6: deciduous shrubs		
	7: C3 grass		
	8: C4 grass		
	9: C3 pasture		
	10: C4 pasture		
	11: C3/C4 crop		

A Fortran program to generate JSBACH initial files for different model setups is available:

`mpiesm/contrib/initial-tarfiles/jsbach_init_file.f90`.

A ksh-script to compile and run the program is associated in the same directory. You can specify here for example the horizontal resolutions, the number of tiles, the simulated year, a scheme for land use change, etc.

#### C.4.2 The initial file variables

Table C.22 lists the variables of the JSBACH initial file `jsbach.nc` together with a short description and the module that corresponds to the configurable initial file names. More information on the boundary data that is used in standard JSBACH experiments is provided in the subsections C.4.3 to C.4.8.

Table C.22: Variables of JSBACH initial file `jsbach.nc`

Variable Name	Description	Unit	Module
dimension variables			
<code>lon</code>	longitudes; only read in standalone JSBACH simulations	[degree E]	grid
<code>lat</code>	latitudes; only read in standalone JSBACH simulation	[degree N]	grid
<code>ntiles</code>	tiles	[ ]	grid
<code>soillev</code>	soil layers (lower boundary)	[m]	soil
<code>soil_layer_depth</code>	soil layer thickness	[m]	soil
boundary conditions			
<code>albedo_soil_vis</code>	soil albedo in the visible range	[ ]	surface
<code>albedo_soil_nir</code>	soil albedo in the near infrared	[ ]	surface
<code>albedo_veg_vis</code>	vegetation albedo in the visible range. Only used if <code>use_albedo-canopy=.TRUE.</code> in <code>albedo_ctl</code>	[ ]	surface
<code>albedo_veg_nir</code>	vegetation albedo in the near infrared. Only used if <code>use_albedo-canopy=.TRUE.</code> in <code>albedo_ctl</code>	[ ]	surface
<code>bclapp</code>	Clapp and Hornberger (1978) exponent b	-	soil
<code>cover_fract</code>	cover fraction of the PFTs defined in <code>cover_type</code> . With dynamic vegetation ( <code>use_dynveg= .TRUE.</code> in <code>jsbach_ctl</code> ), <code>cover_fract</code> only gives initial conditions.	[ ]	surface
<code>cover_type</code>	land cover type (LCT): The indices have to match the land cover type library <code>lctlb.def</code>	-	surface
<code>elevation</code>	mean orography	[m]	surface
<code>fao</code>	map of FAO soil types given as FAO soil type numbers ('flags')	-	soil
<code>heat_capacity</code>	heat capacity of dry soil	[J m <sup>-3</sup> K <sup>-1</sup> ]	soil
<code>heat_conductivity</code>	heat conductivity of dry soil	[J m <sup>-1</sup> s <sup>-1</sup> K <sup>-1</sup> ]	soil
<code>hyd_cond_sat</code>	saturated hydraulic conductivity	[m/s]	soil
<code>lake</code>	fractional land mask (1=lake; 0=no lake)	-	hydrology
<code>maxmoist</code>	maximum soil water capacity	[m]	soil
<code>moisture_pot</code>	saturated matrix potential	[m]	soil
<code>natural_veg</code>	cover fraction of natural PFTs if there was no anthropogenic land use. Only used with land-use transitions or dynamic vegetation ( <code>lcc_forcing_type='TRANSITIONS'</code> or <code>use_dynveg=.TRUE.</code> in <code>jsbach_ctl</code> )	[ ]	surface
<code>orography_std_dev</code>	orographic standard deviation	[m]	surface
<code>pore_size_index</code>	soil pore size distribution index	-	soil
<code>root_depth</code>	soil field capacity	[m]	soil
<code>roughness_length_oro</code>	orographic roughness length (without vegetation)	[m]	soil
<code>slf</code>	fractional sea land mask (0=water; 1=land)	-	hydrology

*table continued on next page*

Table C.22: `jsbach.nc` — continued

Variable Name	Description	Unit	Module
<code>slm</code>	integer sea land mask (0=water; 1=land) alternative names: <code>gridnum</code> , <code>landseemask</code>	-	grid
<code>soil_depth</code>	soil depth until bedrock	[m]	soil
<code>soil_field_cap</code>	volumetric soil field capacity	[m/m]	soil
<code>soil_porosity</code>	volumetric soil porosity	[m/m]	soil
<code>veg_ratio_max</code>	maximum vegetated fraction of the grid box. With dynamic vegetation ( <code>use_dynveg=.TRUE.</code> in <code>jsbach_ctl</code> ), <code>veg_ratio_max</code> only defines initial conditions.	[ ]	surface
<code>wilting_point</code>	volumetric wilting point	[m/m]	soil
initial conditions			
<code>layer_moist</code>	soil wetness of each soil layer	[m]	soil
<code>snow</code>	snow depth	[m (w.eq.)]	soil
<code>surf_temp</code>	surface temperature	[K]	soil
not used in JSBACH standard setups			
<code>albedo</code>	background albedo for ECHAM5 albedo model (section 4.1.2); only used if <code>use_albedo=.FALSE.</code> in namelist <code>jsbach_ctl</code>	[ ]	surface
<code>forest_fract</code>	forest fraction; used throughout the whole run with <code>use_albedo=.FALSE.</code> in namelist <code>jsbach_ctl</code>	[ ]	surface
<code>glac</code>	glacier mask (1=glacier; 0=no glacier); Not used: The glacier mask is determined from variables <code>cover_fract</code> and <code>cover_type</code>	[ ]	-
<code>init_moist</code>	soil wetness (used with bucket scheme)	[m]	soil
<code>lai_clim</code>	climatology of the leaf area index; used throughout the whole run with <code>use_phenology=.FALSE.</code> in namelist <code>jsbach_ctl</code>	[ ]	vegetation
<code>roughness_length</code>	roughness length (including orography and vegetation)	[m]	-
<code>veg_fract</code>	vegetated grid box fraction; used throughout the whole run with <code>use_phenology=.FALSE.</code> in namelist <code>jsbach_ctl</code>	[ ]	surface

### C.4.3 Grid Information

JSBACH is a grid point model. All grid boxes are treated separately, there is no connection between neighboring grid boxes. Longitudes `lon` and latitudes `lat` are needed to calculate the solar angle. They do not need to be stored in two dimensional arrays, it is also possible to provide a vector for the land points. In coupled JSBACH/ECHAM simulations JSBACH does not read the longitude and latitude arrays from the initial file but gets the grid information from the atmosphere model for consistency.

#### C.4.4 Land Sea Mask

The JSBACH land sea mask (`slm`, `landseamask` or `gridnum`) is an integer 1/0 mask. Value 1 represents land, value 0 ocean or lakes. JSBACH takes into account land grid boxes, only.

It turned out to be beneficial to use the same land sea mask in the ECHAM/JSBACH simulations as in coupled setups with the ocean model MPIOM. The standard JSBACH initial files (e.g. as used within the CMIP6 simulations) have a land sea mask determined by MPIOM. The ocean model land sea mask is interpolated to the atmosphere model grid. grid boxes with a land fraction above a certain threshold are defined as land grid boxes, the remaining grid boxes are ocean grid boxes. The threshold is close to 0.5, it is slightly modified for some ocean grids to get a global land fraction that matches observations. Program

```
mpiesm/contrib/initial_tarfiles/cosmos-tarfiles.ksh
```

can be used to generate the ECHAM and JSBACH land sea masks from a given ocean model resolution. grid boxes with a lake fraction of more than 0.5 are non-land grid boxes and are ignored in JSBACH.

#### C.4.5 Orography

The orographic data originally stems from the ECMWF. It is used in all ECHAM versions. JSBACH uses the mean orography (`elevation`) and the standard deviation (`orography_std_dev`). In contrast to former ECHAM versions JSBACH uses the orographic roughness length not including the vegetation roughness length `roughness_length_oro`. This allows the model to adapt the total roughness length with changing vegetation.

#### C.4.6 Vegetation Cover

A land cover type from the LCT library (`lctlib.def`) is defined for each tile of all land grid boxes (variable `cover_type`). The corresponding cover fraction is given in variable `cover_fract`, relative to the vegetated grid box fraction, i.e. the sum of the cover fractions over the tiles is defined to be 1 on all land grid boxes. For numerical reasons the cover fractions of all non glacier tiles have a minimum value of  $10^{-10}$ . The actual vegetated fraction of the grid box is defined by variable `veg_ratio_max`. This tiling concept is described in detail in section 1.3.2.

Variable `natural_veg` also defines cover fractions. Here only the natural vegetation types are taken into account, as if there was no anthropogenic land use. Again, the fractions are calculated relative to the total vegetated grid box fraction. The array is used only in configurations with anthropogenic or natural land cover change.

The vegetation data used in current JSBACH initial files, as e.g. for the CMIP6 simulations is based on landcover maps generated by Julia Pongratz [106]. In runs with land use transitions according to the New Hampshire Land-use Harmonization (LUH) Project [65] the anthropogenic vegetation fractions (crops and pastures) are prescribed from LUH vegetation states (compare chapter 10), and only natural vegetation fractions are based on Pongratz.

In ECHAM5 the vegetation was prescribed with monthly climatological values of vegetation fraction (`veg_fract`), forest fraction (`forest_fract`) and leaf area index (`lai_clim`). These climatologies are described in [60]. They are available in current JSBACH initial files, however in standard JSBACH simulations with dynamic phenology and albedo schemes these climatologies are only used within the initialization phase, if at all.

### C.4.7 Soil Conditions

JSBACH uses a multi layer soil scheme, whose details are described in section 2. In the default setup we use 5 vertical layers, that are identical for thermal and hydrological calculations. Nevertheless, the soil parameters of the jsbach initial file are generally defined as two dimensional horizontal arrays. At run time these arrays are spread over the different soil layers depending on the vertical depth of the different layers. In the following list the soil parameter fields of the jsbach initial file are explained:

`soil_depth` defines the vertical depth of the soil until bedrock. The minimum value is set to 10 cm for deserts and high mountain areas.

`root_depth` gives the maximum depth, that roots are reaching. By definition `root_depth` cannot be greater than `soil_depth`. Detailed information on the rooting depth is provided by T. Stacke (MPI-M) in *mpiesm/src/echam/doc/jsbach/StackeHagemann.DerivationOfRootingDepths4JSBACH.pdf*. To enable runs with dynamic vegetation the rooting depth was modified in desert areas: the values for the Olson Types 'tall grasses and shrubs', 'bare desert', 'semi desert' and 'semi desert sage' were replaced with values for 'semi desert shrubs'.

Please be aware, that in the current JSBACH version the root depth does not depend on the vegetation type and does not change, not even in runs with dynamic vegetation.

`maxmoist` is directly linked to the rooting depth (see above). It defines the maximum soil water capacity of the root zone. More information is given in [60].

`soil_field_cap` and `wilting_point` give the volumetric soil field capacity [m/m] and the volumetric wilting point [m/m] respectively, following [102]

`pore_size_index` gives the soil pore size distribution index defined by [145].

`moisture_pot`, the saturated matrix potential [m], the volumetric soil porosity `soil_porosity` in [m/m], the Clapp and Hornberger (1978) exponent `b` (`bclapp`) and the saturated hydraulic conductivity `hyd_cond_sat` in [m/s] are defined according to [7].

`heat_capacity` and `heat_conductivity`, the heat capacity [J m<sup>-3</sup> K<sup>-1</sup>] and the heat conductivity of dry soil [J m<sup>-1</sup> s<sup>-1</sup> K<sup>-1</sup>] also correspond to [7]. These arrays are only used if the permafrost scheme is switched on. Otherwise the soil heat capacity and conductivity are defined from `fao`, soil flags from the FAO soil classification of [48] based on [40] (variable `fao` below). These values do not depend on soil moisture.

`fao` Soil type map given as FAO soil type numbers ('flag'). See table C.23.

Table C.23: FAO soil types and corresponding FAO classification 'flags' used in the JSBACH initial file

flag	soil type
1	sand (coarse-textured)
2	sandy loam (soil types 1 and 3)
3	loam (medium-textured)
4	mixture of loam and clay (soil types 3 and 5)
5	clay (fine-textured)

### C.4.8 Albedo

As described in section 4.1, surface albedo is calculated from ground albedo maps (`albedo_soil_nir` and `albedo_soil_vis`) and, depending on the albedo scheme, from observation based canopy albedo maps (`albedo_veg_nir` and `albedo_veg_vis`) or from PFT dependent

canopy albedo values prescribed in the LCT library file `lctlib.def`. How the necessary albedo maps can be derived from observational data is explained in appendix F.

The background albedo `albedo` belongs to the ECHAM5 albedo scheme. It is described in [60]. It is not used in standard JSBACH simulations

## C.5 Boundary and Initial Data for the Hydrological Discharge (HD) model

The Hydrological Discharge model runs with a fixed horizontal resolution of 0.5 degrees, thus only one set of initial and boundary condition files is needed, which can be used with all JSBACH horizontal resolutions, at least for today’s orography.

### C.5.1 HD Boundary Conditions: `hdpara.nc`

The HD model reads boundary conditions from file `hdpara.nc`. It contains the HD internal land sea mask, riverflow directions and parameters needed to compute flow speeds and retention times (compare table C.24). Additional information is given in [57].

Table C.24: Variables of the HD parameter file `hdpara.nc`

Variable Name	Description	Unit
<code>lon</code>	longitudes: -179.75, -179.25, ... 179.75	[degree E]
<code>lat</code>	latitudes: 89.75, 89.25, ... -89.75	[degree N]
Boundary Conditions		
<code>FLAG</code>	HD model land sea mask: 0=ocean, 1=land	[]
<code>FDIR</code>	river directions: 0: ocean inflow cell, 1: NW, 2: N, 3: NE, 4: W, 5: internal drainage, 6: E 7: SW, 8: S, 9: SE	[]
<code>ALF_K</code>	overland flow parameter k: retention time	[day]
<code>ALF_N</code>	overland flow parameter n: number of reservoirs	[]
<code>ARF_K</code>	river flow parameter k: retention time	[day]
<code>ARF_N</code>	river flow parameter n: number of reservoirs	[]
<code>AGF_K</code>	ground flow parameter k: retention time	[day]
<code>AGF_N</code>	ground flow parameter n: number of reservoirs	[]
<code>AREA</code>	area of the 0.5 degree grid cells	[m <sup>2</sup> ]
<code>FILNEW</code>	longitude index of flow destination according to <code>FDIR</code> ; only used with <code>lhd_rout = .TRUE.</code>	[]
<code>FIBNEW</code>	latitude index of flow destination according to <code>FDIR</code> ; only used with <code>lhd_rout = .TRUE.</code>	[]

### C.5.2 HD Initial Conditions: `hdstart.nc`

The HD initial file is technically the same as a restart file. It contains the same state variables and no further information. It is thus possible to replace `hdstart.nc` with any restart file, that might be closer to equilibrium with your model setup. The `hdstart.nc` variables are listed in table C.25.

Table C.25: Variables of the HD initial file `hdstart.nc`

Variable Name	Description	Unit
<code>lon</code>	longitudes: -179.75, -179.25, ... 179.75	[degree E]
<code>lat</code>	latitudes: 89.75, 89.25, ... -89.75	[degree N]
Initial Conditions		
<code>FLFMEM</code>	linear overlandflow reservoir content	[m <sup>3</sup> ]
<code>FRFMEM1</code>	content of the 1 <sup>st</sup> reservoir of the inflow cascade	[m <sup>3</sup> ]
<code>FRFMEM2</code>	content of the 2nd reservoir of the inflow cascade	[m <sup>3</sup> ]
<code>FRFMEM3</code>	content of the 3rd reservoir of the inflow cascade	[m <sup>3</sup> ]
<code>FRFMEM4</code>	content of the 4th reservoir of the inflow cascade	[m <sup>3</sup> ]
<code>FRFMEM5</code>	content of the 5th reservoir of the inflow cascade	[m <sup>3</sup> ]
<code>FGMEM</code>	of the linear baseflow reservoir content	[m <sup>3</sup> ]
<code>FINFL</code>	inflow for each gridbox	[m <sup>3</sup> ]

### C.5.3 The HD remapping matrix: `rmp_hd.nc`

As the HD model runs on its own horizontal grid (0.5 degree resolution), runoff and drainage calculated in JSBACH need to be regridded to the HD model grid at each coupling time step. The corresponding remapping matrices are read from file `hd_remap.nc`. A variety of remapping matrices for different JSBACH resolutions are available from the pool directories

```
/pool/data/JSBACH/input/rrev/HD/rmp_res_to_hd.nc.
```

It is however easily possible to generate new remapping matrices for new JSBACH resolutions using the CDOs:

```
cdo gencon,/pool/data/JSBACH/prepare/HD/HD.grid jsbach_file rmp_res_to_hd.nc
```

The file contains detailed information on both, the JSBACH and HD model grids, as well as the remapping matrix. Land use masks are not included.

## C.6 Initial data for the Carbon Pools: `Cpools.nc`

The file `Cpools.nc` is an optional input file to initialize the carbon pools. Whether or not the file is expected is controlled by parameter `read_cpools` in namelist `cbalance_ctl` (compare table C.4). The file contains the state of the different carbon pools for all tiles. It is read within the initialization phase. A list of the variables is given in table C.26.

Typically, the carbon pools stem from a CBALONE run or from a previous JSBACH simulation (compare G). In the first case it is possible to use the cbalance output files `Cbalance.year.cyear.nc` or `restart_Cbalance.year.cyear.nc` as carbon pool file. The file name is configurable with keyword `cpools_file` of namelist `jsbach_ctl` (C.12).

The initialization from a previous JSBACH simulation is straightforward, if the model was running with fixed cover fractions. In this case, the vegetation stream `jsbach_veg_date` contains all carbon pools and can serve as `cpools_file`. If necessary it needs to be converted to netcdf. In runs with varying cover fractions (i.e. with dynamic vegetation or land-use change) the state variables of the carbon pools are not included in the output as the data range spans several orders of magnitude which cannot be handled in GRIB format. In this case the carbon pools have to be taken from the restart file. The program

`npiesm/contrib/restart2cpools.ksh`

helps to convert the data from land-point-only vectors to three dimensional maps as needed in `Cpools.nc`.

If no `Cpools.nc` file is specified, the carbon pools are initialized with constant values in routine `init_cbalance_bethy` in `mo_cbal_bethy.f90`. These values are listed in the third column of table C.26.

Table C.26: Variables of the carbon pool file `Cpools.nc`. The data is expected in unit  $[\text{mol}(\text{C})/\text{m}^2(\text{canopy})]$ . The values listed in the third column give the initial values for the carbon pools if no `Cpool` file is used (`read_cpools=.FALSE.`).

variable name	carbon pool description	value
<code>Cpool_green</code>	green parts of the vegetation	0
<code>Cpool_woods</code>	structural material of the plants	0
<code>Cpool_reserve</code>	reserve carbohydrates (starches, sugars)	0
<code>Cpool_crop_harvest</code>	biomass harvested from crops	0
<code>frac_litter_wood_new</code>	fraction of new litter in woody litter pools	0
only used with the CBALANCE carbon model		
<code>Cpool_litter_green_ag</code>	above ground non-woody litter (leaves)	60
<code>Cpool_litter_green_bg</code>	below ground non-woody litter (fine roots)	60
<code>Cpool_litter_wood_ag</code>	above ground woody litter	180
<code>Cpool_litter_wood_bg</code>	below ground woody litter	180
<code>Cpool_slow</code>	slowly respired soil organic material	2400
only used with the YASSO carbon model		
<code>YCpool_acid_ag1</code>	acid soluble above ground green material	0
<code>YCpool_water_ag1</code>	water soluble above ground green material	0
<code>YCpool_ethanol_ag1</code>	ethanol soluble above ground green material	0
<code>YCpool_nonsoluble_ag1</code>	non-soluble above ground green material	0
<code>YCpool_acid_bg1</code>	acid soluble below ground green material	0
<code>YCpool_water_bg1</code>	water soluble below ground green material	0
<code>YCpool_ethanol_bg1</code>	ethanol soluble below ground green material	0
<code>YCpool_nonsoluble_bg1</code>	non-soluble below ground green material	0
<code>YCpool_humus_1</code>	humus from green material	0
<code>YCpool_acid_ag2</code>	acid soluble above ground woody material	0
<code>YCpool_water_ag2</code>	water soluble above ground woody material	0
<code>YCpool_ethanol_ag2</code>	ethanol soluble above ground woody material	0
<code>YCpool_nonsoluble_ag2</code>	non-soluble above ground woody material	0
<code>YCpool_acid_bg2</code>	acid soluble below ground woody material	0
<code>YCpool_water_bg2</code>	water soluble below ground woody material	0
<code>YCpool_ethanol_bg2</code>	ethanol soluble below ground woody material	0
<code>YCpool_nonsoluble_bg2</code>	non-soluble below ground woody material	0
<code>YCpool_humus_2</code>	humus from woody material	0
only used with the anthropogenic carbon pools		
<code>Cpool_onSite</code>	carbon left on ground from land use change	0
<code>Cpool_paper</code>	wood carbon in short/intermediate term anthropogenic pool from land use change	0
<code>Cpool_construction</code>	wood carbon in long term anthropogenic pool from land use change	0

*table continued on next page*

Table C.26: Variables of the carbon pool file `Cpools.nc` — continued

variable name	carbon pool description	value
only used with the anthropogenic carbon pools and land use transitions		
<code>init_Cpool_onSite_harvest</code>	harvested carbon left on ground	0
<code>init_Cpool_paper_harvest</code>	harvested wood carbon in short/intermediate term anthropogenic pool	0
<code>init_Cpool_construction_harvest</code>	harvested wood carbon in long term anthropogenic pool	0

## C.7 Initial data for the Nitrogen Pools: `Npools.nc`

Similar to the carbon pools, initial nitrogen pool values can be read from file. This is controlled by namelist parameter `read_npools` in namelist `cbalance_ctl` (compare table C.4). Sometimes carbon pools are available from previous simulations, whereas nitrogen pools are missing (e.g. after a first spinup with nitrogen switched off). In that case it is useful to initialize the nitrogen pools consistent to the carbon pools according to the C/N ratios defined for the different pools. Table C.27 shows how nitrogen pools are calculated from the corresponding carbon pools (set namelist key `init_npools=.TRUE.` in namelist `cbalance_ctl`).

If no nitrogen pools are available from restart files, `read_npools=.FALSE.` and `init_npools=.FALSE.`, the nitrogen pools are initialized with zero. This is not recommended, as most plants will die without nitrogen, which prolongs spinup times considerably.

In our normal setups, carbon and nitrogen pools are provided in the same 'cpools' file. Methods on how to generate this file are described in the previous section.

