Term paper Radiative Transfer - Ecosystem and Atmosphere Processes

Simone Massaro

2021-03-31

Table of Contents

Preface

This is the Term Paper for the Ecosystem and Atmosphere Processes course (Winter semester 2020) at Uni Göttingen, prof. Alexander Knohl.

This is a result of a group work between Simone Massaro and Sugam Subedi.

It is part of a course project to develop a canopy model for a canopy ecosystem.

This term paper is also available in html format at https://bookdown.org/massaro_simone_it/Term-paper-Radiative-Transfer-EAP/ which is the suggested way to read it.

Task division

The tasks have been divided inside the group:

- *Simone Massaro*: all code development. In the term paper: model calibration, sensitivity, model results and discussion; Formatting of the term paper.
- *Sugam Subedi*: In the term paper: theory section and first part of model development section.

1 Introduction

Models are an abstraction of reality and is compromise between reality and simplicity. Radiative transfer modeling describes the absorption and separation of solar energy by the ecosystems and their components in evaluating the atmosphere-biosphere interactions. This topic is vital in understanding the functioning of plant and also its whole ecosystem functioning towards radiative difference in atmosphere. The mathematical equations used will assist in getting the sensible outcomes that describes the process like absorption, reflectance , transmittance , scattering by vegetation and soil elements. This topic also gives us great deal of understanding earth science and climate change studies. Models at canopy level helps in understanding production of ecosystem.

Electromagnetic radiation is they only way for the Earth can exchange energy with the rest of the universe. Incoming solar radiation is the source of virtually all energy present on Earth, and outgoing infrared radiation is how Earth can reduce its temperature. This leads the behaviour of radiation at being one of the core aspect that regulates Earth’s climate and biological processes.

The most important interaction between the radiation and the Earth take places at its boundaries, which are often occupied by forest ecosystems. Incoming solar is absorbed, reflected and absorbed by the canopies, which also emit longwave radiation.

The behaviour of the radiation depends on canopy properties: its architecture and leaf optical properties. The amount of available radiation directly influences two key aspects in ecosystems: temperature and photosynthesis. Therefore, understanding how radiation interacts with canopies is crucial to study ecosystem functions.

The aim of this project is developing a model for radiative transfer in canopies to explore the key processes of radiative transfer.

2 Theory

Describe the main set of equations that are central to this topic

Equations of the parameters

Direct beam coefficient

It is the ratio between leaf shadow and real leaf area. It depends on solar zenith angle (Z) and leaf inclination angle(Θ_l). Here we consider for spherical leaf distribution thus,
$Kb=\frac{0.5}{\cos Z}$

Diffuse beam extinction coefficient: It is the optical property that reflects the attenuation in atmosphere and is also known as spectral diffuse attenuation coefficient. The extinction coefficient for diffuse radiation is obtained by substituting the sky zenith angle for the solar zenith angle and integrating over the sky hemisphere.

Diffuse Transmittance $\tau_d=2\sum_{i=1}^{9} exp[-\frac{G(Z_i)}{\cos Z_i}L]\sin Z_i \cos Z_i \Delta Z_i$ whereas $Z_i=5°$ to $85°$ for nine sky zones. Effective Extinction coefficient $K_d=\frac{-\ln T_d}{L}$

For the two stream approximation model, diffuse radiation extinction coefficient: $Kd=1/¿$
Whereas Ross coefficient X_l is between -0.4 and 0.6 which is a leaf angle distribution which quantifies the departure of leaf angles from a spherical distribution.

$\phi_1=0.5-0.6333\chi_l$

$\phi_2=0.877(1-2\phi_1)$

2.0.1 Shortwave radiation

Solar energy enters our atmosphere as shortwave radiation in the form of ultraviolet (UV) rays and visible light. In case of direct beam, scattered flux are as:

$I_b^{\uparrow}=-\gamma_1 e^{-K_b \Omega L}+\mu_1 v e^{-h\Omega L}+\mu_2 v e^{h\Omega L}$

$I_b^{\downarrow}=\gamma_2 e^{-K_b \Omega L}+\mu_1 v e-h\Omega x+\mu_2 v e^{h\Omega L}$

Absorbed flux

Absorbed canopy flux: $\overrightarrow{I_{cb}}=(1-e^{-k_b \Omega L})I_{sky,b}-I_b(0)+I_b(L)-I_b(L)$

Absorbed ground flux: $\overrightarrow{I_{gb}}=(1-\rho_{gd})I_b(L)+(1-\rho_{gb})I_{sky,b}e^{K_b \Omega L}$

On the canopy also, we have sunlit canopy and shaded canopy,
$I_{cSun,b}=(1-\omega_l)[(1-e^{-Kb\Omega L})I_{sky,b}+K_d\Omega(a_1+a_2)]$ $IcSha,b=Icb-I_{cSun,b}$

Now the fluxes of diffuse radiation are: Scattered diffuse flux: Scattered upward: $I_d(x)=\mu_1 v e^{-h\Omega x}+\mu_2 v e^{h\Omega x}$ Scattered downward: $I_d(x)=-\mu_1 v e^{-h\Omega x}-\mu_2 v e^{h\Omega x}$

Absorbed diffuse flux: By canopy: $I_{cd}=I_{sky,d}-I_d(0)+I_d(L)-I_d(L)$ In case of canopy also we have, Absorbed diffuse flux by sunlit canopy:

$$a_1=\mu_1 u[\frac{1-e^{-(K_b+h)\Omega L}}{K_b+h}]+\mu_2 v[\frac{1-e^{(-K_b+h)\Omega L}}{K_b-h}]$$

Absorbed diffuse flux by shaded canopy:

$$a_2=-\mu_1 v[\frac{1-e-(K_b+h)\Omega L}{K_b+h}]-\mu_2 v[\frac{1-e^{(-K_b+h)\Omega L}}{K_b-h}]$$

Absorbed diffuse flux by soil: $I_{gd}=(1-\rho_{gd})I_d(L)$

Parameters of direct/diffuse shortwave radiation:

$$b=[1-(1-\beta)\omega_l]K_d$$

$$c=\beta\omega_l K_d$$

$$h=\sqrt{(b^2-c^2)}$$

$$u=h-b-c/2h$$

$$v=h+b+c/2h$$

For direct beam solution, boundary leaf conditions: $\mu_1=\gamma_2-\mu_2 u/v$

$$\mu_2=\frac{[\mu(\gamma_1+\gamma_2\rho_{gd}+\rho_{gb}I_{sky,b})e^{-K_b\Omega L}-\gamma_2(u+v\rho_{gd})s-h\Omega L]}{v(v+uPgd)e^{h\Omega L}-u(u+vpgd)e^{-h\Omega L}}$$

Where, K_b Direct beam extinction coefficient K_d Diffuse beam extinction coefficient Ω =clumping index γ_1 adn γ_2 parameters χ cumulative leaf area index μ_1 and μ_2 constants obtained from boundary conditions I_{cb} absorbed diffuse flux by canopy I_{gb} absorbed diffuse flux by soil

2.0.2 Longwave radiation

Longwave radiation fluxes can be described similarly to diffuse solar radiation, but dropping the direct beam radiation scattering term and with the addition of a thermal radiation source term emitted by foliage.

Downward longwave radiation:

$$L^{\downarrow}(x)=L_{sky}[1-\varepsilon_l(1-e^{-K_d x})]+\varepsilon_l\sigma T_l^4(1-e^{-K_d x})$$

Upward longwave radiation:

$$L^{\uparrow}(x)=L_g 1-\varepsilon_l[1-e^{-K_d(L-x)}]+\varepsilon_l\sigma T_l^4[1-e^{-K_d(L-x)}]$$

Net absorbed longwave radiation by canopy(per ground area):

$$L_c = \varepsilon_l (L_{sky} + L_g)(1 - e^{-K_d L}) - 2 \varepsilon_l \sigma T_l^4 (1 - e - K_d L)$$

Longwave radiation absorbed by soil

$$L_g = L^{\downarrow}(L) - L_g$$

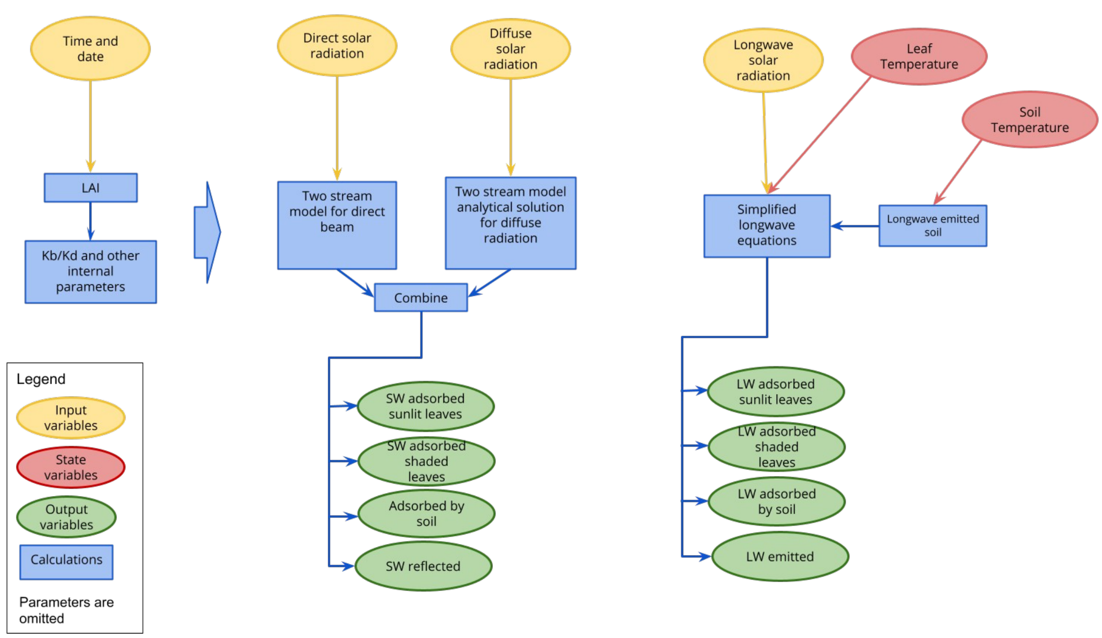
Where, L_{sky} = longwave radiation above the canopy ε_l = longwave emissivity of leaf χ= cumulative leaf area index σ = steffan Boltzmann constant i.e 5.67e -08 T = temperature in Kelvin L = leaf area index $L^{\downarrow}$ = longwave radiation towards downward

3 Model development

Developing model is an important aspect i.e making the model work from all complex equations into simplified version by writing them in code format in R. Developing equations of parameters with optimization, and developing code for short wave through incoming direct and diffuse equations and dividing longwave also in downward and upward, absorbed with canopy and soil. Such simple division of waves into short and long gives us the absorbed radiation passed towards leaf energy balance model and photosynthesis model.

We have to do some assumption so that the model could be developed like we considered only spherical leaf angle distribution and two stream model case in shortwave and forward scattering in case for longwave radiation.

Bonan (2019) has been used as a reference for the all theory of the model development and its example script as a starting point.



Radiative transfer model flow

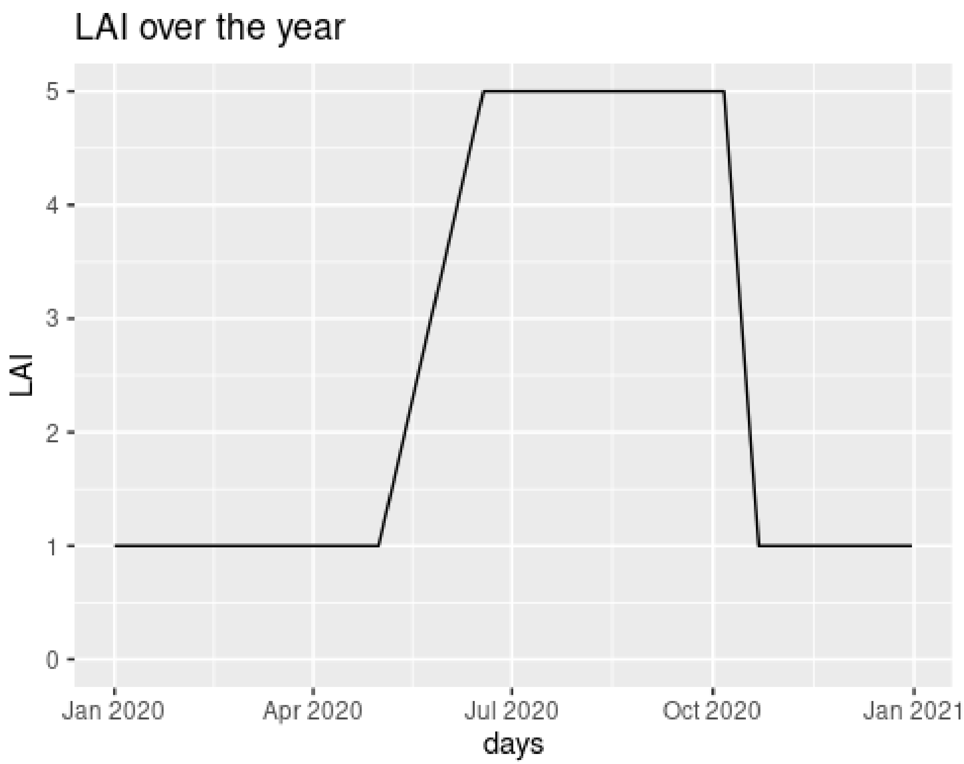
3.1 Parameters

- Leaf reflectance
- Leaf transmittance
- Soil albedo for direct shortwave radiation
- Soil albedo for diffuse shortwave radiation

- Leaf longwave emissivity
- Soil longwave emissivity
- Canopy clumping coefficient
- Max LAI (variation over year)
- Latitude of Hainich
- Longitude of Hainich

3.1.1 LAI

The variation of the LAI over the year has been obtained with a simple model (Figure 3.1) that considers a linear increase of LAI during the spring and a similar process during autumn. During winter the LAI is considered to be 1, even if there are no leaves, as trunks and branches interact with the radiation.



3.2 Sensitivity

The sensitivity analysis of the model has been made using the `FME` package, specifically the function `sensFun`.

The sensitivity analysis has been carried out with data from the July 2018 from the Hainich site. The leaf temperature was assumed equal to the air temperature, while the soil temperature is the average of the first 30 cm.

Table 3.1 represents the sensitivity of each parameter after aggregating on all model outputs. There are three aggregation functions that are used by the `FME` paper Soetaert and Petzoldt (2010) :

- mean
- L1
- L2

Table 3.1: Aggregated model sensitivity.

	value	scale	L1	L2	Mean	Min	Max	N
rho_leaf	0.40	0.40	0.24	0.51	0.14	-0.80	1.80	14880
tau_leaf	0.10	0.10	0.07	0.14	0.04	-0.20	0.53	14880
alb_soil_b	0.10	0.10	0.00	0.01	0.00	-0.05	0.05	14880
alb_soil_d	0.10	0.10	0.00	0.02	0.00	-0.11	0.02	14880
em_leaf	0.97	0.97	0.52	9.89	0.21	-866.64	264.52	14880
em_soil	0.97	0.97	2.89	60.16	-1.00	-1635.34	4966.26	14880

This summary, however, consider both the longwave and shortwave components even if the parameter has no impact on that sub-model. Therefore, the sensitivity has been manually divided and analyzed for each output variable independently.

