SCOPE Documentation

Release 1.8

Egor

Nov 05, 2020

CONTENTS

Soil Canopy Observation, Photochemistry and Energy fluxes (SCOPE, Van der Tol at al. 2009 [11]) radiative transfer model.

CHAPTER

ONE

MODEL CAPABILITIES

1.1 Spectra

For thermal part radiance enable `options.calc_planck` & `options.calc_ebal`.

For soil reflectance simulation with BSM model enable `options.soilspectrum`

1.1.1 Definition

Optical part of spectrum takes into account the ability of objects (leaves, soil) to **reflect** the light.

Thermal part of spectrum is based on Planck's law that is any object that has temperature above 0 K **emits** electromagnetic waves.

SCOPE model uses the theory of radiative transfer describing electromagnetic waves propagation and takes into account absorption and scattering.

There are 3 components in the model:

1. soil
2. leaf
3. canopy

1.1.2 SCOPE soil

SCOPE simulates soil reflectance (optical domain) with *`BSM()`* (if `options.soilspectrum`). Alternatively, a file with soil reflectance spectrum might be provided (several options are in ./data/input/soil_spectrum/soilnew.txt*).

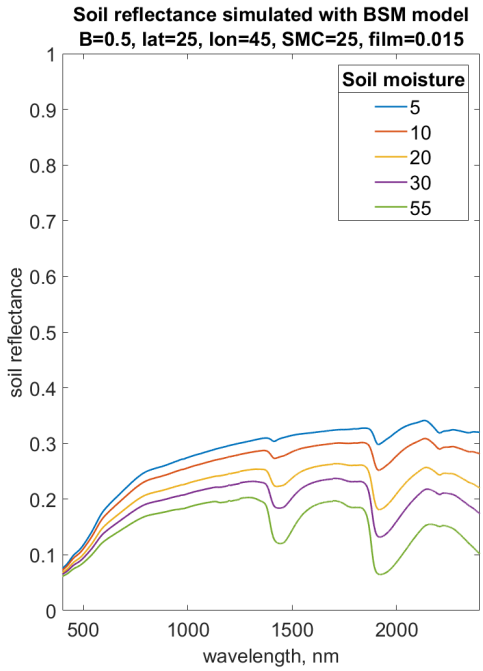
On the graph you can see the difference.

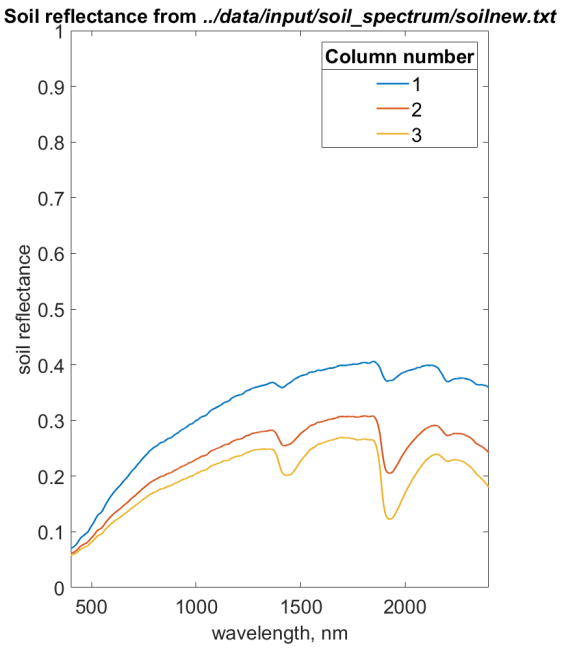
1.1.3 SCOPE leaf

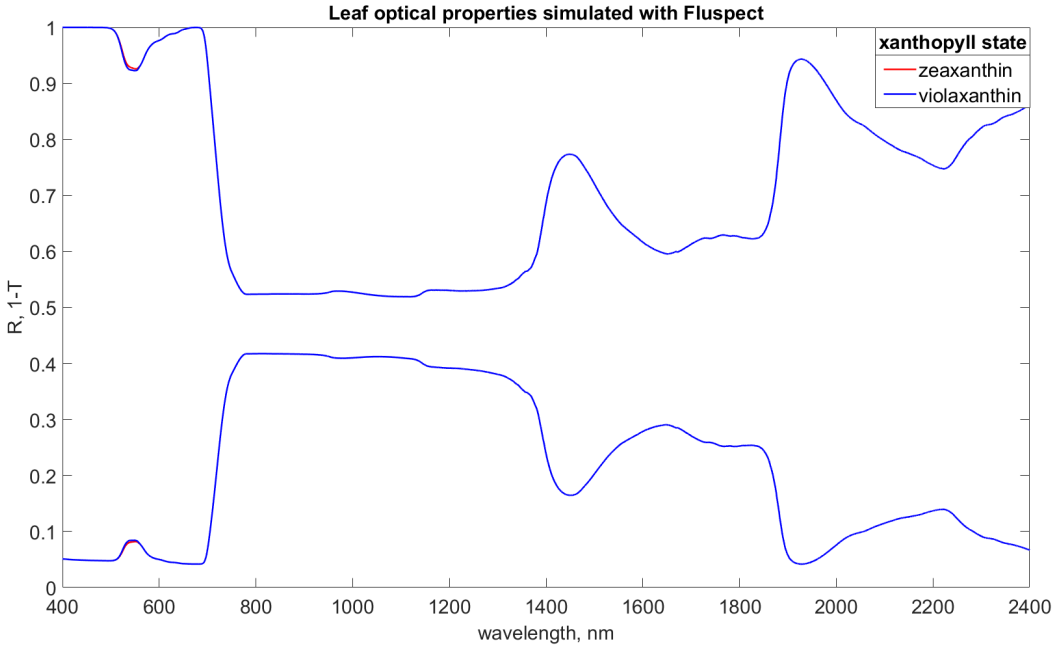
SCOPE model (in particular *`fluspect_B_CX_PSI_PSII_combined()`*) works similar to PROSPECT model and simulated leaf reflectance, transmittance and absorption (the rest). Bonus part is fluorescence.

The other bonus is xanthopyll cycle, available with `option.calc_xanthophyllabs`, that you might see as a tiny pick around 500 nm.

Note: We do not have a model for thermal emission of a single leaf. Any suggestions?


Soil reflectance simulated with BSM model
B=0.5, lat=25, lon=45, SMC=25, film=0.015
Soil moisture
5
10
20
30
55
soil reflectance
wavelength, nm


Soil reflectance from ../data/input/soil_spectrum/soilnew.txt
Column number
1
2
3
soil reflectance
wavelength, nm