Table C.27: Variables of the nitrogen pool file `Npools.nc`. The data is expected in unit  $[\text{mol}(\text{N})/\text{m}^2(\text{canopy})]$ . The entries in the third column show how N pools are initialized from C pools if `init_npools=.TRUE.`. Parameters `cn_green`, `cn_woods`, `cn_litter_green`, `cn_litter_wood` and `cn_slow` define the respective fixed C to N ratios (see namelist C.5). Stars (\*) indicate nitrogen pools that have pre-defined constant ratios to the corresponding carbon pools.

variable name	nitrogen pool description	with <code>init_npools=.TRUE.</code>
<code>Npool_green*</code>	organic N in green (living) parts of vegetation	<code>Cpool_green/cn_green</code>
<code>Npool_woods*</code>	organic N in (dead) structural material of living plants	<code>Cpool_woods/cn_woods</code>
<code>Npool_mobile</code>	plant internal retranslocation of N	<code>Npool_green</code>
<code>Npool_crop_harvest</code>	organic N in biomass harvested from crops	<code>Cpool_crop_harvest/cn_green</code>
<code>SMINN_pool</code>	mineral N in soil	<code>Npool_litter_green.bg + 1</code>
<code>NPP_run_mean</code>	running mean of NPP needed to calculate nitrogen fixation	0

with the CBALANCE carbon model

*table continued on next page*

Table C.27: Variables of the nitrogen pool file Npools.nc — continued

variable name	nitrogen pool description	with init_npools=.TRUE.
Npool_litter_green_ag	organic N in above ground non-woody litter	Cpool_litter_green_ag/cn_litter_green
Npool_litter_green_bg	organic N in below ground non-woody litter	Cpool_litter_green_bg/cn_litter_green
Npool_litter_wood_ag*	organic N in litter from above ground woody parts of the plants	Cpool_litter_wood_ag/cn_litter_wood
Npool_litter_wood_bg*	organic N in litter from below ground woody parts of the plants	Cpool_litter_wood_bg/cn_litter_wood
Npool_slow*	organic N in below ground organic material	Cpool_slow/cn_slow
with the YASSO carbon model		
Npool_litter_green_ag	organic N in above ground non-woody litter	(YCpool_acid_ag1 + YCpool_water_ag1 + YCpool_ethanol_ag1 + YCpool_nonsoluble_ag1)/cn_litter_green
Npool_litter_green_bg	organic N in below ground non-woody litter	(YCpool_acid_bg1 + YCpool_water_bg1 + YCpool_ethanol_bg1 + YCpool_nonsoluble_bg1)/cn_litter_green
Npool_litter_wood_ag*	organic N in litter from above ground woody parts of the plants	(YCpool_acid_ag2 + YCpool_water_ag2 + YCpool_ethanol_ag2 + YCpool_nonsoluble_ag2)/cn_litter_wood
Npool_litter_wood_bg*	organic N in litter from below ground woody parts of the plants	(YCpool_acid_bg2 + YCpool_water_bg2 + YCpool_ethanol_bg2 + YCpool_nonsoluble_bg2)/cn_litter_wood
Npool_slow*	organic N in below ground organic material	(YCpool_humus_1 + YCpool_humus_2)/cn_slow

## C.8 Initial data for foliage projective cover (FPC): fpc.nc

The FPC file can be used to initialize the dynamic vegetation module. Otherwise, FPCs fractions are initialized from the cover fractions defined in the jsbach initial file. The file is mainly needed for CBALONE simulations, where it serves as a restart file. The name is configurable in namelist dynveg.ct1 by keyword `fpc.file_name`. The file is read only if `read_fpc=.TRUE.` in the same name list.

Table C.28: Variables of the FPC file `fpc.nc`. 'Dynamic grid box fraction' means the fraction of the grid box which is covered by natural vegetation types that participate in vegetation dynamics. More information on the fraction definitions can be found in chapter 9.

Variable Name	Description	Unit
<code>act_fpc</code>	actual foliage projective cover	[relative to non-desert fraction of dynamic grid box fraction]
<code>pot_fpc</code>	potential foliage projective cover (without disturbances)	[relative to non-desert fraction of dynamic grid box fraction]
<code>bare_fpc</code>	fraction of bare ground	[relative to non-desert fraction of dynamic grid box fraction]
<code>desert_fpc</code>	desert fraction	[relative to dynamic grid box fraction]
<code>max_green_bio</code>	maximum value of green bio mass within a year	[mol(C) m <sup>-2</sup> (canopy)]
<code>sum_green_bio_memory</code>	vegetated fraction (calculated from green bio mass)	[relative to dynamic grid box fraction]
<code>cover_fract</code>	cover fraction	[relative to vegetated grid box fraction]

## C.9 Initial data for climate buffer variables: `climbuf.nc`

The state of the dynamic vegetation and disturbance calculations depends on several multi day or multi year climate variables. These variables are calculated in the so called climate buffer module. The file is mainly needed for CBALONE simulations, where it serves as restart file. Besides, it is needed in initialized simulations with `mpiesm-as` or `mpiesm-asob`. The filename is configurable in namelist `climbuf_ctl` by keyword `climbuf_file_name`. We typically add the climate buffer variables to the `cpools` file `Cpools.nc`. The climate buffer variables are read if `read_climbuf=.TRUE..`

Table C.29: Variables of the climate buffer file `climbuf.nc`. The file is read in simulations with dynamic vegetation or disturbance calculations.

Variable Name	Description	Unit
<code>ave_npp5</code>	5-year running mean of NPP; used in <code>dynveg</code>	[ $\frac{\text{mol}(\text{CO}_2)}{\text{m}^2(\text{canopy})}$ ]
<code>curr_day_temp_max</code>	maximum air temp of the current day until current time step; internal <code>climbuf</code> variable	[K]
<code>curr_day_temp_min</code>	maximum air temp of the current day until current time step; internal <code>climbuf</code> variable	[K]
<code>gdd_upper_tlim</code>	growing degree days of the previous year	[K]
<code>max_mmtmp20</code>	20-year running mean of highest monthly mean temperature; used in <code>dynveg</code>	[K]
<code>max_mmtmp_of_yr</code>	highest monthly mean temperature of previous year; internal <code>climbuf</code> variable	[K]
<code>max_wind10</code>	typical maximum daily 10m wind speed of the previous years; used in wind throw calculations	[m/s]

*table continued on next page*

Table C.29: Variables of the climate buffer file `climbuf.nc` — continued

Variable Name	Description	Unit
<code>min_mmtemp20</code>	20-year running mean of lowest monthly mean temperature; used in <code>dynveg</code>	[K]
<code>min_mmtemp_of_yr</code>	lowest monthly mean temperature of previous year; internal <code>climbuf</code> variable	[K]
<code>prev_day_max_wind10</code>	maximum 10m wind speed of previous day; internal <code>climbuf</code> variable	[m/s]
<code>prev_day_mean_wind10</code>	mean 10m wind speed of previous day; internal <code>climbuf</code> variable	[kg/m <sup>2</sup> s]
<code>prev_day_precip_mean</code>	mean precipitation of previous day; used in <code>spit-fire</code>	[kg/m <sup>2</sup> s]
<code>prev_day_temp_max</code>	maximum air temp of previous day; used in <code>spit-fire</code>	[K]
<code>prev_day_temp_min</code>	minimum air temp of previous day; used in <code>spit-fire</code>	[K]
<code>prev_year_gdd</code>	growing degree days of the previous year; used in <code>dynveg</code>	[K]
<code>prev_year_npp</code>	NPP of the previous year; internal <code>climbuf</code> variable	$[\frac{\text{mol}(\text{CO}_2)}{\text{m}^2(\text{canopy})}]$
<code>prev_year_precip</code>	precipitation of the previous year; internal <code>climbuf</code> variable	[kg/m <sup>2</sup> ]
<code>prev_year_soilmoist</code>	soil moisture of the previous year	[m]
<code>prev_year_soiltemp</code>	soil temperature of the previous year	[k]
<code>rel_hum_air</code>	relative air humidity, smoothed 14 days; used in <code>jsbach</code> fire module	[%]

## C.10 Forcing data for JSBACH standalone runs: `climate.nc`

In standard setup the forcing data for a JSBACH standalone simulation is read from a file called `climate_startyear.nc`. The name of this NetCDF file is configurable in `namelist forcing_ctl`. It is possible to use several independent forcing files for the different forcing variables (compare C.17). Table C.30 lists the variables expected in the forcing data file(s). Depending on the `namelist` specifications monthly, daily or subdaily forcing data can be read. The forcing frequency is defined for each variable independently. Note that an absolute time axis has to be used, i.e. "day as %Y%m%d.%f" rather than "hours since 1970-01-01 18:00".

Table C.30: Climate variables needed to force a JSBACH standalone simulation. The capital letter keywords refer to parameters in `namelist forcing_ctl`.

Variable	Description	Unit
<code>forcing_temp_file</code>		
<code>tmax</code>	maximum temperature	[°C]
<code>tmin</code>	minimum temperature	[°C]
or <code>air_temperature</code>	temperature (with <code>forcing_temp_freque=</code>	[°C]
	'SUBDAILY')	

*table continued on next page*

Table C.30: Climate variables needed to force a JSBACH standalone simulation — continued

Variable	Description	Unit
<b>forcing_precip_file</b>		
precip	precipitation (liquid and solid)	[kg m <sup>-2</sup> s <sup>-1</sup> ] or [mm/day] depending on <code>forcing_precip_in_mm_per_day</code>
<b>forcing_sw_file</b>		
shortwave or cloud	net shortwave radiation at the surface fractional cloud cover depending on <code>forcing_sw_type</code>	[W/m <sup>2</sup> ] [%]
<b>forcing_sw_pot_file</b>		
mpot	potential net shortwave radiation at the surface (clear sky)	[W/m <sup>2</sup> ]
<b>forcing_lw_file</b>		
longwave or cloud	net longwave radiation at the surface fractional cloud cover depending on <code>forcing_lw_type</code>	[W/m <sup>2</sup> ] [%]
<b>forcing_wind_file</b>		
wspeed	wind speed	[m/s]
<b>forcing_qair_file</b>		
qair	surface specific humidity	[kg/kg]
<b>forcing_co2_file</b>		
CO2	CO <sub>2</sub> concentration	[mol CO <sub>2</sub> /mol air]

### C.11 Forcing data for CBALONE: `exp_date.jsbach_yDay_Mean.nc`

The standalone carbon balance model CBALONE is generally driven by JSBACH output data. The model expects daily data in monthly NetCDF files. Variable `exp` in the file name represents the experiment ID of the driving JSBACH simulation and `date` gives the year and month (yyyymm). Table C.31 lists the variables needed to drive CBALONE. The CBALONE runscript (A.4) can be used to generate the CBALONE forcing files from JSBACH output.

Table C.31: JSBACH variables needed to drive CBALONE.

Variable	Description	Unit
Variables needed for all CBALONE experiments		
LAI_yDayMean	leaf area index of the previous day	[ ]
NPP_yDayMean	mean net primary production rate of the previous day	[mol/m <sup>2</sup> (canopy)s]
topSoilTemp_yDayMean	previous day mean temperature of the uppermost soil layer (not mandatory with YASSO)	[K]
alpha_yDayMean	previous day mean value of root zone water filling relative to maximum filling (not mandatory with YASSO)	[ ]

table continued on next page

Table C.31: JSBACH variables needed to drive CBALONE — continued

Variable	Description	Unit
Variables needed with dynamic vegetation only		
ave_npp5	5-year running mean of the net primary production	[mol/m <sup>2</sup> (canopy)]
bio_exist	flag, whether or not a PFT can survive	[ ]
Variables needed with disturbance only		
prev_day_max_wind10	maximum 10m-wind speed of the previous day	[m/s]
max_wind10	climatological maximum 10m-wind speed	[m/s]
rel_hum_air or rel_hum_air_yDay	relative humidity of the air in the lowest atmosphere level	[%]
Variables needed with YASSO, only		
pseudo_temp_yDay or pseudo_temp2_30d	running mean of temperature	[K]
pseudo_precip_yDay or pseudo_precip_30d	running mean of precipitation	[kg m <sup>-2</sup> s <sup>-1</sup> ]
Variables needed with nitrogen cycle, only		
Runoff_yDayMean	mean of yesterday's runoff	[kg m <sup>-2</sup> s <sup>-1</sup> ]
drainage_yDayMean	mean of yesterday's drainage	[kg m <sup>-2</sup> s <sup>-1</sup> ]
Variables needed with thonicke fire, only		
prev_day_precip_mean or precip	mean of yesterday's precipitation	[kg m <sup>-2</sup> s <sup>-1</sup> ]
topSoilMoisture_yDayMean or SoilMoisture_yDayMean	moisture of the uppermost soil layer (replaced by alpha, if not available)	[ ]
SurfaceTemp_yDayMean	mean of yesterday's temperature	[K]
prev_day_mean_wind10 or WindSpeed_yDayMean	maximum 10m-wind speed of the previous day	[m/s]
prev_day_temp_min or min_2m_temp	minimum of yesterday's temperature	[K]
prev_day_temp_max or max_2m_temp	maximum of yesterday's temperature	[K]

# Appendix D

## Output files

### Contents

---

D.1	Overview	228
D.2	The <code>jsbach</code> stream variables	230
D.3	The land stream variables	230
D.4	The nitrogen ( <code>nitro</code> ) stream variables	230
D.5	The surface ( <code>surf</code> ) stream variables	231
D.6	The vegetation ( <code>veg</code> ) stream variables	231
D.7	The Yasso stream variables	233
D.8	The area weighted ( <code>accw</code> ) stream variables	234
D.9	The carbon dioxide ( <code>co2</code> ) stream variables	234
D.10	The tracer stream variables	234
D.11	CBALONE output	234

---

### D.1 Overview

JSBACH3 shares the ECHAM concept for output stream definition and generation. Output format is GRIB, NetCDF, NetCDF2 or NetCDF4 depending on parameter `file_type` of namelist `jsbach_ctl`. In the standard configuration JSBACH produces separate output files for each month. They generally contain six-hourly data, however this is configurable via namelist. The JSBACH output streams are listed in table D.1. The main JSBACH stream (`jsbach`), contains the output of the photosynthesis and soil schemes for each tile of a grid box. Weighted averages of these quantities are available from the `land` stream. The vegetation (`veg`) stream contains daily output of the phenology and carbon pool modules as well as output from the dynamic vegetation module. The veg-stream quantities are updated only once a day, thus a higher output frequency is not appropriate.

The CBALONE model uses separate output routines supporting NetCDF format, only. Further information on the CBALONE output is available in section D.11.

GRIB output files come with a corresponding `filename.codes` file. This file contains a table of the GRIB code numbers, the corresponding NetCDF variable names and longnames and their units. Amongst others, these files can be used to convert the model output from GRIB to NetCDF using the CDOs, e.g.:

```
cdo -f nc -t exp_date_jsbach.codes copy exp_date_jsbach.grb exp_date_jsbach.nc
```

Tables D.2 to D.7 list the variables of the JSBACH output streams. The tables are generated from *filename.codes* files of a CMIP6 historical simulation, a setup with 11 tiles, active nitrogen cycle, land-use change prescribed from LU-transitions, vegetation dynamics, and SPITFIRE (Thonicke fire scheme). Some variables (and the corresponding grib code numbers) appear in the jsbach stream (on tiles) and in the land stream as weighted mean values integrated over the tiles. These two streams share the same grib code table, otherwise each of the output streams uses their own grib code table.

Table D.1: Output files in the different JSBACH modes: ECH: ECHAM/JSBACH, JSB: JSBACH standalone, CBL: CBALONE. Crosses in brackets indicate files only present for certain configurations. Variable *exp* represents the experiment ID, *date* gives year and month of the simulation. In CBALANCE file names the simulated year is represented by *syear*, the year of the forcing climate by *cyear*.

File Name in the Archive	File Name at Runtime	configuration		
		ECH	JSB	CBL
<b>Output</b>				
<i>exp_date_jsbach</i> . [grb/nc]	<i>exp_date.day1_jsbach</i> [.nc]	x	x	
<i>exp_date_jsbach_land</i> . [grb/nc]	<i>exp_date.day1_land</i> [.nc]	x	x	
<i>exp_date_jsbach_veg</i> . [grb/nc]	<i>exp_date.day1_veg</i> [.nc]	x	x	
<i>exp_date_jsbach_driving</i> . [grb/nc]	<i>exp_date.day1_driving</i> [.nc]		x	
<i>exp_date_jsbach_forcing</i> . [grb/nc]	<i>exp_date.day1_forcing</i> [.nc]		x	
<i>exp_date_jsbach_surf</i> . [grb/nc]	<i>exp_date.day1_surf</i> [.nc]	x		
<i>exp_date_jsbach_yasso</i> . [grb/nc]	<i>exp_date.day1_yasso</i> [.nc]	(x)	(x)	
<i>Cbalone.syear.cyear.nc</i>	<i>Cbalone.syear.cyear.nc</i>			x
<b>Restart</b>				
<i>rerun_exp_jsbach_enddate</i>	<i>rerun_exp_jsbach</i>	x	x	
<i>rerun_exp_veg_enddate</i>	<i>rerun_exp_veg</i>	x	x	
<i>rerun_exp_forcing_enddate</i>	<i>rerun_exp_forcing</i>		x	
<i>rerun_exp_driving_enddate</i>	<i>rerun_exp_driving</i>		x	
<i>rerun_exp_surf_enddate</i>	<i>rerun_exp_surf</i>	x		
<i>rerun_exp_yasso_enddate</i>	<i>rerun_exp_yasso</i>	(x)	(x)	
<i>restart_Cbalone.syear.cyear.nc</i>	<i>restart_Cbalone.syear.cyear.nc</i>			x

For completeness three echam output streams are added: The accw stream (D.8) contains time averaged and area weighted jsbach output variables. The stream was needed in setups with fractional land sea mask (preprocessor key ECHAM\_FRACTIONAL). It contains variables calculated in jsbach, that had formally been in the general echam output stream. To account for the fractional land sea mask, area weighting is needed for these quantities. However, as ECHAM\_FRACTIONAL is generally not defined, no area weighting is needed. Never the less these variables remain in the accw stream.

Among other things the CO<sub>2</sub> stream (D.9) contains the CO<sub>2</sub> fluxes from the different JSBACH compartments to the atmosphere. The tracer stream (D.10) can be used for different atmospheric tracers. In the CMIP6 setup it contains the CO<sub>2</sub> concentration in the atmosphere. The file is not written in setups with prescribed atmospheric CO<sub>2</sub> concentration.

## D.2 The jsbach stream variables

code	tiles	variable	description	units
10	11	cover_fract_pot	Potential Natural Land Cover Fraction	[]
12	11	cover_fract	Land Cover Fraction	[]
14	11	albedo_vis	Surface Albedo in the Visible Range	[]
15	11	albedo_nir	Surface Albedo in the NIR	[]
20	1	veg_ratio_max	Maximum Vegetation Fraction	[]
21	1	sdown_acc	Surface Downwelling Solar Radiation	[W m-2]
22	1	sdown_reflect_acc	Surface Upwelling Solar Radiation	[W m-2]
24	11	box_veg_ratio	vegetated fraction of grid box	[]
25	1	land_fract	Land Fraction	[]
30	1	sdown_absorb_nosno_acc	Surface absorbed Solar Radiation (without snow)	[W m-2]
55	11	skin_reservoir	Water Content of the Skin Reservoir	[m]
56	1	relative_humidity_air_inst	Relative Humidity	[]
67	11	glacier_runoff	Glacier Runoff	[kg m-2 s-1]
77	1	snow_age	Non-dimensional Snow Age	[]
84	5	layer_moisture	Soil Moisture Content	[m]
107	11	lai	Leaf Area Index	[]
109	11	snow_depth_canopy	Snow Depth on Canopy	[m]
110	11	snow_fract_canopy	Snow Fraction on Canopy	[]
116	11	veg_fract_correction	Correction factor for cover fraction 1-exp(-LAI_max/2)	[]
120	11	canopy_conductance	Canopy Conductance	[m/s]
124	11	net_assimilation_acc	Canopy Net Carbon Assimilation	[mol(CO2) m-2(canopy) s-1]
126	11	gross_assimilation_acc	Gross Assimilation (avg.)	[mol(CO2) m-2(canopy) s-1]
148	11	apar_acc	PAR absorbed by Canopy (avg.)	[mol(PHOTONS) m-2(canopy) s-1]
160	1	CO2_flux_net	Net CO2 flux to atmosphere (acc.)	[mol(CO2) m-2(grid box) s-1]
161	1	CO2_flux_herbivory	Grazing CO2 flux to atmosphere (acc.)	[mol(CO2) m-2(grid box) s-1]
162	1	CO2_emission_landcover_change	Land-cover change CO2 flux to the atmosphere (acc.)	[mol(CO2) m-2(grid box) s-1]
163	1	CO2_emission_harvest	Harvest CO2 flux to atmosphere (acc.)	[mol(CO2) m-2(grid box) s-1]
164	1	CO2_flux_dynveg	Disturbance CO2 flux to atmosphere (acc.)	[mol(CO2) m-2(grid box) s-1]
218	1	disch	river discharge	[m/s]
242	1	awfre	P-E over ocean and lakes	[kg/m**2s]

## D.3 The land stream variables

code	level	variable	description	units
36	1	surface_radiative_temp	Surface Radiative Temperature	[K]
40	1	ground_heat_flux	Ground Heat Flux (avg)	[W/m^2]
44	1	evapotranspiration	Evapotranspiration (avg)	[kg/m^2s]
60	1	snow_fract	Snow Cover Fraction	[]
67	1	glacier_runoff	Glacier Runoff	[kg/m^2s]
68	5	soil_temperature	Soil Temperature	[K]
72	1	csat	csat exchange coefficient	[]
73	1	cair	cair exchange coefficient	[]
74	1	csat_transpiration	csat transpiration coefficient	[]
76	1	transpiration	Transpiration (avg)	[kg/m^2s]
77	1	water_storage_pre	Water storage (avg)	[m]
78	1	water_balance	Soil Water Balance	[kg/m^2s]
79	1	bare_soil_evaporation	Bare Soil Evaporation (avg)	[kg/m^2s]
80	1	snow_evaporation	Snow Evaporation (avg)	[kg/m^2s]
81	1	skin_evaporation	Skin Reservoir Evaporation (avg)	[kg/m^2s]
82	1	reduced_evaporation	Potential for reducing Evaporation (avg)	[kg/m^2s]
83	1	precip_acc	Precipitation received by JSBACH	[kg/m^2s]
86	1	wetskin_fract	Mean Wet Skin Fraction	[]
91	1	surface_qair	Surface specific humidity, accu.	[kg/kg]
102	1	evaporation_deficit	Evaporation deficit flux (avg)	[kg/m^2s]
104	1	tair_max	Maximum temperature lowest atmospheric level	[K]
105	1	tair_min	Minimum temperature lowest atmospheric level	[K]
106	1	wind_lal_acc	Windspeed lowest atmospheric level	[m/s]
107	1	lai	Leaf Area Index	[]
109	1	snow_depth_canopy	Snow Depth on Canopy	[m]
110	1	snow_fract_canopy	Snow Fraction on Canopy	[]
124	1	net_assimilation	Canopy Net Carbon Assimilation (avg.)	[mol(CO2) m-2(grid box) s-1]
126	1	gross_assimilation	Gross Assimilation (avg.)	[mol(CO2) m-2(grid box) s-1]
139	1	surface_temperature_acc	Land Surface Temperature accumulated	[K]
148	1	apar_acc	PAR Absorbed by Canopy (avg.)	[mol(PHOTONS) m-2(grid box) s-1]
149	1	par_acc	incoming photosynthetically active radiation (PAR) (avg.)	[mol(PHOTONS) m-2 s-1]