3.2.1 Shortwave

For the shortwave (Tables 3.2 3.3 3.4) the leaf reflectance (ρ) is the parameter with the highest effect on all output variables. Leaf transmittance (τ) has reduced influence, while the soil albedo have a very small impact on the model.

Table 3.2: Shortwave sensitivity aggregated by mean

var	i_down	i_up	ic	ic_sha	ic_sun	ig
rho_leaf	1.37	1.14	-0.30	-0.01	-0.59	1.15
tau_leaf	0.41	0.25	-0.07	0.01	-0.15	0.35
alb_soil_b	0.01	0.00	0.00	0.01	0.00	-0.01
alb_soil_d	0.01	0.00	0.00	0.00	0.00	-0.08

Table 3.3: Shortwave sensitivity aggregated by L1

var	i_down	i_up	ic	ic_sha	ic_sun	ig
rho_leaf	1.37	1.14	0.30	0.26	0.59	1.15
tau_leaf	0.41	0.25	0.07	0.06	0.15	0.35
alb_soil_b	0.01	0.00	0.00	0.01	0.00	0.01
alb_soil_d	0.01	0.00	0.00	0.00	0.00	0.08

Table 3.4: Shortwave sensitivity aggregated by L2

var	i_down	i_up	ic	ic_sha	ic_sun	ig
rho_leaf	1.38	1.14	0.30	0.29	0.59	1.18
tau_leaf	0.42	0.25	0.07	0.07	0.15	0.36
alb_soil_b	0.02	0.00	0.00	0.01	0.00	0.02
alb_soil_d	0.01	0.00	0.00	0.00	0.00	0.08

3.2.2 Longwave

For the longwave (Tables 3.5 3.6 3.7) the parameters are the emissivity of the leaves (ε_l) and the emissivity of the soil (ε_g). The former has a significant impact on all output variables while the latter has a very important influence only on the amount or radiation absorbed by the canopy and emitted by the soil.

Table 3.5: Longwave sensitivity aggregated by mean

var	l_down	l_up	lc	lc_sha	lc_sun	lg
em_leaf	0.17	0.09	1.00	1.00	1.00	0.93
em_soil	0.00	0.05	-6.08	-6.93	-0.91	-6.21

Table 3.6: Longwave sensitivity aggregated by L1

var	l_down	l_up	lc	lc_sha	lc_sun	lg
em_leaf	0.17	0.09	1.00	1.00	1.00	7.08

em_soil	0.00	0.05	6.08	6.93	1.04	43.63

Table 3.7: Longwave sensitivity aggregated by L2

var	l_down	l_up	lc	lc_sha	lc_sun	lg
em_leaf	0.19	0.10	1.00	1.00	1.00	44.21
em_soil	0.00	0.05	19.14	19.47	3.58	267.62

3.3 Model calibration

The sensitvity analysis allows to understand for which parameters the calibration is important.

The observed data at Hainich during July 2018 is used for the calibration.

3.4 Shortwave

The shortwave is calibrated on two parameters: `rho_leaf` and `tau_leaf`.

The obtained values 3.8 are similar of the initial estimates of 0.40 and 0.05 respectively.

Table 3.8: Shortwave parameters after calibration

	x
rho_leaf	0.42
tau_leaf	0.05

3.4.1 Longwave

For the longwave no calibration has been made because the real leaf temperature was not available therefore the calibration would have resulted with not realistic parameter values.

4 Model results

4.1 Model output

The input data for the model was obtained from the fluxnet site at Hainich for the year 2018.

The first step is exploring the main model output both for the shortwave and longwave component.

4.1.1 Shortwave

The shown shortwave variables are:

- Incoming sw (measured) $I^{\downarrow}$

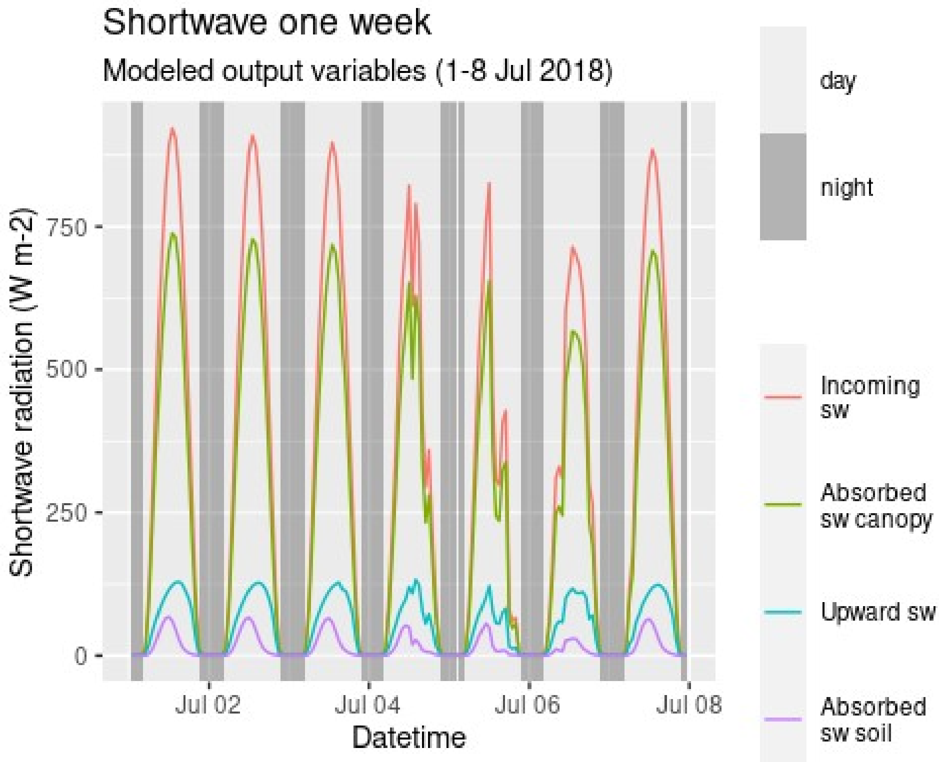
- Absorbed sw by canopy $I^{\uparrow}$
- Upward sw reflected by the canopy $\vec{I}_c$
- Absorbed sw soil $\vec{I}_g$

Due to radiative balance those variables are in the following relationship:

$$I^{\downarrow} = I^{\uparrow} + \vec{I}_c + \vec{I}_g$$

One week

In figure 4.1 the model output for the shortwave is plotted for one week during the summer. The daily cycle can be clearly seen, with all variables having a peak at noon and reach the value of 0 during the night. The total incoming radiation changes depending on the day, mainly due to cloud cover, and the absorbed shortwave follow its pattern.

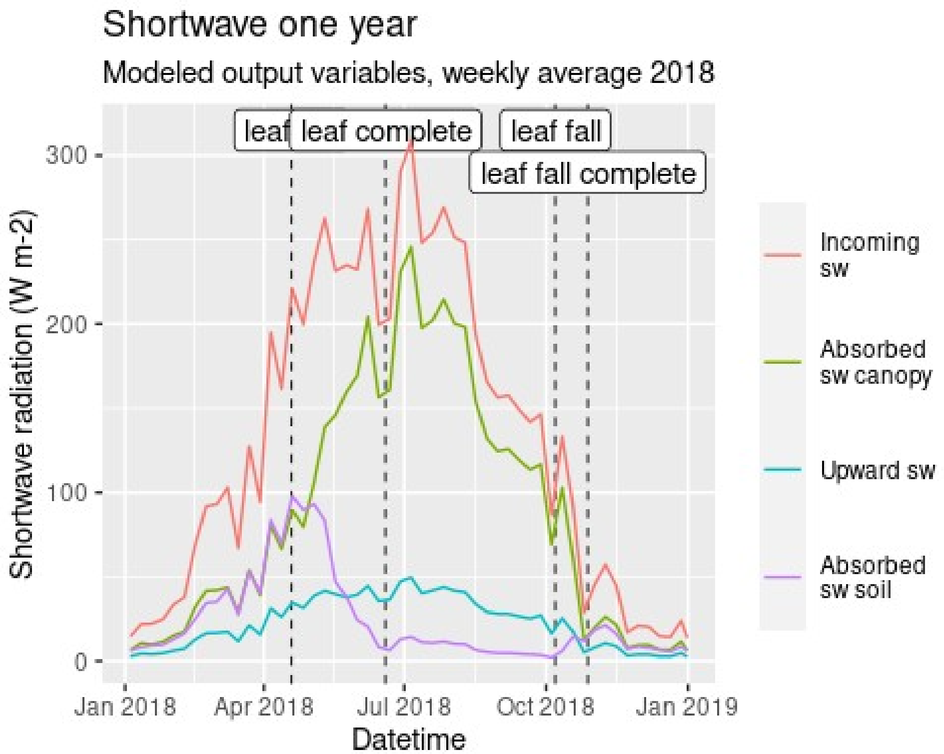


One year

Then the shortwave over one year is analyzed (Figure 4.2), after averaging over one week. There is a yearly cycle in the incoming shortwave radiation with a significant difference, as the averages goes from more than $300\,W\,m^2$ to almost $10\,W\,m^2$ (this is the week average so takes into account also the night when shortwave radiation is zero).

The absorbed by the soil depends on the amount of LAI, in fact during the spring when the radiation is high but there are no leaves yet its values increase constantly. As soon as leaves starts to come out the shortwave absorbed by the soil decreases to reach a stable low value during the summer and eventually increase again when leaves fall.

The radiation reflected by the canopy doesn't have a big change over the year and overall canopy albedo remains for the all year between 0.16 and 0.20. Finally, the radiation absorbed the canopy follows, as expected, the pattern in the incoming radiation. During the spring its value are really similar to the shortwave absorbed by the soil, however this is only a coincidence.



4.1.2 Longwave

The shown longwave variables are:

- Incoming lw (measured) $L^{\downarrow}$
- Absorbed lw by canopy $L^{\uparrow}$
- Upward lw emitted by the canopy $\vec{L}_c$
- Absorbed lw soil $\vec{L}_g$

Due to radiative balance those variables are in the following relationship:

$L^{\downarrow} = L^{\uparrow} + \vec{L}_c + \vec{L}_g$

One week

Longwave radiation has a daily cycle, but the variation is much smaller compared to shortwave (Figure 4.3). The canopy is emitting more longwave radiation than the one that is receiving from the sky, hence the absorbed radiation from the canopy is negative. The incoming shortwave radiation depends on the weather, with cloudy skies resulting in higher level of incoming radiation as it can be seen on the 5-6 of July. The difference can be quite important with an increse of about 30% in radiation levels. The upward longwave radiation, instead, depends on the temperature of the canopy and the soil, which is influenced by heat fluxes and shortwave radiation. Therefore, the incoming and outgoing radiation don't change together, hence there is a variation the radiative balance. In fact it almost reaches zero in the morning of the 6th of July.

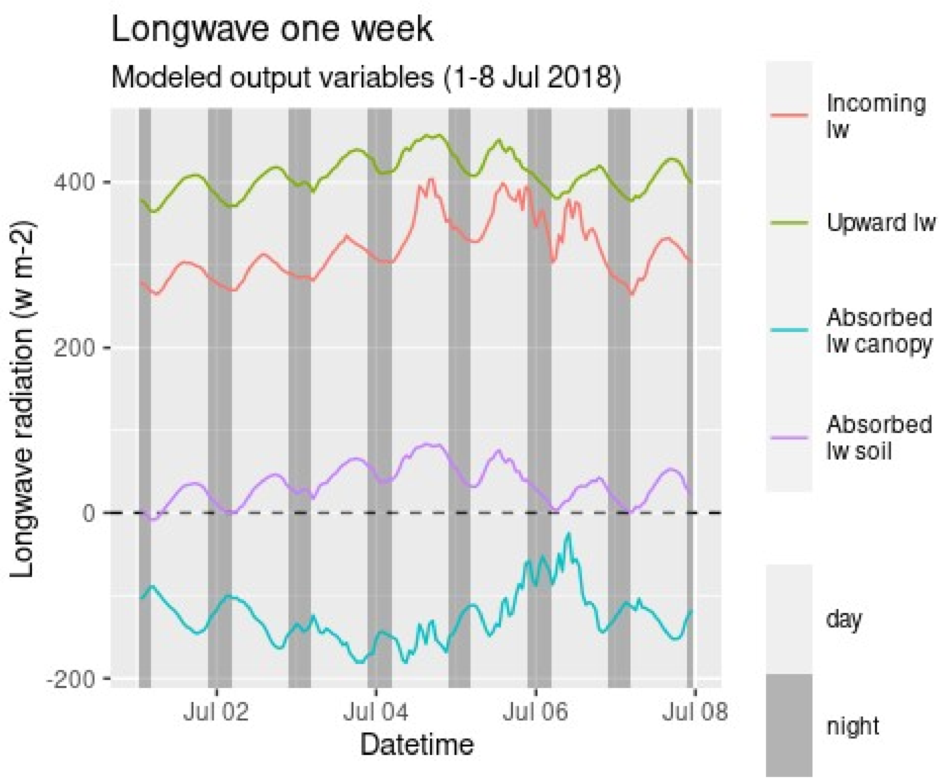
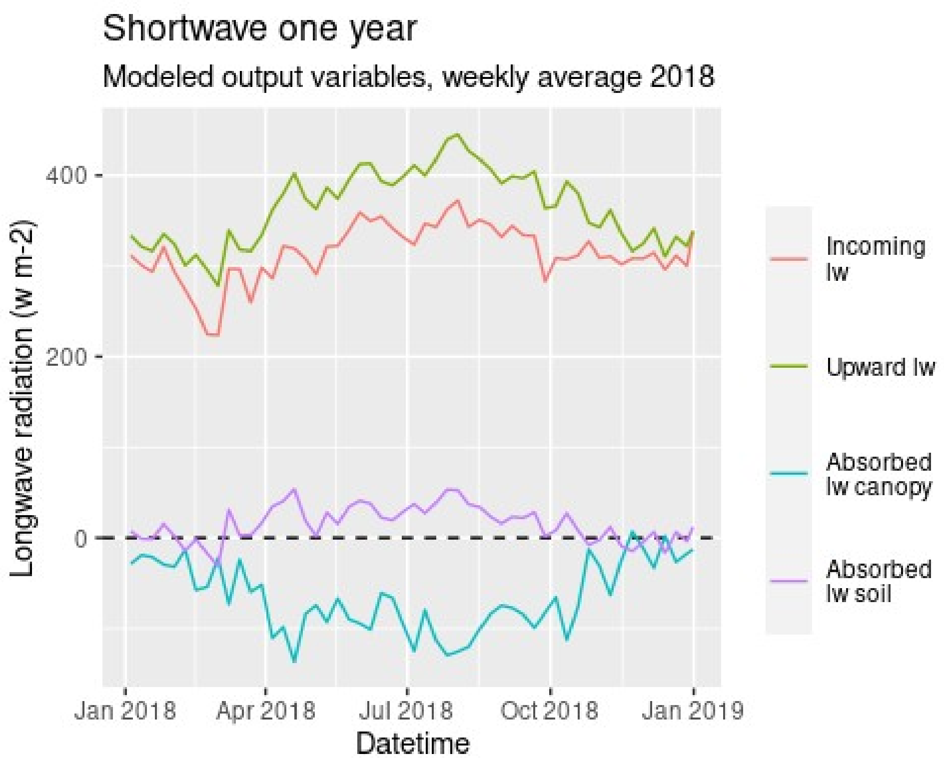


Figure 4.3: Longwave output over one week

One year

During the year there longwave radiation has a cycle with higher values in the summer than in the winter. The difference is, however, limited between seasons. In general the behaviour is comparable with the 1 week period.