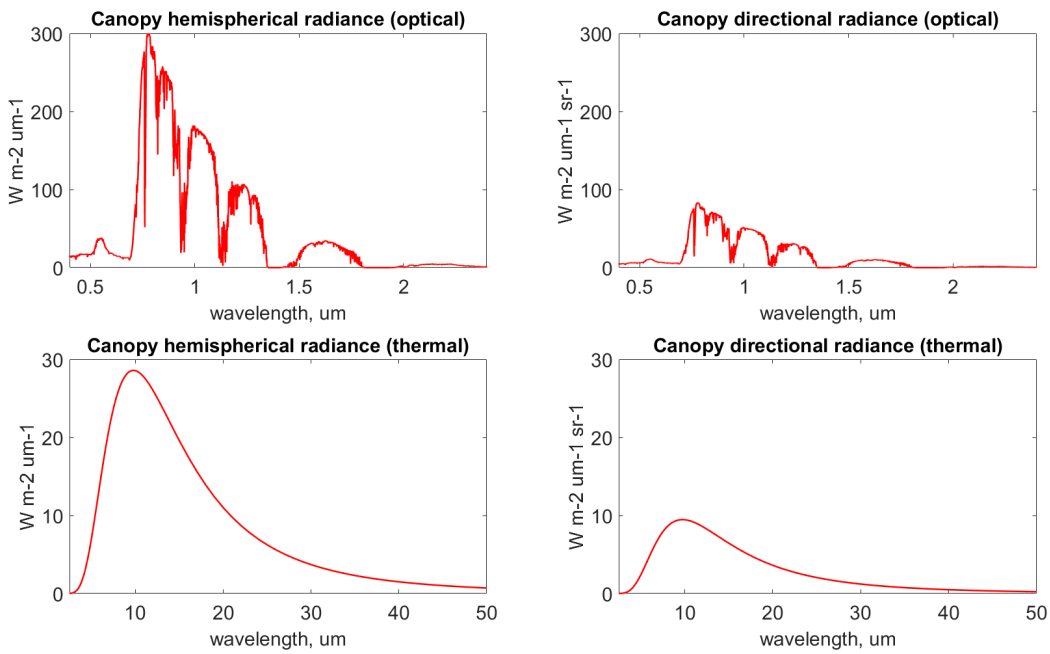

Leaf optical properties simulated with Fluspect
xanthopyll state
zeaxanthin
violaxanthin
R, 1-T
wavelength, nm

1.1.4 SCOPE canopy

SCOPE represents canopy as 60 elementary layers of leaves of two types: sunlit (then we account for leaf inclinations) and shaded.

Leaf and soil optical properties are taken into account here.

Optical properties are calculated by `RTMo()` and thermal properties by either `RTMt_planck()` then the output is radiance (as on the graph) or `RTMt_sb()` - only integrated fluxes.



Note: default configuration uses Stefan-Boltzmann's law (`RTMt_sb()`) that will output **spectrally integrated** thermal fluxes instead of per wavelength, in this case:

- integral of hemispherical thermal radiance is 432.5 W m-2
- integral of direct radiance 140.4 W m-2

1.2 Fluorescence

`options.calc_fluor`

1.2.1 Definition

Light reaching a leaf (pretty much any object) can be reflected, transmitted or absorbed. In plants absorbed light can be spent on three different processes:

1. photochemistry (assimilation CO2)
2. non-photochemical quenching (NPQ): heat dissipation
3. chlorophyll fluorescence excitation

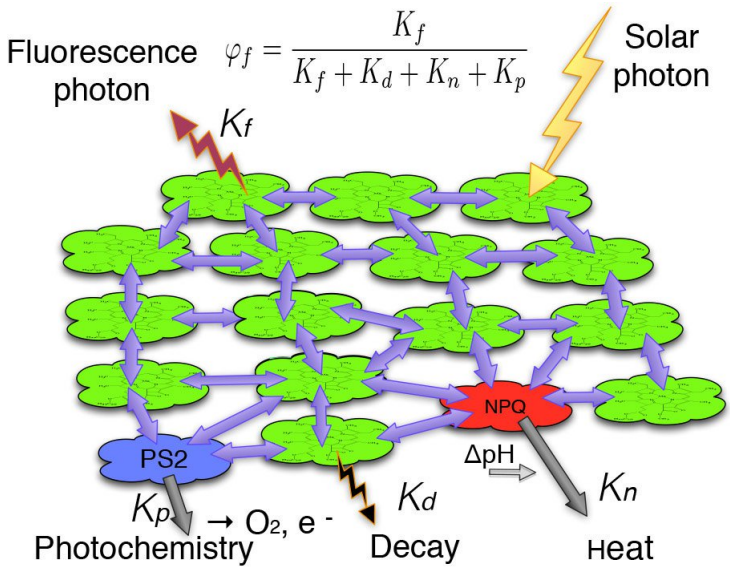


Fig. 1: Source: http://spie.org/newsroom/4725-remote-sensing-of-terrestrial-chlorophyll-fluorescence-from-space?SSO=1

```
>>> Fluorescence is light emitted by chlorophyll molecules in the range 640-800 nm.
```

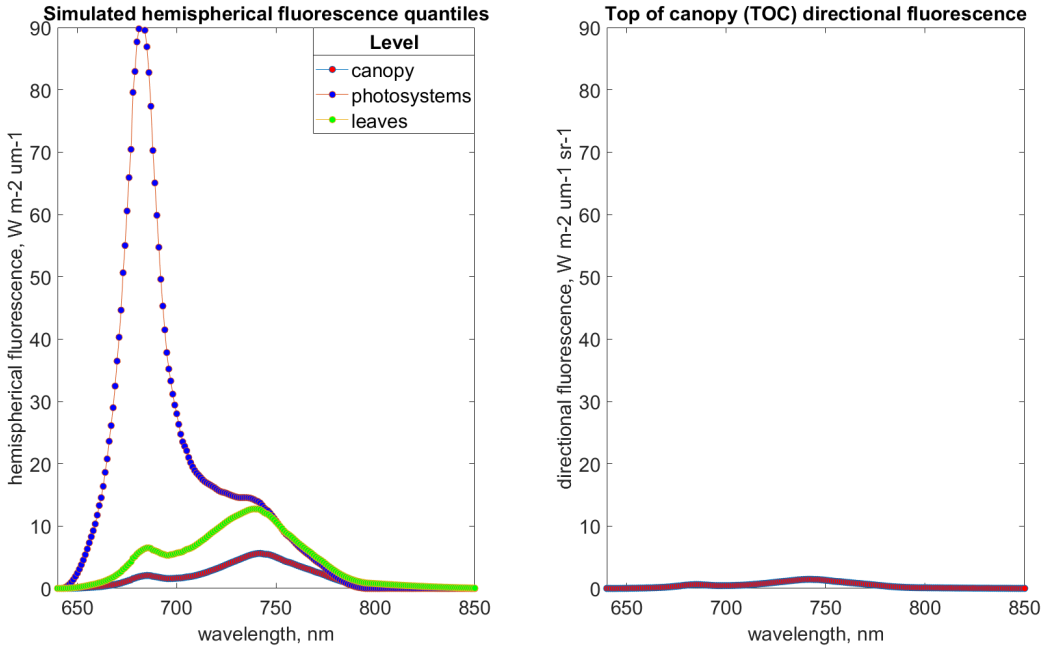
1.2.2 SCOPE

Fluorescence light (pretty much as any light) can be absorbed and scattered on its way to a sensor.

SCOPE model simulates 3 hemispherical fluorescence quantiles:

1. fluorescence emitted by all photosystems without any scattering / re-absorption neither within leaf, nor withing canopy
2. fluorescence emitted by all leaves without any scattering / re-absorption within canopy or from soil
3. fluorescence emitted by canopy (all leaf layers) accounting for all scattering / re-absorption events

Note: Notice the difference in ranges and units between directional and hemispherical fluorescence.



SCOPE model simulates directional fluorescence (the one that actually reaches a sensor) and its components coming from:

1. sunlit leaves
2. shaded leaves
3. scattered by leaves and soil

It is also possible to partition directional fruorescence between photosystem I and II (PSI, PSII) with `options.calc_PSI`, not recommended though.