## D.4 The nitrogen (nitro) stream variables

code	tiles	variable	description	units
70	11	box_minNflux_litter_green	Mineral N flux associated with green litter	[mol(N) m-2(grid box) s-1]
71	11	box_minNflux_litter_wood	Mineral N flux associated with woody litter	[mol(N) m-2(grid box) s-1]
72	11	box_minNflux_slow	Soil mineral N flux from slow pool	[mol(N) m-2(grid box) s-1]
73	11	box_Nplant_uptake	N uptake by plants	[mol(N) m-2(grid box) s-1]
74	11	box_Nsoil_uptake	N uptake by soil	[mol(N) m-2(grid box) s-1]

76	11	box_Ntotal_uptake	total N uptake by plants and soil	[mol(N) m-2(grid box) s-1]
77	11	box_Nfix_to_sminN	N flux to mineral N pool due to N fixation (BNF)	[mol(N) m-2(grid box) s-1]
79	11	box_Ndep_to_sminN	N flux to mineral N pool due to N deposition	[mol (N) m-2(grid box) s-1]
80	11	box_sminN_to_denit	N flux from mineral N pool due to denitrification	[mol(N) m-2(grid box)s-1]
81	11	box_sminN_leach	N flux to mineral N pool due to leaching	[mol(N) m-2(grid box)s-1]
82	11	box_sminN_herbivory	N flux from herbivory via animal faeces to mineral N pool	[mol(N) m-2(grid box) s-1]
84	11	box_sminN_loss	Total N outflux of soil mineral N pool	[mol(N) m-2(grid box)s-1]
86	1	LCC_flux_box_N2litterGreenPools	Nitrogen relocated from green and reserve pools to the green litter pools due to LCC	[mol(N) m-2(vegetated area) s-1]
87	1	LCC_flux_box_N2litterWoodPool	Wood nitrogen relocated by landcover change to woody litter pool	[mol(N) m-2(vegetated area) s-1]
88	1	LCC_flux_box_N2SMINNpool	Wood nitrogen relocated by landcover change to soil mineral N pool	[mol(N) m-2(vegetated area) s-1]
129	1	N2atmos_LUtrans	Nitrogen flux to atmosphere from land use change	[mol(N) m-2(grid box) s-1]
130	1	N2atmos_harvest	Nitrogen flux to atmosphere from harvest	[mol(N) m-2(grid box) s-1]
138	1	jsbachNconserv	jsbach nitrogen conservation test: should be zero.	[mol(N2) m-2(grid box)]
158	1	box_Nflx_2_crop_harvest	N flux from vegetation to crop harvest pool	[mol(N) m-2(grid box) s-1]
212	11	box_N20_total	Sum of all N20 fluxes	[mikro-mol(N) m-2(grid box) s-1]
222	11	boxN_crop_harvest	N-Pool for material harvested from crops	[mol(N) m-2(grid box)]
236	1	Ndep_forc	atmospheric N deposition	[kg (N) m-2(grid box) s-1]
237	11	NetEcosyst_N_flux	Total balance of N gains and losses (positive for ecosystem gain)	[mol(N) m-2(canopy) s-1]
238	11	NPP_run_mean	Exponential running mean of NPP used for N-fixation model	[mol(N) m-2(canopy) s-1]
240	11	boxN_green	N-Pool for Green Parts of Vegetation	[mol(N) m-2(grid box)]
241	11	boxN_woods	N-Pool for Structural Material of Plants	[mol(N) m-2(grid box)]
242	11	boxN_mobile	N-Pool for mobile N	[mol(N) m-2(grid box)]
243	11	boxN_litter_green_ag	N-Pool for above ground non-woody litter (leaves)	[mol(N) m-2(grid box)]
244	11	boxN_litter_green_bg	N-Pool for below ground non-woody litter (fine roots)	[mol(N) m-2(grid box)]
245	11	boxN_litter_wood_ag	N-Pool for above ground woody litter	[mol(N) m-2(grid box)]
246	11	boxN_slow	N-Pool for slowly respired soil organic material	[mol(N) m-2(grid box)]
247	1	box_Npools_total	Sum of nitrogen from all nitrogen pools	[mol(N) m-2(grid box)]
248	11	boxN_litter_wood_bg	N-Pool for below ground woody litter	[mol(N) m-2(grid box)]
250	11	boxN_sminN	N-Pool for soil mineral nitrogen	[mol(N) m-2(grid box)]
251	11	box_N20_emissions_depfix	N20 emissions due to deposition and fixation	[mikro-mol(N) m-2(grid box)]
252	11	box_N20_emissions_mineraliz	N20 emissions due to mineralization	[mikro-mol(N) m-2(grid box)]

## D.5 The surface (surf) stream variables

code	tiles	variable	description	units
203	1	jsswniracc	net surface NIR accumulated	[W/m**2]
204	1	jsswvisacc	net surface visible accumulated	[W/m**2]
205	1	jsswdifniracc	net surface diffuse NIR accumulated	[W/m**2]
206	1	jsswdifvisacc	net surface diffuse visible accumulated	[W/m**2]
207	1	longwave_down_acc	downward longwave radiation	[W/m**2]
212	1	az0lh	roughness length of heat over land	[m]
213	1	az0	roughness length	[m]
244	1	jsswparacc	downward surface PAR accumulated	[W/m**2]
246	1	jsswdifparacc	downward surface diffuse PAR accumulated	[W/m**2]

## D.6 The vegetation (veg) stream variables

code	tiles	variable	description	units
18	11	prev_year_npp	NPP of the previous year	[mol(CO2) m-2(canopy)]
19	11	ave_npp5	five year running mean of annual NPP	[mol(CO2) m-2(canopy)]
20	1	prev_day_max_wind10	maximum 10m wind speed of the previous day	[m/s]
23	1	max_wind10	typical maximum daily 10m wind speed of the previous years	[m/s]
24	1	rel_hum_air_yDay	smoothed (~14 day) relative humidity of the air	[%]
26	11	top_layer_soil_moisture	relative soil moisture of the previous day	[m/m]
29	1	prev_day_temp_max	maximum temperature of the previous day	[K]
30	1	prev_day_temp_min	minimum temperature of the previous day	[K]
31	11	act_fpc	fractional plant cover	[]
34	1	desert_fpc	desert fraction	[]
35	1	bare_fpc	bare soil fraction	[]
37	1	prev_day_precip_mean	mean precipitation of the previous day	[kg m-2 s-1]
38	11	pot_fpc	potential plant cover fraction	[]
39	11	bio_exist	bio exist	[]
40	1	sum_green_bio_memory	vegetated fraction calculated from green biomass	[mol(C) m-2(canopy)]

42	1	prev_day_mean_wind10	mean 10m wind speed of the previous day	[m/s]
50	11	box_burned_frac_diag_avg	average daily burned area fraction including survivors	[m2/m2 (grid box) d-1]
51	1	fuel	fuel for burned area calculations of Thonicke fire scheme	[mol(C)/m2(veg)]
52	11	burned_frac	daily burned area fraction	[m2/m2(veg) d-1]
53	11	box_burned_frac_avg	average daily burned area fraction	[m2/m2(grid box) d-1]
54	11	damaged_frac	daily damaged area fraction	[m2/m2(veg) d-1]
55	11	box_damaged_frac_avg	average daily damaged area fraction	[m2/m2(grid box) d-1]
56	1	box_fire_CO2_flux_2_	atmos average daily CO2 flux to the atmosph.	[kg(CO2) m-2(grid box) d-1]
62	1	dist_nitrogen_2_atmos	N compounds emitted to the atmosphere by fire	[mol(N) m-2(grid box) d-1]
65	11	frac_1hr_wood	fraction of 1hr fuel in wood pool	[]
66	11	frac_10hr_wood	fraction of 10hr fuel in wood pool	[]
67	11	frac_100hr_wood	fraction of 100hr fuel in wood pool	[]
68	11	frac_1000hr_wood	fraction of 1000hr fuel in wood pool	[]
69	1	NI_acc	average Nesterov index	[]
75	11	frac_litter_wood_new	fraction of new litter in woody litter pool	[]
80	1	Box_flux_harvest_2atmos	Harvest flux from natural vegetation emitted from gridbox as CO2 to atmosphere	[mol(C) m-2(gridbox) s-1]
81	11	LCC_coverFract_target	Vegetation cover fractions that should be reached at the end of the year as a result of landcover change.	[]
82	1	LCC_flux_box_C2atmos	Carbon flux to atmosphere from land cover/land use change	[mol(C) m-2(grid box) s-1]
83	1	LCC_flux_box_C2litterGreenPools	Flux of green and reserve carbon relocated by landcover change to the two green litter pools	[mol(C) m-2(grid box) s-1]
84	1	LCC_flux_box_C2litterWoodPool	Wood carbon relocated by landcover change to woody litter pool	[mol(C) m-2(grid box) s-1]
123	1	CROP_2_NATL_ignored	Cover fraction that could not be transferred from croplands to natural vegetation	[-]
124	1	PAST_2_NATL_ignored	Cover fraction that could not be transferred from pastures to natural vegetation	[-]
125	1	Box_harvest	Prescribed harvest from natural vegetation for full gridbox	[mol(C) m-2(gridbox) s-1]
126	1	Box_flux_harvest	Total harvest flux from natural vegetation for full gridbox	[mol(C) m-2(gridbox) s-1]
130	11	excess_NPP	NPP that could not be not stored in vegetation carbon pools	[mol(CO2) m-2(canopy) s-1]
131	11	root_exudates	root exudates	[mol(C) m-2(canopy) s-1]
132	11	Cflux_herbivory	Cflux_herbivory	[mol(C) m-2(canopy) s-1]
133	11	Cflux_herbivory_LG	Cflux_herbivory_LG	[mol(C) m-2(canopy) s-1]
134	11	Cflux_herbivory_2_atm	Cflux_herbivory_2_atm	[mol(C) m-2(canopy) s-1]
135	11	box_root_exudates	root exudates	[mol(C) m-2(grid box) s-1]
136	11	box_Cflux_herbivory	Cflux_herbivory	[mol(C) m-2(grid box) s-1]
138	1	jsbachCconserv	jsbach carbon conservation test: should be zero.	[mol(CO2) m-2(grid box)]
143	11	box_Cflux_herbivory_2_atm	Carbon flux from herbivory to the atmosphere	[mol(C) m-2(grid box) s-1]
157	11	NPP_flux_correction	Flux correction for NPP	[mol(CO2) m-2(canopy) s-1]
158	1	box_Cflx_2_crop_harvest	Carbon flux from crops to crop harvest pool	[mol(C) m-2(grid box) s-1]
160	11	boxC_green	C-Pool for Green Parts of Vegetation	[mol(C) m-2(grid box)]
161	11	boxC_woods	C-Pool for Structural Material of Plants	[mol(C) m-2(grid box)]
162	11	boxC_reserve	C-Pool for reserve carbohydrates (starches, sugars)	[mol(C) m-2(grid box)]
165	11	LAI_yDayMean	Mean Leaf Area Index of the Previous Day	[]
166	11	NPP_yDayMean	Mean NPP Rate of the Previous Day	[mol(CO2) m-2(canopy) s-1]
168	1	topSoilTemp_yDayMean	Previous Day Mean Temperature of the Uppermost Soil Layer	[K]
169	11	alpha_yDayMean	Previous Day Mean Value of the Water Stress Coefficient	[]
170	11	box_soil_respiration	Soil respiration	[mol(C) m-2(grid box) s-1]
171	11	box_NPP_yDayMean	Mean potential NPP of the Previous Day	[mol(CO2) m-2(grid box) s-1]
172	11	box_NPP_flux_correction	Flux correction for NPP	[mol(CO2) m-2(grid box) s-1]
173	11	box_GPP_yDayMean	Mean GPP Rate of the Previous Day	[mol(CO2) m-2(grid box) s-1]
175	11	box_litter_flux	Total litter flux entering the soil pools	[mol(CO2) m-2(grid box) s-1]
176	1	box_Cpools_total	Sum of carbon from all carbon pools	[mol(CO2) m-2(grid box)]
178	11	box_NPP_act_yDayMean	Mean actual NPP the Previous Day	[mol(CO2) m-2(grid box) s-1]
181	1	growth_phase_SG	Growth Phase of Summergreens	[]
182	1	growth_phase_EG	Growth Phase of Evergreens	[]
185	1	previous_day_temp	Air Temperature of the Previous Day	[degC]
186	1	pseudo_soil_temp	Pseudo Soil Temperature	[degC]
188	1	heat_sum_EG	Heat Sum - Evergreens	[degC d]
189	1	heat_sum_SG	Heat Sum - Summergreens	[degC d]
190	1	chill_days_EG	Number of Chill Days - Evergreens	[]
191	1	chill_days_SG	Number of Chill Days - Summergreens	[]
196	1	growth_phase_CRP	Growth Phase of extra-tropical crops	[]
197	1	heat_sum_CRP	Heat Sum - Crops	[degC d]
199	1	heat_sum_winter	Heat Sum - Winter Crops	[degC d]
205	1	previous_day_temp_min	Minimum air Temp. of the Previous Day	[degC]
206	1	previous_day_temp_max	Maximum air Temp. of the Previous Day	[degC]
210	11	NPP_act_yDayMean	NPP_act_yDayMean	[mol(CO2) m-2(canopy) s-1]
211	1	box_Cflx_crop_harvest_2_atm	Carbon flux from crop harvest pool to the atmosphere	[mol(C) m-2(grid box) s-1]
212	11	redFact_Nlimitation	redFact_Nlimitation	[dimensionless]
218	1	boxC_onSite_LCC	Carbon left on ground from land use change	[mol(C) m-2(grid box)]
220	1	boxC_paper_LCC	Wood carbon in short/intermediate term anthropogenic pool from land use change	[mol(C) m-2(grid_box)]
221	1	boxC_construction_LCC	Wood carbon in long term anthropogenic pool	[mol(C) m-2(grid box)]

222	1	boxC_paper_harvest	from land use change Wood carbon in short/intermediate term anthropogenic pool from harvest	[mol(C) m-2(grid box)]
223	1	boxC_construction_harvest	Wood carbon in long term anthropogenic pool from harvest	[mol(C) m-2(grid box)]
224	1	boxC_onSite_harvest	Carbon left on ground from harvest	[mol(C)m-2(grid box)]
225	11	boxC_crop_harvest	C-Pool for material harvested from crops	[mol(C) m-2(grid box)]
228	1	boxC_flux_2_onSite_LCC	Carbon flux from living plants to ground pool from land use change	[mol(C) m-2(grid box) s-1]
230	1	boxC_flux_2_paper_LCC	Wood carbon flux from living plants to short/intermediate term anthropogenic pool from land use change	[mol(C) m-2(grid box) s-1]
231	1	boxC_flux_2_construction_LCC	Wood carbon flux from living plants to long term anthropogenic pool from land use change	[mol(C) m-2(grid box) s-1]
236	1	boxC_flux_onSite_2_atmos_LCC	Carbon flux from ground pool from land use change to	[mol(C) m-2(grid box) s-1]
238	1	boxC_flux_paper_2_atmos_LCC	Carbon flux from short/intermediate term anthropogenic pool from land use change to atmos	[mol(C) m-2(grid box) s-1]
239	1	boxC_flux_construction_2_atmos_LCC	Carbon flux from long term anthropogenic pool from land use change to atmos	[mol(C) m-2(grid box) s-1]
242	1	boxC_flux_2_paper_harvest	Wood carbon flux from living plants to intermediate term anthropogenic pool from harvest	[mol(C) m-2(grid box) s-1]
243	1	boxC_flux_2_construction_harvest	Wood carbon flux from living plants to long term anthropogenic pool from harvest	[mol(C) m-2(grid box) s-1]
246	1	boxC_flux_paper_2_atmos_harvest	Carbon flux intermediate term anthropogenic pool from harvest to atmos	[mol(C) m-2(grid box) s-1]
247	1	boxC_flux_construction_2_atmos_harvest	Carbon flux from long term anthropogenic pool from harvest to atmos	[mol(C) m-2(grid box) s-1]
248	1	drainage_yDayMean	Previous Day Mean drainage	[Kg m-2 s-1]
249	1	boxC_flux_2_onSite_harvest	Wood carbon flux from living plants to short term anthropogenic pool from harvest	[mol(C)m-2(grid box)s-1]
251	1	boxC_flux_onsite_2_atmos_harvest	Carbon flux short term anthropogenic pool from harvest to atmos	[mol(C) m-2(grid box) s-1]
254	11	box_soil_respiration_pot	Soil respiration without N limitation	[mol(C) m-2(grid box) s-1]

## D.7 The Yasso stream variables

code	tiles	variable	description	units
31	11	boxYC_acid_ag1	C-Pool for acid-soluble litte in Yasso (aboveground)	[mol(C) m-2(grid box)]
32	11	boxYC_acid_bg1	C-Pool for acid-soluble litte in Yasso (belowground)	[mol(C) m-2(grid box)]
33	11	boxYC_water_ag1	C-Pool for water-soluble litter in Yasso (aboveground)	[mol(C) m-2(grid box)]
34	11	boxYC_water_bg1	C-Pool for water-soluble litter in Yasso (belowground)	[mol(C) m-2(grid box)]
35	11	boxYC_ethanol_ag1	C-Pool for ethanol-soluble litter in Yasso (aboveground)	[mol(C) m-2(grid box)]
36	11	boxYC_ethanol_bg1	C-Pool for ethanol-soluble litter in Yasso (belowground)	[mol(C) m-2(grid box)]
37	11	boxYC_nonsoluble_ag1	C-Pool for nonsoluble litter in Yasso (aboveground)	[mol(C) m-2(grid box)]
38	11	boxYC_nonsoluble_bg1	C-Pool for nonsoluble litter in Yasso (belowground)	[mol(C) m-2(grid box)]
39	11	boxYC_humus_1	C-Pool for humus in Yasso	[mol(C) m-2(grid box)]
41	11	boxYC_acid_ag2	C-Pool for acid-soluble litte in Yasso (aboveground)	[mol(C) m-2(grid box)]
42	11	boxYC_acid_bg2	C-Pool for acid-soluble litte in Yasso (belowground)	[mol(C) m-2(grid box)]
43	11	boxYC_water_ag2	C-Pool for water-soluble litter in Yasso (aboveground)	[mol(C) m-2(grid box)]
44	11	boxYC_water_bg2	C-Pool for water-soluble litter in Yasso (belowground)	[mol(C) m-2(grid box)]
45	11	boxYC_ethanol_ag2	C-Pool for ethanol-soluble litter in Yasso (aboveground)	[mol(C) m-2(grid box)]
46	11	boxYC_ethanol_bg2	C-Pool for ethanol-soluble litter in Yasso (belowground)	[mol(C) m-2(grid box)]
47	11	boxYC_nonsoluble_ag2	C-Pool for nonsoluble litter in Yasso (aboveground)	[mol(C) m-2(grid box)]
48	11	boxYC_nonsoluble_bg2	C-Pool for nonsoluble litter in Yasso (belowground)	[mol(C) m-2(grid box)]
49	11	boxYC_humus_2	C-Pool for humus in Yasso	[mol(C) m-2(grid box)]
50	1	pseudo_temp_yDay	Pseudo-mean air temperature over previous 15 days	[K]
51	1	pseudo_precip_yDay	Pseudo-mean precipitation rate over previous 15 days	[Kg m-2 s-1]

## D.8 The area weighted (accw) stream variables

code	level	variable	description	units
160	1	runoff	Surface Runoff and Drainage (acc+weighted)	[kg/m <sup>2</sup> s]
161	1	drainage	Drainage (acc+weighted)	[kg/m <sup>2</sup> s]
206	1	ground_heat_flux	Ground Heat Flux (acc+weighted)	[W/m <sup>2</sup> ]
215	1	glacier_runoff	Glacier Runoff (acc+weighted)	[kg/m <sup>2</sup> s]
218	1	snow_melt	Land Snow Melt (acc+weighted)	[kg/m <sup>2</sup> s]
221	1	glacier_precip_minus_evap	P-E over land ice (acc+weighted)	[kg/m <sup>2</sup> s]
222	1	snow_accumulation	Snow Accumulation (acc+weighted)	[kg/m <sup>2</sup> s]
228	1	glacier_melt	Glacier Ice Melt (acc+weighted)	[kg/m <sup>2</sup> s]

## D.9 The carbon dioxide (co2) stream variables

code	level	variable	description	unit
2	1	co2_flux_inst	upward surface CO2 flux	[kg m <sup>-2</sup> s <sup>-1</sup> ]
5	1	co2_flux	total upward surface CO2 flux (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
6	1	co2_flux_land	total upward land CO2 flux (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
7	1	co2_flux_ocean	upward ocean CO2 flux (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
8	1	co2_burden	CO2 content (acc.)	[kg m <sup>-2</sup> ]
14	1	co2_burden_inst	CO2 content	[kg m <sup>-2</sup> ]
17	1	co2ocean	pco2 of surface ocean water	[ppm CO2]
18	1	co2trans	transfer velocity of ocean/atmosphere CO2 flux	[10 <sup>-9</sup> mol s m <sup>-4</sup> ]
20	1	co2_flux_anthro	upward anthropogenic CO2 flux (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
21	1	co2_flux_npp	upward CO2 flux - NPP (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
22	1	co2_flux_resp	upward CO2 flux - soil respiration (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
23	1	co2_flux_herb	upward CO2 flux - herbivory (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
24	1	co2_flux_lcc	upward CO2 flux - land use change (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
25	1	co2_flux_harvest	upward CO2 flux - harvest (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]
26	1	co2_flux_fire	upward CO2 flux - fire (acc.)	[kg m <sup>-2</sup> s <sup>-1</sup> ]

## D.10 The tracer stream variables

code	level	variable	description	units
1	47	CO2	mass fraction of CO2 in air	[kg kg <sup>-1</sup> ]
129	1	geosp	surface geopotential (orography)	[m**2/s**2]
134	1	aps	surface pressure	[Pa]
152	1	lsp	log surface pressure	[ ]
2	1	gboxarea		[m <sup>2</sup> ]

## D.11 CBALONE output

CBALONE is a standalone version of JSBACH for running only those components with long memory (compare table A.1), namely the carbon, nitrogen and landcover change modules of JSBACH3. For historical reasons, CBALONE uses different IO-routines than JSBACH and ECHAM. CBALONE generates NetCDF output files named `Cbalance.syear.cyear.nc`, where *syear* and *cyear* represent the simulated and the forcing climate data years, respectively. Yearly, monthly and daily output is supported and configurable in namelist `cbalone_ctl` (C.2.18). Several output variables are averaged over the output period. The corresponding variable names contain the name-part `avg`. The name-part `box` generally also indicates time averages over the output period. Besides, the `box`-variables give quantities relative to the grid box area rather than to the canopy or vegetated fraction of the grid box.

