

---

# WFLOW\_LINTUL: raster-based simulation of rice growth in the WFLOW/OpenStreams hydrological modelling platform

User manual and description of core model code

Sander C. de Vries



---

# WFLOW\_LINTUL: raster-based simulation of rice growth in the WFLOW/OpenStreams hydrological modelling platform

User manual and description of core model code

Sander C. de Vries

This study was carried out by the Wageningen Research Foundation (WR) Business Unit Agrosystems Research and was commissioned and financed by the Netherlands Space office (NSO).

WR is part of Wageningen University & Research, the collaboration of Wageningen University and Wageningen Research Foundation.

Wageningen, September 2018

---

Report WPR-840

---

De Vries, S.C., 2018. *WFLOW\_LINTUL: raster-based simulation of rice growth in the WFLOW/OpenStreams hydrological modelling platform; User manual and description of core model code*. Wageningen Research, Report WPR-840. 30 pp.; 7 fig.; 24 ref.

This report can be downloaded for free at <https://doi.org/10.18174/461276>

In the context of the G4INDO project, a radically new, raster-based type of crop growth simulation model was developed, specifically with spatial applications in mind. Rather than consecutively and individually simulating the growing season for each grid cell in a raster map, wflow\_lintul simulates crop growth simultaneously for all grid cells, thereby allowing daily interactions between those grid cells, for instance in terms of water flow. The approach also allows for high calculation speeds and facilitates dynamic graphic representation of the output. Wflow\_lintul is based on the LINTUL (Light INterception and UtiLization) family of crop growth models originating from Wageningen University and Research, and simulates rice growth in close conjunction with the spatial hydrological model wflow\_sbm developed at Deltares. Both models are part of the Deltares wflow/OpenStreams suit of simulation models. The present report provides a user manual and description of core model code. Reports on practical applications are to be expected in the foreseeable future.

Keywords: crop growth model, rice, hydrology, yield forecasting, precision agriculture

© 2018 Wageningen, Stichting Wageningen Research, Wageningen Plant Research, Business Unit Agrosystems Research, P.O. Box 16, 6700 AA Wageningen, The Netherlands; T +31 (0)317 48 07 00; [www.wur.eu/plant-research](http://www.wur.eu/plant-research)

Chamber of Commerce no. 09098104 at Arnhem  
VAT NL no. 8065.11.618.B01

Stichting Wageningen Research. All rights reserved. No part of this publication may be reproduced, stored in an automated database, or transmitted, in any form or by any means, whether electronically, mechanically, through photocopying, recording or otherwise, without the prior written consent of the Stichting Wageningen Research.

Stichting Wageningen Research is not liable for any adverse consequences resulting from the use of data from this publication.

Report WPR-840

Photo cover: Deltares

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Differences between wflow_lintul and LINTUL3/other LINTUL versions</b>	<b>6</b>
<b>3</b>	<b>Run options</b>	<b>8</b>
	3.1 Running under Delft-FEWS	8
	3.2 Running wflow_lintul standalone (potential production)	8
	3.3 Running wflow_lintul in conjunction with wflow_sbm	8
<b>4</b>	<b>List of run control parameters in wflow_lintul.ini</b>	<b>9</b>
<b>5</b>	<b>List of model parameters in wflow_lintul.ini</b>	<b>11</b>
<b>6</b>	<b>Model forcing data</b>	<b>17</b>
<b>7</b>	<b>Description of core wflow_lintul model code</b>	<b>18</b>
	7.1 Phenology	18
	7.2 Photosynthesis and crop growth	20
	7.3 Partitioning of overall biomass growth over the various plant organs (calculation of growth rates)	20
	7.4 Daily change in Leaf Area Index (LAI)	23
	7.5 Root (depth) growth	24
	7.6 The influence of water stress on crop growth	25
<b>8</b>	<b>References</b>	<b>27</b>

---

---

# 1 Introduction

`wflow_lintul`, a raster-based crop growth model for rice, is based on `LINTUL3`, a point-based model for simulating nitrogen-limited rice growth (Shibu et al., 2010). `LINTUL3` was parameterized and calibrated for rice by Shibu et al. (2010), based on experimental data from Southeast Asia (Drenth et al., 1994) and drawing on the more complex rice model `ORYZA2000` (Bouman et al., 2001) and on the preceding versions `LINTUL1` (Spitters, 1990) and `LINTUL2` (Spitters and Schapendonk, 1990). In contrast to `LINTUL3`, `wflow_lintul` is primarily intended for simulation of rice production under water-limited conditions, rather than under nitrogen-limited conditions. To that end, it was designed to function in close cooperation with the spatial hydrological model `wflow_sbm`, which operates on a watershed-scale (see Vertessy et al., 1999 for a description of the SBM model).

The `LINTUL` (Light Interception and Utilization) models were the first deviation from the more complex, photosynthesis-based models of the “De Wit school” of crop modelling, also called the “Wageningen school” (Bouman et al., 1996). In the `LINTUL` models, total dry matter production is calculated in a comparatively simple way, using the Monteith approach (Monteith, 1969; 1990), in which crop growth rate is calculated as the product of interception of (solar) radiation by the canopy and a light-use efficiency (LUE; Russell et al., 1989). This way of estimating (daily) biomass production in the `LINTUL` models is reflected in Equation 1 below and may be considered the core of the `wflow_lintul` model:

$$GTOTAL = self.LUE * PARINT * TRANRF \quad (Eq. 1)$$

with:

- `GTOTAL` the overall daily growth rate of the crop ( $\text{g m}^{-2} \text{d}^{-1}$ )
- `self.LUE` a constant light use efficiency<sup>1</sup> ( $\text{g MJ}^{-1}$ )
- `PARINT` the daily intercepted photosynthetically active radiation ( $\text{MJ m}^{-2} \text{d}^{-1}$ )
- `TRANRF` the ‘transpiration reduction factor’, i.e. the reducing (-) impact of water shortage on biomass production (Eq. 36)

For regional studies, `LINTUL`-type models have the advantage that data input requirements are drastically reduced and model parameterization is facilitated (Bouman et al., 1996). `LINTUL` was first developed for potential crop growth (i.e. perfect growing conditions without any water or nutrient shortages and in absence of pests, diseases and adverse soil conditions) as “`LINTUL1`” (Spitters, 1990). Later, it was extended to simulate water-limited conditions (“`LINTUL2`”; Spitters and Schapendonk, 1990) and nitrogen-limited conditions (“`LINTUL3`”; Shibu et al., 2010). Under water-limited conditions, all growth factors except water are assumed non-limiting, i.e. ample nutrient availability, a pest-, disease- and weed-free environment and no adverse soil conditions. Under nitrogen-limited conditions, only nitrogen availability may limit crop growth. `LINTUL` has been successfully applied to different crops such as potato (Spitters and Schapendonk, 1990), grassland (`LINGRA`) (Schapendonk et al., 1998), maize (Farré et al., 2000), oilseed rape (Habekotté, 1997) and rice (Shibu et al., 2010), in potential, water-limited or nitrogen-limited situations.

---

<sup>1</sup> The prefix ‘self’ in `self.LUE` is part of the Python object-oriented syntax and has no mathematical meaning.

---

## 2 Differences between wflow\_lintul and LINTUL3/other LINTUL versions

The main differences between wflow\_lintul and LINTUL3 (Shibu et al., 2010) can be summarized as follows:

- **Potential and water-limited simulations**

Wflow\_lintul presently (spring 2018) simulates potential and water-limited crop growth (the latter in conjunction with wflow\_sbm). First preparations to add simulation of nitrogen-limited rice growth (LINTUL3) have been made in the present version of the model.

- **Water balance outsourced to wflow\_sbm**

The simple water balance, based on Stroosnijder (1982) and Penning de Vries et al. (1989), which was present in LINTUL2 and LINTUL3, is no longer present in wflow\_lintul. All water balance-related simulation tasks are outsourced to the wflow\_sbm model (a contour-based, watershed-scale water balance model; see Vertessy et al., 1999 for a description of the SBM model). On the other hand, several crop growth related tasks in the hydrology model wflow\_sbm, such as the simulation of LAI (leaf area index) are now outsourced to wflow\_lintul. In the wflow framework, wflow\_lintul and wflow\_sbm communicate with each other via an implementation of the (Python) BMI (Basic Model Interface).

- **Written in PCRaster Python**

Whereas the original LINTUL1, LINTUL2 and LINTUL3 models were implemented in the Fortran Simulation Translator (FST) software (Rappoldt and Van Kraalingen, 1996), wflow\_lintul is written in PCRaster Python and fully integrated into the wflow hydrologic modelling framework. PCRaster (Wesseling et al., 1996) is a collection of software targeted at the development and deployment of spatio-temporal environmental models. It contains a scripting model development environment: it allows users to develop their own simulation models. Supported scripting languages include Python (Van Rossum, 1995) and PCRcalc.

- **Integration in the wflow framework**

Integration of wflow\_lintul in the wflow framework means that all timer and input-output related tasks for the model are handled by that framework. The original LINTUL1, LINTUL2 and LINTUL3 models were implemented in the Fortran Simulation Translator (FST) software (Rappoldt and Van Kraalingen, 1996). In FST, timer and input-output related tasks were handled by the TTUTIL library (a collection UTILities originating from the former "Theoretische Teeltkunde" research group of Wageningen University and Research; Rappoldt and Van Kraalingen, 1996).