Note: There are much more outputs of *`biochemical()`* related to Pulse-Amplitude-Modulation (PAM) Fluorometry quantiles that are stored in internal structure `biochem_out`.

1.3 Energy balance

`options.calc_ebal`

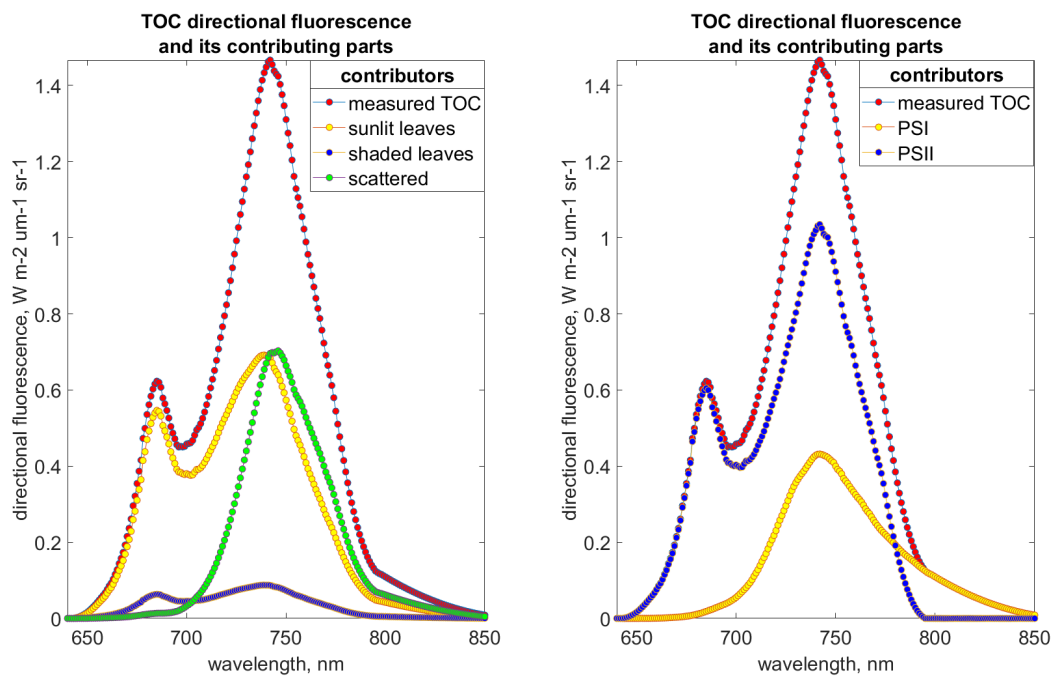
1.3.1 Definition

Net radiation (Rn): .. http://www.indiana.edu/~geog109/topics/04_radiation/4c-RadiationBalance_nf.pdf

```
>>> Rn = (SW_in - SW_out) + (LW_in - LW_out)
```

SW - shortwave radiation (400-2400 nm)
LW - longwave (thermal) radiation

Net radiation can be partitioned into 3 (4) heat fluxes:




```
>>> Rn = H + lE + G
```

H - sensible heat
lE - latent heat
G - ground heat flux

1.3.2 SCOPE

With `options.calc_ebal` energy balance loop is started until the energy balance is closed (net radiation become equal to heat fluxes).

To close energy balance leaf temperatures and Monin-Obukhov length `L` are iteratively adjusted.

This how it looks like:

Leaf temperatures are calculated by *`biochemical()`* or *`biochemical_MD12()`*.

Initial values of soil and leaf temperatures are equal to ambient temperature (`Ta`).

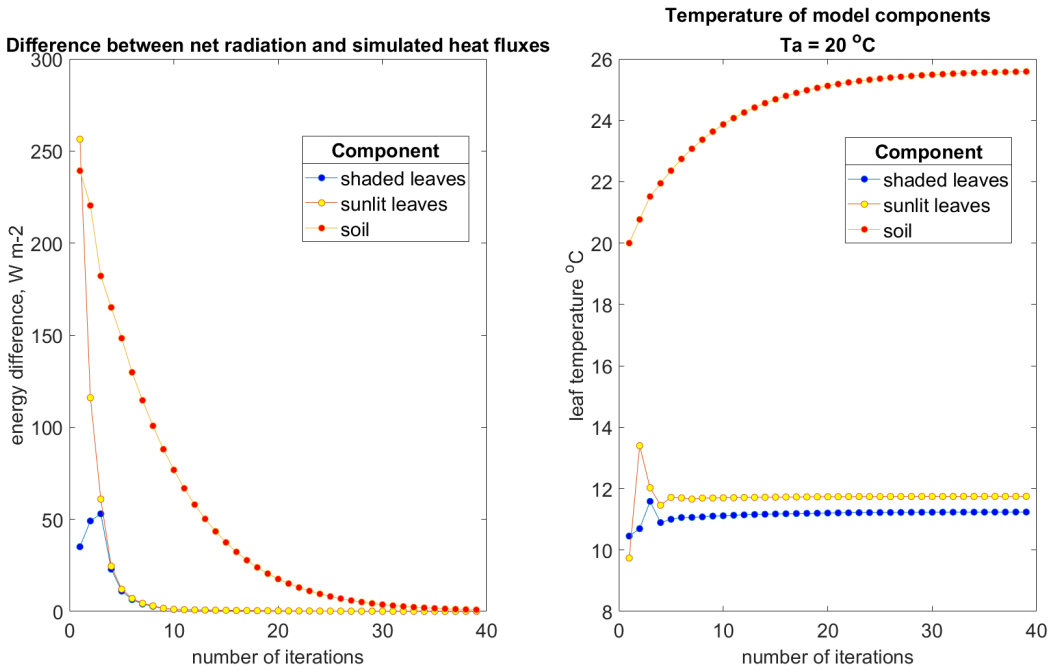
Monin-Obukhov length influences on aerodynamic resistances values.

The results of energy balance calculations are the following:

variable	units	canopy	soil	total
H	W m-2	115.5	19.5	135.0
lE	W m-2	146.2	44.9	191.1
G	W m-2	-	35.0	35.0
Rn	W m-2	**261.7**	**100.1**	**361.8**

Also SCOPE calculates photosynthesis of canopy which in this case was 18.7 umol m-2 s-1

Radiation budget is also calculated by SCOPE:



variable	units	in	out	net
SW	W m-2	600	104.4	495.6
LW	W m-2	300	434.0	-134.0
Rn	W m-2	-	-	**361.8**