4.2 Model evaluation

The model results are compared with observed data to evaluate the model accuracy. It is done independently for the longwave and shortwave components.

In the Hainich dataset there are observed variables that will be used for the evaluation.

The evaluation will be carried out by visually comparing time series and by quantitative means.

4.2.1 Shortwave

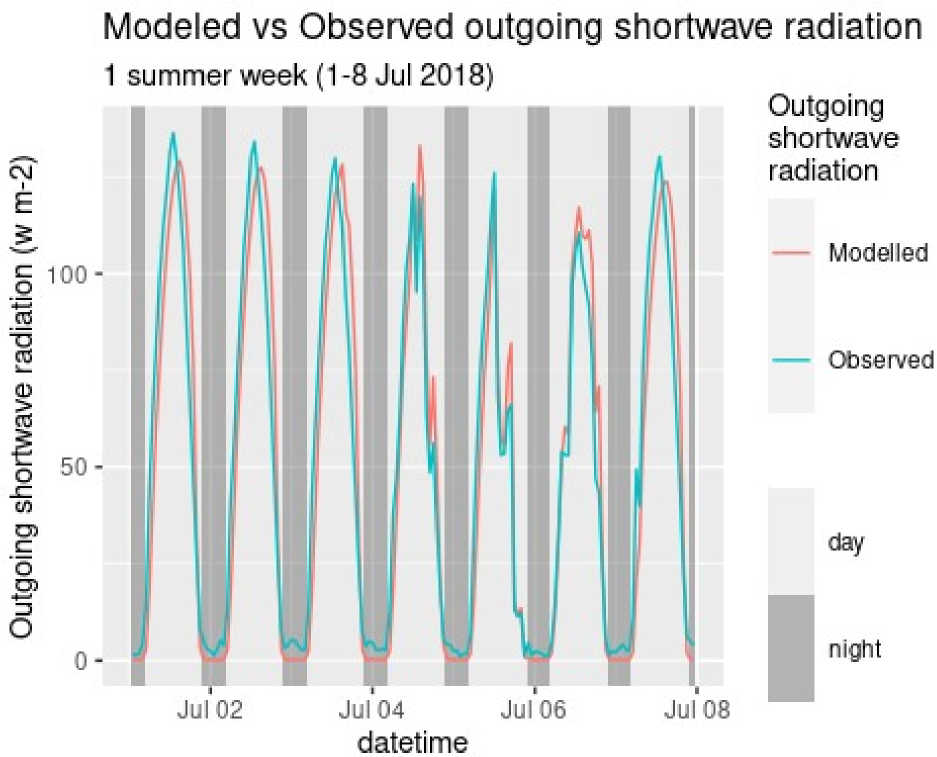
For the shortwave the only observed variable that can be used for the model evaluation is the outgoing shortwave radiation.

Time series

One Week

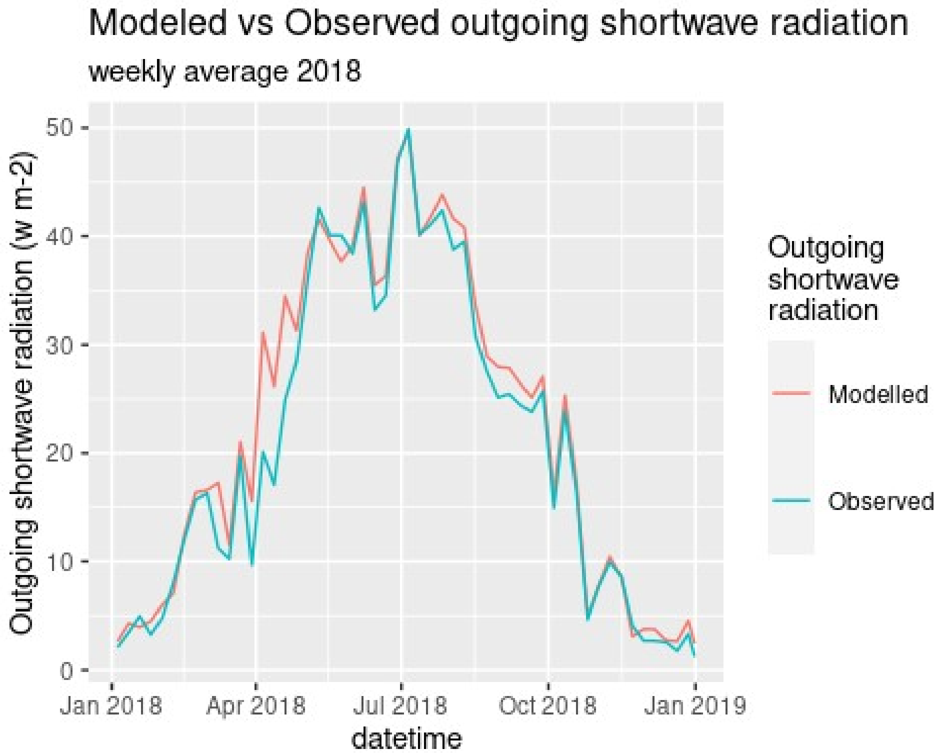
The model has an overall good agreement between the modeled outgoing shortwave, and the measured one (Figure 4.4).

The day peak of the model is often delayed compared to the observed one. This phase shift may connected to the relative long time interval (1 hour) in model outputs. The model generally slightly underestimates the radiation during sunny days (first 3 and last), while overestimates during with cloudy conditions. During the night the observed radiation is slightly above zero, while the model is at zero. This is clearly a measurement error, as during the night there is no shortwave radiation.



One Year

By analyzing a model output over the whole year, there is a good overall performance (Figure 4.5). During spring, and partially during autumn, there is the biggest difference between the model output and the observations. This is probably connected to the fact that in that period the LAI is estimated with a simple linear equation, thus does not reflect completely the real world conditions.



Quantitative evaluation

The performance of the model has been also analyzed in a quantitative way. A linear model has been built between the observations, and the model output (Figure 4.6), if the model matched perfectly the observations the slope should be 1, the intercept 0 and the r^2 1.

The results of the linear model are:

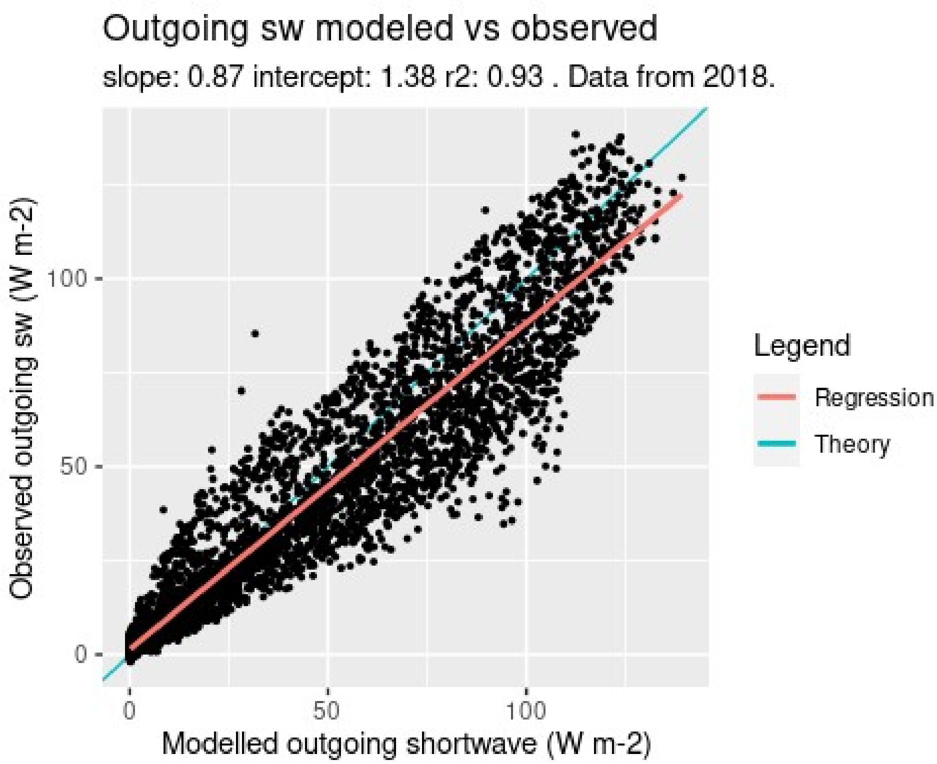
- slope: 0.87
- intercept: 1.38
- r^2: 0.93

The intercept is really close to 0, and the slope and r^2 also indicate a good fit.

The performance of the model has been also evaluated using the Root Mean Square Error (RMSE) and the Nash-Sutcliffe coefficient (NSE). The obtained values are:

- RMSE: 9.51
- NSE: 0.91

The NSE has a value of 1 when there is a perfect model and 0.91 can be considered really good. The RMSE can be interpreted as the amplitude of the error and compared to the range of the radiation. Therefore, in this scenario RMSE of 9.51 can be considered good but not perfect.



####

Conclusion

The performance of the model in more than satisfactory and the some of deviation from the observation can be explained with the inaccurate input data.

4.2.2 Longwave

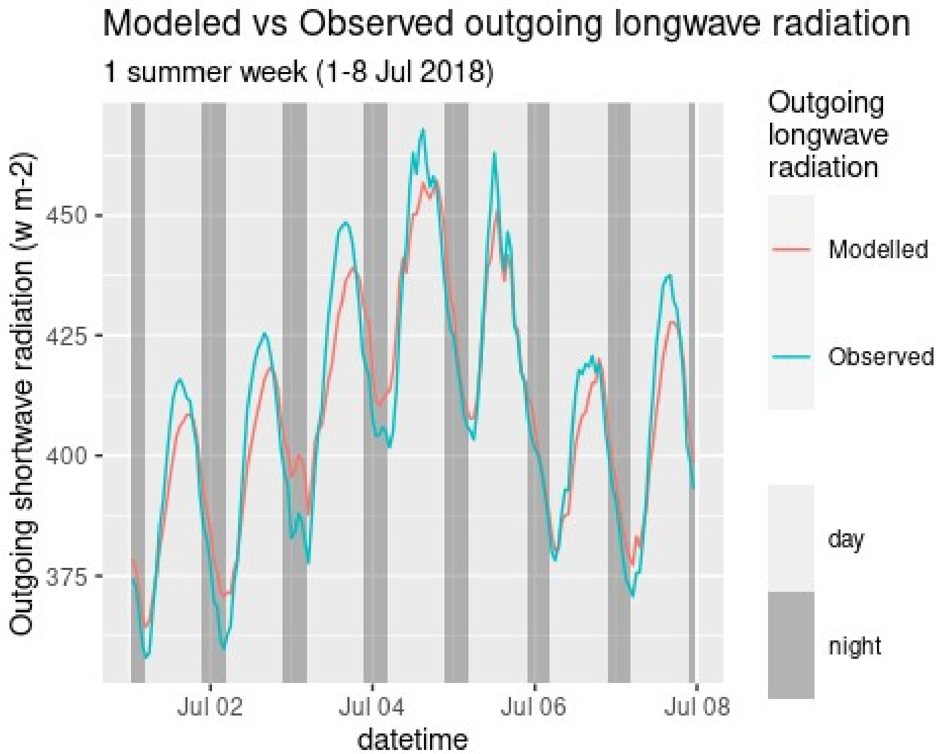
In the same way of shortwave the only observed variable that can be used for the model evaluation is the outgoing longwave radiation.

Time series

One Week

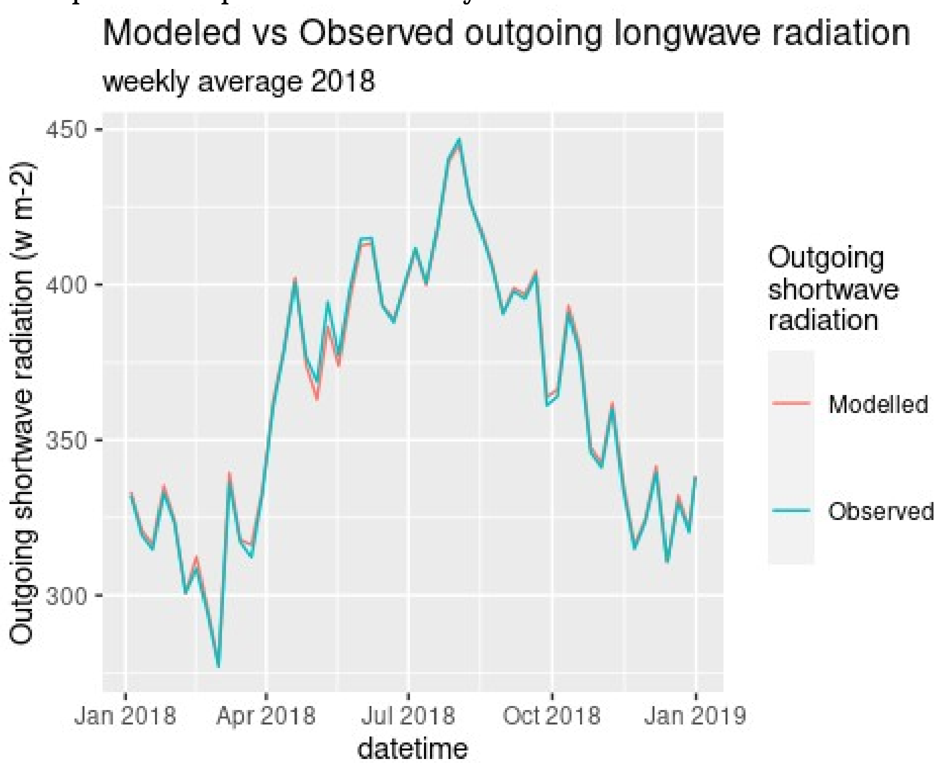
The longwave components is also accurately modeled (Figure 4.7).