The CBALONE restart file `restart_Cbalance.syear.cyear.nc` contains the same list of variables as the standard output files, however the data corresponds to the last day of the simulation, i.e. December 31<sup>st</sup> and the time averaged, diagnostic `avg`- or `box`- variables are incorrect.

Table D.2 lists the CBALONE output variables and gives a short description. As mentioned above, variables of extensive quantities refer to different areas. Especially within the

dynamic vegetation module quantities are scaled to different area fractions. In the table this is indicated as follows:

- \* :Variable relative to grid box fraction (as **box**-variables)
- \*\* :Variable relative to the vegetated fraction of the grid box, defined by **veg\_ratio\_max**
- \*\*\* :Variable relative to dynamic grid box fraction, i.e. relative to the cover fractions of the natural vegetation types that participate in the dynamic vegetation calculations.
- \*\*\*\* :Variable relative to non-desert fraction of dynamic grid box fraction

Table D.2: Variables of the CBALONE output file *Cbalone.syear.cyear.nc*

Variable Name	dim.	Description	Units
Output in all configurations			
<b>Cpool_green</b>	tiles	green parts of the vegetation	[mol(C)/m <sup>2</sup> (canopy)]
<b>Cpool_reserve</b>	tiles	reserve carbohydrates (starches, sugars)	[mol(C)/m <sup>2</sup> (canopy)]
<b>Cpool_crop_harvest</b>	tiles	harvested material from crops	[mol(C)/m <sup>2</sup> (canopy)]
<b>frac_litter_wood_new</b>	tiles	fraction of new litter in woody litter pool	[ ]
<b>Cpool_woods</b>	tiles	structural material of the plants	[mol(C)/m <sup>2</sup> (canopy)]
<b>avg_Cpool_green</b>	tiles	as <b>Cpool_green</b> , relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
<b>avg_Cpool_reserve</b>	tiles	as <b>Cpool_reserve</b> , relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
<b>avg_Cpool_crop_harvest</b>	tiles	as <b>Cpool_crop_harvest</b> , relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
<b>avg_Cpool_woods</b>	tiles	as <b>Cpool_woods</b> , relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
<b>avg_soil_respiration</b>	tiles	soil respiration rate	[mol(C)/m <sup>2</sup> (grid box)s]
<b>avg_NPP_yDayMean</b>	tiles	NPP rate	[mol(C)/m <sup>2</sup> (grid box)s]
<b>avg_NPP_act_yDayMean</b>	tiles	NPP rate of the previous day	[mol(C)/m <sup>2</sup> (grid box)s]
<b>avg_NPP_flux_correction</b>	tiles	flux correction for NPP	[mol(C)/m <sup>2</sup> (grid box)s]
<b>avg_excess_NPP</b>	tiles	NPP dropped into the fast pool	[mol(C)/m <sup>2</sup> (grid box)s]
<b>avg_root_exudates</b>	tiles	root exudates entering to the green litter pools	[mol(C)/m <sup>2</sup> (grid box)s]
<b>avg_Cflux_herbivory</b>	tiles	carbon flux due to herbivory	[mol(C)/m <sup>2</sup> (canopy)s]
<b>avg_Cflux_herbivory_LG</b>	tiles	flux from the green pool to the above ground green litter pool due to herbivory.	[mol(C)/m <sup>2</sup> (canopy)s]
<b>avg_Cflux_herbivory_2_atm</b>	tiles	flux from the green pool to the atmosphere due to herbivory	[mol(C)/m <sup>2</sup> (canopy)s]
<b>avg_box_NEP</b>	1	net ecosystem productivity	[kg(CO <sub>2</sub> )/m <sup>2</sup> (grid box)s]
<b>landindex</b>	1	index of the land grid boxes NE to SW	[ ]
Output with CBALANCE Soil Carbon scheme			
<b>Cpool_litter_green_ag</b>	tiles	above ground non-woody litter (leaves)	[mol(C)/m <sup>2</sup> (canopy)]
<b>Cpool_litter_green_bg</b>	tiles	below ground non-woody litter (fine roots)	[mol(C)/m <sup>2</sup> (canopy)]
<b>Cpool_litter_wood_ag</b>	tiles	above ground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
<b>Cpool_litter_wood_bg</b>	tiles	below ground woody litter	[mol(C)/m <sup>2</sup> (canopy)]

*table continued on next page*

Table D.2: Cbalone.year.cyear.nc — continued

Variable Name	dim.	Description	Units
Cpool_slow	tiles	slowly respired soil organic material	[mol(C)/m <sup>2</sup> (canopy)]
avg_Cpool_litter_green_ag	tiles	as Cpool_litter_green_ag, relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
avg_Cpool_litter_green_bg	tiles	as Cpool_litter_green_bg, relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
avg_Cpool_litter_wood_ag	tiles	as Cpool_litter_wood_ag, relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
avg_Cpool_litter_wood_bg	tiles	as Cpool_litter_wood_bg, relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
avg_Cpool_slow	tiles	as Cpool_slow, relative to grid box area	[mol(C)/m <sup>2</sup> (grid box)]
Output with YASSO Soil Carbon scheme			
YCpool_acid_ag1	tiles	acid soluble aboveground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_water_ag1	tiles	water soluble aboveground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_ethanol_ag1	tiles	ethanol soluble aboveground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_nonsoluble_ag1	tiles	non-soluble aboveground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_acid_bg1	tiles	acid soluble belowground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_water_bg1	tiles	water soluble belowground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_ethanol_bg1	tiles	ethanol soluble belowground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_nonsoluble_bg1	tiles	non-soluble belowground green litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_humus_1	tiles	humus from green carbon	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_acid_ag2	tiles	acid soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_water_ag2	tiles	water soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_ethanol_ag2	tiles	ethanol soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_nonsoluble_ag2	tiles	non-soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_acid_bg2	tiles	acid soluble belowground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_water_bg2	tiles	water soluble belowground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_ethanol_bg2	tiles	ethanol soluble belowground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_nonsoluble_bg2	tiles	non-soluble belowground woody litter	[mol(C)/m <sup>2</sup> (canopy)]
YCpool_humus_2	tiles	humus from woody carbon	[mol(C)/m <sup>2</sup> (canopy)]
avg_YCpool_acid_ag1	tiles	mean acid soluble aboveground green litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_water_ag1	tiles	mean water soluble aboveground green litter	[mol(C)/m <sup>2</sup> (grid box)]

table continued on next page

Table D.2: Cbalone.syear.cyear.nc — continued

Variable Name	dim.	Description	Units
avg_YCpool_ethanol_ag1	tiles	mean ethanol soluble aboveground green litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_nonsoluble_ag1	tiles	mean non-soluble aboveground green litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_acid_bg1	tiles	mean acid soluble belowground green litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_water_bg1	tiles	mean water soluble belowground green litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_ethanol_bg1	tiles	mean ethanol soluble belowground green litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_nonsoluble_bg1	tiles	mean non-soluble belowground green litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_humus_1	tiles	mean humus from green carbon	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_acid_ag2	tiles	mean acid soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_water_ag2	tiles	mean water soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_ethanol_ag2	tiles	mean ethanol soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_nonsoluble_ag2	tiles	mean non-soluble aboveground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_acid_bg2	tiles	mean acid soluble belowground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_water_bg2	tiles	mean water soluble belowground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_ethanol_bg2	tiles	mean ethanol soluble belowground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_nonsoluble_bg2	tiles	mean non-soluble belowground woody litter	[mol(C)/m <sup>2</sup> (grid box)]
avg_YCpool_humus_2	tiles	mean humus from woody carbon	[mol(C)/m <sup>2</sup> (grid box)]
Output with Nitrogen Cycle			
Npool_green	tiles	N in green parts of vegetation	[mol(N)/m <sup>2</sup> (canopy)]
Npool_mobile	tiles	N for plant internal retranslocation	[mol(N)/m <sup>2</sup> (canopy)]
Npool_crop_harvest	tiles	N in harvested material from crops	[mol(N)/m <sup>2</sup> (canopy)]
Npool_woods	tiles	N in for structural material of plants	[mol(N)/m <sup>2</sup> (canopy)]
Npool_litter_green_ag	tiles	N in above ground non-woody litter	[mol(N)/m <sup>2</sup> (canopy)]
Npool_litter_wood_ag	tiles	N in above ground woody litter	[mol(N)/m <sup>2</sup> (canopy)]
Npool_litter_wood_bg	tiles	N in below ground woody litter	[mol(N)/m <sup>2</sup> (canopy)]
Npool_litter_green_bg	tiles	N in below ground non-woody litter	[mol(N)/m <sup>2</sup> (canopy)]
Npool_slow	tiles	N in slowly respired soil organic material	[mol(N)/m <sup>2</sup> (canopy)]
sminN_pool	tiles	soil mineral N	[mol(N)/m <sup>2</sup> (canopy)]
NPP_run_mean	tiles	running mean of NPP used for N-fixation	[mol(N)/m <sup>2</sup> (canopy)s]
avg_Npool_green	tiles	as Npool_green, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_Npool_mobile	tiles	as Npool_mobile, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_Npool_crop_harvest	tiles	as Npool_crop_harvest, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_Npool_woods	tiles	as Npool_woods, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]

table continued on next page

Table D.2: Cbalone.year.cyear.nc — continued

Variable Name	dim.	Description	Units
avg_Npool_litter_green_ag	tiles	as Npool_litter_green_ag, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_Npool_litter_wood_ag	tiles	as Npool_litter_wood_ag, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_Npool_litter_wood_bg	tiles	as Npool_litter_wood_bg, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_Npool_litter_green_bg	tiles	as Npool_litter_green_bg, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_Npool_slow	tiles	as Npool_slow, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_SMINN_pool	tiles	as SMINN_pool, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
avg_SMINN_herbivory	tiles	as SMINN_herbivory, relative to grid box area	[mol(N)/m <sup>2</sup> (grid box)]
redFact_Nlimitation	tiles	reduction factor due to N limitation	[ ]
avg_minNflux_litter_green_ag	tiles	mineral N-flux associated with above ground green litter	[mol(N)/m <sup>2</sup> (grid box)s]
avg_minNflux_litter_wood_ag	tiles	mineral N-flux associated with above ground woody litter	[mol(N)/m <sup>2</sup> (grid box)s]
avg_minNflux_litter_wood_bg	tiles	Soil mineral Nitrogen flux associated with above ground woody litter	[mol(N)/m <sup>2</sup> (grid box)s]
avg_minNflux_litter_green_bg	tiles	Soil mineral Nitrogen flux associated with below ground woody litter	[mol(N)/m <sup>2</sup> (grid box)s]
avg_minNflux_slow	tiles	Mineral N-flux from slow N-pool	[mol(N)/m <sup>2</sup> (grid box)s]
avg_Nplant_demand	tiles	N uptake by plants	[mol(N)/m <sup>2</sup> (grid box)s]
avg_Nsoil_demand	tiles	N uptake for litter decomposition	[mol(N)/m <sup>2</sup> (grid box)s]
avg_Ntotal_demand	tiles	N uptake of plants and for litter decomposition	[mol(N)/m <sup>2</sup> (grid box)s]
ndep_to_sminn	tiles	N flux to mineral N pool due to N deposition	[mol(N)/m <sup>2</sup> (canopy)s]
ndep_forc	1	atmospheric N deposition	[kg(N)/m <sup>2</sup> (canopy)s]
nfert_forc_2d	tiles	N fertilizer forcing	[mg(N)/m <sup>2</sup> (cropland)y]
nfert_forc	1	N fertilizer forcing	[mg(N)/m <sup>2</sup> (cropland)y]
avg_nfert_to_sminn	tiles	N flux to mineral N pool due to fertilizers	[mol(N)/m <sup>2</sup> (canopy)s]
box_Npools_total	1	sum of nitrogen from all nitrogen pools	[mol(N)/m <sup>2</sup> (grid box)]
avg_soil_respiration_pot	tiles	soil respiration without N limitation	[mol(N)/m <sup>2</sup> (grid box)s]
box_test_Ncons	1	should be zero if N is conserved	[ ]
avg_nfix_to_sminn	tiles	N flux to mineral N pool due to N fixation (BNF)	[mol(N)/m <sup>2</sup> (grid box)s]
avg_sminn_to_denit	tiles	N flux from mineral N pool due to denitrification	[mol(N)/m <sup>2</sup> (grid box)s]
avg_sminn_leach	tiles	N flux to mineral N pool due to leaching	[mol(N)/m <sup>2</sup> (grid box)s]
avg_N2O_emissions_ExternalN	tiles	N <sub>2</sub> O emissions due to deposition and fixation	[mol(N)/m <sup>2</sup> (grid box)s]
avg_N2O_emissions_mineraliz	tiles	N <sub>2</sub> O emissions due to mineralization	[mikro-mol(N)/m <sup>2</sup> (grid box)s]
avg_N2O_emissions_nfert	tiles	N <sub>2</sub> O emissions due to N fertilizer use	[mol(N)/m <sup>2</sup> (grid box)s]

table continued on next page

Table D.2: Cbalone.syear.cyear.nc — continued

Variable Name	dim.	Description	Units
avg_N2O_emissions_grazing	tiles	N2O emissions from N input by herbivores	[mol(N)/m <sup>2</sup> (grid box)s]
avg_NetEcosyst_N_flux	tiles	Total balance of N gains and losses (positive for ecosystem gain)	[mol(N)/m <sup>2</sup> (grid box)s]
sum_N_pools	tiles		
test_Nconserv	tiles		
test_Ngreen	tiles		
test_Nwoods	tiles		
test_Nlitter_green_ag	tiles		
test_Nlitter_green_bg	tiles		
test_Nlitter_wood_ag	tiles		
test_Nlitter_wood_bg	tiles		
test_Nslow	tiles		
Output with Land Use Change (maps or transitions) - without Anthropogenic Pools			
LCC_flux_box_C2atmos	1	carbon flux to the atmosphere from landcover change	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C2litterGreen Pools	1	carbon flux from green and reserve pool to fast soil pool by landcover change	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C2litterWood Pool	1	carbon flux from wood pool to woody litter pool by landcover change	[mol(C)/m <sup>2</sup> (grid box)s]
Output with Land Use Change (maps or transitions) - with Anthropogenic Pools			
LCC_flux_C2atmos	1	inst. carbon flux to the atmosphere from landcover change	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_box_C2atmos	1	avg flux from landcover change to atm	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_onSite_2_atmos	1	avg flux from onsite pool to atm	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_paper_2_atmos	1	avg flux from paper pool to atm	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_construction_2_1_atmos	1	avg flux from construction pool to atm	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_C_onSite_2_atmos	1	inst. flux from onsite pool to atm	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_paper_2_atmos	1	inst. flux from paper pool to atm	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_construction_2_atmos	1	inst. flux from construction pool to atm	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_box_C_2_onSite	1	avg flux into onsite pool	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_2_paper	1	avg flux into paper pool	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_2_construction	1	avg flux into construction pool	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_C_2_onSite	1	inst. flux into onsite pool	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_2_paper	1	inst. into paper pool	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_2_construction	1	inst. into construction pool	[mol(C)/m <sup>2</sup> (veg)s]
LCC_box_C_onSite	1	avg onSite pool	[mol(C)/m <sup>2</sup> (grid box)]
LCC_box_C_paper	1	avg paper pool	[mol(C)/m <sup>2</sup> (grid box)]
LCC_box_C_construction	1	avg construction pool	[mol(C)/m <sup>2</sup> (grid box)]
Cpool_onSite_LCC	1	carbon left on ground from land use change	[mol(C)/m <sup>2</sup> (veg)]
Cpool_paper_LCC	1	wood carbon in short/intermediate term anthropogenic pool from land use change	[mol(C)/m <sup>2</sup> (veg)]
Cpool_construction_LCC	1	wood carbon in long term anthropogenic pool from land use change	[mol(C)/m <sup>2</sup> (veg)]

*table continued on next page*

Table D.2: Cbalone.year.cyear.nc — continued

Variable Name	dim.	Description	Units
Output with Land Use Transitions - with Anthropogenic Pools			
LCC_flux_box_C_onSite_harvest_2_atmos	1	avg flux from onSite harvest to atm	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_paper_harvest_2_atmos	1	avg flux from paper harvest to atm	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_construction_harvest_2_atmos	1	avg flux from construction harvest to atm	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_C_onSite_harvest_2_atmos	1	flux from short term anthropogenic harvest pool to atm	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_paper_harvest_2_atmos	1	inst. flux from paper harvest to atm	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_construction_harvest_2_atmos	1	inst. flux from construction harvest to atm	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_box_C_2_onSite_harvest	1	avg flux from living plants to short term anthropogenic harvest pool	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_2_paper_harvest	1	avg flux into paper harvest pool	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_box_C_2_construction_harvest	1	avg flux into construction harvest pool	[mol(C)/m <sup>2</sup> (grid box)s]
LCC_flux_C_2_onSite_harvest	1	inst. flux from living plants to short term anthropogenic harvest pool	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_2_paper_harvest	1	inst. flux into paper harvest pool	[mol(C)/m <sup>2</sup> (veg)s]
LCC_flux_C_2_construction_harvest	1	inst. flux into construction harvest pool	[mol(C)/m <sup>2</sup> (veg)s]
LCC_box_C_onSite_harvest	1	avg onSite harvest pool	[mol(C)/m <sup>2</sup> (grid box)]
LCC_box_C_paper_harvest	1	avg paper harvest pool	[mol(C)/m <sup>2</sup> (grid box)]
LCC_box_C_construction_harvest	1	avg construction harvest pool	[mol(C)/m <sup>2</sup> (grid box)]
Cpool_onSite_harvest	1	Carbon left on ground from harvest	[mol(C)/m <sup>2</sup> (veg)]
Cpool_paper_harvest	1	Wood carbon in short/intermediate term anthropogenic pool from harvest	[mol(C)/m <sup>2</sup> (veg)]
Cpool_construction_harvest	1	Wood carbon in long term anthropogenic pool from harvest	[mol(C)/m <sup>2</sup> (veg)]
Output with Land Use Transitions			
TransMtrx_CROP_2_PAST, TransMtrx_PAST_2_CROP, TransMtrx_NATL_2_PAST, TransMtrx_PAST_2_NATL, TransMtrx_CROP_2_NATL, TransMtrx_NATL_2_CROP, TransMtrx_FRST_2_PAST, TransMtrx_PAST_2_FRST, TransMtrx_GRAS_2_PAST, TransMtrx_PAST_2_GRAS, TransMtrx_FRST_2_CROP, TransMtrx_CROP_2_FRST, TransMtrx_GRAS_2_CROP and TransMtrx_CROP_2_GRAS	1	elements of the reduced (3x3, eq. (10.1)) and extended (4x4, eq. (10.6)) transition matrices . Here, only the landcover types crop (CROP), pasture (PAST), and natural vegetation (NATL) appear. Elements with forest (FRST) and grasses (GRASS) are derived.	[ ]
Grass_coverFract_lastYear	1	grassland fraction	[ ]
NatWood_coverFract_lastYear	1	natural non-grassland vegetation fraction	[ ]

table continued on next page

Table D.2: Cbalone.syear.cyear.nc — continued

Variable Name	dim.	Description	Units
Pasture_coverFract_ lastYear	1	pasture fraction	[ ]
Crop_coverFract_lastYear	1	cropland fraction	[ ]
CROP_2_NATL_ignored and PAST_2_NATL_ignored	1	transitions ignored due to inconsis- tencies with the LUH data	[ ]
Box_harvest	1	carbon harvest	[mol(C)/m <sup>2</sup> (grid box)]
Box_flux_harvest	1	harvest flux	[mol(C)/m <sup>2</sup> (grid box)s]
Box_flux_harvest_2atmos	1	CO <sub>2</sub> flux to atmosphere from har- vest	[mol(C)/m <sup>2</sup> (grid box)s]
Output with Land Use Maps - without anthro. Pools - with nitrogen			
LCC_flux_box_N2atmos	1	Nitrogen released to atmosphere by landcover changes	[mol(N)/m <sup>2</sup> (veg)s]
LCC_flux_box_N2litterGreen Pools	1	Nitrogen released from green and re- serve pool to fast soil pool by land- cover change	[mol(N)/m <sup>2</sup> (veg)s]
LCC_flux_box_N2litterWood Pool	1	Nitrogen released from wood pool to woody litter pool by landcover change	[mol(N)/m <sup>2</sup> (veg)s]
LCC_flux_box_N2SMINNpool	1	surplus nitrogen from wood stub- bing that because of higher C/N- ratio of woody litter as compared to living wood, is fastly freed into the soil mineral N pool.	[mol(N)/m <sup>2</sup> (veg)s]
Output with Dynamic Vegetation or Land Use Transitions <i>(for the *-notation see top of table)</i>			
cover_fract	tiles	cover fraction *	[ ]
veg_ratio_max	1	vegetated fraction **	[ ]
Output with Dynamic Vegetation <i>(for the *-notation see top of table)</i>			
cover_fract_pot	tiles	cover fractions if there was only nat- ural vegetation **	[ ]
act_fpc	tiles	actual foliage projective cover ****	[ ]
pot_fpc	tiles	potential foliage projective cover (without disturbances) ****	[ ]
bare_fpc	1	fraction of bare ground	[ ]
desert_fpc	1	desert fraction ***	[ ]
max_green_bio	tiles	maximum value of green biomass within a year	[mol(C)/m <sup>2</sup> (canopy)]
sum_green_bio_memory	1	vegetated fraction (calculated from green biomass) ***	[ ]
Output with Disturbances (Fire or Windthrow)			
fuel	1	fuel available to wildfires	mol(C)/[m <sup>2</sup> (grid box)]
burned_frac	tiles	inst. fraction of burned vegetation	[fraction of vegetated area/day]
damaged_frac	tiles	inst. fraction of damaged (wind- thrown) vegetation (wind throw)	[fraction of vegetated area/day]
carbon_2_GreenLitterPools	1	carbon relocated to the green litter pools due to wind damage	[mol(C)/m <sup>2</sup> (vegetated area)]
carbon_2_WoodLitterPools	1	carbon relocated to the woody litter pools due to wind damage and fire	[mol(C)/m <sup>2</sup> (vegetated area)]
box_burned_frac_avg	tiles	avg fraction of burned vegetation (SPITFIRE: for carbon calcula- tions)	[grid box fraction/s]

table continued on next page

Table D.2: `Cbalone.syear.cyear.nc` — continued

Variable Name	dim.	Description	Units
<code>box_damaged_frac_avg</code>	tiles	avg fraction of damaged (wind-thrown) vegetation	[grid box fraction/s]
<code>box_fire_CO2_flux_2_atm</code>	1	carbon released to the atmosphere by fire	[mol(C)/m <sup>2</sup> (grid box)]
Output with Thonicke Fire scheme (SPITFIRE)			
<code>box_burned_frac_diag_avg</code>	tiles	burnt fraction of grid box including survivors	[ ]
<code>frac_1hr_wood</code>	tiles	fraction of 1hr fuel in wood pool	[ ]
<code>frac_10hr_wood</code>	tiles	fraction of 10hr fuel in wood pool	[ ]
<code>frac_100hr_wood</code>	tiles	fraction of 100hr fuel in wood pool	[ ]
<code>frac_1000hr_wood</code>	tiles	fraction of 1000hr fuel in wood pool	[ ]
<code>NI_acc</code>	1	average Nesterov index	[ ]
Output for debugging			
<code>test_Cconserv</code>	tiles	carbon conservation check	[mol(C)/m <sup>2</sup> (canopy)]
<code>LCC_testCconserv</code>	1	carbon conservation test (should be zero)	[mol(C)/m <sup>2</sup> (grid box)]
<code>LCC_testNconserv</code>	1	nitrogen conservation test (should be zero)	[mol(N)/m <sup>2</sup> (grid box)]
<code>Test_NATL_2_PAST,</code> <code>Test_PAST_2_NATL,</code> <code>Test_CROP_2_NATL,</code> <code>Test_NATL_2_CROP,</code> <code>Test_CROP_2_PAST,</code> <code>Test_PAST_2_CROP,</code> <code>Test_FRST_2_PAST,</code> <code>Test_PAST_2_FRST,</code> <code>Test_GRAS_2_PAST,</code> <code>Test_PAST_2_GRAS,</code> <code>Test_FRST_2_CROP,</code> <code>Test_CROP_2_FRST,</code> <code>Test_GRAS_2_CROP</code>	and	arrays to test the conversion between the different vegetation types (should be zero). CROP: croplands, PAST: pasture, NATL: natural vegetation, FRST: natural non-grasslands, GRASS: natural grasslands.	[ ]
<code>dynveg_testCconserv_1,</code> <code>dynveg_testCconserv_2,</code> <code>dynveg_testNconserv_1</code> <code>dynveg_testNconserv_2</code>	and	arrays to test carbon and nitrogen conversion within <code>mo_dynveg</code> (should be zero)	[ ]