- **Raster based**

In contrast to LINTUL1, 2 and 3, most input and output variables in wflow\_lintul are represented by maps or arrays, instead of single values (unless a base-map with a dimension of only one grid cell is specified). Each value in such a map or array represents a grid cell on the map of the study area/catchment; wflow-lintul simulates crop growth for all grid cells simultaneously and can therefore be characterized as a grid-based model. The original LINTUL1, LINTUL2 and LINTUL3 models, in contrast, were point-based and only able to simulate crop growth for one grid cell at a time.

- **Runs for multiple consecutive seasons or years**

As opposed to the original LINTUL models, which run for one season/year at a time, wflow\_lintul can continuously run over multiple consecutive cropping cycles and years. At the end of each cropping season, the crop is then harvested automatically, meaning that all variables are reset. At the beginning of each new cropping season, crop growth is automatically re-initiated.



---

- **Advanced and remotely sensed run control**

wflow\_lintul offers various novel options for controlling the onset and termination of crop growth:

1. Crop growth can be initiated automatically, on a pixel-by-pixel basis, based on remotely sensed data. To that end, satellite imagery needs to be processed in such a way that for each pixel/grid cell, a Boolean value indicates whether there is (rice) crop growth, predominantly (value = 1), or not (value = 0). The model starts/terminates crop growth if changes 0->1 or vice versa occur.
2. Crop growth can be (automatically) initiated on a pixel-by-pixel basis if a pre-specified minimum amount of rainfall requirement has accumulated – this amount may be considered necessary for land preparation (soil puddling). In the implementation for central Java, this threshold is normally set at 200 mm rainfall from November 1 on (approximate start of the rainy season), based on Naylor et al. (2007).
3. Similar to the original point-based LINTUL models, the user can enter a single fixed start date and/or a single fixed end date; these dates will then be applied across all grid cells of the simulated catchment area simultaneously. To simulate growth of transplanted rice, which has already emerged and grown for some time in the nursery before being planted out in the main field, a default initial development stage is specified in the model, together with initial weights of leaves, stems and roots.

---

## 3 Run options

### 3.1 Running under Delft-FEWS

In operational systems, `wflow-lintul`, either in combination with the hydrologic model `wflow_sbm` or standalone, can be run under Delft – FEWS (Werner et al., 2013), a powerful open shell system for managing forecasting processes and/or handling time series data. The user is then referred to the online manual of Delft FEWS (<https://publicwiki.deltares.nl/display/FEWSDOC>) and to more specific training materials that have been produced under the umbrella of the Joint Cooperation Programme between Indonesia and the Netherlands (JCP; <https://publicwiki.deltares.nl/display/JCP/Home>).

### 3.2 Running `wflow_lintul` standalone (potential production)

Apart from running under Delft-FEWS, it is also possible to run `wflow_lintul` directly in Python, from the command line (which is in fact what Delft FEWS also does under the bonnet). The `wflow` framework and the PCRaster software for environmental modelling still need to be installed in that case however. It is also necessary to provide the name of a relevant case directory (`-C`; i.e. the directory that contains all the input and output data) and the path to a `wflow_lintul.ini` file (`-c`; a file containing data for correct initialization of the model), for example:

- ```
python c:\continuum\anaconda2\lib\site-packages\wflow-1.0.master-py2.7.egg\wflow\wflow_lintul.py -C d:\Reference_Run\wflow\wflow_lintul -c wflow_lintul.ini
```

For standalone runs, i.e. without linking to the hydrology model `wflow_sbm`, the “`WATERLIMITED`” option in `wflow_lintul.ini` should be set to “`False`”; see also further down (Ad 3). Water limited simulation presently requires the presence of a water balance/hydrology model (i.e. `wflow_sbm`).

### 3.3 Running `wflow_lintul` in conjunction with `wflow_sbm`

For running `wflow_lintul` in conjunction with the hydrological model `wflow_sbm`, which takes care of all water-balance related tasks and which is what the model is really intended for, a custom-built BMI (Basic Model Interface) runner module, exchanging data between `wflow_sbm` and `wflow_lintul`, needs to be run. It is recommended to place this script in the relevant case directory, e.g.:

- ```
python bmi2runner.py -c d:\reference_run\bmi2runner.ini
```

where the location of the relevant ini-file (containing data for initialization) is provided after “`-c`”. It should be ensured that `bmirunner.ini` contains the appropriate run settings before running the models.

---

## 4 List of run control parameters in wflow\_lintul.ini

Wflow\_lintul model runs can be configured by editing the values of a number of parameters in the [model] section of the wflow\_lintul.ini file. Wflow\_lintul reads the values from wflow\_lintul.ini with its parameters function; if a certain parameter or value is not found in wflow\_lintul.ini, a default value (provided for each individual parameter in the parameters function) is returned. The following nine variables in wflow\_lintul.ini have a run control function:

1. CropStartDOY = 0
2. HarvestDAP = 90
3. WATERLIMITED = True
4. AutoStartStop = True
5. RainSumStart\_Month = 11
6. RainSumStart\_Day = 1
7. RainSumReq = 200
8. Pause = 13
9. Sim3rdSeason = True

For each of these control variables, additional explanation is explained below:

1. **CropStartDOY** (integer) - needs to be set to 0 for DEWS runs. Sets the Julian Day Of the Year (DOY) on which crop growth is initiated – for all grid cells simultaneously. If set to 0, crop growth is initiated automatically for each grid cell individually, based on a certain minimum accumulated rainfall quantity (AutoStartStop is set to True) or based on remote sensing information (if AutoStartStop is set to False; see also Run Control variable 4, AutoStartStop).
2. **HarvestDAP** (integer): sets the number of Days After Planting (DAP) on which crop growth is terminated and all crop-related states are reset to 0 – for all grid cells simultaneously. It is advisable to set HarvestDAP = 0, so that crop growth is terminated when the crop is physiologically mature, on a pixel by pixel basis. For rice in Indonesia, this is generally the case after 80-110 DAP. If a positive HarvestDAP value is set and the crop is not yet physiologically mature after the specified number of days, the crop will be harvested immaturely (simultaneously in all pixels). If the crop had already reached physiological maturity before the specified number of DAP, the last simulated values of the state variable (final rice yield, LAI, etc.) will be maintained until the specified number of DAP is reached and then be reset to 0 in all pixels simultaneously (a model artefact, resulting in a blocky output graph).
3. **WATERLIMITED** (Boolean): if WATERLIMITED = False, potential crop growth will be simulated, i.e. crop growth without any water limitation (definition: see Introduction). With this option, wflow\_lintul can also be run standalone (i.e. without coupling to the hydrological model wflow\_sbm). If WATERLIMITED = True (default), evapotranspiration and soil water status will be simulated by the wflow\_sbm model and crop growth will be reduced in case of less-than-perfect water supply (water-limited production - definition: see Introduction). *NOTE: if the latter option is selected without the presence of the wflow\_sbm model and a BMI module for exchanging data between wflow\_lintul and wflow\_sbm, wflow\_lintul will not run properly (due to absence of crop available water, crop growth will not be initiated).*
4. **AutoStartStop** (Boolean): if set to True, rice crop growth in the rainy season will be initiated automatically for each grid cell (unless CropStartDOY is set > 0, see 1), based on the cumulative amount of precipitation that falls in that grid cell from November 1 on (this date can be changed by adjusting RainSumStart\_Month and RainSumStart\_Day in wflow\_lintul.ini, see Section 4, run control variables 5 and 6). If a pre-specified threshold value (RainSumReq, default value 200 mm; see Section 4, run control variable 7) is reached, rice crop growth will be initiated.

---

The second (dry season) crop will be initiated at a specified number of days after termination of the first (wet season) crop. This period allows for harvesting operations and preparation of the new field to take place and can be changed by modifying the value of the variable "Pause" (see 8). Similarly, the third (dry season) crop will be initiated at the specified number of days after termination of the second (dry season) crop.

Crop growth during the second season will, similar to the first season, be terminated when the accumulated temperature sum (model internal variable `self.TSUM`) reaches `TTSUM`, or at the specified number of days after crop establishment (`HarvestDAP`) - whichever occurs first.

For all three simulated seasons, crop growth in a certain grid cell will cease - and a final rice yield estimate will be obtained - when the accumulated temperature sum (model internal variable `self.TSUM`) reaches `TTSUM`, i.e. the total temperature sum required to reach crop maturity. However, if `HarvestDAP` is set > 0 (see Ad 2), crop growth will cease after the specified number of DAP is reached, or when `TTSUM` is reached - whichever occurs first.

5. **RainSumStart\_Month** (integer): the month of the year from which the accumulated amount of precipitation (required for initiating the first, rainy, season) is starting to be calculated. Crop simulation starts if the amount (in mm) meets `RainSumReq` (see 7).
6. **RainSumStart\_Day** (integer): the day of the planting month (`RainSumStart_Month`; parameter 5., above) from which the accumulated amount of precipitation (required for initiating the first, rainy, season) is calculated. Crop simulation starts if the amount (in mm) meets `RainSumReq` (see 7).
7. **RainSumReq** (real, mm): the accumulated amount of precipitation (in mm) that is required for establishment of the irrigated rice crop at the start of the first (rainy) season.
8. **Pause** (integer, days): the number of days between the first (rainy) and second (dry) season and between the second and the third season (if applicable); this period is required for harvesting, land preparation and planting of the second and/or third crop.
9. **Sim3rdSeason** (Boolean): if set to `True`, a third rice crop will be simulated in each simulated year. In case it has not matured on the date specified by Run Control variables 5 and 6, the third crop will be terminated on that date, assuming that farmers will prioritize the first (main) rainy season crop and that in such situations (cooler places), planting 3 rice crops per year is probably not worthwhile.