1.4 Vertical profiles

`options.calc_vert_profiles` & `options.calc_ebal`

1.4.1 Definition

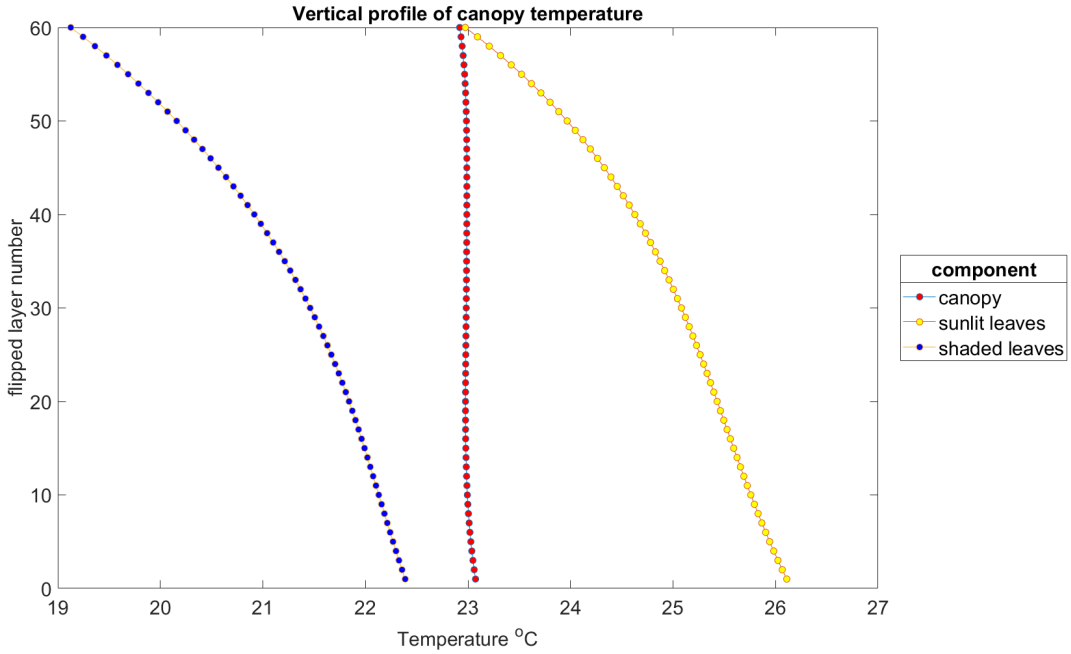
We assume, it's not needed.

1.4.2 SCOPE

SCOPE represents canopy as 60 elementary layers of leaves of two types: sunlit (then we account for leaf inclinations) and shaded.

Components of energy balance, temperature and fluorescence can be calculated for each layer with `options.calc_vert_profiles`

> **Warning:** To produce this graph profiles were flipped so that soil is layer #0 and first canopy layer is layer #1.
>
> This way top of canopy is actually at the top of the graph (layer #60).



1.5 BRDF

`options.calc_directional` & `options.calc_ebal`

Warning: This is an advanced topic, please refer to Schaepman-Strub et al. 2006 [4] for further explanations.

1.5.1 Definition

Light consists of two components **direct** (aka specular) and **diffuse** (aka hemispherical).

To explain reflectance of each light component individually, different reflectance factors are used.

SCOPE model simulates the following reflectance factors:

- **Incoming light is directional**
  - CASE 1: bidirectional (BRF)
  - CASE 2: directional-hemispherical (DHRF)
- **Incoming light is hemispherical**
  - CASE 7: hemispherical-directional (HDRF)
  - CASE 9: bihemispherical (HRF)

After reflectance from a material *direct component* of incoming light contributes to both directional and hemispherical component of reflected light.

After reflectance from a material *diffuse component* of incoming light also contributes to both directional and hemispherical component of reflected light.

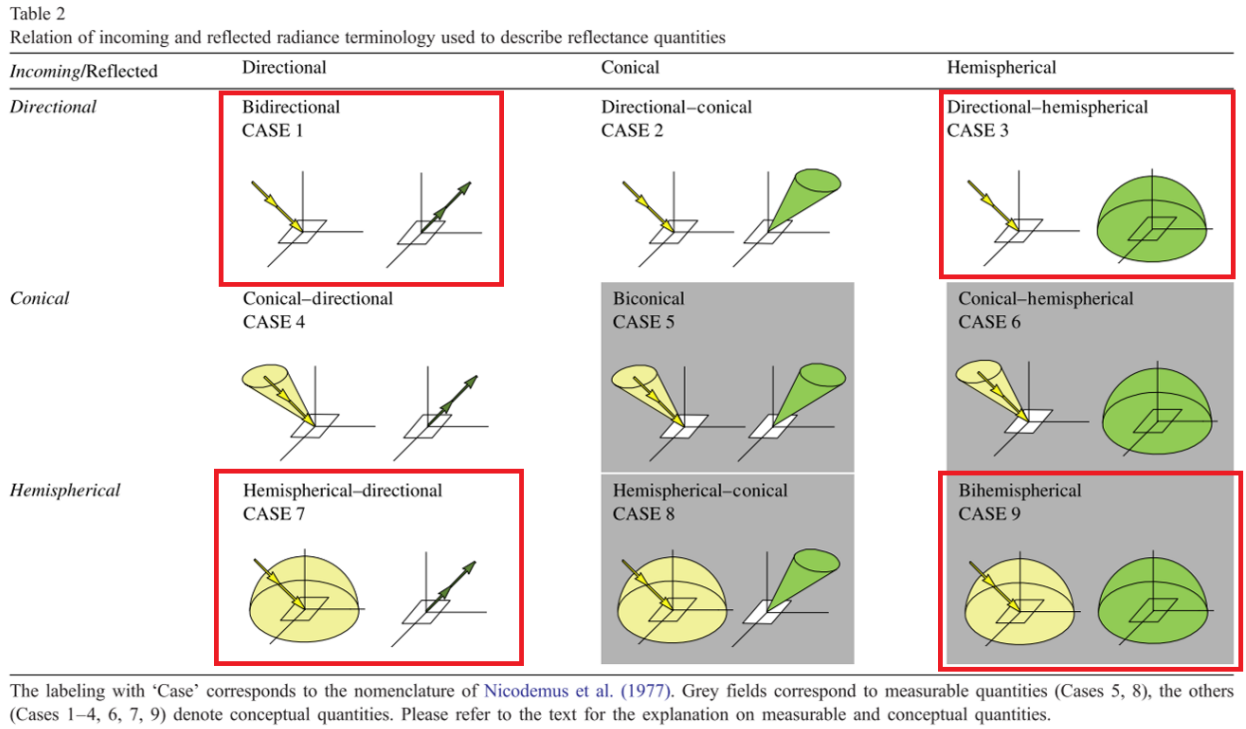


Table 2
Relation of incoming and reflected radiance terminology used to describe reflectance quantities

Incoming/Reflected	Directional	Conical	Hemispherical
Directional	Bidirectional CASE 1	Directional–conical CASE 2	Directional–hemispherical CASE 3
Conical	Conical–directional CASE 4	Biconical CASE 5	Conical–hemispherical CASE 6
Hemispherical	Hemispherical–directional CASE 7	Hemispherical–conical CASE 8	Bihemispherical CASE 9

The labeling with 'Case' corresponds to the nomenclature of Nicodemus et al. (1977). Grey fields correspond to measurable quantities (Cases 5, 8), the others (Cases 1–4, 6, 7, 9) denote conceptual quantities. Please refer to the text for the explanation on measurable and conceptual quantities.

Fig. 2: From Schaepman-Strub et al. 2006 [4].

Note: **Bidirectional Reflectance Distribution Function** is a function describing bidirectional reflectance from a material.

- "input" of BRDF are four angles (solar zenith and azimuth angle (direction of incoming light); viewing zenith and azimuth angle (direction of observation))
- "output" of BRDF is reflectance (BRF)

1.5.2 SCOPE

To simulate BRDF enable `options.calc_directional`.

SCOPE calculates BRDF itself and also directional fluorescence radiance and directional thermal radiance (or brightness temperature).

Directional plots have 3 components:

- viewing zenith angle (towards the centre of the circle)
- viewing azimuth angle (around the circle)
- measured quantity (color)

On all graphs you can see a **hot spot** (red dot) where viewing azimuth angle is 0º and *viewing* zenith angle is 30º.