There is a daily pattern in the difference between the modeled and the observed outgoing longwave. This is probably due to the fact that the model doesn't use the true leaf temperature, as it needs to be calculated by other models, but uses the air temperature as a proxy.



One Year {-}

By comparing the week averages over the year (Figure 4.7) the model produces almost perfect output for the whole year.



Quantitative evaluation

The same procedure for shortwave has been followed for the longwave of the quantitative evaluation of model performance.

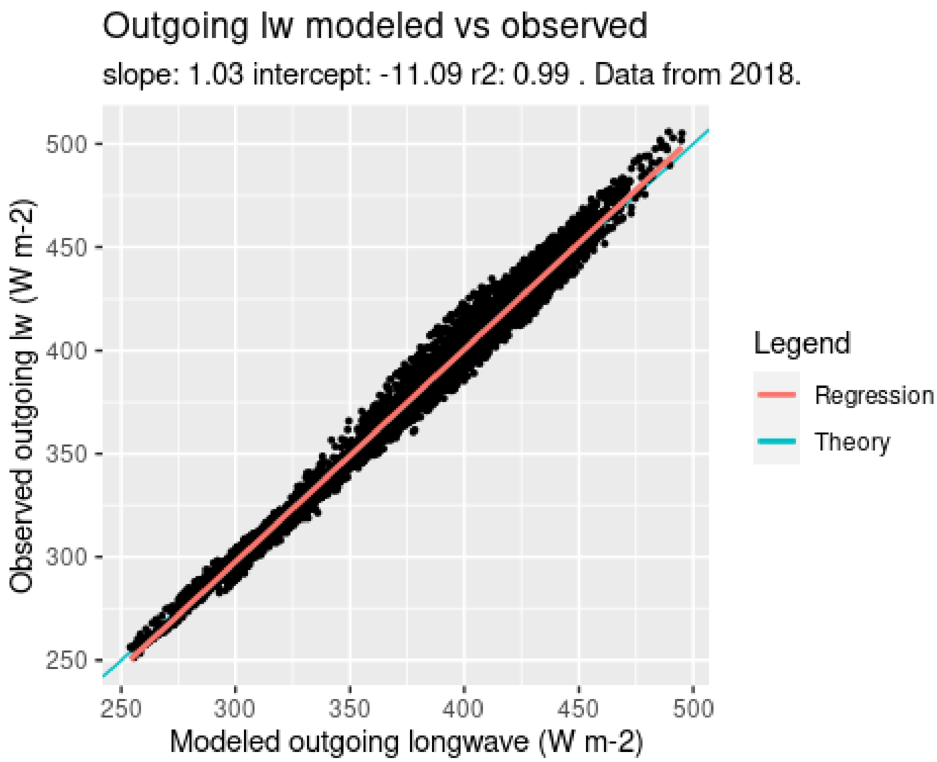
The longwave model has a very good performance (Figure 4.9) The results of the linear model are:

- slope: 1.03
- intercept: −11.09
- r^2: 0.99

Both the slope and the r 2 have a value of 1 when rounded at the first decimal digit.

This is confirmed by the high value of the Nash-Sutcliffe coefficient and the low Root Mean square error:

- RMSE: 5.33
- NSE: 0.99



####

Conclusion

The longwave model performs very well. This is true in spite of the inaccurate leaf temperature that is used as input.

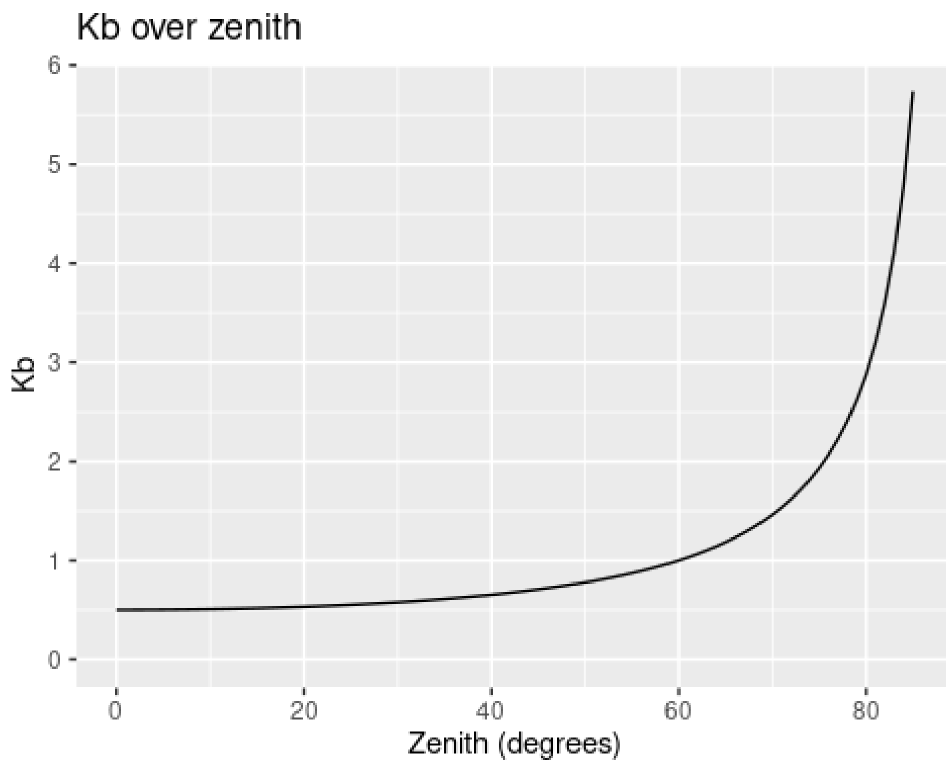
5 Discussion

5.1 Model components

After the analyzing the model results and evaluating its performance a more the behaviour of the model components is analyzed.

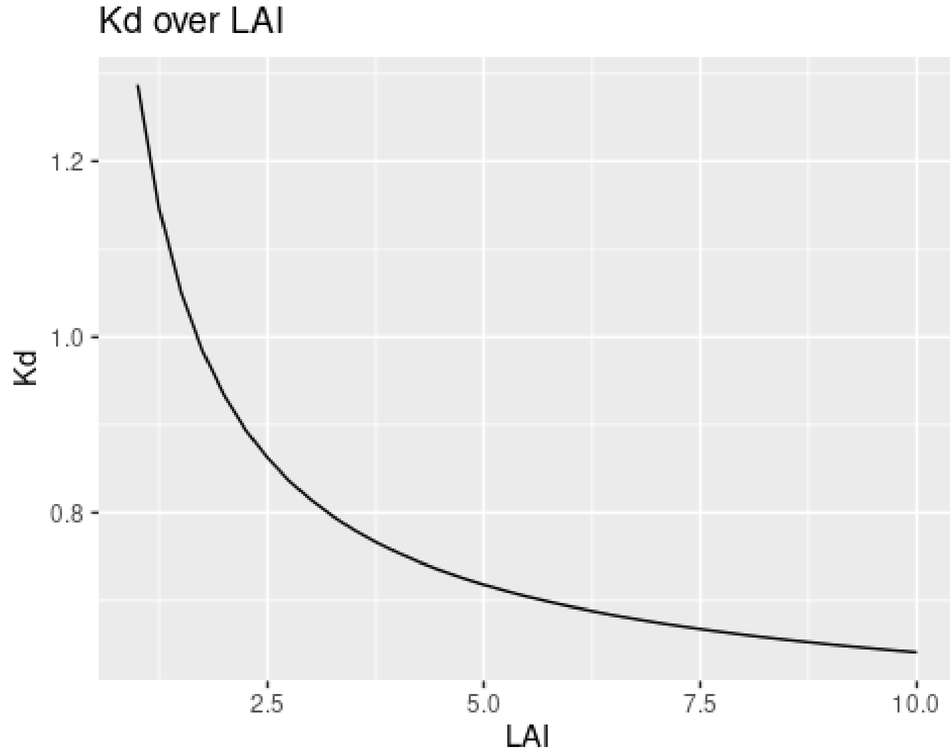
Kb

The direct beam extinction coefficient depends on the leaf angle distribution and the zenith. The leaf angle distribution doesn't change hence the variation of the zenith is the only aspect that can influences the penetration of direct shortwave radiation. For zenith inferior to $30°$ the K_b has a value of about 0.5 (Figure 5.1), then reaches 1 with a $60°$ zenith and eventually start to grow exponentially reaching the theoretical value of infinity when the sun is completely horizontal.



Kd {-} The diffuse radiation extinction coefficient does not depend on directly on the solar zenith, as the radiation comes from all directions, but depends on the LAI. The bigger the LAI the smaller the K_d, thus more light penetrates through the canopy (Figure 5.2). The range of the K_d has a smaller than the K_b. The K_d showed in this plot is used only by the longwave

model, as the 2 stream approximation uses a different K_d, that depends only on leaf angle

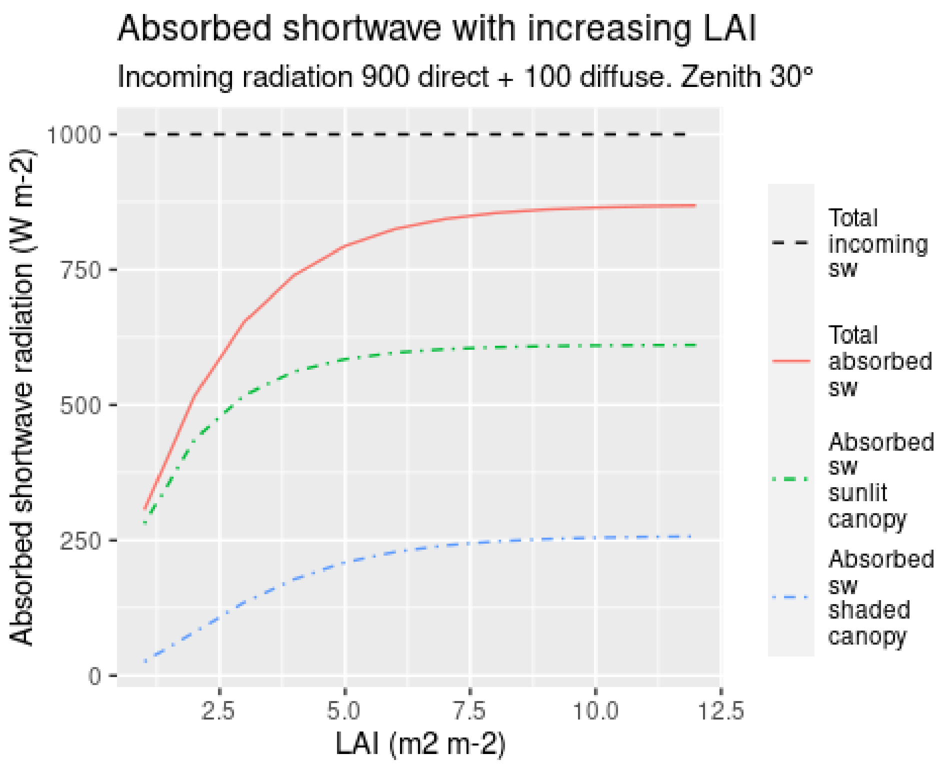


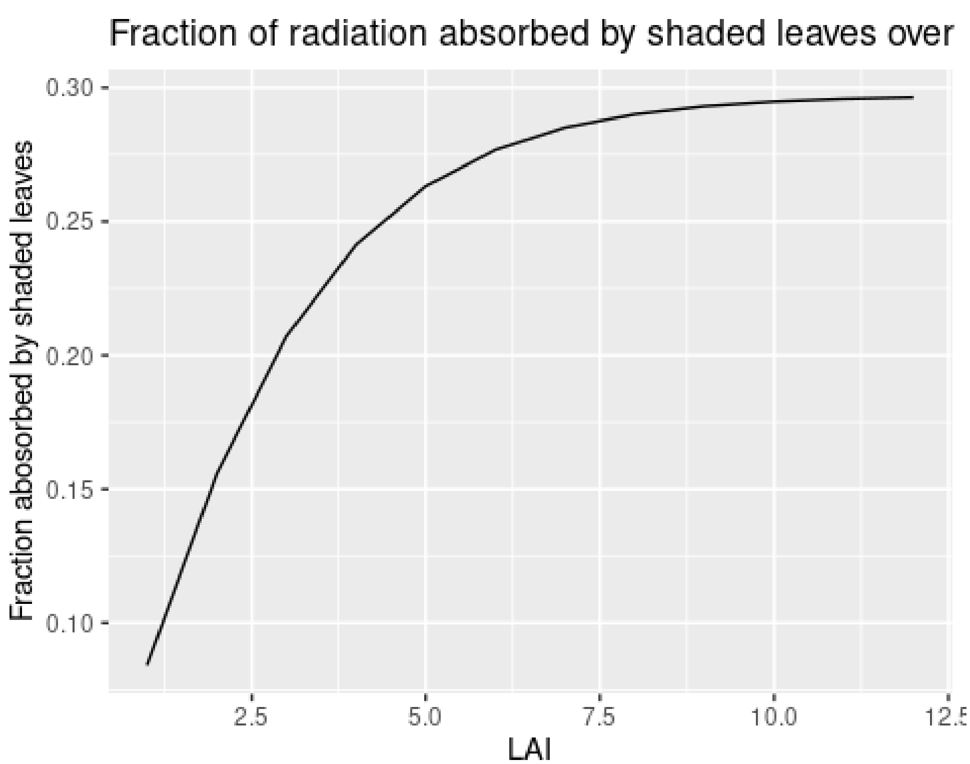
distribution.

Absorbed radiation over LAI

The amount LAI influences directly the amount of radiation absorbed. The increase in LAI initially increase the amount of radiation absorbed to then reach an asymptote (Figure 5.3).

The sunlit leaves absorb the majority of the radiation, ranging from 70% at high LAI values to over 90% with low LAI (Figure 5.4).





The variation of LAI influences also the amount of radiation absorbed by the soil, ranging from over 50% to almost 0% with high LAI values (Figure 5.5).

The LAI has virtually no impact on the total canopy albedo.