---

## 5 List of model parameters in wflow\_lintul.ini

```
10. LAT = 3.16
11. LUE = 2.47
12. TSUMI = 362
13. K = 0.6
14. SLAC = 0.02
15. TSUMAN = 1420
16. TSUMMT = 580.
17. TBASE = 8.
18. RGRL = 0.009
19. WLVGI = 0.86
20. WSTI = 0.71
21. WRTLI = 1.58
22. WSOI = 0.00
23. DVSDR = 0.8
24. RDRRT = 0.03
25. RDRSHM = 0.03
26. LAICR = 4.0
27. ROOTDM_mm = 1000.
28. RRDMAX_mm = 10.
29. ROOTDI_mm = 50.
30. RDRTB = [0.0,0.0, 0.6,0.00, 1.0,.015, 1.6,0.025, 2.1,0.05, "RDRTB"]
31. PHOTB = [0.0,0.0, 8.0,0.0, 10.0,1.0, 12.0,1.0, 13.0,0.8, 14.0,0.6,
18.,0.0, "PHOTB"]
32. SLACF = [0.0,1.72, 0.21,1.72, 0.24,1.72, 0.33,1.32 , 0.7,1.20
,1.01,1.00, 2.0,0.75, 2.1,0.75, "SLACF"]
33. FRTTB = [0.0,0.300, 0.48,0.30, 0.62,0.12, 0.69,0.11 , 0.84,0.11 ,
0.92,0.10, 1.00,0.08, 1.38,0.00, 2.11,0.0, "FRTTB"]
34. FLVTB = [0.0,0.315, 0.48,0.35, 0.62,0.44, 0.69,0.463, 0.84,0.463,
0.92,0.45, 1.00,0.00, 1.38,0.00, 2.10,0.0, "FLVTB"]
35. FSTTB = [0.0,0.385, 0.48,0.35, 0.62,0.44, 0.69,0.427, 0.84,0.427,
0.92,0.27, 1.00,0.00, 1.38,0.00, 2.10,0.0, "FSTTB"]
36. FSOTB = [0.0,0.00, 0.48,0.00, 0.62,0.00, 0.69,0.00, 0.84,0.00 ,
0.92,0.18, 1.00,0.92, 1.38,1.00, 2.10,1.00, "FSOTB"]
```

For each of the above parameters, additional explanation is explained below:

- LAT** (real, degrees): the geographic latitude, required for calculation of the correct astronomic day length for each day of the simulation; day length may influence crop-physiological processes such as the allocation of carbohydrates, the initialization of flowering, etc. It assumed that day length is identical throughout the simulated catchment area; in very large catchments it might, theoretically, be preferable to calculate the day length for each individual grid cell. Actual day length calculations are based on the FORTRAN subroutine SASTRO (Van Kraalingen, 1995) which is, in turn, based on Goudriaan and van Laar (1994, around p.30).
- LUE** (real, g biomass MJ<sup>-1</sup> of intercepted solar radiation): the light use efficiency of the crop or variety that is simulated. For rice, the default value is 3.0 (Shibu et al., 2010). To account for different varieties, management imperfections or adverse conditions, a different value may be obtained by local calibration with observed rice yields, preferably of well-managed experiments that approach potential or water limited growing conditions, i.e. in absence of nutrient shortages, toxic elements or pests and diseases. The potential yield level is determined by the growth-defining factors, i.e. incoming solar radiation, temperature and characteristics of the crop when

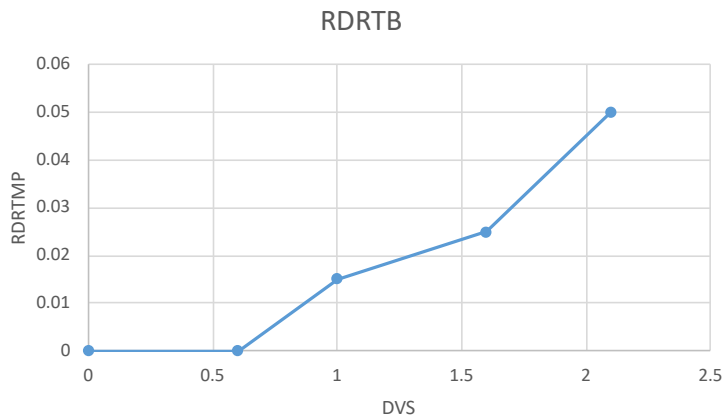
---

the crop is optimally supplied with water and nutrients and is completely protected against growth-reducing factors. Water-limited and nutrient-limited yield levels are lower than the potential, due to a suboptimal supply of water and/or nutrients, respectively. The actual production level is determined by actual supplies of water and nutrients, and by the degree that the crop is protected against growth-reducing factors or escapes their effects (Van Ittersum and Rabbinge, 1997).

Alternatively, if the LUE of wflow\_lintul is calibrated against actual farm yields, the model may be used to predict *actual* crop yields, instead of *water-limited* or *potential* crop yields, taking into the account the average occurrence of nutrient shortages, toxic elements or pests and diseases in the relevant region and over the relevant period of time. The latter was carried out for the Brantas catchment in central Java, Indonesia, in the context of the G4INDO project, resulting in LUE = 2.47.

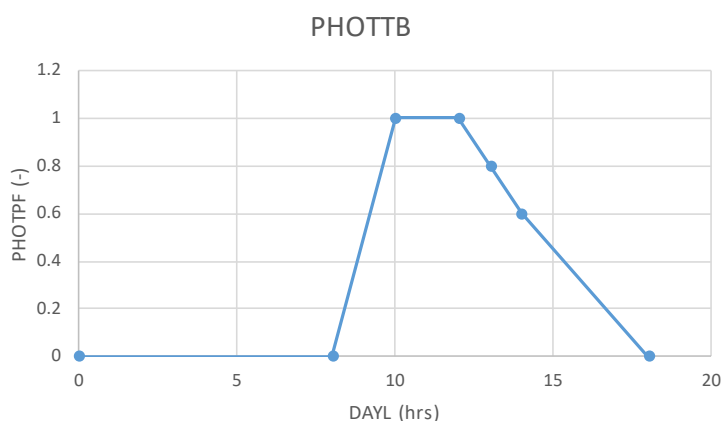
12. **TSUMI** (real, degree days): the initial temperature sum (at the moment of transplanting, only relevant for transplanted rice).
13. **K** (real, dimensionless): the light extinction coefficient (for rice leaves). Light interception in LINTUL increases with LAI according to a negative exponential pattern, characterized by a crop-specific light extinction coefficient ( $K$ ,  $\text{m}^2$  (ground)  $\text{m}^{-2}$  (leaf)): an implementation of Lambert-Beer's law.
14. **SLAC** (real,  $\text{m}^2$  (leaf)  $\text{g}^{-1}$  (leaf)): specific leaf area constant, introduced by Shibu et al. (2010). SLAC is multiplied by *SLACF*, a Leaf area correction function as a function of development stage to obtain *SLA*, the specific leaf area (real,  $\text{m}^2$  (leaf)  $\text{g}^{-1}$  (leaf)).
15. **TSUMAN** (real, degree days): the temperature sum required for the crop to reach anthesis.
16. **TSUMMT** (real, degree days): the temperature sum required for the crop to develop from anthesis to maturity (rice ready for harvest).
17. **TBASE** (real, degrees centigrade): since many growth processes are temperature dependent above a certain threshold temperature, only temperatures above a certain minimum temperature (*TBASE*, real, degrees centigrade) lead to increase in temperature sum/crop phenological development. For rice, *TBASE* is 8 °C (Shibu et al., 2010) hence below that temperature, no crop development takes place.
18. **RGRL** (real, dimensionless): the relative (daily) growth rate of leaf area during the exponential growth phase, expressed per degree-day.
19. – 22.: **WLVTI**, **WSTI**, **WRTLI**, **WSOI**: the initial weight (at transplanting time) of green leaves, stems, roots and storage organs (i.e. rice grains), respectively, in  $\text{g DM m}^{-2}$ .
23. **DVSDR** (real, dimensionless): value of the crop development stage (*DVS*) above which the death of leaves and roots sets in.
24. **RDRRT** (real, dimensionless): relative (daily) death rate of roots.
25. **RDRSHM** (real, dimensionless): maximum relative (daily) death rate of leaves due to shading.
26. **LAICR** (real, dimensionless): value of the crop Leaf Area Index (*LAI*) above which mutual shading of leaves occurs which may in turn influence the death rate of leaves.
27. **ROOTDM\_mm** (real, mm): maximum rooting depth for a rice crop. Instead of m in LINTUL3 (Shibu et al., 2010), the unit was changed to mm in wflow\_lintul, for easy exchange with wflow\_sbm.

28. **RRDMAX\_mm** (real, mm d<sup>-1</sup>): maximum daily increase in rooting depth (m) for a rice crop. Instead of m in LINTUL3 (Shibu et al., 2010), the unit was changed to mm in wflow\_lintul, for easy exchange with wflow\_sbm.
29. **ROOTDI\_mm** (real, mm): initial rooting depth (after transplanting). Instead of m in LINTUL3 (Shibu et al., 2010), the unit was changed to mm in wflow\_lintul, for easy exchange with wflow\_sbm.
30. **RDRTB** (real, dimensionless): interpolation table defining the relative (daily) death rate of leaves (RDRTMP) as a function of Developmental Stage (DVS, Figure 1). Adopted by Shibu et al. (2010) from Bouman et al. (2001).



**Figure 1** The *RDRTB* parameter (list) defines the relative (daily) death rate of leaves (*RDRTMP*) as a function of crop Developmental Stage (*DVS*).