Hot spot occurs when the observation direction coincides with the illumination direction. Indeed, for this example *solar* zenith angle of 30º was used.

Directional plots are made per wavelength.

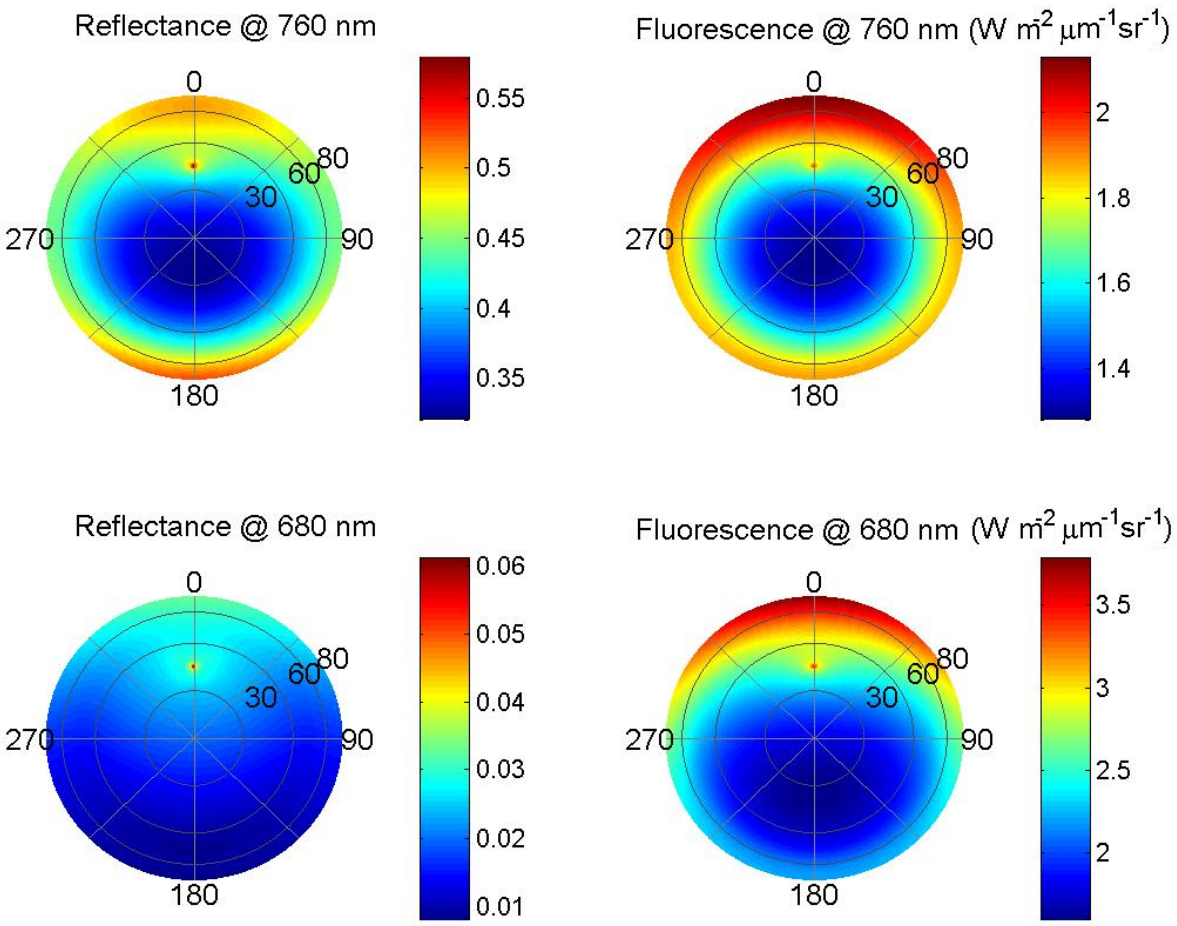
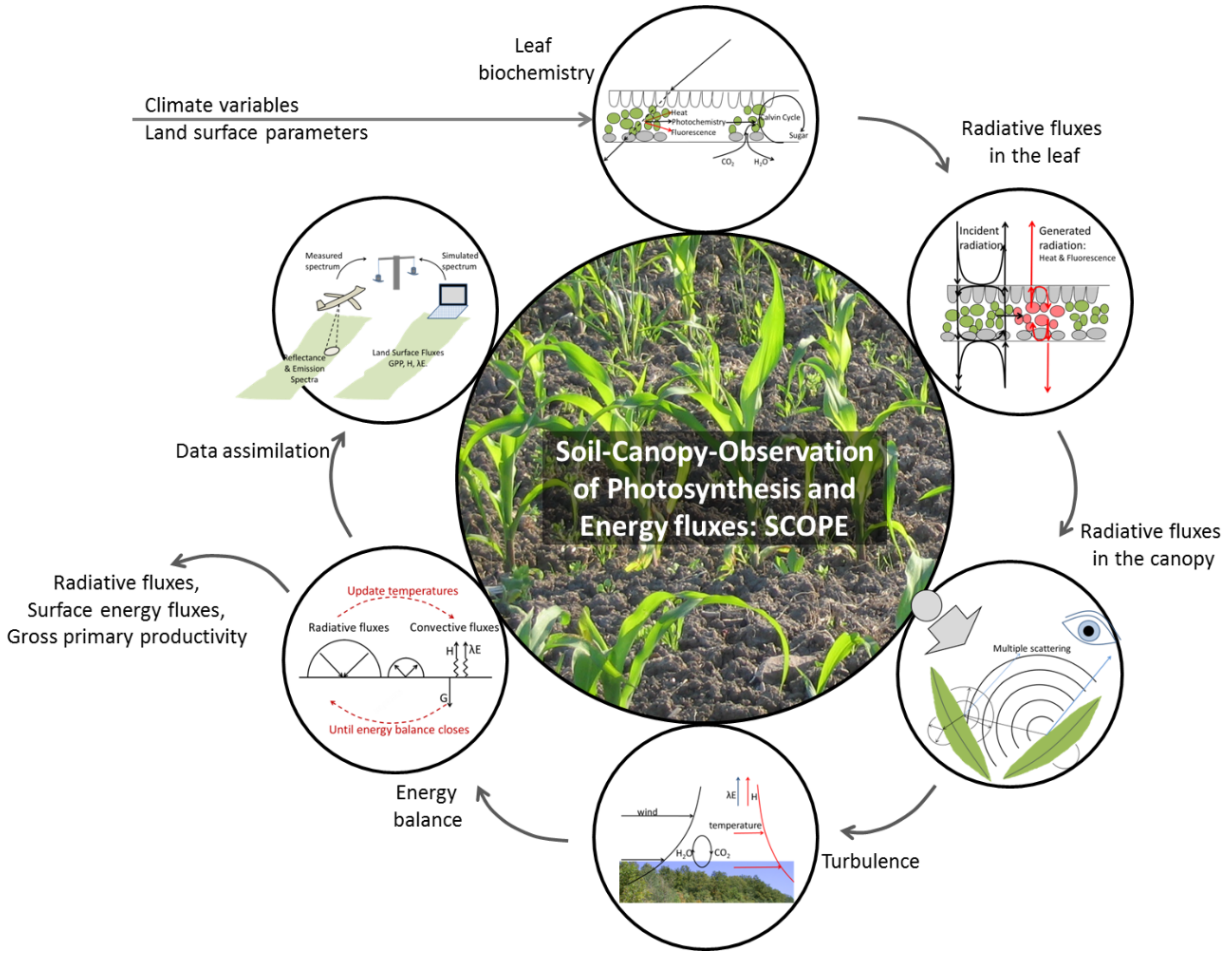