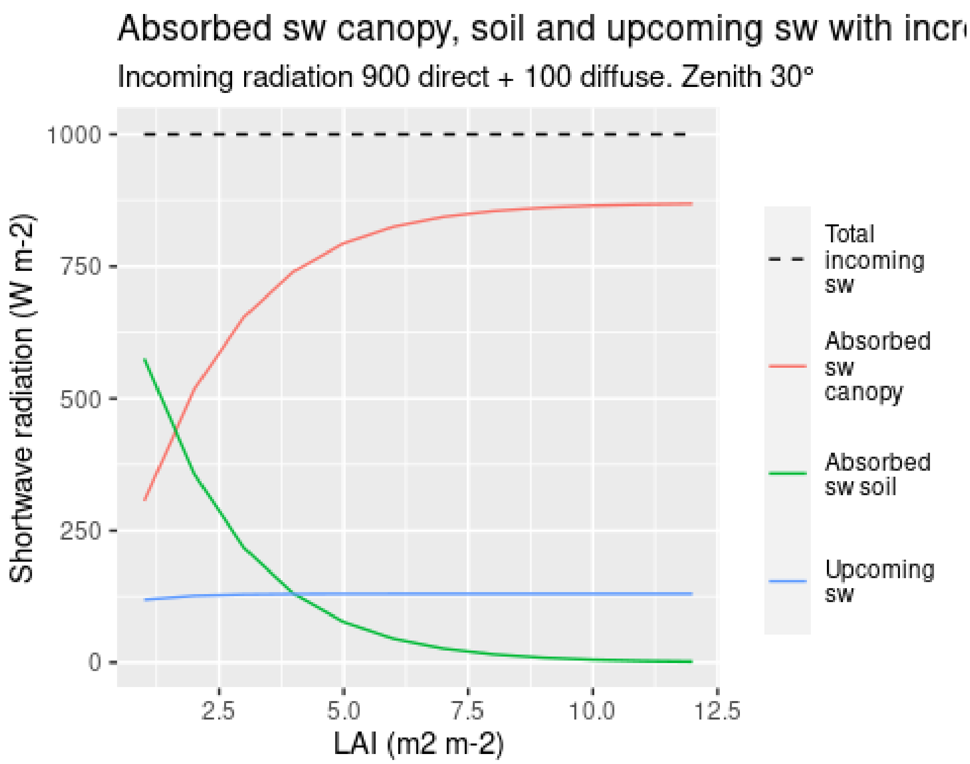


Figure 5.5: Absorbed sw canopy, soil and upcoming sw with increasing LAI

Emitted longwave radiation at different temperature

The leaf temperature have a direct influence on the emitted radiation.

In figure 5.6 soil and leaves temperature change together between 260 and 310 K. This results in an almost linear increase in the emitted radiation.

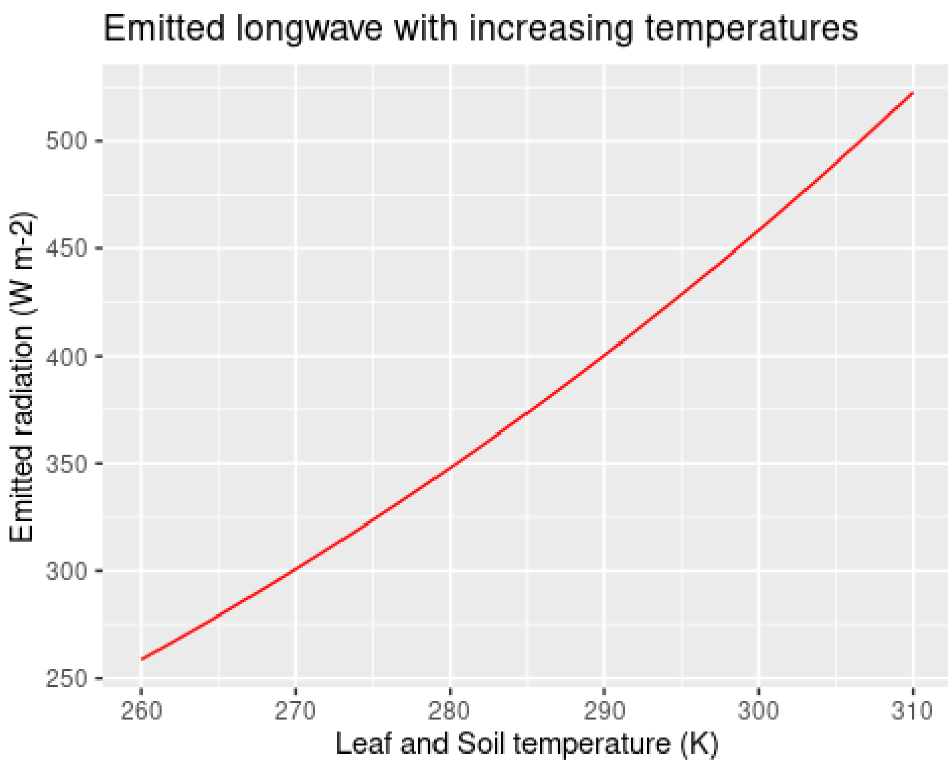


Figure 5.6: Emitted longwave with increasing temperatures

Absorbed radiation vs zenith

The solar zenith influences the fraction of the solar radiation absorbed 5.7, it increases with the zenith, reaching peak at 60 ° and then rapidly decreases. The variation of absorbed radiation with the zenith is limited ranging from about 70 % to 80.

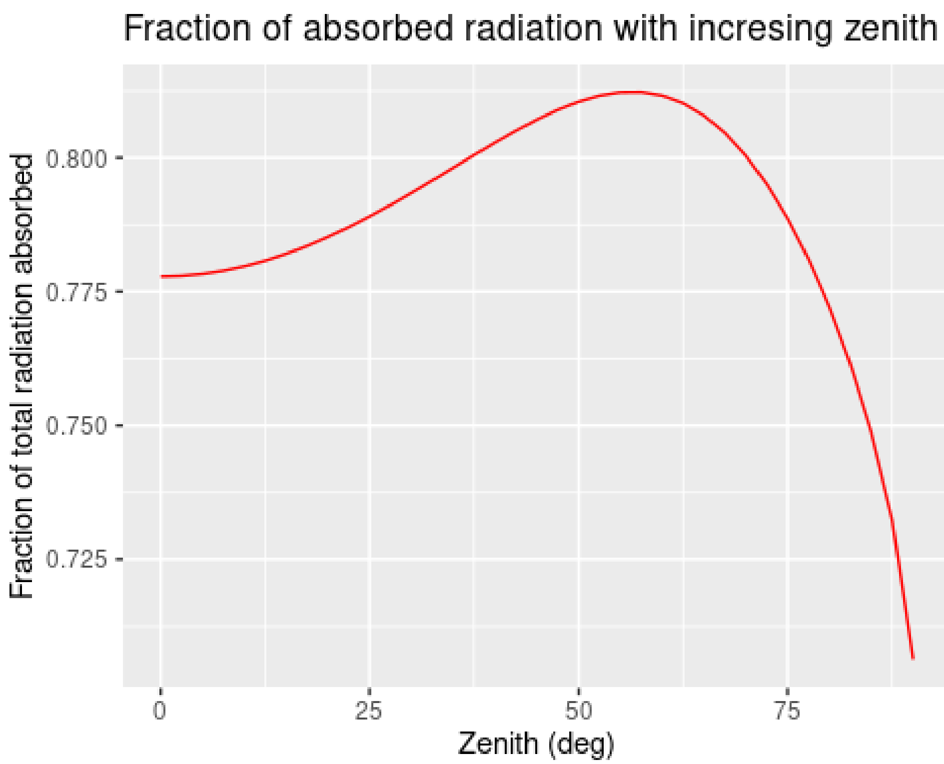


Figure 5.7: Fraction of absorbed radiation with incresing zenith

5.2 Conclusion

The model was developed with the goal to be as simple as possible to be able to understand the radiative transfer processes. The model was not intended to be used in real data conditions and its main aim has been achieved with the current version.

Nonetheless, improvements are possible in particular:

- solve the shortwave radiations for the visible and near infrared radiation, as leaves have different optical properties.
- use more accurate equation than the two-streams approximation, like Norman model.
- optimize the performance of the model in order to allow for faster iteration during the analysis.

Regarding the last point an experimental port has been made using the Julia language https://github.com/mone27/canopy-model.jl obtaining promising results (roughly 1000 times faster).

6 Attachements

Here all the source code is collected. It is also available on github.

6.1 Model code

This is the code of the model

6.1.1 Model entry point

```
# Radiative transfer model

source("radiative_transfer/shortwave.R")
source("radiative_transfer/longwave.R")
source("radiative_transfer/calc_parameters.R")



#' Radiative transfer model step
#'
#' This the core routine of the radiative trasfer model. It calls all
the models function
#'
#' @param input A data frame row (or list) containing at least the
following elements
#' - datetime
#' - sw_in total incoming shortwave radiation
#' - sw_dif diffuse shortwave radiation incoming
#' - lw_in longwave radiation incoming
#'
#' @param state A data frame row (or list) containing at least the
following elements
#' - t_soil temperature of soil (in Kelvin)
#' - t_leaf temperature of leaves (in Kelvin)
#'
#' @param pars a list of the model parameters containing at least the
following elements:
#'
#' - max LAI value in the summer
#' - min_LAI min value of LAI during winter, it is an aproximation
that consider the total Plant Area Index as LAI
#' - leaf_out day leaves start in spring
#' - leaf_full day leaves reach max LAI
#' - leaf_fall day leaves start to fall
#' - leaf_fall_complete day all leaves are fallen
#'
#' - lat latidude
#' - lon longitude
#'
#' - rho_leaf Reflencance of leaf
```

```r
#' - tau_leaf trasmissivity of leaf
#' - omega_leaf scattering coefficient of leaf
#' - clump_OMEGA canopy clumping coefficient
#' - alb_soil_b soil albedo direct beam
#' - alb_soil_d soil albedo diffuse
#'
#' - em_leaf emittivity of leaves
#' - em_soil emittivity of soil
#'
#' @return One row data Dataframe with
#' - ic Absorbed shortwave radiation from the canopy
#' - ic_sun Absorbed shortwave radiation from sunlit canopy
#' - ic_sha Absorbed shortwave radiation from shaded canopy
#' - ig Absorbed shortwave radiation from soil
#' - i_up Reflected shortwave radiation above the canopy
#' - i_down Transmitted shortwave radiation below the canopy
#' - lc Absorbed longwave radiation from the canopy
#' - lc_sun  Absorbed longwave radiation from sunlit canopy
#' - lc_sha  Absorbed longwave radiation from shaded canopy
#' - lg Absorbed longwave radiation from the soil
#' - l_up Emitted longwave radiation above the canopy
#' - l_down Transmitted longwave radiation below the canopy
#' - LAI Leaf Area Index
#' - LAI_sunlit LAI of sunlit canopy
fun_calc_radiative_transfer <- function(input, state, pars, dt){
    # Calc all the intermediate parameters
    # Possible optimization here as not all the paramaters changes
every step
    LAI <- get_day_LAI(input$datetime, pars$max_LAI, pars$leaf_out,
pars$leaf_full, pars$leaf_fall, pars$leaf_fall_complete)
    radiation_PAI <- max(LAI, pars$min_radiation_PAI) # During winter
the are no leaves but there are still branches that interact with
light
    avg_datetime <- input$datetime - duration(dt/2) # calculating the
zenith at the mid of the interval
    zenith <- get_zenith(avg_datetime, pars$lat, pars$lon)
    Kb <- get_Kb(zenith, max_Kb = 1000) # 1000 is an arbitrary high
number
    Kd <- get_Kd(LAI)
    omega_leaf <- pars$rho_leaf + pars$tau_leaf
    beta <- get_beta(pars$rho_leaf, pars$tau_leaf)
    beta0 <- get_beta0(zenith, Kb, Kd_2stream, omega_leaf)

    # the incoming shortwave is the total diffure + direct. Due to
sensor errors the difference can be negative so the min possible value
is set to 0
    sw_sky_b <- max(input$sw_in - input$sw_dif, 0)
    shortwave <- shortwave_radiation(sw_sky_b, input$sw_dif,
radiation_PAI, Kb, Kd_2stream, beta, beta0 , omega_leaf,
                                     pars$clump_OMEGA,
```

```
pars$alb_soil_b, pars$alb_soil_d)
    longwave <- longwave_radiation(input$lw_in, radiation_PAI,
state$t_leaf, state$t_soil, Kb, Kd, pars$em_leaf, pars$em_soil)

    LAI_sunlit <- get_LAI_sunlit(LAI, Kb, pars$clump_OMEGA)
    LAIs <- c(LAI=LAI, LAI_sunlit=LAI_sunlit)

    return(data.frame(c(shortwave, longwave, LAIs)))
}
# The Kd in the Two Stream model has a different value
Kd_2stream <- get_two_stream_Kd() # This is a costant value that
depends only on the leaf angle distribution
```