31. **PHOTTB** (real, dimensionless): interpolation table defining the modifying effect of photoperiodicity on the phenological development rate (from LINTUL3/Shibu et al. 2010 - original source unclear) as a function of daylength (*DAYL*, hours).



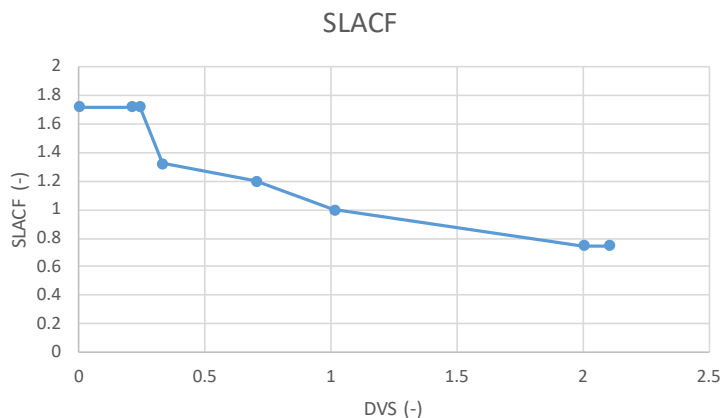
**Figure 2** The *PHOTTB* parameter (table) defines the modifying effect of photoperiodicity on the phenological development rate as a function of day length (*DAYL*).

32. **SLACF** (real, dimensionless): interpolation table defining the leaf area correction factor (*SLACF*) as a function of development stage (*DVS*; Drenth et al., 1994). The Specific Leaf Area Constant (*SLAC*, see parameter 14.) is multiplied with a correction factor to obtain the specific leaf area

(SLA); this correction factor, in turn, is obtained by linear interpolation in the SLACF table (Figure 3), using the relevant value of DVS as independent variable.

$$SLA = self.SLAC * self.SLACF.lookup\_linear(self.DVS) \quad (Eq. 2)$$

where `self.SLACF.lookup_linear(self.DVS)` results in the relevant value of the correction factor by linear interpolation based on DVS. Linear interpolation in this way is possible since SLACF is changed from a parameter (list) into an instance of the class `Interpol_Obj` (not described in this document) upon model initialization.

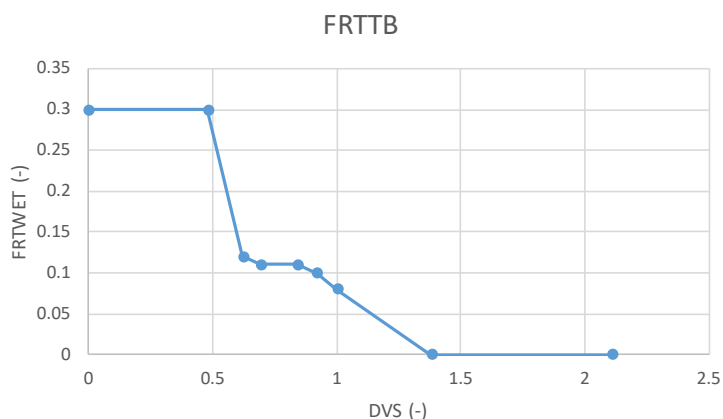


**Figure 3** The SLACF interpolation table defines the leaf area correction factor (SLACF) as a function of development stage (DVS).

33. **FRTTB** (real, dimensionless): interpolation table defining the fraction of daily dry matter production allocated to root growth (FRTWET), in absence of water shortage, as a function of development stage (DVS; Figure 4). FRTWET is obtained by linear interpolation in FRTTB, with DVS as the independent variable.

$$FRTWET = self.FRTTB.lookup\_linear(self.DVS) \quad (Eq. 3)$$

Linear interpolation is possible since FRTTB is changed from a parameter (list) into an instance of the class `Interpol_Obj` (not described in this document) upon model initialization.



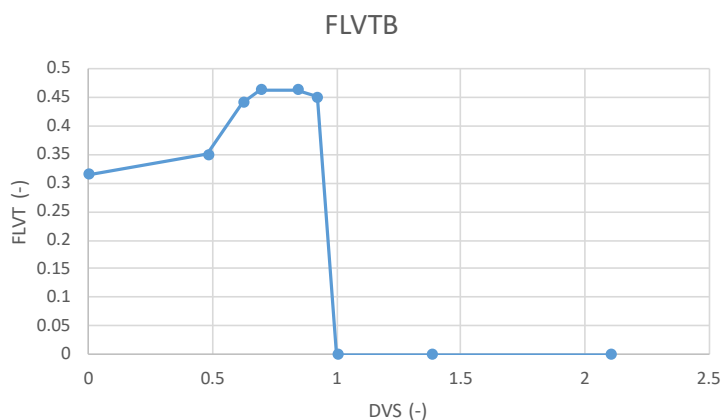
**Figure 4** The FRTTB interpolation table defines the fraction of daily dry matter production allocated to root growth (FRTWET) in absence of water shortage as a function of the crop development stage (DVS).



34. **FLVTB** (real, dimensionless): interpolation table defining the fraction of daily dry matter production allocated to growth of leaves (FLVT), in absence of water shortage, as a function of development stage (DVS; Figure 5). FLVT is obtained by linear interpolation in FLVTB, with DVS as the independent variable:

$$\text{FLVT} = \text{self.FLVTB.lookup\_linear}(\text{self.DVS}) \quad (\text{Eq. 4})$$

Linear interpolation is possible since FLVTB is changed from a parameter (list) into an instance of the class `Interpol_Obj` (not described in this document) upon model initialization.

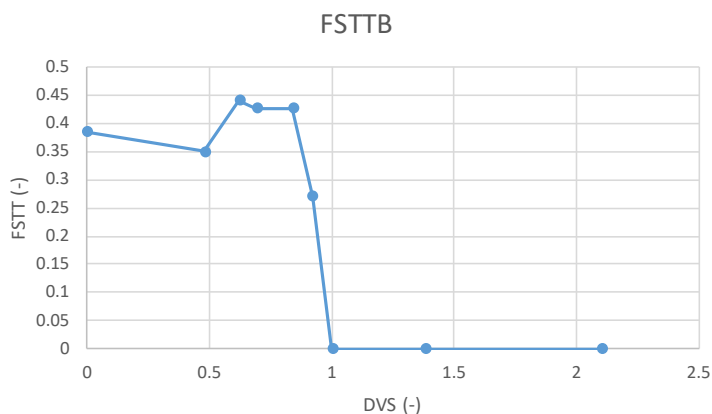


**Figure 5** The FLVTB interpolation table defines the fraction of daily dry matter production allocated to growth of leaves (FLVT) in absence of water shortage, as a function of development stage.

35. **FSTTB** (real, dimensionless): interpolation table defining the fraction of daily dry matter production allocated to growth of stems (FSTT), in absence of water shortage, as a function of development stage (DVS; Figure 6). FSTT is obtained by linear interpolation in FSTTB, with DVS as the independent variable:

$$\text{FSTT} = \text{self.FSTTB.lookup\_linear}(\text{self.DVS}) \quad (\text{Eq. 5})$$

Linear interpolation is possible since FSTTB is changed from a parameter (list) into an instance of the class `Interpol_Obj` (not described in this document) upon model initialization.

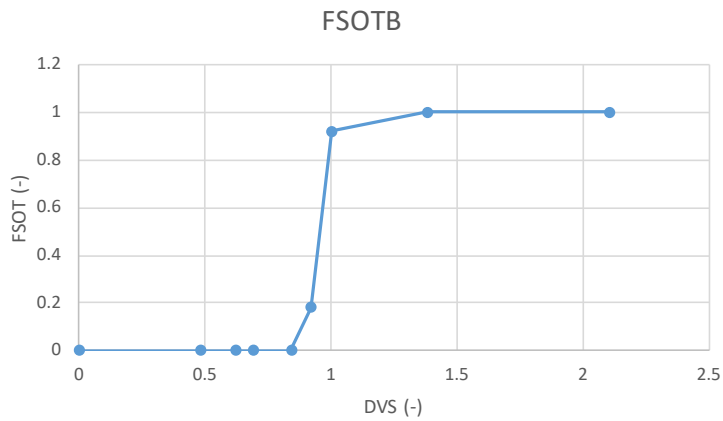


**Figure 6** The FSTTB interpolation table defines the fraction of daily dry matter production allocated to growth of stems (FSTT) in absence of water shortage, as a function of the crop development stage (DVS).

36. **FSOTB** (real, dimensionless): interpolation table defining the fraction of daily dry matter production allocated to growth of stems (*FSOT*), in absence of water shortage, as a function of development stage (*DVS*; Figure 7). *FSOT* is obtained by linear interpolation in *FSOTB*, with *DVS* as the independent variable:

$$FSOT = self.FSOTB.lookup\_linear(self.DVS) \quad (Eq. 6)$$

Linear interpolation is possible since *FSOTB* is changed from a parameter (list) into an instance of the class *Interpol\_Obj* (not described in this document) upon model initialization.



**Figure 7** The *FSOTB* interpolation table defines the fraction of daily dry matter production allocated to growth of stems (*FSOT*), in absence of water shortage, as a function of the crop development stage (*DVS*).

---

## 6 Model forcing data

In terms of meteorological data, `wflow_lintul` requires daily values of:

- Solar radiation (`IRRAD`;  $\text{kJ m}^{-2} \text{d}^{-1}$ )
- Average temperature (`T`; °C)
- Precipitation (`RAIN`, mm)

Due the raster-based nature of `wflow_lintul`, forcing data should be presented in the **PCRaster map format**, for each simulated day.