Fig. 3: Courtesy of Peiqi Yang


Leaf
biochemistry
Climate variables
Land surface parameters
Radiative fluxes
in the leaf
Incident
radiation
Generated
radiation:
Heat & Fluorescence
Measured
spectrum
Simulated
spectrum
Reflectance
& Emission
Spectra
Land Surface Fluxes
GPP, H, λE
Data assimilation
Soil-Canopy-Observation
of Photosynthesis and
Energy fluxes: SCOPE
Radiative fluxes
in the canopy
Multiple scattering
Radiative fluxes,
Surface energy fluxes,
Gross primary productivity
Update temperatures
Radiative fluxes
Convective fluxes
Until energy balance closes
Energy
balance
wind
temperature
Turbulence

CHAPTER

TWO

GETTING STARTED

Contents

2.1 0. Software requirements

The model SCOPE_v1.73 is written in Matlab R2015b running on a Windows operating system. We took care not to use functions that are available in all recent Matlab versions, but we cannot give any warranty that it works under other operating systems and other Matlab versions.

> **Warning:** If you do **not** have Matlab on your computer you can use `SCOPE.exe` with Matlab Runtime **only R2017b (version 9.3)**
>
> Compiled version `SCOPE.exe` can be run only with the Excel file input (`input_data.xlsx`).

SCOPE consists of several scripts and functions (modules), which can be used separately or as parts of the integrated SCOPE model (`SCOPE.m`).

When the modules are used separately, then it is important to provide input in the structures specified in *Structs*.

When the integrated model is called, then the input is automatically loaded from the spreadsheet *input_data.xlsx* and from the files specified in ./*data*/input.

Basic knowledge of the use of Matlab is required to operate the model.

The application of the model involves the following steps:

2.2 1. Unpack the zip file

Unpack the model, and **leave the directory structure intact**.

2.3 2. Run the model once

Run the model once, before modifying the parameters and input. It will check whether the software works under your system. The model runs with an example data set (`options.verify`), and the output is automatically compared to output that it should produce. If there is any difference in the results, messages will show up.

- **Navigate to the directory where the matlab code is** ./SCOPE_v1.73/*src*
- Open `SCOPE.m` in Matlab
- **in Matlab command window type:**

```
SCOPE
```

Running the model may take a while because almost all options are switched on. If the output of the model is not as expected, then messages will appear. There will also be graphs appearing showing the freshly produced output together with the expected output. If all is ok then no graphs or warnings are produced.

2.4 3. Set the input in `input_data.xlsx`

Main input file i``input_data.xlsx`` with 4 sheets is located in ./SCOPE_v1.73. In the documentation we refer to this file, although text alternatives are also possible.

Note: If Excel is not available, it is possible to use input from text files (.m and .txt). See **alternative**.

To specify which input to use (text or excel) comment / uncomment lines in `set_parameter_filenames.m` with ``% sign.

Warning: Compiled version `SCOPE.exe` can be run only with the Excel file input (`input_data.xlsx`).

sheet (tab)	content	alternative
readme	sheets description of `input_data.xlsx` explanation of leaf inclination distribution function (LIDF) parameters recommended values for plan functional types (PFTs) some parameter ranges	-
options	*Options*	`setoptions.m`
filenames	filenames for current simulation and for time-series	`filenames.m`
inputdata	values for *input structs*	`inputdata.txt`

To find out ranges and units of input parameters take a look into *input structs*.

Pay extra attention to the *simulation*

2.5 4. Analyse the output

All output files and their content (variables, units) are available at *Output files*.

Some output files are available for each run, the others can be written with various *Options*.

To plot the output either select `options.makeplots` or use function from *`plots()`*

Note: Radiation, spectral and fluorescence output usually has two quantiles:

- outgoing diffuse light (**hemispherical**) W m-2 um-1
- outgoing light in observation directions (**directional**, the one that actually reaches the sensor) W m-2 um-1 sr-1

To get further information see: *Definition*

2.6 5. Going further

`SCOPE.m` is a script, thus after a run all matlab structures that were generated during the run (input, output, constants) are available in the workspace. You can get some extra variables that are not written to output files. You can find out available variables at *Structs*.

All functions are documented within the code and also at *API*.

For any questions, please, use SCOPE_model SCOPE_model group.

CHAPTER

THREE

OPTIONS

Effectively this are (almost[1][2]) all capabilities of the SCOPE model.

- *Initialized*
- *Rules of input reading*
  - `simulation`
- *Variations in input*
  - `rt_thermal`
  - `calc_zo`
  - `soilspectrum`
  - `soil_heat_method`
  - `calc_rss_rbs`
- *Variations in output*
  - `calc_ebal`
  - `calc_planck`
  - `calc_directional`
  - `calc_xanthophyllabs`
  - `calc_vert_profiles`
  - `calc_fluor`
  - `calc_PSI`
  - `Fluorescence_model`
  - `apply_T_corr`
- *For users' comfort*
  - `verify`
  - `save_headers`
  - `makeplots`

[1] extra output variables that are not saved to files (see *Structs*) are available in the workspace after the model run.
[2] model can be varied by user, please, consult *API* to learn signatures of functions

This is an input structure that controls the workflow.

The values have binary (or tertiary) logic thus equal to 0 or 1 (or 2).

Influence on the output files is highlighted in the corresponding section *Output files*

Note: Not all combinations can bring to the desired result

3.1 Initialized

`SCOPE.m`: read from `input_data.xlsx` or `setoptions.m`

3.2 Rules of input reading

3.2.1 `simulation`

Defines rules of input reading

Switch in `SCOPE.m` (multiple)

0 individual run(s): specify one value for fixed input parameters, and an equal number (> 1) of values for all parameters that vary between the runs.

1

time series (uses text files with meteo input as time series from *"../data/input/dataset X"* with files similar to ../data/input/*dataset for_verification* specified on the `filenames` sheet of `input_data.xslx` *`load_timeseries()`*

2 Lookup-Table: specify a number of values in the row of input parameters. All possible combinations of inputs will be used.

Let us illustrate what the difference is in details.

It is possible to specify several values in a row on `inputdata` sheet of `input_data.xslx`. Suppose we have an the following combination of input parameters. Notice, we provide two values for Cab and Cca parameters.