# Appendix E

## Post-processing

### Contents

---

E.1	Monthly means	243
E.2	Post-processing and visualization with <code>visualize_all.ksh</code>	243
E.2.1	Carbon and Nitrogen inventories	244
E.2.2	Vegetation fractions	244
E.2.3	CMOR-like output	244
E.3	Monitoring	251

---

### E.1 Monthly means

The generation of monthly means from the JSBACH output streams is part of the standard post-processing. The post-processing script `expid.post` is created together with the run script from `Create_TASKS.frm` (compare section A.3). Monthly mean albedo (code 13) and faPAR (code 125) are not generated from the corresponding 6-hourly values. Instead, monthly mean albedo is the quotient of monthly mean down-welling and upwelling solar radiation. Monthly faPAR is calculated as quotient of monthly mean absorbed PAR (code 148) and incoming PAR (code 149).

### E.2 Post-processing and visualization with `visualize_all.ksh`

As the name suggests the script

```
mpiesm/contrib/plots/cosmos-asob/visualize_all.ksh
```

is a comprehensive tool for visualization and post-processing. It is designed to analyze the output of the Earth system model MPI-ESM. However, parts of it are also useful for ECHAM or JSBACH data from standalone runs. Detailed instructions on how to run the script are given in the header.

It is possible, to run `visualize_all.ksh` on a local PC. The program checks for new model output and transfers the data from the supercomputer to your local machine. This makes it very efficient for the analysis of experiments already at run-time.

The program comprises three main tasks. It calculates carbon inventories (E.2.1), vegetation and soil cover fractions (E.2.2), and converts some of the output to a CMOR-like format, that allows a direct comparison of your experiment with CMIP6 experiments (E.2.3).

### E.2.1 Carbon and Nitrogen inventories

Carbon inventories of the different compartments of the earth system model (atmosphere, ocean, sediment, vegetation, soil) are calculated in `co2inventory.sh`, which is called from the main script `visualize_all.ksh`. The program has useful comments on how to convert the carbon units used in the different compartments. Time-series are calculated and visualized with `R` (`plot_carbon_inventory`). The graph is especially useful to check carbon conservation within the ESM as a whole. You find it in `plots/invent_expid.pdf`.

Inventories of the different land carbon pools are visualized with `carbon_land.sh` and `plot_land_carbon_pool`. Timeseries of the amount of carbon in living vegetation (`cVeg`), in litter (`cLitter`), in the soil (`cSoil`) and in anthropogenic product pools (`cProduct`) are shown. Analogously, timeseries of the land nitrogen pools are generated with `nitrogen_land.sh` and `plot_land_nitrogen_pool`. The plots can be found in directory `plots`.

### E.2.2 Vegetation fractions

Whether or not cover fractions are calculated `visualize_all.ksh` is controlled by parameter

```
dynveg=yes # Plot vegetation fractions for dynveg runs (yes/no).
```

Nevertheless, the derived cover fractions and time-series are not only interesting for runs with dynamic vegetation but also for runs with anthropogenic land cover change. In contrast to variable `cover_fractions` of the jsbach output stream (compare D.2) the program calculates cover fractions relative to the total grid box. The data files can be found in directory `outdata_land`. The derived cover fractions are calculated in `forestfract.sh`. The script is called from `visualize_all.ksh`, it can however be used as offline tool, as well. Time-series of forest, shrub, natural grass, crops, pastures and desert fractions are visualized with `R`. The graph is archived as `plots/dynveg_expid.pdf`.

### E.2.3 CMOR-like output

The CMOR format (Climate Model Output Rewriter, PCMDI) was designed to facilitate model intercomparison in international projects as CMIP. Data in CMOR format comes with detailed metadata on the models used, the output variables, the performed experiment, on the institution that performed the simulation, and more. Here, we generate “CMOR-like” output, i.e. the output is useful for comparison with ‘real’ CMOR data; however a lot of metadata is missing. Monthly mean JSBACH output and output of selected ECHAM variables are converted to files that have the same structure, variable names and units and the same grid definitions as the data in the CMIP archives (e.g. <https://esgf-data.dkrz.de/search/cmip6-dkrz>). Additionally, time series of global means or global sums are generated for all variables. To enable the generation of CMOR-like output set `cmor=yes` in `visualize_all.ksh`. Particularly variables requested by the LUMIP [84] and C<sup>4</sup>MIP [69] intercomparison projects are provided in this way. For details on the exact methods to calculate the CMOR-output please consult the script `c6_cmor_land` and the corresponding README file `cmor.txt`. A list of the CMOR variables calculated by `c6_cmor_land` is given in Table

E.1. Each variable is stored in a separate netcdf-file whose name is the variable name. You find the output in directory `outdata_land`.

Table E.1: CMOR-like output variables generated by `c6_cmor_land.ksh`. Most variables are needed for CMIP, others are generated for CRESCENDO.

CMOR name	Description	CMIP	CRESC	Comment
<code>albs</code>	surface albedo		x	
<code>alpha</code>	root zone water filling relative to maximum filling			
<code>ares</code>	aerodynamic resistance	x		cannot be calculated from standard output
<code>baresoilFrac</code>	bare soil (desert)	x		only with dynveg
<code>burntArea</code>	burnt area			not inline with CMOR definition
<code>c3PftFrac</code>	C3 plant fraction	x		only with dynveg
<code>c4PftFrac</code>	C4 plant fraction	x		only with dynveg
<code>cLand</code>	Total land carbon	x		$cLand = cVeg + cLitter + cSoil + cProduct$
<code>cLitter</code>	carbon in litter	x	x	only YASSO above ground litter
<code>cLitterCrop</code>	litter carbon of crops			
<code>cLitterGrass</code>	litter carbon of grass	x		
<code>cLitterLut</code>	litter carbon on land use tiles	x		
<code>cLitterPast</code>	litter carbon of pasture			
<code>cLitterShrub</code>	litter carbon of shrubs	x		
<code>cLitterTree</code>	litter carbon of trees	x		
<code>clt</code>	cloud cover	x		
<code>cnc</code>	canopy fraction	x		
<code>co2mass</code>	co2mass	x		
<code>cProduct</code>	carbon in produc pools	x	x	$d/dt(cProduct) = fDeforestToProduct + fHarvestToProduct - fProductDecomp$
<code>cropFrac</code>	crop fraction	x		only with dynveg $cropFrac = cropFracC3 + cropFracC4$
<code>cropFracC3</code>	C3 crop fraction	x		only with dynveg
<code>cropFracC4</code>	C4 crop fraction	x		only with dynveg
<code>cSoil</code>	carbon in soil pools	x	x	including YASSO below ground litter
<code>cSoilCrop</code>	soil carbon of crops			
<code>cSoilFast</code>	carbon in fast soil pools			not inline with CMOR definition
<code>cSoilGrass</code>	soil carbon of grass	x		
<code>cSoilLut</code>	soil carbon on land use tiles	x		
<code>cSoilPast</code>	soil carbon of pasture			

*table continued on next page*

Table E.1: CMOR-like output variables generated by `c6_cmor_land.ksh` — continued

CMOR name	Description	CMIP	CRESC	Comment
<code>cSoilPools</code>	carbon in different soil pools	x		SUM( <code>cSoilPools</code> ) = <code>cSoil</code>
<code>cSoilShrub</code>	soil carbon of shrubs	x		
<code>cSoilSlow</code>	carbon in slow soil pools (= humus pools)			not inline with CMOR definition
<code>cSoilTree</code>	soil carbon of trees	x		
<code>cVeg</code>	total vegetation carbon	x	x	
<code>cVegCrop</code>	vegetation carbon in crops			
<code>cVegGrass</code>	vegetation carbon in grass	x		
<code>cVegLut</code>	vegetation carbon of land use tiles	x		
<code>cVegPast</code>	vegetation carbon in pastures			
<code>cVegShrub</code>	vegetation carbon in shrubs	x		
<code>cVegTree</code>	vegetation carbon in trees	x		
<code>es</code>	bare Soil evaporation	x		
<code>et</code>	evapotranspiration	x		
<code>evapotrans</code>	evapotranspiration		x	same as <code>et</code>
<code>evspsbl</code>	evaporation	x		
<code>fapar</code>	fapar		x	
<code>fBNF</code>	nitrogen fixation	x	x	
<code>fco2antt</code>	carbon flux to the atmosphere due to anthropogenic emissions	x		only with emission driven scenarios
<code>fco2fos</code>	carbon flux to the atmosphere due to fossil fuel emissions	x		only with emission driven scenarios
<code>fco2nat</code>	carbon flux to the atmosphere from natural sources	x		
<code>fDeforestToProduct</code>	carbon flux to product pool due to deforestation	x		
<code>fFire</code>	carbon flux due to fire	x	x	
<code>fGrazing</code>	carbon flux due to grazing	x	x	
<code>fHarvest</code>	carbon flux due to crop harvest	x	x	crop and wood harvest ( $\neq$ <code>fHarvestToAtmos</code> )
<code>fHarvestToProduct</code>	harvested biomass going to product pools	x		
<code>fLuc</code>	carbon flux due to land use change	x	x	flux from the product pools to the atmosphere ( $\neq$ <code>fDeforestToAtmos</code> , which is zero in simulation with anthropogenic C pools)

*table continued on next page*

Table E.1: CMOR-like output variables generated by `c6_cmor_land.ksh` — continued

CMOR name	Description	CMIP	CRESC	Comment
<code>fN2O</code>	N2O flux	x	x	
<code>fNAnthDisturb</code>	nitrogen flux to the atmosphere due to human activities	x		in contrast to the C flux <code>fAnthDisturb</code> the N flux is non zero, as there are no nitrogen product pools, except for the crop harvest pool.
<code>fNdep</code>	nitrogen deposition	x	x	
<code>fNgas</code>	nitrogen lost to the atmosphere	x	x	<code>fNgas = fNgasFire + fNgasNonFire</code>
<code>fNgasFire</code>	nitrogen flux to the atmosphere due to fire	x	x	
<code>fNgasNonFire</code>	nitrogen flux to the atmosphere without fire	x		
<code>fNleach</code>	nitrogen lost by leaching	x	x	
<code>fNloss</code>	nitrogen loss	x	x	<code>fNloss = fNgas + fNleach</code>
<code>fNnetmin</code>	net nitrogen release	x	x	
<code>fNProduct</code>	N flux to product pool	x		
<code>fNup</code>	plant nitrogen uptake	x	x	
<code>fProductDecomp</code>	decomposition of product pools	x		
<code>fracInLut</code>	grid box fraction transferd into LU tile	x		
<code>fracLut</code>	fraction of LUMIP land use tiles	x		
<code>fracOutLut</code>	grid box fraction transferd from LU tile into another tile	x		
<code>gpp</code>	total GPP	x	x	<code>gpp = npp + ra</code>
<code>gppCrop</code>	GPP of crops			
<code>gppGrass</code>	GPP of grass	x		
<code>gppLut</code>	GPP on land use tiles	x		
<code>gppPast</code>	GPP of pastures			
<code>gppShrub</code>	GPP of shrub	x		
<code>gppTree</code>	GPP of trees	x		
<code>grassFrac</code>	natural grass fraction	x		only with dynveg <code>grassFrac = grassFracC3 + grassFracC4</code>
<code>grassFracC3</code>	C3 grass fraction (only natural)	x		only with dynveg
<code>grassFracC4</code>	C4 grass fraction (only natural)	x		only with dynveg

*table continued on next page*

Table E.1: CMOR-like output variables generated by `c6_cmor_land.ksh` — continued

CMOR name	Description	CMIP	CRESC	Comment
<code>grassFracPot</code>	potential grass fraction			
<code>hfls</code>	latent heat flux	x	x	
<code>hfss</code>	sensible heat flux	x		
<code>hurs</code>	2m relative humidity	x		
<code>huss</code>	2m specific humidity	x		
<code>lai</code>	LAI	x		
<code>lai_pft</code>	LAI per PFT		x	also called <code>lai</code>
<code>laiLut</code>	LAI on land use tiles	x		
<code>landCoverFrac</code>	land cover fractions	x	x	only with <code>dynveg</code>
<code>mrro</code>	total runoff	x	x	<code>mrro = mrrob + mrros</code>
<code>mrrob</code>	drainage	x		
<code>mrso</code>	soil moisture content	x	x	
<code>mrsol</code>	water content of soil layer	x		
<code>mrros</code>	surface runoff	x		
<code>mrsos</code>	water content of uppermost 10 cm	x		
<code>msl</code>	soil moisture of each layer		x	same as CMOR var. <code>mrsol</code>
<code>nbp</code>	net biospheric production	x	x	<code>nbp = gpp - ra - rh - fFire - fHarvest - fLuc</code> same as <code>netAtmosLandC02Flux</code>
<code>nep</code>	net ecosystem productivity	x		natural fluxes from the atmosphere to the land <code>nep = gpp - ra - rh - fFire</code>
<code>netAtmosLandC02Flux</code>	net biospheric production	x		<code>netAtmosLandC02Flux = d/dt(cLand)</code> same as <code>nbp</code>
<code>nLand</code>	total land nitrogen	x		<code>nLand = nVeg + nLitter + nSoil + nProduct + nMineral</code>
<code>Nlimitation</code>	nitrogen limitation			
<code>nLitter</code>	nitrogen in litter pools	x	x	only above ground pools; below ground litter is part of soil
<code>nMineral</code>	mineral nitrogen	x		
<code>npp</code>	total NPP	x	x	<code>npp = gpp - ra</code>
<code>nppCrop</code>	NPP of crops			
<code>nppGrass</code>	NPP of grass	x		
<code>nppLut</code>	NPP of land use tiles	x		
<code>nppPast</code>	NPP of pastures			
<code>nppShrub</code>	NPP of shrubs	x		

*table continued on next page*

Table E.1: CMOR-like output variables generated by `c6_cmor_land.ksh` — continued

CMOR name	Description	CMIP	CRESC	Comment
<code>nppTree</code>	NPP of trees	x		
<code>nProduct</code>	nitrogen in product pools	x	x	
<code>nSoil</code>	nitrogen in soil pools	x	x	includes N from below ground litter pools
<code>nSoilFast</code>	nitrogen in below ground litter (fast soil pools)			
<code>nSoilSlow</code>	nitrogen in slow soil pool			
<code>nVeg</code>	nitrogen in vegetation pools	x	x	
<code>nwdFracLut</code>	non woody fraction of land use tile	x		
<code>pastureFrac</code>	pasture fraction	x		<code>pastureFrac = pastureFracC3 + pastureFracC4</code>
<code>pastureFracC3</code>	C3 pasture fraction	x		only with dynveg
<code>pastureFracC4</code>	C4 pasture fraction	x		only with dynveg
<code>pr</code>	precipitation	x	x	
<code>prra</code>	rainfall rate	x		
<code>ra</code>	total autotrophic respiration	x	x	<code>npp = gpp - ra</code>
<code>raCrop</code>	autotrophic respiration of crops			
<code>raGrass</code>	autotrophic respiration of grass	x		
<code>raLut</code>	autotrophic respiration of land use tiles	x		
<code>raPast</code>	autotrophic respiration of pastures			
<code>raShrub</code>	autotrophic respiration of shrubs	x		
<code>raTree</code>	autotrophic respiration of trees	x		
<code>residualFrac</code>	residual fraction (glaciers)	x		<code>vegFrac + baresoilFrac + residualFrac = landFrac</code>
<code>rh</code>	heterotrophic respiration from ALL consumers, including herbivory	x	x	
<code>rhCrop</code>	heterotrophic respiration of crops			
<code>rhGrass</code>	heterotrophic respiration of grass	x		
<code>rhLut</code>	soil heterotrophic respiration of land use tiles	x		herbivory included as in other <code>rh</code> variables
<code>rhPast</code>	heterotrophic respiration of pastures			

*table continued on next page*

Table E.1: CMOR-like output variables generated by `c6_cmor_land.ksh` — continued

CMOR name	Description	CMIP	CRESC	Comment
<code>rhShrub</code>	heterotrophic respiration of shrubs	x		
<code>rhSoil</code>	heterotrophic soil respiration	x		
<code>rhTree</code>	heterotrophic respiration of tree	x		
<code>rlds</code>	downward longwave radiation	x		
<code>rlus</code>	upward longwave radiation	x		
<code>rsds</code>	downward shortwave radiation	x	x	
<code>rsds_nir</code>	surface downwelling near-IR radiation	x		only with LS3M_TYPE=1
<code>rsds_vis</code>	surface downwelling visible radiation	x		only with LS3M_TYPE=1
<code>rss_nir</code>	net surface near-IR radiation	x		
<code>rss_nir_ns</code>	net surface near-IR radiation without snow	x		only with LS3M_TYPE=1
<code>rss_ns</code>	net surface solar radiation without snow	x		only with LS3M_TYPE=1
<code>rss_vis</code>	net surface visible radiation	x		
<code>rss_vis_ns</code>	net surface visible radiation without snow	x		only with LS3M_TYPE=1
<code>rsus</code>	upward shortwave radiation	x		
<code>rzwc</code>	root zone soil moisture	x		
<code>sbl</code>	snow and ice sublimation	x		
<code>sfcWind</code>	10m wind speed	x		
<code>sfls</code>	sensible heat flux		x	same as CMOR var. <code>hfss</code>
<code>shrubFrac</code>	shrub fraction	x		only with dynveg
<code>shrubFracPot</code>	potential shrub fraction			only with dynveg
<code>snc</code>	snow fraction	x		
<code>snm</code>	snow melt	x		
<code>snw</code>	snow amount	x		
<code>snwc</code>	snow on vegetation	x		
<code>tas</code>	2m temperature	x	x	
<code>tr</code>	radiative temperature	x		
<code>tran</code>	transpiration	x	x	
<code>treeFrac</code>	forest fraction	x		only with dynveg
<code>treeFracPot</code>	potential forest fraction			
<code>ts</code>	surface temperature	x		
<code>tsl</code>	soil temperature	x	x	temperature of each soil layer
<code>vegFrac</code>	vegetated fraction	x		only with dynveg <code>vegFrac = c3PftFrac + c4PftFrac</code>

Within CMIP several variables are requested for different vegetation types (e.g. trees, shrubs and (natural) grasses). To represent all vegetation types, we added two more types: crops and pastures. The cover fraction of all vegetation types sum up to the vegetated part of each grid box.

$$\text{vegFrac} = \text{treeFrac} + \text{shrubFrac} + \text{grassFrac} + \text{cropFrac} + \text{pastureFrac}$$

The requested sub-tile variables show the respective quantity valid for the specified vegetation types. The following equations gives the link between the grid box value *var* and the corresponding sub-tile values:

$$\begin{aligned} \text{var} = & \text{varGrass} * \text{grassFrac} + \text{varShrub} * \text{shrubFrac} + \\ & \text{varTree} * \text{treeFrac} + \text{varCrop} * \text{cropFrac} + \text{varPast} * \text{pastureFrac} \end{aligned}$$

LUMIP requests several variables on so called land use tiles. Here the first tile represents primary and secondary land (including bare land), the second tiles crops, and the third tile pastures. The fourth tile is meant for urban land, however in the current JSBACH version urban land is ignored. So the fourth tile of the land use tiles is filled with missing values. The following equation holds for the LUMIP land use tile variables.

$$\text{var} = \text{SUM}(\text{varLut}(:) * \text{fracLut}(:))$$

Here again, *var* represents the overall grid box variable.

### E.3 Monitoring

The monitoring allows to inspect the performance of a running experiment on the basis of time series of a few key variables from the different earth system components. The generated plots are available in html-format accessible via internet. Thus the monitoring is a comfortable tool to share the results with co-workers, remotely. For private experiments, however, the monitoring directory is generally not on a web server. In this case you should open a web-browser on the machine where the data reside.

Currently the monitoring is only available for MPI-ESM simulations with mkexp (compare appendix A), and not with the running environment described here.

## Appendix F

# Generation of maps for leaf and soil albedo

The albedo calculations described in section 4.1 need as basic input maps for vegetation albedo  $\alpha_{leaf}$  and ground albedo  $\alpha_{soil}$ , each separately for the visible and near infrared range. This appendix describes how they can be derived from linear regression of the fraction of absorbed photosynthetically active radiation  $fapar(t)$  against total surface albedo  $\alpha(t)$ . For the current version of JSBACH the albedo maps were generated from published data sets of  $fapar(t)$  and albedo  $\alpha(t)$  derived from measurements with the Moderate Resolution Imaging Spectroradiometer (MODIS) aboard the TERRA satellite in the years 2001-2004.