For water-limited simulations, LINTUL-type models normally also require data of **vapour pressure** and **wind speed** for calculating evapotranspiration. However, in the case of `wflow_lintul`, actual (rice) crop transpiration (`Transpiration`;  $\text{mm d}^{-1}$ ) and potential (rice) crop transpiration (`PotTrans`;  $\text{mm d}^{-1}$ ) under water-limited crop growth are calculated by the `wflow_sbm` hydrologic model which then supplies these variables to `wflow_lintul` via the BMI (Basic Model Interface). They are then used for calculating `TRANRF`, a factor that describes the reducing effect of water shortage on biomass production, according to Eq. 36 (Section 7.7).

`wflow_sbm`, in turn, requires reference evapotranspiration as a forcing variable. Water content in the rooted zone (`self.WA`) is simulated by `wflow_sbm` and then transferred to `wflow_lintul` via the BMI. For `wflow_sbm` being able to perform these calculations, `wflow_lintul` supplies it with daily values of `LAI` ( $\text{m}^2 \text{leaf m}^{-2} \text{ground}$ ) and rooting depth to `wflow_sbm`, again via the BMI.

---

# 7 Description of core wflow\_lintul model code

## 7.1 Phenology

Phenological development of crops is generally closely related to thermal time, i.e. the accumulated number of degree-days after emergence. Instead of simply adding up degree-days, a daily effective temperature ( $DTEFF$ , °C d) is used in wflow\_lintul however, since many growth processes are only temperature dependent, or only occur, above a certain threshold temperature.  $DTEFF$  is calculated according to:

$$DTEFF = \text{ifthenelse}(\text{Warm\_Enough}, \text{DegreeDay}, 0.) \quad (\text{Eq. 7})$$

with

- $DTEFF$  the daily effective temperature (°C d)
- `ifthenelse` a PCRaster statement that, based on the value of a Boolean variable (first expression between parentheses), assigns the value of the second or the third expression between parentheses to its result, if the value of the Boolean variable is `True` or `False`, respectively.
- `Warm_Enough` a Boolean variable that equals `True` when the daily average temperature  $T$  is larger than or equal to  $TBASE$ , the base temperature for a rice crop (8°C).
- `DegreeDay` is the daily average temperature reduced with  $TBASE$ :  $DegreeDay = self.T - self.TBASE$

Thus, if  $T$  is great than  $TBASE$ ,  $DTEFF$  is set equal to `DegreeDay`; in all other cases it is set to 0. In addition, before  $DTEFF$  is added to  $TSUM$  (state variable), it is corrected for the potentially modifying effect of day length (photoperiodicity) resulting in the daily increase in temperature sum  $RTSUMP$  (°C d), according to:

$$RTSUMP = DTEFF * PHOTPF \quad (\text{Eq. 8})$$

where:

- $RTSUMP$  is the daily increase in temperature sum (°C d), modified by the influence of photoperiod
- $DTEFF$  the daily effective temperature (°C d)
- $PHOTPF$  is a correction factor that accounts for the modifying influence of day length on crop development via  $DTEFF$  (see also Figure 2); it is defined as a function of day length ( $DAYL$ ).  $PHOTPF$  is less than 1 if day length ( $DAYL$ , hours) is shorter than 10 hours or longer than 12 hours. Day lengths between 10-12 hours have no modifying influence on  $DTEFF$ ; phenological development is thus slowed down in such cases.  $DAYL$  in wflow\_lintul is calculated by the function `astro_py` (not described here), a Python implementation of a FORTRAN subroutine from the Wageningen school of models, dating back to Spitters et al. (1989) and likely even further.

Calculation of  $TSUM$  (model state variable) by accumulating daily values of  $RTSUMP$  is then done according to:

$$\begin{aligned} self.TSUM = & (self.TSUM + \text{ifthenelse}(\text{CropStartNow}, \\ & \text{scalar}(self.TSUMI), 0.) + \text{ifthenelse}(\text{EMERG}, RTSUMP, 0.)) * \\ & \text{ifthenelse}(\text{CropHarvNow}, \text{scalar}(0.), 1.) \end{aligned} \quad (\text{Eq. 9})$$

with:

- `self.TSUM` the temperature sum (a crop state variable, °C d)
- `ifthenelse` a PCRaster statement that, based on the value of a Boolean variable (first expression between parentheses), assigns the value of the second or the third expression between parentheses to its result, if the value of the Boolean variable is `True` or `False`, respectively.
- `CropStartNow`: a Boolean variable that equals `True` on the moment of crop growth initiation

- `scalar(self.TSUMI)`: a piece of Python PCRaster code, specifying that the variable type of `TSUMI` is a scalar
- `TSUMI` (real, degree days): the initial temperature sum (at the moment of transplanting, only relevant for transplanted rice).
- `EMERG` is a Boolean variable that indicates whether crop growth occurs; it equals `True`, when three conditions are met:
  - crop growth has been initiated
  - the water content of the soil is larger than the water content at permanent wilting point
  - `LAI` is greater than 0.
- `CropHarvNow` a Boolean variable that only equals `True` on the day that the crop is being harvested

Thus, in Eq. 9, the state of `TSUM` of the previous day is increased (on the current day) with the initial temperature at the moment of transplanting/initiation of crop growth, and increased with the daily increase in (effective) temperature during the subsequent period of crop growth. At the moment of harvest (when `CropHarvNow = True`), the third `ifthenelse` statement in Eq. 9 will return a zero value, hence multiplying `TSUM` with zero (effectively resetting it).

Regarding the influence of `TSUM` on rice crop development two crop-specific parameters are of paramount importance, since assimilates are partitioned differently over the different plant organs before flowering (vegetative growth) than after flowering (generative growth):

- `TSUMAN`, the value of `TSUM` at which crop anthesis is initiated (1420 °C d for rice variety IR72)
- `TSUMMT`, the change in `TSUM` required for the crop to develop from anthesis to maturity (hence ripened rice grain can be harvested) - 580 °C d for rice variety IR72.

In `LINTUL1` and `LINTUL2`, `TSUM` controls all processes that are influenced by crop phenological development. However, Shibu et al. (2010) derived important parts of `LINTUL3` (for rice) from `ORYZA2000` (Bouman et al., 2001). In `ORYZA2000`, most processes are steered by a variable called `DVS` (development stage; -). The development stage of a plant defines its physiological age and is characterized by the formation of the various organs and their appearance. The key development stages for rice distinguished in `ORYZA2000` are emergence (`DVS = 0`), panicle initiation (`DVS = 0.65`), flowering (`DVS = 1`), and physiological maturity (`DVS = 2`; Bouman et al., 2001). Shibu et al. incorporated `DVS` into `LINTUL` which is why some processes are controlled directly by `TSUM` and others are controlled by its derived variable `DVS` – a situation that seems to offer scope for future improvement. In `wflow_lintul`, the key development stages emergence, flowering and physiological maturity coincide with `TSUM = 0`, `TSUM = 1420` (flowering) and `TSUM = 2000` (140+580; crop maturity), respectively. Panicle initiation (at `DVS = 0.65`) is not a significant event in `wflow_lintul`. `DVS` is calculated from `TSUM` according to the following three equations:

$$\text{DVS\_veg} = \text{self.TSUM} / \text{self.TSUMAN} * \text{ifthenelse}(\text{CropHarvNow}, \text{scalar}(0.), 1.) \quad (\text{Eq. 10})$$

$$\text{DVS\_gen} = (1. + (\text{self.TSUM} - \text{self.TSUMAN}) / \text{self.TSUMMT}) * \text{ifthenelse}(\text{CropHarvNow}, \text{scalar}(0.), 1.) \quad (\text{Eq. 11})$$

$$\text{self.DVS} = \text{ifthenelse}(\text{Vegetative}, \text{DVS\_veg}, 0.) + \text{ifthenelse}(\text{Generative}, \text{DVS\_gen}, 0.) \quad (\text{Eq. 12})$$

where:

- `DVS_veg`: `DVS` during the vegetative crop stage
- `CropHarvNow` a Boolean variable that equals `True` when the crop is mature or has reached a fixed pre-defined harvest date.
- `scalar(0.)` is a Python PCRaster statement which will return zeros for all simulated raster cells.
- `DVS_gen`: `DVS` during the generative crop stage
- `TSUMAN`, the value of `TSUM` at which crop anthesis is initiated (1420 °C d for rice variety IR72 – parameter 15. in Section 5)

- `TSUMMT`, the change in `TSUM` required for the crop to develop from anthesis to maturity (hence ripened rice grain can be harvested) - 580 °C d for rice variety IR72 (parameter 16. in Section 5).
- `DVS`, the crop (phenological) development stage

Hence, `DVS` is calculated as the sum of `DVS_veg` and `DVS_gen`; it equals 1 (flowering) if `TSUM` reaches `TSUMAN` and equals 2 (maturity) if `TSUM` reaches the sum `TSUMAN + TSUMMT`. `DVS_veg` and `DVS_gen` are both reset (multiplied with zero) when the crop is harvested.