If **individual run(s)** (`options.simulation == 0`) was chosen the given combination will end up in **two** simulations:

- Cab=80, Cca=20
- Cab=40, Cca=10

If **Lookup-Table** (`options.simulation == 2`) was chosen the given combination will end up in **four** simulations:

- Cab=80, Cca=20
- Cab=80, Cca=10
- Cab=40, Cca=20
- Cab=40, Cca=10

3.3 Variations in input

3.3.1 `rt_thermal`

Leaf and soil emissivity in thermal range

Switch in `SCOPE.m`

0

provide emissivity values as input *leafbio* (rho_thermal, tau_thermal), *soil*.rs_thermal

1 use values from fluspect and soil at 2400 nm for the TIR range

---

3.3.2 `calc_zo`

roughness length for momentum of the canopy (zo) and displacement height (d)

Switch in *`select_input()` `load_timeseries()`*

0

zo and d values provided in the inputdata *canopy*

1 calculate zo and d from the LAI, canopy height, CD1, CR, CSSOIL (recommended if LAI changes in time series) `zo_and_d()`

3.3.3 `soilspectrum`

Calculate soil reflectance or use from a file in ../data/input/*soil_spectrum*

Switch in `SCOPE.m`

0

use soil spectrum from the file with *soil*.spectrum
default file is `soilnew.txt`, can be changed on the `filenames` sheet `soil_file` cell
variable name is `rsfile`

1 simulate soil spectrum with the BSM model (`BSM()`) parameters are fixed in code

---

3.3.4 `soil_heat_method`

Method of ground heat flux (G) calculation

Switch in `SCOPE.m`, *`select_input()`*, *`ebal()`*

0

standard calculation of thermal inertia from soil characteristic
`Soil_Inertia0()` in *`select_input()`*

1

empirically calibrated formula from soil moisture content `Soil_Inertia1()` in `select_input()`

2

as constant fraction (0.35) of soil net radiation
`Soil_Inertia0()` in `select_input()`

3.3.5 calc_rss_rbs

soil resistance for evaporation from the pore space (rss) and soil boundary layer resistance (rbs)

Switch in `select_input()`

0

use resistance rss and rbs as provided in inputdata *soil*

1 calculate rss from soil moisture content and correct rbs for LAI `calc_rssrbs()`

3.4 Variations in output

`RTMo()` (SAIL) is executed in any valid run. Other functions may be included with these options.

3.4.1 calc_ebal

Switch in `SCOPE.m`

0

Only `RTMo()` is run (with `RTMf()` if `options.calc_fluor`)

1

Calculate the complete energy balance.

> **Warning:** required for `calc_planck`, `calc_directional`, `calc_xanthophyllabs`

3.4.2 calc_planck

Calculate spectrum of thermal radiation with spectral emissivity instead of broadband

> **Warning:** only effective with `calc_ebal == 1`

Switch in `SCOPE.m`, `calc_brdf()`

0

RTMt_sb() - broadband brightness temperature is calculated in accordance to Stefan-Boltzman's equation.

1

RTMt_planck() is launched in `SCOPE.m` and *calc_brdf()* (if `calc_directional`). Calculation is done per each wavelength thus takes more time than Stefan-Boltzman.

3.4.3 calc_directional

Calculate BRDF and directional temperature for many angles specified in the file: *directional*.

> **Warning:**
> - only effective with `calc_ebal == 1`
> - Be patient, this takes some time

Switch in `SCOPE.m`, *calc_brdf()*

0

-

1

struct *directional* is loaded from the file *directional*
calc_brdf() is launched in `SCOPE.m`

3.4.4 calc_xanthophyllabs

Calculate dynamic xanthopyll absorption (zeaxanthin) for simulating PRI (photochemical reflectance index)

> **Warning:**
> - only effective with `calc_ebal == 1`

Switch in `SCOPE.m`

0

-

1 *RTMz()* is launched in `SCOPE.m` and *calc_brdf()* (if `calc_directional`)

3.4.5 calc_vert_profiles

Calculation of vertical profiles (per 60 canopy layers).

Corresponding structure *profiles*

Switch in `SCOPE.m`, *`RTMo()`* and *`ebal()`*

0

Profiles are not calculated

1

Photosynthetically active radiation (PAR) per layer is calculated in *`RTMo()`*
Energy, temperature and photosynthesis fluxes per layer are calculated in *`ebal()`*
Fluorescence fluxes are calculated in *`RTMf()`* if (`calc_fluor`)

3.4.6 calc_fluor

Calculation of fluorescence

Switch in `SCOPE.m`, *`calc_brdf()`*

0

No fluorescence

1

`RTMf()` is launched in `SCOPE.m` and *`calc_brdf()`* (if `calc_directional`)
total emitted fluorescence is calculated by `SCOPE.m`

3.4.7 calc_PSI

Separate fluorescence of photosystems I and II (PSI, PSII) or not

Switch in `SCOPE.m`, *`select_input()`*

0

recommended
treat the whole fluorescence spectrum as one spectrum (new calibrated optipar)
fluspect version *`fluspect_B_CX_PSI_PSII_combined()`*

1

differentiate PSI and PSII with Franck et al. spectra (of SCOPE 1.62 and older)
fluspect version *`fluspect_B_CX()`*
fluorescence quantum efficiency of PSI is set to 0.2 of PSII in *`select_input()`*

3.4.8 `Fluorescence_model`

Fluorescence model

Switch in *`ebal()`*

0

empirical, with sustained NPQ (fit to Flexas' data)

1 empirical, with sigmoid for Kn: *`biochemical()`* (Berry-Van der Tol)

2 *`biochemical_MD12()`* (von Caemmerer-Magnani)

3.4.9 `apply_T_corr`

correct Vcmax and rate constants for temperature

> **Warning:** only effective with `Fluorescence_model != 2` i.e. for *`biochemical()`*

Switch in *`ebal()`*

0

-

1 correction in accordance to Q10 rule

3.5 For users' comfort

3.5.1 `verify`

verify the results (compare to saved 'standard' output) to test the code for the first time

Switch in `SCOPE.m`

0

-

1 runs *`output_verification()`*

3.5.2 save_headers

write header lines in output files

Switch in *create_output_files()*

0

-

1 runs additional section in *create_output_files()* which writes two lines (names, units) in output files

---

3.5.3 makeplots

plot the results

Switch in `SCOPE.m`

0

-

1 launches *plots()* for the results of the last run

CHAPTER

FOUR

DIRECTORIES

4.1 SCOPE_v1.73

Contents

4.1.1 Files

input_data.xlsx

Main input file is `input_data.xlsx` with 4 sheets. In the documentation we refer to this file, although text alternatives are also possible.

Note: If Excel is not available, it is possible to use input from text files (.m and .txt). See **alternative**.

To specify which input to use (text or excel) comment / uncomment lines in `set_parameter_filenames.m` with `` % sign.

sheet (tab)	content	alternative
readme	sheets description of `input_data.xlsx` explanation of leaf inclination distribution function (LIDF) parameters recommended values for plan functional types (PFTs) some parameter ranges	-
options	*Options*	`setoptions.m`
filenames	filenames for current simulation and for time-series	`filenames.m`
inputdata	values for *input structs*	`inputdata.txt`

4.1.2 Directories

output

The function `output_data()` saves the output of SCOPE in an output directory.

In SCOPE, output_data is called after each calculation.

The data are stored in the following directory: `SiteName_yyyy-mm-dd-hh-mm`

In which `yyyy` refers to the Julian year,

`mm` to the month,

`dd` the day,

`hh` the hour and

`mm` the minutes

of the time when the simulation was started.

for files see *Output files*

src

.m files with the code.