Here we used the white sky albedo of the visible and the NIR range from the product MOD43C1 [47], which specifies the albedo on a 0.05 degree grid in 16 day periods. White sky albedo (also referred to as bi-hemispherical reflectance) is the reflectance of a surface under diffuse illumination (same radiance for all viewing directions). It is considered to be a good proxy of the daily average albedo, which is the decisive parameter in the context of climate modeling. Only at high latitudes it slightly underestimates albedo as the solar zenith angle is large throughout the whole day, so that the direct beam is reflected more than diffuse light, as for nearly all kind of surfaces the reflectance is higher at large zenith angles than at small zenith angles.

The  $fapar$  data were taken from the product MOD15A2, which provides 8 day  $fapar$  composites on a 1km sinusoidal grid [94]. Both, albedo and  $fapar$ , data sets are remapped to a 0.25 degree grid excluding pixels with snow cover and the  $fapar$  fields are averaged over the 16 day periods of the albedo data set.

The linear regression is done separately for the visible range (incl. UV radiation, 0.3 – 0.7 $\mu m$ ) and the NIR range (0.7 – 3 $\mu m$ ) in the following way:

$$(F.1) \quad \alpha(t) = fcover(t) \alpha_{leaf} + (1 - fcover(t)) \alpha_{soil}$$

Here  $fcover(t)$  is the fraction of the grid box covered by a green canopy. For the visible range it can be assumed that approximately  $fcover(t) = fapar(t)/(1 - \alpha_{leaf})$ . This implies that the difference in the reflectivity of UV radiation and the photosynthetically active radiation (i.e. the visible radiation) has no substantial influence on the total reflectivity and that the radiation reflected at the soil beneath the canopy and penetrating the canopy thereafter is negligible. Thereby one obtains from (F.1)

$$(F.2) \quad \alpha_{vis}(t) = \frac{fapar(t)}{1 - \alpha_{leaf,vis}} \alpha_{leaf,vis} + \left(1 - \frac{fapar(t)}{1 - \alpha_{leaf,vis}}\right) \alpha_{soil,vis}.$$

Taking this as a linear relation  $\alpha_{vis}(fapar) = a * fapar + b$ , the coefficients  $a$  and  $b$  can be determined from the observational data by a linear regression of  $\alpha_{vis}$  onto  $fapar$ . Then  $\alpha_{leaf,vis}$  and  $\alpha_{soil,vis}$  are obtained from  $a$  and  $b$  by

$$(F.3) \quad \alpha_{vis}(fapar = 0) = b = \alpha_{soil,vis}$$

$$\alpha_{vis}(fapar = 1) = a + b$$

$$= \frac{1}{1 - \alpha_{leaf,vis}} \alpha_{leaf,vis} + \left(1 - \frac{1}{1 - \alpha_{leaf,vis}}\right) \alpha_{soil,vis}$$

$$(F.4) \quad \implies \alpha_{leaf,vis} = \frac{a + b}{1 + a}$$

After  $\alpha_{leaf,vis}$  and  $\alpha_{soil,vis}$  have been determined by linear regression,  $\alpha_{leaf,nir}$  and  $\alpha_{soil,nir}$  are calculated in a similar way from

$$(F.5) \quad \alpha_{nir}(t) = \frac{fapar(t)}{1 - \alpha_{leaf,vis}} \alpha_{leaf,nir} + \left(1 - \frac{fapar(t)}{1 - \alpha_{leaf,vis}}\right) \alpha_{soil,nir}.$$

This simple regression scheme is suitable to disentangle the albedo of the soil surface and the albedo of the canopy covered surface for most of the 0.25 degree land grid boxes, but a particular handling has to be applied in the following cases (for the visible range as well as the NIR range):

Variations in  $\alpha(t)$  or  $fapar(t)$  are too small:

If  $\alpha(t)$  varies less than 0.02 in time, it is assumed that no reliable information on the time evolution of surface albedo is contained in this MODIS product as this limit is in the order of the noise introduced by e.g. imperfect atmospheric correction. Therefore,  $\alpha_{soil}$  as well as  $\alpha_{leaf}$  are both set to the time average of  $\alpha(t)$ . If additionally the standard deviation of  $fapar(t)$  is smaller than 0.08, then the low variability of  $\alpha(t)$  does not ensure a similar albedo of the soil surface and the canopy. So,  $\alpha_{soil}$  is set to 'undefined' if the time average of  $fapar(t)$  is larger than 0.5 or  $\alpha_{leaf}$  is set to 'undefined' if the time average of  $fapar(t)$  is smaller than 0.5.

In the cases  $fapar(t)$  varies less than 0.05 in time it is assumed that variations in albedo cannot be related to changes in  $fapar$  due to the noise in this variable. Accordingly,  $\alpha_{soil}$  and  $\alpha_{leaf}$  are handled as in the case with low variability in  $\alpha(t)$  and a small standard deviation in  $fapar(t)$ . Similarly, if  $fapar(t)$  never exceeds 0.2 the value of  $\alpha_{soil}$  is set to the time average of  $\alpha(t)$  and  $\alpha_{leaf}$  is set to undefined. After excluding all these cases the linear regression of albedo onto  $fapar$  is performed for all remaining grid boxes.

The correlation between  $\alpha(t)$  and  $fapar(t)$  is not significant:

At grid boxes with no significant linear correlation between  $\alpha(t)$  and  $fapar(t)$  (the threshold is set to the 80% significance level according to a students t-test) it is not possible to reliably disentangle  $\alpha_{soil}$  and  $\alpha_{leaf}$  by this regression method. So, both variables are again set to the time average of  $\alpha(t)$ .

$fapar(t)$  is too high in the tropics:

Generally, it is challenging to determine the albedo of the soil surface under dense evergreen vegetation. In tropical rain forest areas it is in addition difficult to mask spurious clouds. This results in unrealistically high values of  $\alpha_{soil}$  in some regions. To minimize the consequences of these artefacts a parameter  $\alpha_{dense} = 0.15$  is introduced as a representative value for the albedo of the soil surface under dense tropical forest in

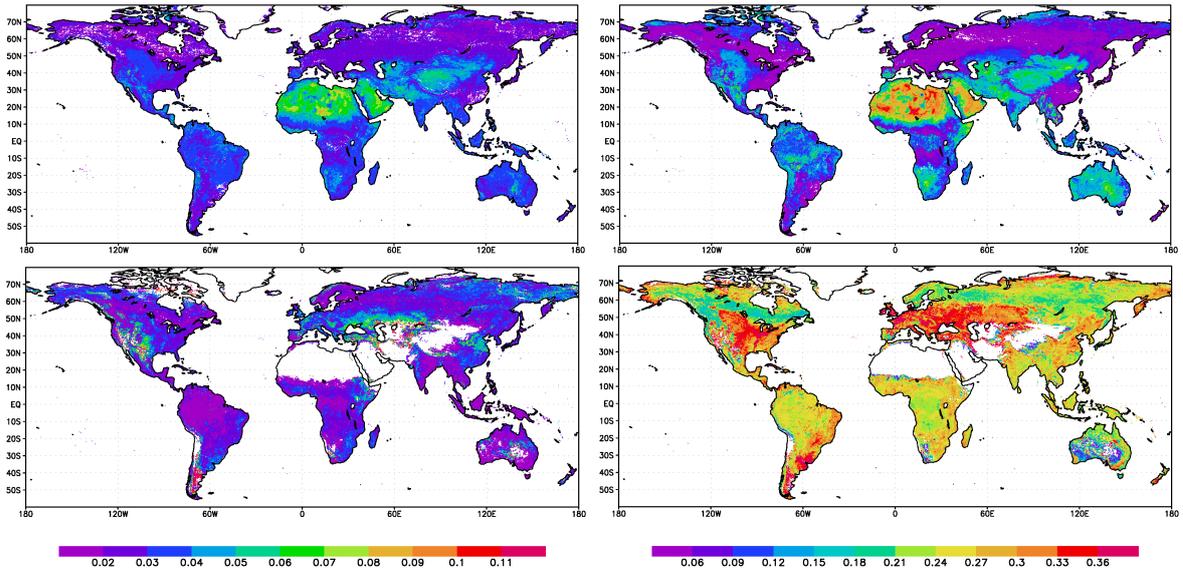


Figure F.1: Maps of surface albedo:  $\alpha_{soil,vis}$  (top left),  $\alpha_{soil,nir}$  (top right),  $\alpha_{leaf,vis}$  (bottom left),  $\alpha_{leaf,nir}$  (bottom right). Grid boxes with undefined value are white.

the NIR range. Within the zone 20°S to 28°N and with the time average of  $fapar(t)$  larger than 0.75 the value of  $\alpha_{soil,nir}$  is adjusted to  $\alpha_{dense}$  by taking the average of the hitherto calculated  $\alpha_{soil,nir}$  and  $\alpha_{dense}$ . Thereafter  $\alpha_{leaf,nir}$  is recomputed accordingly.

$\alpha_{soil}$  or  $\alpha_{leaf}$  are out of range:

Finally, it is checked whether the calculated albedo values are in a reasonable range. If an albedo value calculated by the regression method is out of the range of values specified in table F.1 then it is reset to the upper margin or the lower margin of this range, respectively. Furthermore, albedo values are set to 'undefined' if they are negative or exceed the upper limit in the right column of table F.1 as the outcome of the regression is obviously unrealistic in these cases.

The resulting maps of  $\alpha_{soil,vis}$ ,  $\alpha_{leaf,vis}$ ,  $\alpha_{soil,nir}$ , and  $\alpha_{leaf,nir}$  are shown in Fig. F.1. They are aggregated to the considered JSBACH grid. In the case a JSBACH grid box is covered by less than 5% of its area by valid 0.25 degree grid boxes, the fill value specified in table F.1 is applied. Finally, to compensate for white sky albedo systematically underestimating daily average surface albedo at high latitudes, all four albedo maps are modified: 0.01 is added at latitudes 40° - 50°, 0.02 is added at 50° - 60°, 0.03 is added at 60° - 70°, and 0.04 at latitudes beyond 70°. This is done as part of the generation of the the initial data file `jsbach.nc` (see section C.4).

Table F.1: Limits of albedo values that have been specified by linear regression of  $\alpha(t)$  onto  $fapar(t)$  (left column and middle column), fill value for grid boxes of the JSBACH grid with less than 5% valid values from the regression method (right column).

	range of values	upper limit	fill value
$\alpha_{soil,vis}$	0.02 - 0.52	0.6	0.1
$\alpha_{leaf,vis}$	0.015 - 0.12	0.2	0.04
$\alpha_{soil,nir}$	0.015 - 0.68	0.75	0.2
$\alpha_{leaf,nir}$	0.06 - 0.38	0.45	0.2

## Appendix G

# Equilibration of carbon, nitrogen, and vegetation distribution

A crucial issue for climate simulations that include components with long internal time scales is finding an equilibrium state to have well defined conditions for starting further simulations. As turnover times in soils are 100 years or longer [77], several hundred simulation years are required to equilibrate the soil carbon pools. The equilibration time gets even longer, when interactions with the nitrogen cycle and/or dynamical changes in biogeography come into play. For example, a change from one to another forest type might span only several decades but the resulting drift in carbon and nitrogen storages might last much longer. For this reason a 'spinup' run with ECHAM/JSBACH including dynamic vegetation (DYNVEG) might easily take several thousand years until equilibrium is reached.

To speedup the simulation time needed for equilibration, JSBACH provides two tools. The first is the CBALONE driver (appendix A.4). It uses a certain set of forcing data from previous JSBACH simulations to run only those submodels of JSBACH that have long internal time scales (carbon cycle, nitrogen cycle, dynamic vegetation). The second tool is the script `equilibrate_humus.ksh` to obtain an equilibrium for simulations with Yasso (see section 6.4). While the CBALONE driver obtains the speedup from running only part of the model, the equilibration script achieves the speedup by analytically predicting the equilibrium values of the Yasso pools with the longest time scales (humus pools) from their transient behaviour in short preparatory simulations.

The following instructions describe step-by-step how to equilibrate carbon pools, nitrogen pools, and vegetation distribution using `equilibrate_humus.ksh`. The script can be used with all JSBACH setups as long as Yasso is active. The equilibration is illustrated by example of a JSBACH standalone simulation (compare Fig. G.1).

1. Create a runscript for your spinup simulation (see e.g. A.3). This runscript should use cyclic climate forcing. JSBACH runscripts provide a corresponding flag (`cyclic_forcing`) and the variables `climyear1` and `climyear2` should be set such that the first decades of the climate forcing time series are used (e.g. 1901-1930). In addition, the planetary orbit should be kept constant (namelist flag `lyr_perp`), which can be achieved by setting `piControl=.TRUE..` Also other transient forcings like the concentrations of greenhouse gases (in particular CO<sub>2</sub>) and anthropogenic land use change should be kept constant or switched off.
2. In principle, a spinup can either start from scratch, or, if available, from equilibrium pools (compare section C.6), or from restart files resulting from a previous similar simulation.
3. Depending on the selected settings (DYNVEG, nitrogen, ...), the spinup simulation needs

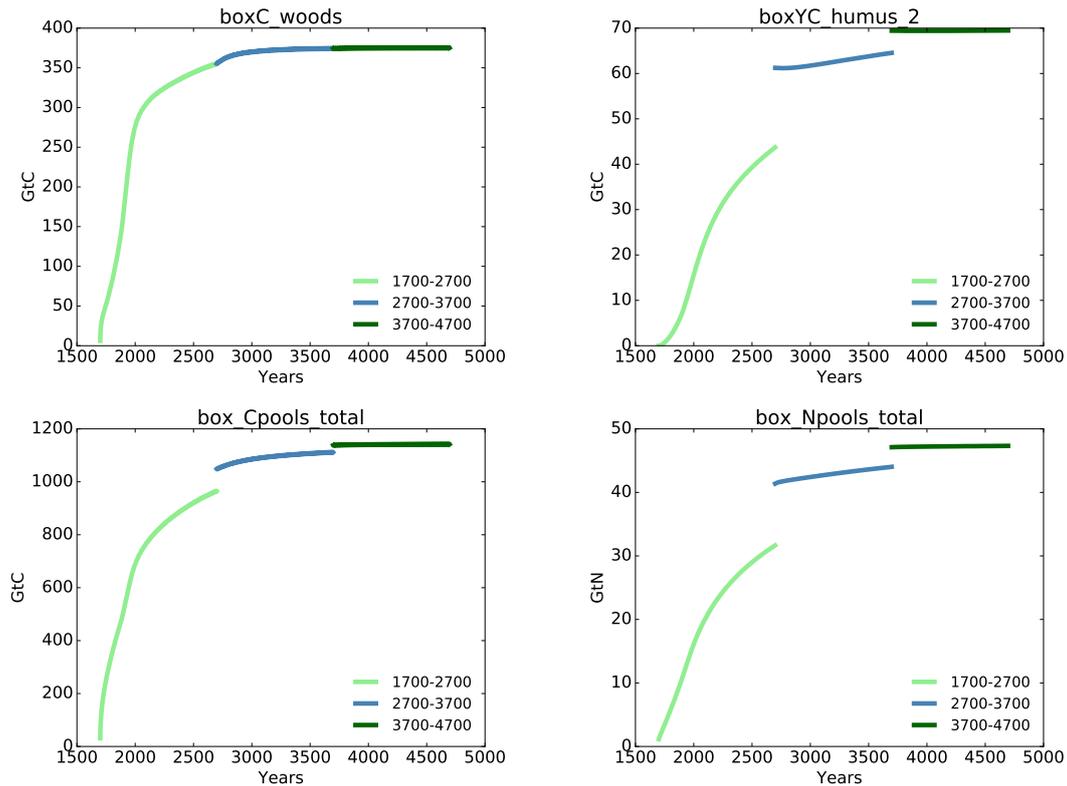


Figure G.1: Temporal development of selected global sums of carbon and nitrogen pools for an example equilibration using JSBACH with nitrogen but without dynamically changing vegetation distribution (namelist `use_dynveg=.FALSE.`). According to the stepwise instructions, a simulation was conducted for 1000 years (step 3), until the non-humus carbon pools indicated signs of saturation, demonstrated here by example of `boxC_woods`. Subsequently, the equilibration script was applied (step 4) to predict equilibrium values for the humus pools (e.g. `boxYC_humus_2`). Following step 5, the simulation was carried on using the updated restart files for another 1000 years (2700-3700). Since the carbon pools still increased in this second simulation, the equilibration script was applied a second time, followed by another 1000 years of simulation. The plots of total carbon (`box_Cpools_total`) and total nitrogen (`box_Npools_total`) demonstrate that a global equilibrium was reached after the second application of the equilibration script. However, this does not exclude that regionally equilibrium has not been fully reached, which also needs to be checked.

to be run for several hundred or thousand years until all carbon pools except the two humus pools indicate saturation. When simulating with dynamically changing vegetation distribution (namelist `use_dynveg=.TRUE.`) also the vegetation distribution should show approach to equilibrium (check the variable `cover_fract` of the `jsbach` stream).

4. Once these pools have reached sufficiently stable values (see e.g. `cWood` in Fig. G.1 after 1000 simulation years), the equilibration script `equilibrate_humus.ksh` can be applied. The script uses the two FORTRAN programs `unpack.f90` and `pack.f90`, also located in the `contrib` directory. The programs need to be compiled before application, instructions are given in the code preambles. The equilibration script reads in- and out-fluxes of the humus pools from restart files representing the last climate forcing cycle of the spinup run. From the imbalance of the in- and out-fluxes the script predicts equilibrium values for the humus pools and overwrites their values in the last of the specified restart files. The corresponding nitrogen pools are adjusted according to their given C:N ratios (compare section 7.2).

5. Subsequently, the JSBACH spinup should be carried on. Since the equilibration script works directly on the restart file, the current runscript can simply be used to continue the spinup simulation by increasing the final simulation date (`final_date`). Depending on the selected settings, the simulation should be run for another several hundred or thousand years. It might be beneficial to apply `equilibrate_humus.ksh` once again, as illustrated in Fig. G.1.
6. When a sufficiently stable equilibrium is reached, the restart files can be used for production runs.

## Appendix H

# Regional and site-level Simulations

In addition to its main application for global simulations, JSBACH (and also CBALONE) can be run for pre-defined sets of grid boxes that may form a connected region (e.g. Europe) or may be scattered around the world. Simulations for such unconnected sets of grid boxes are particularly interesting to compare simulation results with single or multi site measurements so that such simulations are also called 'site-level' simulations. The only technical requirement is that the considered set of grid boxes has an even number of latitudes; this has to do with the way grids are managed in the ECHAM6 infra-structure, which is also used in JSBACH standalone simulations. This section provides an overview on the scripts in the JSBACH software package supporting the preparation and running of such local simulations.

### Scripts for regional and site-level simulations

To conduct JSBACH simulations, the gridded input files, i.e. the jsbach initial file `jsbach.nc` and the climate forcing data file `climate.nc`, must be prepared such that they contain exactly those grid boxes where simulations shall be performed. Considering a certain lat-lon region, one can use the script `mpiesm/contrib/prepare_jsbach_site.ksh/` to prepare suitable input data by extracting the relevant information from the standard global JSBACH input files and then run JSBACH as usual (see appendix A.3).

To perform site level simulations one can use either of the two run scripts

```
run_jsbach-site.ksh or run_ISLv5.c1-rev8047.ksh
```

found in the directory

```
mpiesm/contrib/jsbach-site/
```

Besides preparing the gridded input files for a defined region or a list of grid points, these scripts execute a chain of simulations using CBALONE and JSBACH followed by some post-processing. Concerning the simulations, the first aim is to achieve an equilibrium state for carbon, nitrogen and/or the distribution of vegetation, using the input climate cyclically over and over again. Then this equilibrium state is used as starting point to perform a production run. The approach is similar to the example of a global equilibration given in Appendix G. The scripts are prepared for single site simulations, but for the above mentioned reasons they use input data files with two grid-points. The scripts can be used as templates for running JSBACH also on larger subsections of the global grid. More information is found in the script headers and in the file `README_jsbach-site` also located in the above mentioned directory.

### DICE - An example for a single site simulation

As an example for a site-level simulation the **DI**urnal land/atmosphere **C**oupling **E**xperiment in Kansas is considered. The files specific to this example case are found in the directory `mpiesm/contrib/jsbach-site/DICE/`

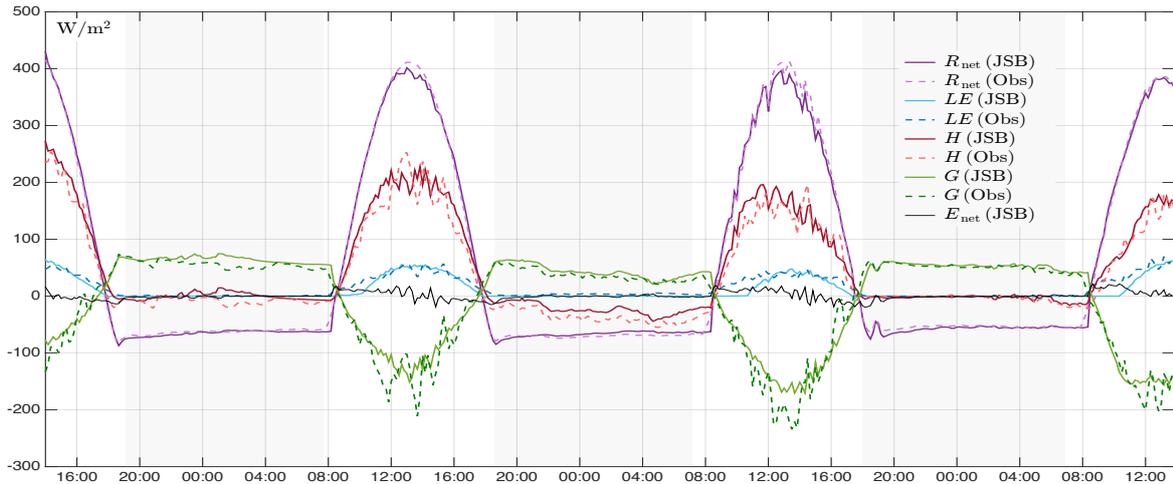


Figure H.1: Representation of the energy balance for the **DI**urnal land/atmosphere **C**oupling **E**xperiment in Kansas for the period 23rd to 26th October 1999 - Comparison between measured data and results obtained with JSBACH-SkIn<sup>+</sup> standalone (see [64]).