## 7.2 Photosynthesis and crop growth

As outlined in the Introduction, overall assimilate production rate in `wflow_lintul` is calculated as the product of the Photosynthetically Active (solar) Radiation (PAR) that is intercepted by the crop canopy and a fixed Light-Use Efficiency (LUE; Russell et al., 1989). This way of estimating (daily) biomass production in the LINTUL models is reflected in Equation 1 and may be considered the core of the `wflow_lintul` model:

$$GTOTAL = self.LUE * PARINT * TRANRF \quad (\text{Eq. 1, p.1})$$

with:

- `GTOTAL` the overall daily growth rate of the crop (g m<sup>-2</sup> d<sup>-1</sup>)
- `self.LUE` a constant light use efficiency<sup>2</sup> (g MJ<sup>-1</sup>)
- `PARINT` the daily intercepted photosynthetically active radiation (MJ m<sup>-2</sup> d<sup>-1</sup>)
- `TRANRF` the 'transpiration reduction factor', i.e. the reducing (-) impact of water shortage on biomass production (Eq. 36)

The daily intercepted PAR (`PARINT`, MJ m<sup>-2</sup> d<sup>-1</sup>), is calculated as:

$$PARINT = \text{ifthenelse}(\text{Not\_Finished}, 0.5 * self.IRRAD * 0.001 * (1. - \exp(-self.K * self.LAI)), 0.) \quad (\text{Eq. 13})$$

with:

- `ifthenelse` a PCRaster statement that, based on the value of a Boolean variable (first expression between parentheses), assigns the value of the second or the third expression between parentheses to its result, if the value of the Boolean variable is `True` or `False`, respectively.
- `Not_Finished` a Boolean variable that indicates whether the crop is still growing and developing (`True`) or not (`False`).
- 0.5 a factor to account for the fraction of photosynthetically active radiation in the incident solar radiation. About 50% (in terms of energy) of the frequency spectrum of incident solar radiation can be used for photosynthesis by green plants.
- `IRRAD`: incident solar radiation (MJ m<sup>-2</sup> d<sup>-1</sup>) as measured by e.g. a weather station.
- `K` a crop and variety-specific light extinction coefficient (-)
- `LAI` the leaf area index (m<sup>2</sup> leaf m<sup>-2</sup> ground).

Hence, as long as the crop is still growing (`Not_Finished = True`) the daily intercepted PAR is calculated according to the middle part (between the commas) of the `ifthenelse` statement in Eq. 12. In all other cases, it equals zero.

## 7.3 Partitioning of overall biomass growth over the various plant organs (calculation of growth rates)

The overall daily growth of the crop (`GTOTAL`, g m<sup>-2</sup> d<sup>-1</sup>) is partitioned over growth of leaves, stems, storage organs (grains in the case of rice) and roots:

<sup>2</sup> The prefix 'self' in `self.LUE` is part of the Python object-oriented syntax and has no mathematical meaning.

$$\text{RWLVG} = \text{ifthenelse}(\text{EMERG}, \text{GTOTAL} * \text{FLV} - \text{DLV}, \text{scalar}(0.)) \quad (\text{Eq. 14})$$

$$\text{RWST} = \text{ifthenelse}(\text{EMERG}, \text{GTOTAL} * \text{FST}, \text{scalar}(0.)) \quad (\text{Eq. 15})$$

$$\text{RWSO} = \text{ifthenelse}(\text{EMERG}, \text{GTOTAL} * \text{FSO}, \text{scalar}(0.)) \quad (\text{Eq. 16})$$

$$\text{RWRT} = \text{ifthenelse}(\text{EMERG}, \text{GTOTAL} * \text{FRT} - \text{DRRT}, \text{scalar}(0.)) \quad (\text{Eq. 17})$$

where:

- `ifthenelse` a PCRaster statement that, based on the value of a Boolean variable (first expression between parentheses), assigns the value of the second or the third expression between parentheses to its result, if the value of the Boolean variable is `True` or `False`, respectively.
- `RWLVG`, `RWST`, `RWSO` and `RWRT` are the daily growth rates of leaves, stems, storage organs (i.e. rice grains) and roots, respectively ( $\text{g m}^{-2} \text{d}^{-1}$ ); their joint values should add up to `GTOTAL`, given the law of mass conservation.
- `GTOTAL` is the overall daily growth rate of the crop ( $\text{g m}^{-2} \text{d}^{-1}$ )
- `EMERG` is a Boolean variable that indicates whether crop growth occurs; it equals `True`, when three conditions are met:
  - crop growth has been initiated
  - the water content of the soil is larger than the water content at permanent wilting point
  - `LAI` is greater than 0.
- `scalar(0.)` is a Python PCRaster statement which will return zeros for all simulated raster cells.
- `FLV`, `FST`, `FSO` and `FRT` are the fractions (-) of the overall biomass growth rate (`GTOTAL`;  $\text{g m}^{-2} \text{d}^{-1}$ ) allocated to leaves, stems, storage organs (rice grains) and roots, respectively. They are related to the phenological development stage of the crop (`DVS`, see under "phenology"), following relationships defined in the parameter interpolation tables `FLVTB`, `FSTTB`, `FSOTB`, and `FRTTB` (see list of model parameters, parameter 33.-36.). If water shortage occurs, growth of belowground and aboveground crop parts is modified with the factors `FRTMOD` and `FSHMOD`, respectively (calculation of `FRTMOD` and `FSHMOD` is discussed further down, in Eqs. 37. and 38.):

$$\text{FLV} = \text{FLVT} * \text{FSHMOD} \quad (\text{Eq. 18})$$

$$\text{FST} = \text{FSTT} * \text{FSHMOD} \quad (\text{Eq. 19})$$

$$\text{FSO} = \text{FSOT} * \text{FSHMOD} \quad (\text{Eq. 20})$$

$$\text{FRT} = \text{FRTWET} * \text{FRTMOD} \quad (\text{Eq. 21})$$

where `FLVT`, `FSTT`, `FSOT` and `FRTWET` are the (raw) interpolated allocation fractions for leaves, stems, storage organs (rice grains) and roots, respectively and `FLV`, `FST`, `FSO` and `FRT` are the final allocated fractions, after modification for water shortage (if any).

- `DLV` and `DRRT` are the death rates of leaves and roots, respectively ( $\text{g m}^{-2} \text{d}^{-1}$ ); their calculation is described by Eqs. 22. and 23 below.

Summarizing: as long as crop growth occurs (i.e. if `EMERG = True`), growth rates of the different plant organs are calculated by multiplying overall crop growth with certain organ-specific fractions, depending on crop development stage via pre-defined interpolation functions, and on water availability. Net growth rates are obtained by diminishing the growth with the weight of the foliage that has died.

The daily change in leaf weight as a consequence of the dying of foliage is calculated according to:

$$\text{DLV} = \text{self.WLVG} * \text{RDR} \quad (\text{Eq. 22})$$

with:

- `self.WLVG` the total green leaf biomass ( $\text{g m}^{-2}$ );

- RDR the relative (daily) death rate of leaves (-). The calculation of RDR is described in Section 1.3 below (Daily change in Leaf Area Index). Calculation of RDR by Eqs. 31.-33. Below.

The daily change in root weight as a consequence of the dying of roots is calculated according to:

$$DRRT = \text{ifthenelse}(\text{Roots\_Dying}, \text{self.WRT} * \text{self.RDRRT}, \text{scalar}(0.)) \quad (\text{Eq. 23})$$

with:

- self.WLVG the total roots biomass ( $\text{g m}^{-2}$ );
- RDRRT the relative (daily) death rate of roots (-, see list of model parameters, parameter 24.).

Now that the net (mass) growth rates of leaves, stems, storage organs (rice grain) and roots are known, their respective masses (model state variables) can also be calculated, according to:

$$\text{self.WLVG} = (\text{self.WLVG} + \text{ifthenelse}(\text{CropStartNow}, \text{self.WLVGI}, \text{scalar}(0.)) + \text{RWLVG}) * (1. - \text{scalar}(\text{CropHarvNow})) \quad (\text{Eq. 24})$$

$$\text{self.WST} = (\text{self.WST} + \text{ifthenelse}(\text{CropStartNow}, \text{self.WSTI}, \text{scalar}(0.)) + \text{RWST}) * (1. - \text{scalar}(\text{CropHarvNow})) \quad (\text{Eq. 25})$$

$$\text{self.WSO} = (\text{self.WSO} + \text{ifthenelse}(\text{CropStartNow}, \text{self.WSOI}, \text{scalar}(0.)) + \text{RWSO}) * (1. - \text{scalar}(\text{CropHarvNow})) \quad (\text{Eq. 26})$$

$$\text{self.WRT} = (\text{self.WRT} + \text{ifthenelse}(\text{CropStartNow}, \text{self.WRTL I}, \text{scalar}(0.)) + \text{RWRT}) * (1. - \text{scalar}(\text{CropHarvNow})) \quad (\text{Eq. 27})$$

with:

- WLVG, WST, WSO and WRT the weights of green leaves, dead leaves, stem, storage organs and roots ( $\text{g m}^{-2}$ )
- ifthenelse a PCRaster statement that, based on the value of a Boolean variable (first expression between parentheses), assigns the value of the second or the third expression between parentheses to its result, if the value of the Boolean variable is True or False, respectively.
- CropStartNow: a Boolean variable that equals True on the moment of crop growth initiation
- WLVGI, WSTI, WSOI, WRTL I the initial weights of green leaves, dead leaves, stem, storage organs and roots ( $\text{g m}^{-2}$ ), i.e. at the moment of transplanting
- RWLVG, RWST, RWSO and RWRT the daily growth rates of leaves, stems, storage organs (i.e. rice grains) and roots, respectively ( $\text{g m}^{-2} \text{d}^{-1}$ ); their joint values should add up to GTOTAL, given the law of mass conservation.
- CropHarvNow a Boolean variable that only equals True on the day that the crop is being harvested

Eqs. 21-24 can be described as follows: upon the initiation of crop growth (when CropStartNow = True), the initial weights of the different crop organs are added (remark: the 'real' initial values at the start of the simulation, even before these initial weights are added, are defined by a number of PCRaster maps that are present for each model state variable in the "inmaps" folder of the relevant wflow case directory). Furthermore, the daily increase in weight of each plant organ is added to the relevant state variable – it should be noted that the daily increase in weight is greater than nil if EMERG = TRUE (see Eqs. 13.-16.). Finally, on the moment of harvest (CropHarvNow = True), all organ weights are multiplied with zero, thereby effectively harvesting the crop / resetting the state variables to zero