6.1.2 Parameter calculation

```
library(pracma) # for rad2deg and deg2rad
library(solartime) # for calculating zenith
library(lubridate) # datetime utils



#' Simple model to get current LAI
#'
#' Use a simple model to get the LAI of the different day of the year.
#' It has 4 phases:
#'  - Winter: from leaf_fall_complete to leaf_out -> LAI = 0
#'  - Spring: from leaf_out to leaf_full -> linear growth from 0 to
max_LAI
#'  - Summer: from leaf_full to leaf_fall -> LAI = max_LAI
#'  - Fall: from leaf_fall to leaf_fall_complete -> linear decrease
from max_LAI to 0
#' The 4 paramenters (leaf_out...) are the day of the year
#'
#' @param time a datetime object
#' @param max_LAI max LAI value in the summer
#' @param leaf_out day leaves start in spring
#' @param leaf_full day leaves reach max LAI
#' @param leaf_fall day leaves start to fall
#' @param leaf_fall_complete day all leaves are fallen
#'
#' @return LAI Leaf Area Index value for the day of the year
get_day_LAI <- function(datetime, max_LAI, leaf_out, leaf_full,
leaf_fall, leaf_fall_complete) {

  yday <- yday(datetime)
  if (yday < leaf_out) { # before leaves are out LAI is min
    return(0)
  }
  if (yday >= leaf_out & yday < leaf_full) {
    ndays <- leaf_full - leaf_out # n days of the transition
    return(max_LAI * (yday - leaf_out) / ndays)
  }
```

```
  if (yday >= leaf_full & yday < leaf_fall) {
    return(max_LAI)
  }
  if (yday >= leaf_fall & yday < leaf_fall_complete) {
    ndays <- leaf_fall_complete - leaf_fall # n days of the transition
    return(max_LAI * (leaf_fall_complete - yday) / ndays)
  }
  if (yday >= leaf_fall_complete) {
    return(0)
  }
}

#' Solar zenith from datetime and geographical coordinates
#'
#' @param time Datetime object with the current time
#' @param lon Longitude
#' @param lat Latidute
#'
#' @return solar zenith (in degrees) between 0 and 90
get_zenith <- function(time, lat, lon) {
  elevation <- computeSunPosition(time, lat, lon)[3]
  Z <- 90 - rad2deg(elevation)
  Z <- min(90, Z) # avoid zenith values below horizon
  return(Z)
}


#'  All the following function assumes a SPHERICAL leaves distribution
#' Chapter 2.2


#' Direct beam extiction coefficient
#' @param zenith in degrees
#' @return Kb
get_Kb <- function(zenith, max_Kb = 20) {
  # Eq. 14.29
  Kb <- 0.5 / cos(deg2rad(zenith)) # extinction coefficient
  Kb <- min(Kb, max_Kb) # Prevent the Kb to become too large at low
sun angles.
  # The default value of 20 is from the Bonan matlab code script
sp_14_03 line 150
  return(Kb)
}

#' Diffuse radiation extiction coefficient
#' @param LAI
#' @return Kd
get_Kd <- function(LAI) {
  G_z <- 0.5
```

```
  # Eq. 14.33
  td <- 0
  for (z in seq(0, pi / 4, pi / 18)) { # make 9 steps from 0 till pi/2
    td <- td + exp(-G_z / cos(z) * LAI) *
      sin(z) *
      cos(z) *
      (pi / 18)
  }

  # Eq 14.34
  Kd <- -log(2 * td) / LAI
  return(Kd)
}

#' Diffuse radiation extiction coefficient for the 2 stream
aproxmiation
#' This depends only on the leaf angle distribution
#' @param LAI
#' @return Kd
get_two_stream_Kd <- function() {
  # Eq. 14.31
  ross <- 0.01 # should be zero but if is zero it mess up the
computations.
  # See Bonan matlab code script sp_14_03 line 130
  phi_1 <- 0.5 - 0.633 * ross - 0.333 * (ross)^2
  phi_2 <- 0.877 * (1 - 2 * phi_1)
  # Eq 14.80
  Kd <- 1 / ((1 - phi_1 / phi_2 * log((phi_1 + phi_2) / phi_1)) /
phi_2)
  return(Kd)
}


#' Fraction of diffuse light scattered backward
#' @param rho_leaf
#' @param tau_leaf
#'
#' @return beta
get_beta <- function(rho_leaf, tau_leaf) {
  # Derived from equations 14.81 following the book approximation for
sperical distribution
  beta <- (0.625 * rho_leaf + 0.375 * tau_leaf) / (rho_leaf +
tau_leaf)
  return(beta)

}

#' Fraction of direct light scattered backward
```

```r
#' @param zenith in degrees
#' @param Kb
#' @param Kd
#' @param omega_leaf
#'
#' @return beta0
get_beta0 <- function(zenith, Kb, Kd, omega_leaf) {

  # Eq. 14.31
  ross <- 0
  phi_1 <- 0.5 - 0.633 * ross - 0.333 * (ross)^2
  phi_2 <- 0.877 * (1 - 2 * phi_1)

  G_mu <- 0.5 #mu is cos(Z) but G(Z) for sperical leaves distribution
is costant
  mu <- cos(deg2rad(zenith))

  # Equation 14.84

  #defining commonly used terms
  mphi_1 <- mu * phi_1
  mphi_2 <- mu * phi_2

  a_s <- (
    (omega_leaf / 2) * (G_mu) / (G_mu + mphi_2) *
      (1 - (mphi_1 / (G_mu + mphi_2) * log((G_mu + mphi_1 + mphi_2) /
mphi_1)))
  )

  beta_0 <- (((Kb + Kd) / Kb) * a_s) / omega_leaf
  return(beta_0)
}

get_LAI_sunlit <- function(LAI, Kb, clump_OMEGA) {
  # Eq.14.18 integrated in the same way of Eq. 14.12 (also line in
Bonan Matlab code line script sp_14_03 line 167)
  LAI_sunlit <- (1 - exp(-clump_OMEGA * Kb * LAI)) / Kb
  return(LAI_sunlit)
}
```

6.1.3 Shortwave

```r
## solving for direct/diffuse shortwave using the two stream
approximation

#' Calculate the direct shortwave radiation absorbed by the canopy
with sunlit and shaded components
#' @param sw_sky_b direct beam radiation above the canopy in W m-2
#' @param LAI Leaf Area Index
#' @param Kb Direct beam extiction coefficient
```

```r
#' @param Kd Diffuse exiction coefficient
#' @param beta Fraction of diffuse radiation upward scattered
#' @param beta0 Fraction of direct beam radiation upward scattered
#' @param omega_leaf Leaf scattering coefficient (reflectace +
trasmittance)
#' @param clump_OMEGA Canopy clumping coefficient
#' @param alb_soil_b Albedo soil for direct beam radiation
#'
#' @return list of ic, ic_sun, ic_sha
direct_beam_radiation <- function(sw_sky_b, LAI, Kb, Kd, beta, beta0,
omega_leaf, clump_OMEGA, alb_soil_b, alb_soil_d){

  # defining common terms between direct/diffuse
  # --- Common terms: Eqs. (14.87) - (14.91)

  b <- (1 - (1 - beta) * omega_leaf) * Kd
  c <- beta * omega_leaf * Kd
  h <- sqrt(b*b - c*c)
  u <- (h - b - c) / (2 * h)
  v <- (h + b + c) / (2 * h)
  #d = omega(p,ib) * Kb(p) * atmos.swskyb(p,ib) / (h*h - Kb(p)*Kb(p));
  g1 <- ((beta0 * Kb - b * beta0 - c * (1 - beta0)) *
         omega_leaf * Kb * sw_sky_b / (h^2 - Kb^2))
  g2 <- ((1 - beta0) * Kb + c * beta0 + b * (1 - beta0)) * omega_leaf
* Kb * sw_sky_b / (h*h - Kb^2)

  # --- Exponential functions of leaf area

  s1 <- function(x) exp(-h * clump_OMEGA * x);
  s2 <- function(x) exp(- Kb * clump_OMEGA * x)



  # --- Direct beam solution
  # n1 (Eq. 14.92) and n2 (14.93)

  num1 <- v * (g1 + g2 * alb_soil_b + alb_soil_b * sw_sky_b) * s2(LAI)
  num2 <- g2 * (u + v * alb_soil_b) * s1(LAI)
  den1 <- v * (v + u * alb_soil_b) / s1(LAI)
  den2 <- u * (u + v * alb_soil_b) * s1(LAI)
  n2b <- (num1 - num2) / (den1 - den2)
  n1b <- (g2 - n2b * u) / v

  # Scattered direct beam fluxes:
  # iupwb - direct beam flux scattered upward above cumulative LAI
(W/m2); Eq. (14.94)
  # idwnb - direct beam flux scattered downward below cumulative LAI
(W/m2); Eq. (14.95)

  i_upw_b <-  function(x) -g1 * s2(x) + n1b * u * s1(x) + n2b * v /
```

```
s1(x)
  i_dwn_b <-  function(x)  g2 * s2(x) - n1b * v * s1(x) - n2b * u /
s1(x)

  # icb - direct beam flux absorbed by canopy (W/m2); Eq. (14.97)

  ic_b <- sw_sky_b * (1 - s2(LAI)) - i_upw_b(0) + i_upw_b(LAI) -
i_dwn_b(LAI)

  # ig_b - direct beam flux absorbed by the soil; Eq 14.98

  ig_b <- ((1- alb_soil_d) * i_dwn_b(LAI)) + ((1 - alb_soil_b) *
sw_sky_b * s2(LAI))

  # icsunb - direct beam flux absorbed by sunlit canopy (W/m2); Eq.
(14.114)
  # icshab - direct beam flux absorbed by shaded canopy (W/m2); Eq.
(14.115)

  a1b <- -g1 *      (1 - s2(LAI)*s2(LAI)) / (2 * Kb) +
    n1b * u * (1 - s2(LAI)*s1(LAI)) / (Kb + h) + n2b * v * (1 -
s2(LAI)/s1(LAI)) / (Kb - h)
  a2b <-  g2 *      (1 - s2(LAI)*s2(LAI)) / (2 * Kb) -
    n1b * v * (1 - s2(LAI)*s1(LAI)) / (Kb + h) - n2b * u * (1 -
s2(LAI)/s1(LAI)) / (Kb - h)

  ic_sun_b <- (1 - omega_leaf) * ((1 - s2(LAI)) * sw_sky_b + Kd * (a1b
+ a2b) * clump_OMEGA)
  ic_sha_b <- ic_b - ic_sun_b

  i_up_b <- i_upw_b(0)
  i_down_b <- i_dwn_b(LAI)

  return(list(ic_b = ic_b, ic_sun_b=ic_sun_b, ic_sha_b=ic_sha_b, ig_b
= ig_b, i_up_b = i_up_b, i_down_b = i_down_b))
}

#' Calculate the diffuse shortwave radiation absorbed by the canopy
with sunlit and shaded components
#' @param sw_sky_d diffuse radiation above the canopy in W m-2
#' @param LAI Leaf Area Index
#' @param Kb Direct beam extiction coefficient
#' @param Kd Diffuse exiction coefficient
#' @param beta Fraction of diffuse radiation upward scattered
#' @param beta0 Fraction of direct beam radiation upward scattered
#' @param omega_leaf Leaf scattering coefficient (reflectace +
trasmittance)
#' @param clump_OMEGA Canopy clumping coefficient
#' @param alb_soil_d Albedo soil for diffuse radiation
#'
```

```
#' @return list of ic, ic_sun, ic_sha, ig, i_up_d, i_down_d
diffuse_radiation <- function(sw_sky_d, LAI, Kb, Kd, beta, beta0,
omega_leaf, clump_OMEGA, alb_soil_d){

  # defining common terms between direct/diffuse
  # --- Common terms: Eqs. (14.87) - (14.91)

  b <- (1 - (1 - beta) * omega_leaf) * Kd
  c <- beta * omega_leaf * Kd
  h <- sqrt(b*b - c*c)
  u <- (h - b - c) / (2 * h)
  v <- (h + b + c) / (2 * h)
  #g1 <- ((beta0 * Kb - b * beta0 - c * (1 - beta0))
  #       * omega_leaf * Kb * sw_sky_d / (h^2 - Kb^2))
  #g2 <- ((1 - beta0) * Kb + c * beta0 + b * (1 - beta0)) * omega_leaf
* Kb * sw_sky_d / (h*h - Kb^2)

  # --- Exponential functions of leaf area

  s1 <- function(x) exp(-h * clump_OMEGA * x);
  s2 <- function(x) exp(-Kb * clump_OMEGA * x)

  # --- Diffuse solution

  # n1d and n2d (Eq. 14.99) and n2 (14.100)
  num <- sw_sky_d * (u + v * alb_soil_d) * s1(LAI)
  den1 <- v * (v + u * alb_soil_d) / s1(LAI)
  den2 <- u * (u + v * alb_soil_d) * s1(LAI)
  n2d <- num / (den1 - den2)
  n1d <- -(sw_sky_d + n2d * u) / v

  # Scattered diffuse fluxes:
  # iupwd - diffuse flux scattered upward above cumulative LAI (W/m2);
Eq. (14.101)
  # idwnd - diffuse flux scattered downward below cumulative LAI
(W/m2); Eq. (14.102)

  i_upw_d <- function(x)  n1d * u * s1(x) + n2d * v / s1(x)
  i_dwn_d <- function(x) -n1d * v * s1(x) - n2d * u / s1(x)


  #' icd - diffuse flux absorbed by canopy (W/m2); Eq. (14.104)
  ic_d <- sw_sky_d - i_upw_d(0) + i_upw_d(LAI) - i_dwn_d(LAI)

  #' ig_b - diffuse flux absorbed by the soil; Eq 14.105
  ig_d <- (1 - alb_soil_d) * i_dwn_d(LAI)

  # icsund - diffuse flux absorbed by sunlit canopy (W/m2); Eq.
(14.120)
  # icshad - diffuse flux absorbed by shaded canopy (W/m2); Eq.
```

```r
(14.121)

  a1d <-  n1d * u * (1 - s2(LAI)*s1(LAI)) / (Kb + h) + n2d * v * (1 -
s2(LAI)/s1(LAI)) / (Kb - h)
  a2d <- -n1d * v * (1 - s2(LAI)*s1(LAI)) / (Kb + h) - n2d * u * (1 -
s2(LAI)/s1(LAI)) / (Kb - h)

  ic_sun_d <- (1 - omega_leaf) * Kd * (a1d + a2d) * clump_OMEGA
  ic_sha_d <- ic_d - ic_sun_d

  i_up_d <-  i_upw_d(0)
  i_down_d <-  i_dwn_d(LAI)

  return(list(ic_d = ic_d, ic_sun_d = ic_sun_d, ic_sha_d = ic_sha_d,
ig_d = ig_d , i_up_d = i_up_d, i_down_d = i_down_d))
}

#' Calculate the shortwave radiation absorbed by the canopy with
sunlit and shaded components
#'
#' @param sw_sky_b direct beam radiation above the canopy in W m-2
#' @param sw_sky_d diffuse radiation above the canopy in W m-2
#' @param LAI Leaf Area Index
#' @param Kb Direct beam extiction coefficient
#' @param Kd Diffuse exiction coefficient
#' @param beta Fraction of diffuse radiation upward scattered
#' @param beta0 Fraction of direct beam radiation upward scattered
#' @param omega_leaf Leaf scattering coefficient (reflectace +
trasmittance)
#' @param clump_OMEGA Canopy clumping coefficient
#' @param alb_soil_b Albedo soil for direct beam radiation
#' @param alb_soil_d Albedo soil for diffuse radiation
#'
#' @return list of ic, ic_sun, ic_sha
shortwave_radiation <- function(sw_sky_b, sw_sky_d, LAI, Kb, Kd, beta,
beta0, omega_leaf, clump_OMEGA, alb_soil_b, alb_soil_d){

  ib <- direct_beam_radiation(sw_sky_b, LAI, Kb, Kd, beta, beta0,
omega_leaf, clump_OMEGA, alb_soil_b, alb_soil_d)
  id <- diffuse_radiation(sw_sky_d, LAI, Kb, Kd, beta, beta0,
omega_leaf, clump_OMEGA, alb_soil_d)

  ic <- ib$ic_b + id$ic_d
  ic_sun <- ib$ic_sun_b + id$ic_sun_d
  ic_sha <- ib$ic_sha_b + id$ic_sha_d
  ig <- ib$ig_b + id$ig_d
  i_up <- ib$i_up_b + id$i_up_d
  i_down <- ib$i_down_b + id$i_down_d

  return(list(ic = ic, ic_sun = ic_sun, ic_sha = ic_sha, ig=ig, i_up =
```

```
i_up, i_down = i_down))

}
```