DICE provides forcing data (temperature, wind, specific humidity, precipitation and radiation) for a 10-year period (1990 to 1999) that is used here to obtain an equilibrium in soil temperature and soil moisture, and detailed verification data from a 3-day experiment (23rd to 26th October 1999), all in the form of time series for sensible and latent heat flux, and surface temperature. The data are open access and can be found at <http://appconv.metoffice.com/dice/dice.html>. The measurement height for air temperature and specific air humidity is 2 m, while for wind speed the height is 10 m.

An additional preparatory task is to convert the DICE forcing data (`DICE.LSM.screen.nc` and `DICE.LSM.spinup.nc`) into a format suitable for JSBACH (e.g. by including a second dummy grid point; see above). For this purpose also some meta data of the netcdf-files have to be modified: Since CDO commands are not suitable in this case, for this conversion the Matlab program `forcing_dice2jsb.m` is provided in the subfolder `Forcing`. It reads the DICE data files and creates a new netcdf-file compatible with the JSBACH specifications. Besides, it also contains a dynamic method which interpolates or averages the data according to different time steps, if requested. Additionally, the initial file `jsbach.nc` (see appendix C.4) needs to be prepared. For this purpose the Matlab program `jsbach_nc_new.m` is provided in the DICE subfolder `InitialFile`.

To run JSBACH for the DICE case, the run script `run_jsbach-site.dice.ksh` has been prepared (see DICE folder). It mainly creates the `namelist` and specifies the start and end date of the simulation. This script was derived from the above-mentioned run script `run_jsbach-site.ksh` keeping only those parts necessary to run a DICE simulation and adding some DICE-specific settings. A special feature of this simulation setup is the sequence of the initial 10 year equilibration run followed by the 3-day production simulation, that because of the available forcing data cannot start as usual at the end of a year, but must start not only at a certain day in the year but also during the course of a day (7 pm), which poses a technical challenge. This is solved in the run script by adjusting the time stamp `vtime` and `nstep` of the restart file written at the end of equilibration run so that it is suitable to start the 3-day production simulation. The results of the simulation are finally found in the file `jsbach.10min.nc`. The run script defines which output fields it contains. Fig. H.1 shows as an example the simulated components of the surface energy balance together with the observation data for those three days in October 1999.

# Appendix I

## Some Abbreviations

AMIP	Atmospheric Model Intercomparison Projekt
AMIP run	An atmosphere-only simulation with prescribed ocean surface temperatures provided by the AMIP
C	Carbon
C <sup>4</sup> MIP	Coupled Climate-Carbon Cycle Model Intercomparison Project
CMIP	Coupled Model Intercomparison Project
CRESCENDO	European project on Coordinated Research in Earth Systems and Climate Experiments, Knowledge, Research and Outreach
DGVM	Dynamic Global Vegetation Model
DKRZ	Deutsches Klimarechenzentrum (the German climate computing center)
ECHAM	The MPI-M Atmosphere Model
fAPAR	fraction of Absorbed PAR
FAO	Food and Agriculture Organization of the United nations
FPC	Foliage Projective Cover
GPP	Gross Primary Productivity
GRIB	GRIdded Binary – a digital format for meteorological data
HAMOCC	Hamburg Ocean Carbon Cycle model
Holocene	MPI internal project for simulation the holocene climate development
HD	Hydrological Discharge model
IPCC	Intergovernmental Panel on Climate Change
JSBACH	Jena Scheme for Biosphere Atmosphere Coupling in Hamburg
LAI	Leaf Area Index
LCC	Land Cover Change
LUC	Land Use Change
LUH	New Hampshire Land-use Harmonization Project [65]
LUMIP	Land Use Modelling Intercomparison Project
MPI	Max Planck Institute (for meteorology)
MPI-ESM	The Max Planck Institute Earth System model comprising ECHAM, JSBACH, MPIOM and HAMOCC
MPIOM	Max Planck Institute Ocean Model
N	Nitrogen
NetCDF	A self-explaining format for climate data
NPP	Net Primary Productivity
PalMod	The German paleo-modelling initiative
PAR	Photosynthetically Active Radiation (400-700nm)
PFT	Plant Functional Type
SLA	Specific leaf area

# Bibliography

- [1] A. Anav, G. Murray-Tortarolo, P. Friedlingstein, S. Sitch, S. Piao and Z. Zhu, *Evaluation of Land Surface Models in Reproducing Satellite Derived Leaf Area Index over the High-Latitude Northern Hemisphere. Part II: Earth System Models*, Remote Sens. 5 (2013) 3637-3661; doi:10.3390/rs5083637.
- [2] R. Asselin, *Frequency filter for time integrations*, Monthly Weather Review 100 (1972) 487-490.
- [3] S. Bathiany, M. Claussen, V. Brovkin, T. Raddatz and V. Gayler, *Combined biogeophysical and biogeochemical effects of large-scale forest cover changes in the MPI earth system model* Biogeosciences 7 (2010) 1383-1399, doi:10.5194/bg-7-1383-2010.
- [4] F.A. Bazzaz and J. Grace (eds.), *Plant resource allocation* (Academic, San Diego, 1997).
- [5] K. Böttcher, T. Markkanen, T. Thum, T. Aalto, M. Aurela, C. Reick, P. Kolari, A. Arslan and J. Pulliainen, *Evaluating biosphere model estimates of the start of the vegetation active season in boreal forests by satellite observations*, Remote Sensing 8 (2016) 580, doi:10.3390/rs8070580.
- [6] M.J. Begon, J.L. Harper and C.R. Townsend, *Ecology – individuals, populations, and communities* (Blackwell Science, Oxford, UK, 1999), 3<sup>rd</sup> edition.
- [7] J. Beringer, A. H. Lynch, F. S. Chapin III, M. Mack and G. B. Bonan, *The representation of Arctic soils in the land surface model: The importance of mosses*, J. Climate 14 (2001) 3324-3335.
- [8] C.J. Bernacchi, E.L. Singaas, C. Pimentel, A.R. Portis Jr. and S.P. Long, *Improved temperature response functions for models of Rubisco-limited photosynthesis*, Plant, Cell & Environment 24 (2001) 253-259.
- [9] M.J. Best, A. Beljaars, J. Polcher and P. Viterbo, *A Proposed Structure for Coupling Tiled Surfaces with the Planetary Boundary Layer*, J. Hydrometeorology 5 (2004) 1271-1278, doi:10.1175/JHM-382.1.
- [10] W.J. Bond, *What Limits Trees in C<sub>4</sub> Grasslands and Savannas?*, Annual Review of Ecology, Evolution, and Systematics, 39 (2008) 64159, doi:10.1146/annurev.ecolsys.39.110707.173411.
- [11] A. Botta, N. Viovy, P. Ciais, P. Friedlingstein and P. Monfray, *A global prognostic scheme of leaf onset using satellite data*, Global Change Biol. 6 (2000) 709-725.
- [12] C.C. Bresson, A.S. Kowalski, A. Kremer and S. Delzon, *Evidence of altitudinal increase of photosynthetic capacity: gas exchange measurements at ambient and constant CO<sub>2</sub> partial pressures*, Annals of Forest Science 66 (2009) 505.
- [13] S. Brinkop and E. Roeckner, *Sensitivity of a general circulation model to parameteri-*

- zations of cloud-turbulence interactions in the atmospheric boundary layer*, *Tellus* 47A (1995) 197-220.
- [14] A. Brooks and G. D. Farquhar, *Effect of temperature on the CO<sub>2</sub>/O<sub>2</sub> specificity of ribulose-1,5-bisphosphate carboxylase/oxygenase and the rate of respiration in the light*, *Planta* 165 (1985) 397-406.
- [15] V. Brovkin, T. Raddatz, C.H. Reick, M. Claussen, V. Gayler, *Global biogeophysical interactions between forest and climate*, *Geophysical Research Letters* 36 (2009) L07405, doi:10.1029/2009GL037543.
- [16] S. von Caemmerer, *Biochemical models of leaf photosynthesis*, (CSIRO, Collingwood, Australia, 2000).
- [17] E.A. Catchpole, W.R. Catchpole, N.R. Viney, W.L. McCaw and J.B. Marsden-Smedley, *Estimating fuel response time and predicting fuel moisture content from field data*, *International Journal of Wildland Fire* 10 (2001) 215-222.
- [18] J. Cebrian, *Patterns in the fate of production in plant communities*, *American Naturalist* 154 (1999) 449-468.
- [19] J.Q. Chambers, E.S. Tribuzy, L.C. Toledo, B.F. Crispim, N. Higuchi, J. dos Santos, A.C. Araújo, B. Kruijt, A.D. Nobre and S.E. Trumbore, *Respiration from a tropical forest ecosystem: partitioning of sources and low carbon use efficiency*, *Ecological Applications* 14 (2004) 7288.
- [20] F.S. Chapin, A.J. Bloom, C.B. Field and R.H. Waring, *Plant responses to multiple environmental factors*, *BioScience* 37 (1987) 49-57.
- [21] F.S. Chapin, E.-D. Schulze and H.A. Mooney, *The ecology and economy of storage in plants*, *Annu. Rev. Ecol. Syst.* 21 (1990) 423-47.
- [22] N.P. Cheney, J.S. Gould and W.R. Catchpole, *Prediction of fire spread in grasslands*, *International Journal of Wildland Fire* 8 (1998) 1-13.
- [23] R.B. Clapp and G.M. Hornberger, *Empirical Equations for Some Soil Hydraulic Properties*, *Water Resources Research* 14 (1978) 601-604.
- [24] M. Claussen, (1995), *Flux aggregation at large scale: on the limits of validity of the concept of blending height*, *Journal of Hydrology* 166 (1995) 371-382.
- [25] C.C. Cleveland, A.R. Townsend, D.S. Schimel, H. Fisher, R.W. Howarth, L.O. Hedin, S.S. Perakis, E.F. Latty, J.C. von Fischer, A. Elseroad and M.F. Wasson, *Global patterns of terrestrial biological nitrogen (N<sub>2</sub>) fixation in natural ecosystems*, *Global Biogeochemical Cycles* 13 (1999) 623-645.
- [26] D.C. Coleman, *A review of root production processes and their influence on soil biota in terrestrial ecosystems*. In: *The Role of Terrestrial and Aquatic Organisms in Decomposition Processes*, edited by J.M. Anderson and A. Macfadyem, pp. 417-434 (Blackwell Scientific, Oxford, 1976).
- [27] G.J. Collatz, J.T. Ball, C. Grivet and J.A. Berry, *Physiological and environmental regulation of stomatal conductance, photosynthesis and transpiration: a model that includes a laminar boundary layer*, *Agricultural and Forest Meteorology* 54 (1991) 107-136.
- [28] G.J. Collatz, M. Ribas-Carbo and J.A. Berry, *Coupled photosynthesis-stomatal conductance model for leaves of C<sub>4</sub> plants*, *Aust. J. Plant Physiology* 19 (1992) 519-38.

- [29] R.T. Conant, *Grazer-dominated Ecosystems*, Encyclopedia of Life Sciences, Wiley, 2003.
- [30] J.R. Crush, G.C. Waghorn and M.P. Rolston, *Greenhouse gas emissions from pasture and arable crops grown on a Kairanga soil in the Manawatu, North Island, New Zealand*, New Zealand J. Agri. Res., 35 (1992) 253-257.
- [31] C. Dahlke, A. Loew and C. Reick, 2012: *Assessing the seasonality of multi-source fA-  
PAR time-series*, poster, [www.mpimet.mpg.de/fileadmin/land/trs/ILEAPS\\_Dahlke.pdf](http://www.mpimet.mpg.de/fileadmin/land/trs/ILEAPS_Dahlke.pdf).
- [32] D. Dalmonech, S. Zaehle, G.J. Schürmann, V. Brovkin, C. Reick and R. Schnur, *Separation of the Effects of Land and Climate Model Errors on Simulated Contemporary Land Carbon Cycle Trends in the MPI Earth System Model version 1*, J. Climate 28 (2014) 272-291, doi: 10.1175/JCLI-D-13-00593.1.
- [33] D. Dalmonech and S. Zaehle, *Towards a more objective evaluation of modelled land-carbon trends using atmospheric CO<sub>2</sub> and satellite-based vegetation activity observations*, Biogeosciences 10 (2013) 4189-4210, doi:10.5194/bg-10-4189-2013.
- [34] R.E. Dickinson, *Land surface processes and climate — Surface albedos and energy balance*. Advances in Geophysics 25 (1983) 305-353.
- [35] R.E. Dickinson, A. Henderson-Sellers, P.J. Kennedy, M.F. Wilson, *Biosphere-Atmosphere Transfer Scheme (BATS) for the NCAR Community Climate Model*, NCAR Technical Note (1986), NCAR/TN-275+STR, doi:10.5065/D67W6959.
- [36] M. Disse, *Modellierung der Verdunstung und der Grundwasserneubildung in ebenen Einzugsgebieten*, Mitteilungen des Inst. f. Hydrologie u. Wasserwirtschaft d. Universität Karlsruhe 53 (1995) 95-107.
- [37] L. Dümenil and E. Todini, *A rainfall-runoff scheme for use in the Hamburg climate model*. In: *Advances in Theoretical Hydrology, A Tribute to James Dooge*, edited by J.O. Kane, European Geophysical Society Series on Hydrological Sciences, Elsevier (1992) 129-157.
- [38] J.R. Ehleringer and T.E. Cerling, *Photosynthetic pathways and climate*. In: *Global biogeochemical cycles in the climate system*, edited by E.-D. Schulze et al., Academic Press (2001) 267-277.
- [39] H.S. Eggleston, L. Buendia, K. Miwa, T. Ngara and K. Tanabe (eds.), *N<sub>2</sub>O emissions from managed soils and CO<sub>2</sub> emissions from lime and urea application*. In: *IPCC guidelines for national greenhouse gas inventories*, vol. 4, ch. 11. (Hayama, Japan: IGES).
- [40] FAO/Unesco, *Soil maps of the world*, Vols. 1-10, Unesco, Paris.
- [41] G.D. Farquhar, S. Caemmerer and J.A. Berry, *A biochemical model of photosynthesis in leaves of C<sub>3</sub> species*, Planta 149 (1980) 78-90.
- [42] G.D. Farquhar, *Models relating subcellular effects of temperature to whole plant responses*. In: *Plants and Temperature*, ed. by S.P. Long and F.I. Woodward, Biologists Limited, Cambridge (1988) 395-409.
- [43] G.D. Farquhar, *Models of integrated photosynthesis of cells and leaves*, Phil. Trans. Roy. Soc. London 323 (1989) 357-367.
- [44] S. Fatichi, S. Leuzinger and C. Körner, *Moving beyond photosynthesis: From carbon source to sink-driven vegetation modeling*, New Phytologist 201 (2014) 1086-1095.

- [45] P. Friedlingstein, P. Cox, R. Betts, L. Bopp L., W. Von Bloh, V. Brovkin, P. Cadule, S. Doney, M. Eby, I. Fung, G. Bala, J. John, C. Jones, F. Joos, T. Kato, M. Kawamiya, W. Knorr, K. Lindsay, H. D. Matthews, T. Raddatz, P. Rayner, C. Reick, E. Roeckner, K.-G. Schnitzler, R. Schnur, K. Strassmann, A. J. Weaver, C. Yoshikawa and N. Zeng, *Climate-carbon cycle feedback analysis: Results from the C4MIP model intercomparison*, Journal of Climate 19 (2006) 3337-3353, doi: 10.1175/JCLI3800.1.
- [46] J.N. Galloway, F.J. Dentener, D.G. Capone, E.W. Boyer, R.W. Howarth, S.P. Seitzinger, G.P. Asner, C.C. Cleveland, P.A. Green, E.A. Holland, D.M. Karl, A.F. Michaels, J.H. Porter, A.R. Townsend, C.J. Vöosmarty, *Nitrogen cycles: Past, present, and future*, Biogeochemistry 70 (2004) 153-226.
- [47] F. Gao, C. Schaaf, A.H. Strahler, A. Roesch, W. Lucht, R. Dickinson, *MODIS bidirectional reflectance distribution function and albedo climate modeling grid products and the variability of albedo for major global vegetation types*, Journal of Geophysical Research 110 (2005) D01104, doi:10.1029/2004JD005190.
- [48] M.P. Gildea and B. Moore, *FAOSOL – A global soils archive*, Complex Systems Research Center, University of New Hampshire, Durham, New Hampshire, (1986), unpublished data tape and documentation.
- [49] M. Giorgetta, J. Jungclaus, C. Reick, S. Legutke, J. Bader, M. Bttinger, V. Brovkin, T. Crueger, M. Esch, K. Fieg, K. Glushak, V. Gayler, H. Haak, H.-D. Hollweg, T. Ilyina, S. Kinne, L. Kornblueh, D. Matei, T. Mauritsen, U. Mikolajewicz, W. Mueller, D. Notz, F. Pithan, T. Raddatz, S. Rast, R. Redler, E. Roeckner, H. Schmidt, R. Schnur, J. Segschneider, K. Six, M. Stockhause, C. Timmreck, J. Wegner, H. Widmann, K.-H. Wieners, M. Claussen, J. Marotzke and B. Stevens, *Climate and carbon cycle changes from 1850 to 2100 in MPI-ESM simulations for the coupled model intercomparison project phase 5*, JAMES 5 (2013) 572-597, doi:10.1002/jame.20038.
- [50] D.S. Goll, V. Brovkin, B.R. Parida, C.H. Reick, J. Kattge, P.B. Reich, P.M. van Bodegom and Ü. Niinemets, *Nutrient limitation reduces land carbon uptake in simulations with a model of combined carbon, nitrogen and phosphorus cycling*, Biogeosciences 9 (2012) 3547-3569, doi:10.5194/bg-9-3547-2012.
- [51] D.S. Goll, *The influence of phosphorus cycling and temperature acclimation of photosynthesis on the land carbon cycle*, Reports on Earth System Science 126 (MPI for Meteorology, Hamburg, 2013), ISSN 1614-1199.
- [52] D.S. Goll, V. Brovkin, J. Liski, T. Raddatz, T. Thum and K.E.O. Todd-Brown, *Strong dependence of CO<sub>2</sub> emissions from anthropogenic land cover change on initial land cover and soil carbon parametrization*, Global Biogeochemical Cycles 29 (2015) 1511-1523.
- [53] D.S. Goll, A.J. Winkler, T. Raddatz, N. Dong, I.C. Prentice, P. Ciais and V. Brovkin, *Carbon-nitrogen interactions in idealized simulations with JSBACH (version 3.10)*, Geoscientific Model Development 10 (2017) 2009-2030, doi:10.5194/gmd-10-2009-2017.
- [54] *The ECHAM 3 Atmospheric General Circulation Model*, Technical Report 6 (Deutsches Klimarechenzentrum, Hamburg, 1994).
- [55] M.A. Giorgetta, E. Roeckner, T. Mauritsen, J. Bader, T. Crueger, M. Esch, S. Rast, L. Kornblueh, H. Schmidt, S. Kinne, C. Hohenegger, B. Möbis, T. Krismer, K.H. Wieners, B. Stevens, *The atmospheric general circulation model ECHAM6 – Model description*, Berichte zur Erdsystemforschung 135, (MPI for Meteorology, Hamburg, 2013).

- [56] S. Hagemann and L. Dümenil, *A parameterization of the lateral waterflow for the global scale*, *Climate Dynamics* 14 (1998) 17-31.
- [57] S. Hagemann and L. Dümenil, *Documentation for the Hydrological Discharge Model*, Technical Report 17 (Deutsches Klimarechenzentrum, Hamburg, 1998).
- [58] S. Hagemann, M. Botzet, L. Dümenil and B. Machenhauer, *Derivation of global GCM boundary conditions from 1 km land use satellite data*, MPI Report 289 (MPI for Meteorology, Hamburg, 1999).
- [59] S. Hagemann and L. Dümenil Gates, *Validation of the hydrological cycle of ECMWF and NCEP reanalyses using the MPI hydrological discharge model*, *J. Geophys. Res.* 106 (2001) 1503-1510.
- [60] S. Hagemann, *An improved land surface parameter dataset for global and regional climate models*, MPI Report 336 (MPI for Meteorology, Hamburg, 2002).
- [61] S. Hagemann and T. Stacke, *Impact of the soil hydrology scheme on simulated soil moisture memory*, *Climate Dynamics* 44 (2014) 1731-1750, doi:10.1007/s00382-014-2221-6.
- [62] J.C. Halliday, K.R. Tate, R.E. McMurtrie and N.A. Scott, *Mechanisms for changes in soil carbon storage with pasture to Pinus radiata land-use change*, *Global Change Biology* 9 (2003) 1294-1308.
- [63] S.P. Harrison, I.C. Prentice, D. Barboni, K.E. Kohfeld, J. Ni and J.-P. Sutra, *Ecophysiological and bioclimatic foundations for a global plant functional classification*, *Journal of Vegetation Science* 21 (2010) 300-317, doi:10.1111/j.1654-1103.2009.01144.x.
- [64] M. Heidkamp, A. Chlond and F. Ament, *Closing the energy balance using a canopy heat capacity and storage concept – a physically based approach for the land component JS-BACHv3.11*, *Geoscientific Model Development* 11 (2018) 3465-3479, doi:10.5194/gmd-11-3465-2018.
- [65] G.C. Hurtt, S.E. Frohking, M.G. Fearon, B. Moore, E. Shevliakova, S. Malyshev, S.W. Pacala and R.A. Houghton, *The underpinnings of land-use history: Three centuries of global gridded land-use transitions, wood harvest activity, and resulting secondary lands*, *Global Change Biology* 12 (2006) 1208-1229.
- [66] G.C. Hurtt, L.P. Chini, S. Frohking, R.A. Betts, J. Feddema, G. Fischer, J.P. Fisk, K. Hibbard, R.A. Houghton, A. Janetos, C.D. Jones, G. Kindermann, T. Kinoshita, K. Klein-Goldewijk, K. Riahi, E. Shevliakova, S. Smith, E. Stehfest, A. Thomson, P. Thornton, D.P. van Vuuren and Y.P. Wang, *Harmonization of land-use scenarios for the period 1500-2100: 600 years of global gridded annual land-use transitions, wood harvest, and resulting secondary lands*, *Climatic Change* 109 (2011) 117-161, doi:10.1007/s10584-011-0153-2.
- [67] G.C. Hurtt, L. Chini, R. Sahajpal, S. Frohking, B.L. Boudirsky, K. Calvin, J.C. Doelman, J. Fisk, S. Fujimori, K.K. Goldewijk, T. Hasegawa, P. Havlik, A. Heinemann, F. Humpenöder, J. Jungclaus, J. Kaplan, J. Kennedy, T. Kristzin, D. Lawrence, P. Lawrence, L. Ma, O. Mertz, J. Pongratz, A. Popp, B. Poulter, K. Riahi, E. Shevliakova, E. Stehfest, P. Thornton, F.N. Tubiello, D.P. van Vuuren, and K. Zhang, *Harmonization of Global Land-Use Change and Management for the Period 850-2100 (LUH2) for CMIP6*, *Geoscientific Model Development* 13 (2020) 5425-5464. See also [luh.umd.edu](http://luh.umd.edu).
- [68] R.E. Ingham and J. K. Detling, *Plant-herbivore interactions in a North American mixed-grass prairie*, *Oecologia* 63 (1984) 307313.