---

## 7.4 Daily change in Leaf Area Index (LAI)

Whereas the daily increase in leaf *biomass* was already explained above, the daily increase in leaf *area* (Leaf Area Index, LAI,  $\text{m}^2 \text{m}^{-2}$ ) is simulated as:

$$\begin{aligned} \text{GLAI} = & \text{ifthenelse}(\text{LetsGro}, \text{self.LAII}, \text{scalar}(0.)) + & \text{(Eq. 28)} \\ & \text{ifthenelse}(\text{Juv\_or\_Harv}, \text{self.LAI} * (\exp(\text{self.RGRL} * \text{DTEFF}) - 1.)) \\ & * \text{TRANRF} + \text{ifthenelse}(\text{Adt\_or\_Harv}, \text{SLA} * \text{GLV}, \text{scalar}(0.)) \end{aligned}$$

with:

- GLAI the (daily) growth in LAI ( $\text{m}^2 \text{m}^{-2} \text{d}^{-1}$ )
- LetsGro a Boolean variable that, as soon as it becomes "True", triggers the initiation of crop growth;
- LAII ( $\text{m}^2 \text{m}^{-2}$ ) the initial value of LAI (leaf area index)
- scalar (0.) is a Python PCRaster statement which will return zeros for all simulated raster cells.
- Juv\_or\_Harv a Boolean variable indicating that the crop is juvenile or is being harvested
- RGRL the relative (daily) growth rate of leaf area (-), expressed per degree-day ( $(^\circ\text{Cd})^{-1}$ )
- DTEFF the daily effective temperature ( $^\circ\text{C d}$ , see section 2.1 Phenology).
- TRANRF the 'transpiration reduction factor', i.e. the relative impact of water shortage on biomass production (Eq. 36)
- Adt\_or\_Harv a Boolean variable indicating whether the crop is adult or is being harvested (True) or none of those two (False).
- SLA the specific Leaf area ( $\text{m}^2/\text{g}$ ). The model calculates the growth of leaf area by multiplying the simulated increase in leaf weight (GLV,  $\text{g m}^{-2} \text{d}^{-1}$ ) by the specific leaf area of new leaves (SLA,  $\text{m}^2 \text{g}^{-1}$ ).
- GLV the simulated increase in leaf weight (GLV,  $\text{g m}^{-2} \text{d}^{-1}$ )

In short: upon the initiation of crop growth (when the internal auxiliary variable LetsGro equals 'True'), the daily leaf growth rate GLAI is set equal to the initial leaf area index (LAII). Subsequently, during the juvenile stage, LAI grows exponentially, with the exponential growth rate being determined by RGRL and the daily effective temperature. During the adult stage, the daily net growth in leaf area follows the daily net (mass) growth rates of leaves ( $\text{g m}^{-2} \text{d}^{-1}$ , see also Eq. 2) by multiplication with SLA, the specific leaf area ( $\text{m}^2/\text{g}$ ; see Ad 32).

Now that the daily increase in LAI is known, it seems in principle possible to calculate the LAI itself (model state variable), according to:

$$\text{self.LAI} = (\text{self.LAI} + \text{GLAI} - \text{DLAI}) * \text{ifthenelse}(\text{CropHarvNow}, \text{scalar}(0.), 1.) \quad \text{(Eq. 29)}$$

with:

- LAI the leaf area index ( $\text{m}^2 \text{m}^{-2}$ )
- GLAI the (daily) growth in LAI ( $\text{m}^2 \text{m}^{-2} \text{d}^{-1}$ )
- DLAI the daily **decrease** in LAI (DLAI,  $\text{m}^2 \text{m}^{-2} \text{d}^{-1}$ ) from dying of leaves – its calculation is discussed under Eq. 30 below
- CropHarvNow a Boolean variable that only equals True on the day that the crop is being harvested

The daily **decrease** in LAI (DLAI,  $\text{m}^2 \text{m}^{-2} \text{d}^{-1}$ ) from dying of leaves is, analogous to the calculation of DLV (the death rate of leaves in terms of *mass*), calculated as:

$$\text{DLAI} = \text{self.LAI} * \text{RDR} \quad \text{(Eq. 30)}$$

with:

- LAI the leaf area index ( $\text{m}^2 \text{m}^{-2}$ ; state variable);

- RDR the relative (daily) decline in LAI due dying of leaves (-). RDR in turn depends on two terms, the relative (daily) death rate of leaves due to aging (RDRDV, -) and the relative (daily) death rate of leaves due to mutual shading (RDRSH, -), according to:
- $RDR = \max(RDRDV, RDRSH)$  (Eq. 31)

The relative (daily) death rate of leaves due to aging (RDRDV, -), in turn, is calculated following a number of steps, starting with:

$$RDRDV = \text{ifthenelse}(\text{AtAndAfterAnthesis}, RDRTMP, \text{scalar}(0.)) \quad (\text{Eq. 32})$$

with:

- `AtAndAfterAnthesis` a Boolean variable that equals `True` if the crop has reached anthesis. Hence, if `AtAndAfterAnthesis` equals `True`, `RDRDV` is set equal to `RDRTMP`; in all other cases it is set to 0.
- `RDRTMP` is the relative (daily) death rate obtained by interpolation in the `RDRTB` table, with `DVS` as the independent variable.
- `scalar(0.)` is a Python `PCRaster` statement which will return zeros for all simulated raster cells.

The relative (daily) death rate of leaves due to mutual shading (RDRSH, -) is calculated according to:

$$RDRSH = \max(0., \text{self.RDRSHM} * (\text{self.LAI} - \text{self.LAICR}) / \text{self.LAICR}) \quad (\text{Eq. 33})$$

with:

- `RDRSHM` is a fixed daily relative death rate due to shading (-);
- `LAICR` is the critical LAI above which mutual shading starts to occur.

## 7.5 Root (depth) growth

Root depth growth can only occur if a number of (logical) conditions are simultaneously met:

$$\text{RootGrowth} = \text{Enough\_water} \ \& \ \text{BeforeAnthesis} \ \& \ \text{EMERG} \ \& \ \text{CanGrowDownward} \quad (\text{Eq. 34})$$

where:

- `RootGrowth` is a Boolean variable indicating whether root depth growth occurs (`True`) or not (`False`).
- `Enough_water` is a Boolean variable indicating whether there is water for crop growth (`True`) or not (`False`, this is the case when soil moisture content is at permanent wilting point).
- `BeforeAnthesis` is a Boolean variable indicating whether the crop growth has reached the anthesis stage in its phenological development (`True`) or not yet (`False`). It is calculated as:
- `BeforeAnthesis = self.TSUM < self.TSUMAN` with `self.TSUM` the accumulated temperature sum of the crop and `self.TSUMAN` the temperature sum required for the crop to reach anthesis. Hence, `BeforeAnthesis` remains `True` as long as the crop has not yet accumulated the temperature sum required for it to reach the anthesis development stage.
- `EMERG` is a Boolean variable that indicates whether crop growth occurs; it equals '`True`', when three conditions are met:
  - crop growth has been initiated
  - the water content of the soil is larger than the water content at permanent wilting point
  - LAI is greater than 0.
- `CanGrowDownward` is a Boolean variable indicating whether roots can grow further downward (`True`) or not (`False`). The latter is the case is the maximum rooting depth has been reached. It is calculated as:

---

- CanGrowDownward = self.ROOTD\_mm <= self.ROOTDM\_mm with self.ROOTD\_mm the rooting depth in mm and self.ROOTDM\_mm is the maximum rooting depth in mm

Hence, as long as the roots have not yet reached their maximum depth, CanGroDownward remains True. Actual root growth only occurs where RootGrowth equals True:

$$RROOTD\_mm = \text{ifthenelse}(\text{RootGrowth}, \text{self.RRDMAX\_mm}, \text{scalar}(0.)) \quad (\text{Eq. 35})$$

where:

- RROOTD\_mm is the daily increase in rooting depth (mm)
- RootGrowth is a Boolean variable indicating whether root depth growth occurs (True) or not (False).
- self.RRDMAX\_mm is the maximum daily increase in rooting depth (mm)
- scalar (0.) is a Python PCRaster statement which will return zeros for all simulated raster cells.
- So, if RootGrowth is True, the daily increase in rooting depth (mm) will be set equal to the maximum daily increase in rooting depth (mm); in all other cases it will be set to zero.

There is no simulation of a reducing effect of root death on rooting depth. Root death only impacts the weight of (living) roots as they are diminished with the (daily) death rate of roots (Eq. 23).

## 7.6 The influence of water stress on crop growth

As wflow\_lintul is a much simpler rice model than e.g. ORYZA2000 (Bouman et al., 2001), the effect of water stress on crop growth is also modelled in a simpler way - there is no mechanistic simulation of crop drought responses such as leaf rolling, or of events such as spikelet sterility. These events can only be taken into account indirectly, by calibrating model drought response to match observed yield and biomass data - a process that has presently (June 2018) not yet been entirely completed.