6.1.4 Longwave

```
## simplified longwave model (assumes there is no upward scattering)

#' Calculate the longwave radiation absorbed by the canopy with sunlit
and shaded components
#' @param lw_sky_d longwave radiation above the canopy in W m-2
#' @param t_leaf temperature leaves (in Kelvin)
#' @param t_soil temperature of soil (in Kelvin)
#' @param LAI Leaf Area Index
#' @param Kb Direct beam extiction coefficient
#' @param Kd Diffuse exiction coefficient
#' @param em_leaf Longwave emissivity of leaf
#' @param em_soil Longwave emissivity of soil
#'
#' @return list of lc, lg, lc_sun, lc_sha, l_up, l_down
longwave_radiation <- function(lw_sky, LAI, t_leaf, t_soil, Kb, Kd,
em_leaf, em_soil){
  ## commonly used terms--
  sigma <- 5.67e-08   # Stefan-Boltzmann constant W m-2 K-4

  lw_soil_emit <- em_soil * sigma * t_soil^4
  lw_leaf_emit <- em_leaf * sigma  * t_leaf^4

  ## Equation 14.134
  lw_down_trans <- function(x)
    lw_sky * (1 - em_leaf * (1-exp(-Kd * x)))
  lw_down_emit <- function(x)
    lw_leaf_emit *(1-exp(-Kd * x))

  lw_down <- function(LAI) lw_down_emit(LAI) + lw_down_trans(LAI)


  ## Equation 14.135

  lw_up_trans <- function(x) {
    lw_soil_emit * (1 - em_leaf * (1-exp(-Kd * (LAI - x))))
  }

  lw_up_emit <- function(x)
    lw_leaf_emit * (1-exp(-Kd * (LAI - x)))

  lw_up <- function (x) lw_up_trans(x) + lw_up_emit(x)
  ## Equation 14.137
  perc_abs <-  1-exp(-Kd * LAI) # amount absorbed
  lc <- perc_abs * (em_leaf * (lw_sky + lw_soil_emit)
```

```
        - 2 * lw_leaf_emit)

  ## Equation 14.138
  lg<- lw_down(LAI) - lw_soil_emit # Lw adboserbed by the soil


  ### --- Sunlit and shaded leaves ---

  # Sunlit Eq. 14.140

  lc_sun <-
  (
      (
        (em_leaf * (lw_sky - sigma * t_leaf ^ 4 ) * Kd )
        / (Kd + Kb)
        * (1 - exp(-(Kd + Kb) * LAI))
      )
      +
      (
        (em_leaf *(lw_soil_emit - sigma * t_leaf ^ 4 ) * Kd)
        / (Kd - Kb)
        * (exp(-Kb * LAI)  - exp(-Kd * LAI))
      )
  )

  # Shaded Eq. 14.141

  lc_sha <- lc - lc_sun

  return(list(lc = lc,                       # Lw radiation absorbed
by the canopy
              lg = lg,                       # Lw radiation absorbed
by the soil
              lc_sun = lc_sun,               # Lw radiation absorbed
by the sunlit canopy
              lc_sha = lc_sha,               # Lw radiation absorbed
by the shaded canopy
              l_up = lw_up(0),            # Lw emitted into the sky
              l_down = lw_down(LAI)          # Lw reaching the soil
  ))
}
```

6.2 Report code

This is the code used for writing this report

6.2.1 Setup

```
# setup input and fluxes for 2016 (because in 2018 lw is NA)
library(tidyverse)
library(progress)
```

```
library(scales)
dt <- 3600
source("setup_parameters.R")

source("fun_calc_radiative_transfer.R")

input <- read.csv(file.path('data', 'Hainich_2018_input.csv'))
fluxes <- read.csv(file.path('data', 'Hainich_2018_fluxes.csv'))

# Initial variable selection, renaming and conversion
input <- input %>% mutate(
  time = 1:nrow(input),
  datetime = force_tz(as_datetime(Date.Time), "Etc/GMT+1"),
  tair = TA_F + 273.15,  # Celsius to Kelvin
  p = P_F/1000,          # mm time_step-1 to m3 m-2 time_step-1
  sw_in = SW_IN_F,       # W m-2
  lw_in = LW_IN_F,       # W m-2
  ppfd_in = PPFD_IN,     # µmol m-2 s-1
  vpd = VPD_F * 100,     # hPa to Pa
  pa = PA_F * 1000,      # kPa to Pa
  ws = WS_F,             # m s-1
  rh = RH / 100,         # percent to fraction
  sw_dif = SW_DIF,       # W m-2
  co2 = CO2_F_MDS,       # µmol mol-1
  night = as_factor(as.integer(NIGHT)),
  .keep = "unused"
)


# Initial variable selection, renaming and conversion
# tsoil and swc means across soil depths don't take into account layer
thickness or properties.
# We ignore this for the purpose of this excersice.
fluxes <- fluxes %>% mutate(
  time = 1:nrow(fluxes),
  sw_out = SW_OUT,          # W m-2
  tsoil = ((TS_F_MDS_1 + TS_F_MDS_2 + TS_F_MDS_3 + TS_F_MDS_4 +
TS_F_MDS_5) / 5) + 273.15, # 30cm depth mean. Celsius to Kelvin
  swc = ((SWC_F_MDS_1 + SWC_F_MDS_2 + SWC_F_MDS_3) / 3) / 100, # 30cm
depth mean. Percent to fraction
  g = G_F_MDS,              # W m-2
  le = LE_F_MDS,            # W m-2
  h = H_F_MDS,              # W m-2
  nee = NEE_VUT_REF * 12 / 1000000 / 1000 * dt,        # µmol m-2 s-1
to kg m-2 dt-1
  reco = RECO_NT_VUT_REF * 12 / 1000000 / 1000 * dt,   # µmol m-2 s-1
to kg m-2 dt-1
  gpp = GPP_NT_VUT_REF * 12 / 1000000 / 1000 * dt,     # µmol m-2 s-1
to kg m-2 dt-1
  TIMESTAMP_START = NULL,
```

```r
  TIMESTAMP_END = NULL,
  NIGHT = NULL,
  lw_out = LW_OUT,
  TS_F_MDS_5 = NULL,
  LE_RANDUNC = NULL,
  H_RANDUNC = NULL,
  NEE_VUT_REF_JOINTUNC = NULL,
  .keep = "unused"
)


state <- tibble(t_leaf = input$tair, t_soil = fluxes$tsoil)

# Data for 1 month used for calibration and sensitivity

input_1m <- read.csv(file.path("data", "Hainich_2018-07_input.csv"))

fluxes_1m <- read.csv(file.path("data", "Hainich_2018-07_fluxes.csv"))

# Initial variable selection, renaming and conversion
input_1m <- input_1m %>% mutate(
  time = 1:nrow(input_1m),
  datetime = force_tz(as_datetime(Date.Time), "Etc/GMT+1"),
  tair = TA_F + 273.15,  # Celsius to Kelvin
  p = P_F,               # mm / time_step = l m-2 / time_step
  sw_in = SW_IN_F,       # W m-2
  lw_in = LW_IN_F,       # W m-2
  ppfd_in = PPFD_IN,     # µmol m-2 s-1
  vpd = VPD_F * 100,     # hPa to Pa
  pa = PA_F * 1000,      # kPa to Pa
  ws = WS_F,             # m s-1
  rh = RH / 100,         # percent to fraction
  sw_dif = SW_DIF,       # W m-2
  co2 = CO2_F_MDS,       # µmol mol-1
  NIGHT = NULL,
  .keep = "unused"
)


# Initial variable selection, renaming and conversion
# tsoil and swc means across soil depths don't take into account layer
thickness or properties.
# We ignore this for the purpose of this excersice.
fluxes_1m <- fluxes_1m %>% mutate(
  time = 1:nrow(fluxes_1m),
  sw_out = SW_OUT,          # W m-2
  lw_out = LW_OUT,          # W m-2
  tsoil = ((TS_F_MDS_1 + TS_F_MDS_2 + TS_F_MDS_3 + TS_F_MDS_4) / 4) +
273.15, # 30cm depth mean. Celsius to Kelvin
```

```r
  swc = ((SWC_F_MDS_1 + SWC_F_MDS_2 + SWC_F_MDS_3) / 3) / 100, # 30cm
depth mean. Percent to fraction
  g = G_F_MDS,              # W m-2
  le = LE_F_MDS,            # W m-2
  h = H_F_MDS,              # W m-2
  nee = NEE_VUT_REF * 12 / 1000000 / 1000 * dt,        # µmol m-2 s-1
to kg m-2 dt-1
  reco = RECO_NT_VUT_REF * 12 / 1000000 / 1000 * dt,   # µmol m-2 s-1
to kg m-2 dt-1
  gpp = GPP_NT_VUT_REF * 12 / 1000000 / 1000 * dt,     # µmol m-2 s-1
to kg m-2 dt-1
  TIMESTAMP_START = NULL,
  TIMESTAMP_END = NULL,
  NIGHT = NULL,
  TS_F_MDS_5 = NULL,
  LE_RANDUNC = NULL,
  H_RANDUNC = NULL,
  NEE_VUT_REF_JOINTUNC = NULL,
  .keep = "unused"
)

state_1m <- tibble(t_leaf = input_1m$tair, t_soil = fluxes_1m$tsoil)

# Utility funcs

#' Full radiative transfer model with potentially new paramameters
that overwrite the defaults
rad_transf_new_p <- function (new_p=list()){
  pars <- merge_lists(pars, new_p)
  rad_transf(pars = pars)
}

#' Full radiative transfer model with potentially new paramameters
that overwrite the defaults. Using 1 month data
rad_transf_new_p_1m <- function (new_p=list()){
  pars <- merge_lists(pars, new_p)
  rad_transf(pars = pars, input= input_1m, state=state_1m)
}

#' Full radiative transfer model

d_input <- input
d_state <- state
d_pars <-  pars
d_dt <- dt
rad_transf <- function(input = d_input, state = d_state, pars =
d_pars, dt = d_dt){
  out <- data.frame()
  pb <- setup_pb()
  for (i in seq_along(input$datetime)){
```

```r
    rad<- radiative_transfer_step_debug(input[i,], state[i,], pars,
dt)
    out[i, names(rad)] <- rad
    pb$tick()
  }
  return(out)
}

setup_pb <- function(){
  progress_bar$new(format = "(:spin) [:bar] :percent
[Elapsed: :elapsedfull | ETA: :eta]",
                       total = length(input$datetime),
                       clear = FALSE)
}

#' merges two list, in case of conflicts uses the second list value
merge_lists <- function (list1, list2){
  l <- list1
  for (name in names(list2)){
    l[name] <- list2[name]
  }
  return(l)
}

#' transpose an aggreagated sensisivity dataframe
invert_sens <- function(df){
    vars <- df$var
    df <- select(df, -var)
    pars <- names(df)
    df <- as_tibble(t(df))
    df <- cbind(pars, df)
    names(df) <- c("var", vars)
    return(df)
}

filter_sens <- function(sens_model, vars = NULL, pars = NULL){
  sens_model <- select(sens_model, -x)
  if (!is.null(vars)) {sens_model <- filter(sens_model, var %in%
vars)}
  if (!is.null(pars)) {sens_model <- select(sens_model,
all_of(c("var", pars)))}
  return(sens_model)
}

detailed_sens <- function(sens_model, func){
  sens_model %>%
        group_by(var) %>%
        summarize_all(func) %>%
        invert_sens
}
```

```
Kd_2stream <- get_two_stream_Kd() # This is a costant value that
depends only on the leaf angle distribution
#' This is the rad_transf function that can take custom parameters
from the input
radiative_transfer_step_debug <- function(input, state, pars, dt){
    # Calc all the intermediate parameters
    # Possible optimization here as not all the paramaters changes
every step
    LAI <- ifelse("LAI" %in% names(input), input$LAI,
get_day_LAI(input$datetime, pars$max_LAI, pars$leaf_out,
pars$leaf_full, pars$leaf_fall, pars$leaf_fall_complete))
    radiation_PAI <- max(LAI, pars$min_radiation_PAI) # During winter
the are no leaves but there are still branches that interact with
light
    avg_datetime <- input$datetime - duration(dt/2) # calculating the
zenith at the mid of the interval
    zenith <- ifelse("zenith" %in% names(input), input$zenith,
get_zenith(avg_datetime, pars$lat, pars$lon))
    Kb <- ifelse("Kb" %in% names(input), input$Kd, get_Kb(zenith,
max_Kb = 1000)) # 1000 is an arbitraty high number
    Kd <- ifelse("kd" %in% names(input), input$Kb, get_Kd(LAI))
    omega_leaf <- pars$rho_leaf + pars$tau_leaf
    beta <- get_beta(pars$rho_leaf, pars$tau_leaf)
    beta0 <- get_beta0(zenith, Kb, Kd_2stream, omega_leaf)

    # the incoming shortwave is the total diffure + direct. Due to
sensor errors the difference can be negative so the min possible value
is set to 0
    sw_sky_b <- max(input$sw_in - input$sw_dif, 0)
    shortwave <- shortwave_radiation(sw_sky_b, input$sw_dif,
radiation_PAI, Kb, Kd_2stream, beta, beta0 , omega_leaf,
                                     pars$clump_OMEGA,
pars$alb_soil_b, pars$alb_soil_d)
    longwave <- longwave_radiation(input$lw_in, radiation_PAI,
state$t_leaf, state$t_soil, Kb, Kd, pars$em_leaf, pars$em_soil)

    LAI_sunlit <- get_LAI_sunlit(LAI, Kb, pars$clump_OMEGA)
    vars <- c(LAI=LAI, LAI_sunlit=LAI_sunlit, radiation_PAI =
radiation_PAI, Kb=Kb, Kd=Kd, beta=beta, beta0= beta0, zenith=zenith)

    return(data.frame(c(shortwave, longwave, vars)))
}

#' Adds a white and black background when is night
#' data should be a dataframe with datetime and the night column
without repetitions (eg. not in tidy format)
```

```r
 night_bg <- function(data, y_mean=0){
  list(
    geom_tile(aes(x= datetime, width = dt, y = y_mean, height = Inf,
fill = night), alpha = .4, linetype = 0, data=data),
    scale_fill_manual(name = NULL, values = c("#ececec", "#555555"),
labels = c("day", "night"),
                      guide = guide_legend(override.aes = list(colour =
c("#ececec", "#b9b9b9"), alpha = .4)))
  )
}

legend_labels <- function(labels, name="", max_width = 10){
  list(
  scale_color_hue(labels = str_wrap(labels, max_width), str_wrap(name,
max_width)),
  theme(legend.key.height = unit(40, "pt"))
  )
}
# breaks_12hours <- function (limits){
#   seq(limits[1], limits[2], by="12 hours")
# }
#
# labels_12hours <- function (limits){
#   format(breaks_12hours(limits), "%d %b %Ih")
# }
```