- [69] C.D. Jones, V. Arora, P. Friedlingstein, L. Bopp, V. Brovkin, J. Dunne, H. Graven, F. Hoffman, T. Ilyina, J.G. John, M. Jung, M. Kawamiya, C. Koven, J. Pongratz, T. Raddatz, J.T. Randerson and S. Zaehle, *C4MIP – The Coupled Climate-Carbon Cycle Model Intercomparison Project: Experimental protocol for CMIP6*, Geosci. Model Dev. 9 (2016) 2853-2880, doi:10.5194/gmd-9-2853-2016.
- [70] H.G. Jones, *Plants and microclimate* (Cambridge University Press, Cambridge, 1992).
- [71] J. H. Jungclaus, S. J. Lorenz, C. Timmreck, C. H. Reick, V. Brovkin, K. Six, J. Segschneider, M. A. Giorgetta, T. J. Crowley, J. Pongratz, N. A. Krivova, L. E. Vieira, S. K. Solanki, D. Klocke, M. Botzet, M. Esch, V. Gayler, H. Haak, T. J. Raddatz, E. Roeckner, R. Schnur, H. Widmann, M. Claussen, B. Stevens and J. Marotzke, *Climate and carbon-cycle variability over the last millennium*, Climate of the Past 6 (2010) 723-737, doi:10.5194/cp-6-723-2010.
- [72] T. Kaminski, W. Knorr, G. Schürmann, M. Scholze, P.J. Rayner, S. Zaehle, S. Blessing, W. Dorigo, V. Gayler, R. Giering, N. Gobron, J.P. Grant, M. Heimann, A. Hooker-Stroud, S. Houweling, T. Kato, J. Kattge, D. Kelley, S. Kemp, E. N. Koffi, C. Köstler, P.-P. Mathieu, B. Pinty, C.H. Reick, C. Rödenbeck, R. Schnur, K. Scipal, C. Sebald, T. Stacke, A. Terwisscha van Scheltinga, M. Vossbeck, H. Widmann and T. Ziehn, *The BETHY/JSBACH Carbon Cycle Data Assimilation System: Experiences and challenges*, J. Geophys. Res. Biogeosciences 118 (2013) 1414-1426, doi:10.1002/jgrg.20118.
- [73] J. Kattge and W. Knorr, *Temperature acclimation in a biochemical model of photosynthesis: A reanalysis of data from 36 species*, Plant, Cell and Environment 30 (2007) 1176-1190, doi: 10.1111/j.1365-3040.2007.01690.x.
- [74] A.K. Knapp and G.A. Carter, *Variability in leaf optical properties among 26 species from a broad range of habitats*, Am. J. of Botany, 85 (1998) 940-946.
- [75] W. Knorr, *Satellitengestützte Fernerkundung und Modellierung des globalen CO<sub>2</sub>-Austauschs der Landvegetation*, Examensarbeit 49, (MPI for Meteorology, Hamburg, 1998). This thesis was also published under the title *Satellite remote sensing and modelling of the global CO<sub>2</sub> exchange of land vegetation: A synthesis study*.
- [76] W. Knorr, *Annual and Internannual CO<sub>2</sub> Exchanges of the Terrestrial Biosphere: Process-Based Simulations and Uncertainties*, Global Ecology and Biogeography 9 (2000) 225-252.
- [77] W. Knorr, I.C. Prentice, J.I. House and E.A. Holland, *Long-term sensitivity of soil carbon turnover to warming*, Nature 433 (2005) 298-301.
- [78] W. Knorr, T. Kaminski, M. Scholze, N. Gobron, B. Pinty, R. Giering and P.-P. Mathieu, *Carbon cycle data assimilation with a generic phenology model*, J. Geophys. Res. 115 (2010) G04017, doi:10.1029/2009JG001119.
- [79] R.D. Koster and M. J. Suarez, *A comparative analysis of two land surface heterogeneity representations*, Journal of Climate 5 (1992) 1379-1390.
- [80] W. Larcher, *Ökologie der Pflanzen* (Ulmer Verlag, Stuttgart, 1976).
- [81] G. Lasslop, K. Thonicke and S. Kloster, *SPITFIRE within the MPI Earth system model: Model development and evaluation*, JAMES 6 (2014) 740-755, doi:10.1002/2013MS000284.
- [82] G. Lasslop, V. Brovkin, C.H. Reick, S. Bathiany and S. Kloster, *Multiple stable states*

- of tree cover in a global land surface model due to a fire-vegetation feedback*, Geophysical Research Letters 43 (2016) 6324-6331, doi:10.1002/2016GL069365.
- [83] W.K. Lauenroth and D.G. Milchunas, *Short-grass steppe*. In: R.T. Coupland (ed.), *Ecosystems of the World*, Vol. 8A, (Elsevier, New York, 1992), Chapter II, pp. 183-226.
- [84] D.M. Lawrence, G.C. Hurtt, A. Arneth, V. Brovkin, K.V. Calvin, A.D. Jones, C.D. Jones, P.J. Lawrence, N. de Noblet-Ducoudré, J. Pongratz, S.I. Seneviratne, and E. Shevliakova, *The Land Use Model Intercomparison Project (LUMIP) contribution to CMIP6: rationale and experimental design*, Geoscientific Model Development 9 (2016) 2973-2998, doi:10.5194/gmd-9-2973-2016.
- [85] C.S. Li, J. Aber, F. Stange, K. Butterbach-Bahl and H. Papen, *A process-oriented model of N<sub>2</sub>O and NO emissions from forest soils: 1. Model development*, Journal of Geophysical Research 105 (2000) 4369-4384.
- [86] J. Liski, T. Palosuo, M. Peltoniemi and R. Sievänen, *Carbon and decomposition model Yasso for forest soils*, Ecological Modelling 189 (2005) 168-182.
- [87] A. Loew, P.M. van Bodegom, J.-L. Widlowski, J. Otto, T. Quaife, B. Pinty and T. Raddatz, *Do we (need to) care about canopy radiation schemes in DGVMs? Caveats and potential impacts*, Biogeosciences 11 (2014) 1873-1897, doi:10.5194/bg-11-1873-2014.
- [88] N. Mahowald, F. Lo, Y. Zheng, L. Harrison, C. Funk and D. Lombardozzi, *Projections of leaf area index in Earth system models*, Earth System Dynamics 7 (2016) 211-229, doi:10.5194/esd-7-211-2016.
- [89] T. Mauritsen et al., *Developments in the MPIM Earth System Model version 1.2 (MPI-ESM 1.2) and its response to increasing CO<sub>2</sub>*, JAMES 11 (2019) 998-1038, doi:10.1029/2018MS001400.
- [90] A.D. McGuire, S. Sitch, J.S. Clein, R. Dargaville, G. Esser, J. Foley, M. Heimann, F. Joos, J. Kaplan, D.W. Kicklighter, R.A. Meier, J.M. Melillo, B. Moore III, I.C. Prentice, N. Ramankutty, T. Reichenau, A. Schloss, H. Tian, L.J. Williams and U. Wittenberg, *Carbon balance of the terrestrial biosphere in the twentieth century: Analyses of CO<sub>2</sub>, climate and land use effects with four process-based ecosystem models*, Global Biogeochem. Cycles 15 (2001) 183-206, doi:10.1029/2000GB001298.
- [91] S.J. McNaughton, *Serengeti migratory wildbeest: Facilitation of energy flow by grazing*, Science 191 (1976) 92-94.
- [92] S.J. McNaughton, *Ecology of a grazing system: the Serengeti*, Ecol. Monogr. 55 (1985) 259-294.
- [93] M.B. Murray, M.G.R. Cannell and R.I. Smith, *Date of budburst of fifteen tree species in Britain following climatic warming*, J. Appl. Ecology 26 (1989) 693-700.
- [94] R.B. Myneni, S. Hoffman, Y. Knyazikhin, J.L. Privette, J. Glassy, Y. Tian, Y. Wang, X. Song, Y. Zhang, G.R. Smith, A. Lotsch, M. Friedl, J.T. Morisette, P. Votava, R.R. Nemani, S.W. Running, *Global products of vegetation leaf area and fraction of absorbed PAR from year one of MODIS data*, Remote Sensing of Environment 83 (2002) 214-231.
- [95] E.A. Newell, S.S. Mulkey and S.J. Wright, *Seasonal patterns of carbohydrate storage in four tropical tree species*, Oecologia 131 (2002) 333-342.
- [96] T. Nilson, *A theoretical analysis of the frequency of gaps in plant stands*, Agricultural Meteorology 8 (1971) 25-38.

- [97] A.T. O'Geen, *Soil Water Dynamics*, Nature Education Knowledge 3 (2012) 12.
- [98] J. Otto, T. Raddatz, M. Claussen, V. Brovkin and V. Gayler *Separation of atmosphere-ocean-vegetation feedbacks and synergies for mid-Holocene climate*, Geophysical Research Letters 36 (2009) L09701, doi:10.1029/2009GL037482.
- [99] J. Otto, T. Raddatz and M. Claussen *Strength of forest-albedo feedback in mid-Holocene climate simulations*, Climate of the Past 7 (2011) 1027-1039, doi:10.5194/cp-7-1027-2011.
- [100] T. Parnik, H. Ivanova and O. Keerberg, *Photorespiratory and respiratory decarboxylations in leaves of  $C_3$  plants under different  $CO_2$  concentrations and irradiances*, Plant, Cell and Environment 30 (2007) 1535-1544.
- [101] B. Parida, *The influence of plant nitrogen availability on the global carbon cycle and  $N_2O$  emissions*, Reports on Earth System Science 92 (2011), (MPI for Meteorology, Hamburg), ISSN 1614-1199, [www.mpimet.mpg.de/fileadmin/publikationen/Reports/WEB\\_BzE\\_92.pdf](http://www.mpimet.mpg.de/fileadmin/publikationen/Reports/WEB_BzE_92.pdf).
- [102] K. A. Patterson, 1990: *Global distributions of total and total-available soil water holding capacities*. M.S. thesis, (Dept. of Geography, University of Delaware, 1990).
- [103] J. Polcher, B. McAvaney, P. Viterbo, M.-A. Gaertner, A. Hahmann, J.-F. Mahfou, J. Noilhan, T. Phillips, A. Pitman, C.A. Schlosser, J. -P. Schulz, B. Timbal, D. Versegny and Y. Xue, *A proposal for a general interface between land surface schemes and general circulation models*, Global and Planetary Change 19 (1998) 261-276.
- [104] A.J. Pitman, N. de Noblet-Ducoudré, F.T. Cruz, E.L. Davon, G.B. Bonan, V. Brovkin, M. Claussen, C. Delire, L. Ganzefeld, V. Gayler, B.J.J.M. van den Hurk, P.J. Lawrence, M.K. van der Molen, C. Mller, C.H. Reick, S.I. Seneviratne, B.J. Strengers and A. Voldoire, *Uncertainties in climate responses to past land cover change: First results from the LUCID intercomparison study*, Geophysical Research Letters 36 (2009) L14814, doi:10.1029/2009GL039076
- [105] J. Pongratz, C. H. Reick, T. Raddatz and M. Claussen, *Effects of anthropogenic land cover change on the carbon cycle of the last millennium*, Global Biogeochemical Cycles 23 (2009) GB4001, doi:10.1029/2009GB003488
- [106] J. Pongratz, C. Reick, T. Raddatz and M. Claussen, *A Global Land Cover Reconstruction AD 800 to 1992 - Technical Description*, Reports on Earth System Science 51 (MPI for Meteorology, Hamburg, 2008), ISSN 1614-1199.
- [107] A. Quarteroni, R. Sacco and F. Saleri, *Numerical Mathematics*, (Springer, New York, 2007).
- [108] J.W. Raich and C.S. Potter, *Global patterns of carbon dioxide emissions from soils*, Global Biogeochemical Cycles 9 (1995) 23-36.
- [109] S. Rast, R. Brokopf, S.-K. Cheedela, M. Esch, V. Gayler, I. Kirchner, Luis Kornblüh, A. Rhodin, H. Schmidt, U. Schulzweida and K.-H. Wieners, *User manual for ECHAM6*, Berichte zur Erdsystemforschung 136, (MPI for Meteorology, Hamburg, 2013).
- [110] C. Reick, T. Raddatz, J. Pongratz and M. Claussen, *Contribution of anthropogenic land cover change emissions to preindustrial atmospheric  $CO_2$* , Tellus B62 (2010) 329-336, doi:10.1111/j.1600-0889.2010.00479.x.
- [111] C. Reick, T. Raddatz, V. Brovkin and V. Gayler, *Representation of natural and an-*

- thropogenic land cover change in MPI-ESM*, JAMES 5 (2013) 1-24, doi:10.1002/jame.20022.
- [112] Z. Ren-Jun, Z.Yi-Lin, F. Le-Run, L. Xin-Ren and Z. Quan-Sheng, *The Xinanjiang model*, Hydrological forecasting (Proceedings of the Oxford Symposium, April 1980), IAHS-AISH Publ. no. 129.
- [113] L.A. Richards, *Capillary conduction of liquids through porous mediums*. J. Appl. Phys. 1 (1931) 318-333, doi:10.1063/1.1745010.
- [114] R. Richtmyer and K. Morton, *Difference Methods for Initial Value Problems* (Interscience, 1967).
- [115] E. Roeckner, K. Arpe, L. Bengtsson, M. Christoph, M. Claussen, L. Dümenil, M. Esch, M. Giorgetta, U. Schlese and U. Schulzweida, *The atmospheric general circulation model ECHAM-4: model description and simulation of present-day climate*, MPI Report 218 (MPI for Meteorology, Hamburg, 1996).
- [116] E. Roeckner, G. Bäuml, L. Bonaventura, R. Brokopf, M. Esch, M. Giorgetta, S. Hagemann, I. Kirchner, L. Kornbluh, E. Manzini, A. Rhodin, U. Schlese, U. Schulzweida and A. Tompkins, *The atmospheric general Circulation model ECHAM 5. Part I: Model description*, MPI Report 349 (MPI for Meteorology, Hamburg, 2003).
- [117] E. Roeckner, *update\_surfacetemp*, personal notes, Hamburg 2013.
- [118] A. Roesch, M. Wild, H. Gilgen and A. Ohmura, *A new snow cover fraction parameterization for the ECHAM4 GCM*, Climate Dynamics 17 (2001) 933-946.
- [119] O.E. Sala, W.K. Lauenroth and I.C. Burke, *Carbon budgets of temperate grasslands and the effects of global change*. In: A.I. Breymeyer, D.O. Hall, J.M. Melillo and G. Ågren (eds.), *Global Change: Effects on Coniferous Forests and Grasslands*, (Wiley & Sons, Chichester, 1996), pp. 101-120.
- [120] J.-P. Schulz, *On the role of the land surface representation and the numerical coupling to the atmosphere for the simulated climate of the global ECHAM4 model*, Examensarbeit 57 (MPI for Meteorology, Hamburg, 1998).
- [121] J.-P. Schulz, L. Dümenil and J. Polcher, *On the land surface-atmosphere coupling and its impact in a single-column atmospheric model*, J. Appl. Meteorol. 40 (2001) 642-663.
- [122] E. Schulze, F.M. Kelliher, C. Körner, J. Lloyd and R. Leuning, *Relationships among Maximum Stomatal Conductance, Ecosystem Surface Conductance, Carbon Assimilation Rate, and Plant Nitrogen Nutrition: A Global Ecology Scaling Exercise*, Annual Review of Ecology and Systematics, 25 (1994) 629-662.
- [123] J.M.O. Scurlock, W. Cramer, R.J. Olson, W.J. Parton and S.D. Prince, *Terrestrial NPP: Toward a consistent data set for global model evaluation*, Ecol. Appl. 9 (1999) 913-919.
- [124] P.J. Sellers, *Canopy reflectance, photosynthesis and transpiration*, Int. J. Remote Sensing 6 (1985) 1335-1372.
- [125] P.J. Sellers, J.A. Berry, G.J. Collatz, C.B. Field and F.G. Hall, *Canopy reflectance, photosynthesis and transpiration. III. A reanalysis using improved leaf models and canopy integration scheme*, Remote Sensing of Environment 42 (1992) 187-216.
- [126] P.J. Sellers, D.A. Randall, G.J. Collatz, J.A. Berry, C.B. Field, D.A. Dazlich, C. Zhang, G.D. Collelo and L. Bounoua, *A revised land surface parametrization (SiB2) for Atmo-*

- spheric GCMs. Part I: Model Formulation*, J. of Climate 9 (1996) 676-705.
- [127] J. Sierra, *Nitrogen mineralization and nitrification in a tropical soil: Effects of fluctuating temperature conditions*, Soil Biology and Biochemistry 34 (2002) 1219-1226.
- [128] V.P. Singh, *Hydrologic systems*, Vol. 1, *Rainfall-runoff modeling* (Prentice Hall, 1988).
- [129] S. Sitch, C. Prentice, A. Arneth, A. Bondeau, W. Cramer, J. Kaplan, S. Levis, W. Lucht, M. Sykes, K. Thonicke and S. Venevsky, *Evaluation of ecosystem dynamics, plant geography and terrestrial carbon cycling in the LPJ dynamic global vegetation model*, Global Change Biology 9 (2003) 161-185.
- [130] P. Sitte, E.W. Weiler, J.W. Kadereit, A. Bresinsky and C. Körner, *Lehrbuch der Botanik für Hochschulen*, 35<sup>th</sup> edition, (Spektrum Akademischer Verlag, Heidelberg-Berlin, 2002).
- [131] M.T. Sykes, I.C. Prentice and W. Cramer, *A bioclimatic model for the potential distributions of North European tree species under present and future climates*, Journal of Biogeography 23 (1996) 203-233.
- [132] K. Thonicke, A. Spessa, I.C. Prentice, S.P. Harrison, L. Dong and C. Carmona-Moreno, *The influence of vegetation, fire spread and fire behaviour on biomass burning and trace gas emissions: Results from a process-based model*, Biogeoscience, 7 (2010) 1991-2011, doi:10.5194/bg-7-1991-2010.
- [133] P.E. Thornton, J.-F. Lamarque, N.A. Rosenbloom and N.M. Mahowald, *Influence of carbon-nitrogen cycle coupling on land model response to CO<sub>2</sub> fertilization and climate variability*, Global Biogeochemical Cycles 21 (2007) GB4018, doi:10.1029/2006GB002868. See in particular the supplement S1.
- [134] T. Thum, P. Risnen, S. Sevanto, M. Tuomi, C. Reick, T. Vesala, T. Raddatz, T. Aalto, H. Jaervinen, N. Altimir, K. Pilegaard, Z. Nagy, S. Rambal and J. Liski, *Soil carbon model alternatives for ECHAM5/JSBACH climate model: Evaluation and impacts on global carbon cycle estimates*, Journal of Geophysical Research 116 (2011) 1-14.
- [135] E. Todini, *The ARNO rainfall-runoff model*, Journal of Hydrology 175 (1996) 339-382, doi:10.1016/S0022-1694(96)80016-3.
- [136] M. Tuomi, P. Vanhala, K. Karhu, H. Fritze and J. Liski, *Heterotrophic soil respiration – comparison of different models describing its temperature dependence*, Ecological Modelling 211 (2008) 182-190.
- [137] M. Tuomi, T. Thum, H. Järvinen, S. Fronzek, B. Berg, M. Harmon, J. Trofymow, S. Sevanto, and J. Liski, *Leaf litter decomposition – estimates of global variability based on Yasso07 model*, Ecological Modelling 220 (2009) 3362-3371.
- [138] M. Tuomi, R. Laiho, A. Repo and J. Liski, *Wood decomposition model for boreal forests*, Ecological Modelling 222 (2011) 709-718.
- [139] M. Tuomi, J. Rasinmäki, A. Repo, P. Vanhala and J. Liski, *Soil carbon model Yasso07 graphical user interface*, Environmental Modelling & Software 26 (2011) 1358-1362.
- [140] F. Vamborg, V. Brovkin and M. Claussen, *The effect of a dynamic background albedo scheme on Sahel/Sahara precipitation during the mid-Holocene*, Climate of the Past 7 (2011) 117-131, doi:10.5194/cp-7-117-2011.
- [141] M.T. Van Genuchten, *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, Soil Science Society of America Journal 44 (1980) 892-898.

- [142] V. Westhoff and E. van der Maarel, *The Braun-Blanquet approach*. In: R.H. Whittaker (ed.), *Ordination and classification of communities*, Handbook of Vegetation Science 5 (1973) 617-726.
- [143] M.A. White, P.E. Thornton, S.W. Running and R.R. Nemani, *Parameterization and sensitivity analysis of the BIOME-BGC terrestrial ecosystem model: Net primary production controls*, Earth Interactions 4 (2000) 1-85.
- [144] S. Wilkenskjaeld, S. Kloster, J. Pongratz, T. Raddatz and C. H. Reick, *Comparing the influence of net and gross anthropogenic land-use and land-cover changes on the carbon cycle in the MPI-ESM*, Biogeosciences, 11 (2014) 4817-4828, doi:10.5194/bg-11-4817-2014.
- [145] R. D. Williams and L. R. Ahuja, *Scaling and estimating the soil water characteristic using a one-parameter model*. In: Y. Pachepsky, D.E. Radcliffe and H.M. Selim (eds.), *Scaling Methods in Soil Physics*, (CRC Press, Boca Raton, Florida, 2003), pp. 35-48.
- [146] A. Wolf, P. Ciais, V. Bellassen, N. Delbart, C. B. Field and J. A. Berry, *Forest biomass allometry in global land surface models*, Global Biogeochemical Cycles 25 (2011) GB3015, doi:10.1029/2010GB003917.
- [147] S. Zaehle, B.E. Medlyn, M.G. De Kauwe, A.P. Walker, M.C. Dietze, T. Hickler, Y. Luo, Y.-P. Wang, B. El-Masri, P. Thornton, A. Jain, S. Wang, D. Warlind, E. Weng, W. Parton, C.M. Iversen, A. Gallet-Budynek, H. McCarthy, A. Finzi, P.J. Hanson, I.C. Prentice, R. Oren and R.J. Norby, *Evaluation of 11 terrestrial carbon-nitrogen cycle models against observations from two temperate free-air CO<sub>2</sub> enrichment studies*, New Phytologist 201 (2014) 803-822, doi:10.1111/nph.12697.



Die gesamten Veröffentlichungen in der Publikationsreihe des MPI-M  
„Berichte zur Erdsystemforschung“,  
„Reports on Earth System Science“,  
ISSN 1614-1199

sind über die Internetseiten des Max-Planck-Instituts für Meteorologie erhältlich:

<http://www.mpimet.mpg.de/wissenschaft/publikationen.html>