A central parameter in modelling the response to water shortage in wflow\_lintul is TRANRF, a factor that describes the reducing effect of water shortage on biomass production, having a direct impact on overall crop production (via Eq. 1) and leaf area growth (via Eq. 28). It is calculated according to:

$$\text{TRANRF} = \text{self.Transpiration} / \text{NOTNUL\_pcr}(\text{self.PotTrans}) \quad (\text{Eq. 36})$$

with:

- self.Transpiration the actual (rice) crop transpiration (mm d<sup>-1</sup>)
- self.PotTrans the potential (rice) crop transpiration (mm d<sup>-1</sup>)
- NOTNUL\_pcr a Python PCRaster implementation of a FST intrinsic function (Rappoldt and Van Kraalingen, 1996), returning 1 if the value between parentheses equals 0, to prevent zero division errors. In all others cases, it returns the unchanged value.

Both self.Transpiration and NOTNUL\_pcr(self.PotTrans) are calculated by the wflow\_sbm hydrologic model which then supplies these variables to wflow\_lintul via the BMI (Basic Model Interface).

Indirectly, TRANRF also impacts crop growth through influencing the crop's root/shoot ratio. If crops are subjected to drought or nutrient stress, it is commonly observed that they are inclined to invest more in root growth (in search of water or nutrients, arguably). In wflow\_lintul, TRANRF modifies root growth as soon as it drops below the value of 0.5 (i.e. as soon as crop transpiration is half or less its potential, under the given circumstances). Root growth is modified with a factor FRTMOD, via Eq. 21. FRTMOD is calculated according to:

$$\text{FRTMOD} = \max(1., 1. / (\text{TRANRF} + 0.5)) \quad (\text{Eq. 37})$$

---

with

- $TRANRF$  the 'transpiration reduction factor', i.e. the reducing impact of water shortage on biomass production

Similarly, shoot growth is modified with a factor  $FSHMOD$ , via Eqs. 18.20.  $FSHMOD$  is calculated according to:

$$FSHMOD = (1 - FRT) / (1 - FRT/FRTMOD) \quad (Eq. 38)$$

with

- $FRT$  the final allocated fraction of total crop growth allocated to the roots, after modification for water shortage (Eq. 21).
- $FRTMOD$  a factor describing the modifying effect of drought stress on root growth (Eq. 37).

---

## 8 References

- Bouman, B.A.M., van Keulen, H., van Laar, H.H., Rabbinge, R., 1996. The 'School of de Wit' crop growth simulation models: a pedigree and historical overview. *Agric. Syst.* 52, 171/198.
- Bouman, B.A.M., M.J. Kropff, T.P. Tuong, M.C.S. Wopereis, H.F.M. ten Berge, and H.H. van Laar, 2001. *ORYZA2000: Modelling lowland rice*. 235 p. International Rice Research Institute, Los Baños, Philippines, Wageningen University and Research Centre, Wageningen, The Netherlands.
- Drenth, H., Ten Berge, H.F.M. and Riethoven, J.J.M.(Editors). *ORYZA simulation modules for potential and nitrogen limited rice production*. SARP Research Proceedings-December, 1994. DLO-Research Institute for Agrobiological and Soil fertility, Wageningen, WAU-Department of Theoretical Production Ecology, Wageningen, IRRI-International Rice Research Institute, Los Banos, Pages: 197-210.
- Farré, I., Van Oijen, M., Leffelaar, P.A., Faci, J.M., 2000. Analysis of maize growth for different irrigation strategies in northeastern Spain. *Eur. J. Agron.* 12, 225-238.
- Goudriaan, J. and van Laar, H.H., 1994. *Modelling Potential Crop Growth Processes*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994. pp. 238.
- Habekotté, B., 1997. Description, parameterization and user guide of LINTULBRASNAP 1.1. A crop growth model of winter oilseed rape (*Brassica napus* L.). In: *Quantitative Approaches in Systems Analysis No. 9*. Wageningen Agricultural University, Wageningen, The Netherlands, 40 pp.
- Van Ittersum, M.K., Rabbinge, R., 1997. Concepts in production ecology for analysis and quantification of agricultural input-output combinations. *Field Crops Research* 52 (1997) 197-208
- Van Kraalingen, D.W.G., 1995. The FSE system for crop simulation, version 2.1. *Quantitative Approaches in Systems Analysis, No. 1*. C.T. de Wit Graduate School for Production Ecology and Resource Conservation, Wageningen University, The Netherlands, pp. 58.
- Monteith, J. L., 1969. Light interception and radiative exchange in crop stands. In *Physiological aspects of crop yield*, eds J. D. Easton, F. A. Haskins, C. Y. 194 B. A. M. Bouman, H. van Keulen, H. H. van Laar, R. Rabbinge Sullivan & C. H. M. van Bavel. American Society of Agronomy, Madison, Wisconsin. pp. 89-111.
- Monteith, J. L., 1990. Conservative behaviour in the response of crops to water and light. In *Theoretical Production Ecology: reflection and prospects*, eds R. Rabbinge, J. Goudriaan, H. van Keulen, F. W. T. Penning de Vries & H. H. van Laar. Simulation Monographs, PUDOC, Wageningen, The Netherlands. pp. 3-16.
- Naylor, R. L. Battisti, D.S., Vimont, D.J., Falcon, W.P., Burke, M.B., 2007. Assessing risks of climate variability and climate change for Indonesian rice agriculture. *PNAS* 104-19, p. 7752-7757
- Penning de Vries, F.W.T., D.M. Jansen, H.F.M. ten Berge & A. Bakema, 1989. Simulation of ecophysiological processes of growth of several annual crops. *Simulation Monographs* 29, Pudoc, Wageningen, 271 pp.
- Rappoldt, C., Van Kraalingen D.W.G., 1996. The FORTRAN Simulation Translator. FST version 2.0. Introduction and Reference Manual, in: B. A. M. Bouman and M. K. Van Ittersum (Eds.), *Quantitative Approaches in Systems Analysis No. 5*, Wageningen University and Research Center, Wageningen. pp. 178.
- Russell, G., Jarvis, P. G. & Monteith, J. L., 1989. Absorption of solar radiation and stand growth. In *Plant canopies, their growth form and function*. eds G. Russell, P. G. Jarvis & B. Marshall. Cambridge University Press. Cambridge. UK. pp. 21-39.
- Shibu, M.E., Leffelaar, P.A., van Keulen, H., Aggarwal, P.K., 2010. LINTUL3, a simulation model for nitrogen-limited situations: application to rice. *European Journal of Agronomy* 32, 255-271.
- Schapendonk, A.H.C.M., Stol, W., Van Kraalingen, D.W.G., Bouman, B.A.M., 1998. LINGRA, a sink/source model to simulate grassland productivity in Europe. *Eur. J. Agron.* 9, 87-100.
- Spitters C.J.T., Van Keulen H., Van Kraalingen, D.W.G., 1989. A simple and universal crop growth simulator: SUCROS87, in: Rabbinge R., Ward S.A., van Laar H.H.(Eds.), *Simulation and Systems Management in Crop Protection, III*, Simulation Monographs 32 (1) (1989)47-177.
- Spitters, C.J.T., 1990. Crop growth models: their usefulness and limitations. *Acta Hort.* 267, 349-368.
- Spitters, C.J.T., Schapendonk, A.H.C.M., 1990. Evaluation of breeding strategies for drought tolerance in potato by means of crop growth simulation. *Plant Soil* 123, 193-203.

- 
- Stroosnijder, L., 1982. Simulation of the soil water balance. In: Eds F.W.T. Penning de Vries & H.H. van Laar, Simulation of plant growth and crop production. Simulation Monographs, Pudoc, Wageningen, pp. 175-193.
- Van Rossum, G., 1995. Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.
- Vertessy, R.A. and H. Elsenbeer, "Distributed modelling of storm flow generation in an Amazonian rainforest catchment: effects of model parameterization," Water Resources Research, vol. 35, no. 7, pp. 2173-2187, 1999.
- Werner, M., Schellekens, J., Gijsbers, P., Van Dijk, M., Van den Akker, O. and Heynert K, 2013. The Delft-FEWS flow forecasting system, Environmental Modelling & Software; 40:65-77. DOI: 10.1016/j.envsoft.2012.07.010
- Wesseling, C.G., Karssenber, D.-J., Burrough, P.A., Van Deursen, W.P.A., 1996. Integrated dynamic environmental models inGIS: the development of a Dynamic Modelling language. Transactions in GIS, 11, 4048.



---

Corresponding address for this report:

P.O. Box 16  
6700 AA Wageningen  
The Netherlands  
T +31 (0)317 48 07 00  
[www.wur.eu/plant-research](http://www.wur.eu/plant-research)

Report WPR-840

---

The mission of Wageningen University & Research is "To explore the potential of nature to improve the quality of life". Under the banner Wageningen University & Research, Wageningen University and the specialised research institutes of the Wageningen Research Foundation have joined forces in contributing to finding solutions to important questions in the domain of healthy food and living environment. With its roughly 30 branches, 5,000 employees and 10,000 students, Wageningen University & Research is one of the leading organisations in its domain. The unique Wageningen approach lies in its integrated approach to issues and the collaboration between different disciplines.







To explore  
the potential  
of nature to  
improve the  
quality of life



---

Corresponding address for this report:  
P.O. Box 16  
6700 AA Wageningen  
The Netherlands  
T +31 (0)317 48 07 00  
[www.wur.eu/plant-research](http://www.wur.eu/plant-research)

Report WPR-840

The mission of Wageningen University & Research is "To explore the potential of nature to improve the quality of life". Under the banner Wageningen University & Research, Wageningen University and the specialised research institutes of the Wageningen Research Foundation have joined forces in contributing to finding solutions to important questions in the domain of healthy food and living environment. With its roughly 30 branches, 5,000 employees and 10,000 students, Wageningen University & Research is one of the leading organisations in its domain. The unique Wageningen approach lies in its integrated approach to issues and the collaboration between different disciplines.