6.2.2 Markdown

```r
source("radiative_transfer/term paper/radiation_utils_test_data.R")
library(FME)
days <- seq.Date(as.Date("2020-01-01"), as.Date("2020-12-31"), by=1)
LAI <- pmax(Vectorize(get_day_LAI, "datetime")
            (days, pars$max_LAI, pars$leaf_out, pars$leaf_full,
pars$leaf_fall, pars$leaf_fall_complete), pars$min_radiation_PAI)
ggplot() +
  geom_line(aes(x = days, y = LAI)) +
  ylim(c(0, pars$max_LAI)) +
  labs(title = "LAI over the year")
sens_p <- pars[c("rho_leaf", "tau_leaf", "alb_soil_b", "alb_soil_d",
"em_leaf", "em_soil")]
sens_model <- sensFun(rad_transf_new_p_1m, sens_p, map=NULL)
sens_sum <- summary(sens_model)
# cbind(par = attr(sens_sum, "row.names"), sens_sum) #little hack
because pycharm doesn't show row names properly
knitr::kable(
  sens_sum, booktabs = TRUE,
  digits = 2,
  caption = 'Aggregated model sensitivity.'
)
sens_sw <- filter_sens(sens_model, vars = c("i_down", "i_up", "ic",
"ic_sha", "ic_sun", "ig"),
```

```r
                      pars = c("rho_leaf", "tau_leaf" ,"alb_soil_b",
"alb_soil_d"))
knitr::kable(detailed_sens(sens_sw, mean), caption = "Shortwave
sensitivity aggregated by mean", digits = 2)
knitr::kable(detailed_sens(sens_sw, function(x) mean(abs(x))), caption
= "Shortwave sensitivity aggregated by L1", digits = 2)
knitr::kable(detailed_sens(sens_sw, function(x) sqrt(mean(x^2))),
caption = "Shortwave sensitivity aggregated by L2", digits = 2)
sens_lw <- filter_sens(sens_model, vars = c("l_down", "l_up", "lc",
"lc_sha", "lc_sun", "lg"),
                      pars = c("em_leaf", "em_soil"))
knitr::kable(detailed_sens(sens_lw, mean), caption = "Longwave
sensitivity aggregated by mean", digits = 2)
knitr::kable(detailed_sens(sens_lw, function(x) mean(abs(x))), caption
= "Longwave sensitivity aggregated by L1", digits = 2)
knitr::kable(detailed_sens(sens_lw, function(x) sqrt(mean(x^2))),
caption = "Longwave sensitivity aggregated by L2", digits = 2)
cal_p_sw <- c(rho_leaf = 0.4, tau_leaf = 0.1)
cal_p_sw_lower <- c(rho_leaf = 0.38, tau_leaf = 0.05 )
cal_p_sw_upper <- c(rho_leaf = 0.42, tau_leaf = 0.2)

model_cost_sw <- function (params){
  out <- rad_transf_new_p_1m(params)
  return(out$i_up - fluxes$sw_out)
}
mfit_sw <- modFit(model_cost_sw, cal_p_sw, cal_p_sw_lower,
cal_p_sw_upper)
knitr::kable(
  summary(mfit_sw)$par[,1],
  caption = "Shortwave parameters after calibration"
)
source('radiative_transfer/term paper/radiation_utils_test_data.R')
# library(scales)
library(tsibble)
library(hydroGOF)
out <- rad_transf_new_p()

out_plot <- tibble(datetime = input$datetime, night = input$night,
mod_sw_out = out$i_up, obs_sw_out = fluxes$sw_out, mod_lw_out =
out$l_up, obs_lw_out = fluxes$lw_out)
out_all <- cbind(input, select(fluxes, -time, -Date.Time), out)
out_1week <- filter(out_all, datetime > as_date("2018-07-1") &
datetime < as_date("2018-07-8"))

out_1year_wk <- out_all %>% # to day average
  select(-c(night, time)) %>%
  as_tsibble(index = datetime) %>%
  index_by(week = ~ yearweek(.)) %>%
  summarise_all(mean, na.rm = TRUE)
out_1week %>%
```

```
        gather(key = "type", value = "radiation", sw_in, ic, i_up, ig, 
factor_key = T) %>%
        ggplot() +
        night_bg(out_1week) +
        geom_line(aes(x = datetime, y = radiation, color = type)) +
        legend_labels(c("Incoming sw", "Absorbed sw canopy", "Upward 
sw", "Absorbed sw soil")) +
        labs(x = "Datetime", y = "Shortwave radiation (W m-2)", title 
= "Shortwave one week", subtitle = "Modeled output variables (1-8 Jul 
2018)")
lai_breaks <- as_datetime(c("2018-04-19", "2018-06-19", "2018-10-7", 
"2018-10-28"))
lai_breaks_lbl <- c("leaf out", "leaf complete", "leaf fall", "leaf 
fall complete")
pos <- function (x) {x+10}
out_1year_wk %>%
        gather(key = "type", value = "radiation", , sw_in, ic, i_up, 
ig, factor_key = T) %>%
        ggplot() +
        geom_vline(xintercept = lai_breaks, linetype=2, alpha = .8, 
size=.4) +
        annotate('label', x=lai_breaks, y=c(315, 315, 315, 290), label 
= lai_breaks_lbl) + #make the last lable lower to avoid overlap
        geom_line(aes(x = datetime, y = radiation, color = type)) +
        legend_labels(c("Incoming sw", "Absorbed sw canopy", "Upward 
sw", "Absorbed sw soil")) +
        labs(x= "Datetime", y = "Shortwave radiation (W m-2)",
             title="Shortwave one year", subtitle = "Modeled output 
variables, weekly average 2018")
out_1week %>%
        gather(key = "type", value = "radiation", lw_in, l_up, lc, lg, 
factor_key = T) %>%
        ggplot() +
        geom_hline(yintercept = 0, linetype = 2) +
        night_bg(out_1week) +
        geom_line(aes(x = datetime, y = radiation, color = type)) +
        labs(x = "Datetime", y = "Longwave radiation (w m-2)",
             title = "Longwave one week", subtitle = "Modeled output 
variables (1-8 Jul 2018)") +
        legend_labels(c("Incoming lw", "Upward lw", "Absorbed lw 
canopy", "Absorbed lw soil"))
out_1year_wk %>%
  gather(key = "type", value = "radiation", lw_in, l_up, lc, lg, 
factor_key = T) %>%
  ggplot() +
  geom_hline(yintercept = 0, linetype=2) +
  geom_line(aes(x = datetime, y = radiation, color = type)) +
  labs(x="Datetime", y = "Longwave radiation (w m-2)",
   title="Shortwave one year", subtitle = "Modeled output variables, 
weekly average 2018") +
```

```
  legend_labels(c("Incoming lw", "Upward lw", "Absorbed lw canopy",
"Absorbed lw soil"))
out_1week %>%
        gather(key = "type", value = "radiation", i_up, sw_out,
factor_key = T) %>%
        ggplot() +
        night_bg(out_1week) +
        geom_line(aes(x = datetime, y = radiation, color = type)) +
        labs(y = "Outgoing shortwave radiation (w m-2)",
        title= "Modeled vs Observed outgoing shortwave radiation",
subtitle = "1 summer week (1-8 Jul 2018)") +
        legend_labels(labels = c("Modelled", "Observed"), "Outgoing
shortwave radiation")
out_1year_wk %>%
        gather(key = "type", value = "radiation", i_up, sw_out,
factor_key = T) %>%
        ggplot() +
        geom_line(aes(x = datetime, y = radiation, color = type)) +
        labs(y = "Outgoing shortwave radiation (w m-2)",
             title= "Modeled vs Observed outgoing shortwave
radiation", subtitle = "weekly average 2018") +
        legend_labels(labels = c("Modelled", "Observed"), "Outgoing
shortwave radiation")
sw_lm <- lm(obs_sw_out ~ mod_sw_out, data = out_plot)
sw_r2 <- round(summary(sw_lm)$r.squared, 2)
sw_slope <- round(coef(sw_lm)[2], 2)
sw_intercept <- round(coef(sw_lm)[1], 2)
sw_rmse <- round(sqrt(mean((out_plot$obs_sw_out -
out_plot$mod_sw_out)^2)), 2)
sw_nse <- round(NSE(out_plot$mod_sw_out, out_plot$obs_sw_out), 2) #
Nash-Sutcliffe Efficiency
ggplot(out_plot) +
  geom_abline(aes(intercept = 0, slope = 1, color = "Theory"),
key_glyph = "path") +
  geom_point(aes(x = mod_sw_out, y = obs_sw_out), size=.7) +
  geom_smooth(aes(x = mod_sw_out, y = obs_sw_out, color =
"Regression"), formula = y ~ x, method = 'lm', se = F) +
  labs(title = "Outgoing sw modeled vs observed",
       subtitle = paste(c("slope: ", sw_slope, " intercept: ",
sw_intercept, " r2: ", sw_r2, " . Data from 2018."), collapse = ""),
       x = "Modelled outgoing shortwave (W m-2)", y = "Observed
outgoing sw (W m-2)", colour = "Legend")
out_1week %>%
        gather(key = "type", value = "radiation", l_up, lw_out,
factor_key = T) %>%
        ggplot() +
        night_bg(out_1week, y_mean = 400) +
        geom_line(aes(x = datetime, y = radiation, color = type)) +
        labs(y = "Outgoing shortwave radiation (w m-2)",
             title= "Modeled vs Observed outgoing longwave radiation",
```

```
subtitle = "1 summer week (1-8 Jul 2018)") +
        legend_labels(labels = c("Modelled", "Observed"), "Outgoing
longwave radiation")
out_1year_wk %>%
        gather(key = "type", value = "radiation", l_up, lw_out,
factor_key = T) %>%
        ggplot() +
        geom_line(aes(x = datetime, y = radiation, color = type)) +
        labs(y = "Outgoing shortwave radiation (w m-2)",
             title= "Modeled vs Observed outgoing longwave radiation",
subtitle = "weekly average 2018") +
        legend_labels(labels = c("Modelled", "Observed"), "Outgoing
shortwave radiation")
lw_lm <- lm(obs_lw_out ~ mod_lw_out, data = out_plot)
lw_r2 <- round(summary(lw_lm)$r.squared, 2)
lw_slope <- round(coef(lw_lm)[2], 2)
lw_intercept <- round(coef(lw_lm)[1], 2)

lw_rmse <- round(sqrt(mean((out_plot$obs_lw_out -
out_plot$mod_lw_out)^2)), 2)
lw_nse <- round(NSE(out_plot$mod_lw_out, out_plot$obs_lw_out), 2)#
Nash-Sutcliffe Efficiency
ggplot(out_plot) +
  geom_abline(aes(intercept = 0, slope = 1, color = "Theory"),
key_glyph = "path") +
  geom_point(aes(x = mod_lw_out, y = obs_lw_out), size=.7) +
  geom_smooth(aes(x = mod_lw_out, y = obs_lw_out, color =
"Regression"), formula = y ~ x, method = 'lm', se = F) +
  labs(title = "Outgoing lw modeled vs observed",
       subtitle = paste(c("slope: ", lw_slope, " intercept: ",
lw_intercept, " r2: ", lw_r2, " . Data from 2018."), collapse = ""),
       x = "Modeled outgoing longwave (W m-2)", y = "Observed outgoing
lw (W m-2)", colour = "Legend")
zenith <- seq(0,85)
Kb <- map_dbl(zenith, get_Kb)

ggplot() +
  geom_line(aes(x = zenith , y = Kb)) +
  labs(title = "Kb over zenith", x="Zenith (degrees)")+
  scale_y_continuous(n.breaks = 7, limits = c(0,NA))
LAI <- seq(1,10,.25)
Kd <- map_dbl(LAI, get_Kd)

ggplot() +
  geom_line(aes(x = LAI, y = Kd)) +
  labs(title = "Kd over LAI")
fake_input_5 <- tibble(
  datetime = as.Date("2020-07-15 12:00"), sw_in = 1000, sw_dif = 100,
lw_in = 0, t_soil = 0, t_leaf = 0, zenith = 30,
  LAI = 1:12
```

```
)

out_2 <- cbind(rad_transf(input = fake_input_5), sw_in =
fake_input_5$sw_in)
labels <- str_wrap(c("Total incoming sw", "Total absorbed sw",
"Absorbed sw sunlit canopy", "Absorbed sw shaded canopy"), 8)

out_2 %>%
  gather("type", "radiation", sw_in, ic, ic_sun, ic_sha, factor_key =
T) %>%
  ggplot() +
  geom_line(aes(x = LAI, y = radiation, color = type, linetype =
type)) +

  labs(title = "Absorbed shortwave with increasing LAI", subtitle =
"Incoming radiation 900 direct + 100 diffuse. Zenith 30°",
       x = "LAI (m2 m-2)", y = "Absorbed shortwave radiation (W m-2)",
color = "", linetype = "") +
  scale_color_manual(values = c("black", hue_pal()(3)), labels =
labels) +
  scale_linetype_manual(values = c(2, 1, 4, 4), labels = labels) +
  theme(legend.key.height = unit(44, "pt"))

ggplot(out_2) +
geom_line(aes(x=LAI, y=ic_sha/ic))+
labs(title="Fraction of radiation absorbed by shaded leaves over LAI",
y="Fraction abosorbed by shaded leaves")
labels <- str_wrap(c("Total incoming sw", "Absorbed sw canopy",
"Absorbed sw soil", "Upcoming sw"), 8)
out_2 %>%
  gather("Radiation_type", "Radiation", sw_in, ic, ig, i_up,
factor_key = T) %>%
  ggplot() +
  geom_line(aes(x = LAI, y = Radiation, color = Radiation_type,
linetype = Radiation_type)) +
  scale_color_manual("", values = c("black", hue_pal()(3)), labels =
labels) +
  scale_linetype_manual("", values = c(2, 1, 1, 1), labels = labels) +
  theme(legend.key.height = unit(40, "pt")) +
  labs(title = "Absorbed sw canopy, soil and upcoming sw with
increasing LAI", subtitle = "Incoming radiation 900 direct + 100
diffuse. Zenith 30°",
       x = "LAI (m2 m-2)", y = "Shortwave radiation (W m-2)")
fake_input_3 <- tibble(
  datetime = as.Date("2020-07-15 12:00"), # Summer day
  sw_in = 800,
  sw_dif = 100,
  lw_in = 200,
  t_soil = 260:310,
  t_leaf = 260:310,
```

```r
  zenith = 30,
)

out_3 <- rad_transf(fake_input_3, fake_input_3) # the second time is
for the state

out_3 %>%
  select(-zenith) %>%
  cbind(fake_input_3) %>%
  ggplot() +
  geom_line(aes(x = t_leaf, y = l_up), color = "red") +
  labs(title = "Emitted longwave with increasing temperatures",
   x="Leaf and Soil temperature (K)", y="Emitted radiation (W m-2)",
color="")
fake_input_4 <- tibble(
  datetime = as.POSIXct("2016-07-21 00:00"), # Summer day
  zenith = seq(0, 90, 2.5),
  sw_in = 1000,
  sw_dif = 100,
  lw_in = 200,
  t_soil = 300,
  t_leaf = 300,
)
out_4 <- rad_transf(fake_input_4)

out_4 %>%
  select(-zenith) %>%
  cbind(fake_input_4) %>%
  mutate(absorbed_fraction = ic / sw_in) %>%
  ggplot() +
  geom_line(aes(x = zenith, y = absorbed_fraction), color = "red") +
  labs(x = "Zenith (deg)", y = "Fraction of total radiation absorbed",
title="Fraction of absorbed radiation with incresing zenith")
```

References

Bonan, Gordon. 2019. *Climate Change and Terrestrial Ecosystem Modeling*. Cambridge University Press. https://doi.org/10.1017/9781107339217.

Soetaert, Karline, and Thomas Petzoldt. 2010. "Inverse Modelling, Sensitivity and Monte Carlo Analysis in R Using Package FME." *Journal of Statistical Software* 33 (3): 1–28.