- *+equations*
- *+helpers*
- *+io (input output)*
- *+plot*
- *not_used*

4.2 data

Contents

4.2.1 input

dataset for_verification

'Dataset for_verification' contains time series of meteorological data. In this case, half-hourly data are provided. It is possible to work with any time interval, but due to the thermal inertia of the soil, the calculation of soil temperature may not be accurate when the time interval is longer than three hours. It is recommended to name your own dataset 'dataset sitename or projectname'. The directory contains the following compulsory files (all in ASCII format):

- A time vector (`t_.dat`): a vertical array of time values, in decimal days of the year [1:366.99]. For example, 10 January 2009, 12:00 would be: 10.5. All other files (see below) should correspond to this time vector (and thus have the same size).
- A year vector (`year_.dat`): the year corresponding to the time vector. For example, 10 January 2009, 12:00 would be: 2009
- TOC incoming shortwave radiation (`Rin_.dat`): a broadband (0.3 to 2.5 m) measurement of incoming shortwave radiation (W m-2), perpendicular to the surface.
- TOC incoming long wave radiation (`Rli_.dat`): a broadband (2.5 to 50 m) measurement of incoming long wave radiation (W m-2), perpendicular to the surface.
- Air pressure (`p_.dat`): air pressure (hPa or mbar)
- Air temperature measured above the canopy (`Ta_.dat`): air temperature above the canopy in °C.
- Vapour pressure measured above the canopy (`e_.dat`): vapour pressure above the canopy (hPa or mbar).
- Wind speed (`u_.dat`): wind speed measured above the canopy (m s-1)

The following additional files (not compulsory) can be added:

- Carbon dioxide concentration measured above the canopy (mg m-3)

And the following tables (not compulsory):

- Leaf area index (`LAI_.dat`)
- Measurement height (`z_.dat`) (m)

- Vegetation height (`h_.dat`) (m)
- Maximum carboxylation capacity (`Table_Vcmax_.dat`)
- Chlorophyll content file (`Table_Cab_.dat`)

If a table is not present, then the corresponding a priori value specified in the file input_data.xlsx file is used instead. It is only useful to create the tables LAI_dat etc. if the leaf area index, measurement height, vegetation height etc. change with time during the measurement period.

A table has a slightly different format than the other input files. A table has two columns: the first column contains the decimal DOY, the second column contains the value of the variable. The reason why tables have a different format is that the variables in the table are usually not measured at the same time interval as the meteorological input. For example, the LAI may be measured only once per month.

An example of a table can be found in 'dataset for_verification'. The measurement height is only relevant for wind speed, vapour pressure and the carbon dioxide concentration. It is currently not possible to specify separate measurement heights for each of these variables.

The carbon dioxide concentration must be provided in mg m-3. This is a commonly used unit in most data sets. SCOPE automatically converts this to ppm and to umol m-3 internally. If the carbon dioxide concentration file is not provided, SCOPE assumes a constant concentration corresponding to 380 ppm.

Note: It is important that all files except for the tables have equal length, and that all measurements correspond to the time vector. A Julian calendar is used. The time zone should be provided (the difference between the local time in the file and UTC or GMT. Input files should be comma separated, space separated or tab separated ASCII files. They should not contain empty lines or comment lines.

In case SCOPE is run in individual mode, then the meteorological input files are not used. In that case, all meteorological data are taken from the spreadsheet.

directional

The input in the directory 'directional' is only used for multi-angle simulations (if the option 'directional' is switched on in parameters. In this directory one can provide the observer's zenith and azimuth angles. The files in this directory have two columns: the first column is the observer's zenith angle, the second the observer's azimuth (relative to that of the sun, counterclockwise), both in degrees. If the option `directional` is switched on, SCOPE will calculate the radiance spectrum in all directions given in the input file.

fluspect_parameters

In this directory, absorption spectra of different leaf components are provided, according to PROSPECT 3.1, as well as Fluspect input: standard spectra for PSI and PSII.

leafangles

In this folder, example leaf inclination distribution data are provided. It is possible to use these distributions instead of the leafinclination model of Verhoef (1998 [6]) with the two parameters LIDFa and LIDFb. In that case, provide a filename in the `input_data.xlsx` tab **filenames** or the file `filenames.m`.

radiationdata

`RTMo.m` calculates spectra based on MODTRAN5 outputs. One .atm (atmospheric) file is provided in the data, 12 more are provided separately in a different .zip folder (in order to minimize the size of the SCOPE package, these are not provided standard). Note that in the input data (files as well as the spreadsheet), the broadband input radiation may be provided. SCOPE linearly scales the input spectra of the optical and the thermal domain in such a way, that the spectrally integrated input shortwave and long wave radiation matches with the measured values. A limitation of this approach is that the same shape of the input spectrum is used independent on the atmospheric conditions. If this scaling is not wanted, then leave 'Rin' and 'Rli' empty in the spreadsheet.

Note: In earlier versions of the model (1.34 and older), two input spectra of solar and sky radiation were provided (rad.txt and rad2.txt) in this directory. The data were calculated with MODTRAN4. The ASCII file in this directory consisted of three columns containing the following. The first column contained the wavelength in nm, the second column the solar radiation in W m-2 m-1, and the third column the sky radiation in W m-2 m-1. These data are now obsolete (since version 1.40).

soil_spectrum

In this directory, the soil spectrum is provided. The ASCII file in this directory consists of two columns containing the following: The first column contains the wavelength in m, the following columns reflectance spectra. Note that it is also possible to simulate a soil reflectance spectrum with the BSM model. In that case the values for the BSM model parameters are taken from the input data, and the archived spectra in this folder are not used.

4.2.2 measured

The validation data are stored in directory 'measured'. It is up to the user to organize this directory.

Warning: Do not change directory names or file names inside them!

```
SCOPE-master.zip
├── SCOPE_v1.73
│   ├── output
│   │   ├── example_directional_run
│   │   │   ├── Directional
│   │   │   └── Parameters
│   │   └── verificationdata
│   │       └── Parameters
│   └── src
│       ├── +equations
│       ├── +helpers
│       ├── +io
│       ├── +plot
│       └── not_used
```

(continues on next page)

(continued from previous page)

```
├── data
│   ├── input
│   │   ├── dataset for_verification
│   │   ├── directional
│   │   ├── fluspect_parameters
│   │   ├── leafangles
│   │   ├── PFT
│   │   ├── radiationdata
│   │   └── soil_spectrum
│   └── measured
│       ├── dataset Duke
│       ├── dataset for_verification
│       ├── dataset Merzenhausen
│       ├── dataset reading 1
│       ├── dataset reading 2
│       └── dataset yarnton 2013
└── docs
```

CHAPTER

FIVE

OUTPUT FILES

5.1 In each simulation

Contents

5.1.1 aerodyn.dat

rows - time (simulation number)

columns - variables

variable	units	description
raa	s m-1	total aerodynamic resistance above canopy
rawc	s m-1	canopy total aerodynamic resistance below canopy
raws	s m-1	soil total aerodynamic resistance below canopy
ustar	m s-1	friction velocity

5.1.2 BOC_irradiance.dat

BOC - bottom of canopy (61st layer)

rows - timestep

First 2162 columns: shaded fraction.

Last 2162 columns: average BOC irradiance.

variable	units	description
Emin_(61, :)	W m-2 um-1	irradiance at the bottom of the canopy for the shaded fraction
Emin_(61, :) + Esun_(61, :) * gap.Ps(61, :)	W m-2 um-1	average BOC irradiance (sunlit + shaded fraction)

5.1.3 fluxes.dat

rows - time (simulation number)

columns - variables

variable	units	description
timestep	-	time step counter
counter	-	number of iterations in energy balance
year	-	year
T	-	decimal day of year (DOY)
Rntot	W m-2	total net radiation
lEtot	W m-2	total latent heat flux
Htot	W m-2	total sensible heat
Rnctot	W m-2	net radiation of canopy
lEctot	W m-2	latent heat flux of canopy
Hctot	W m-2	sensible heat of canopy
Actot	umol m-2 s-1	net photosynthesis of canopy
Rnstot	W m-2	net radiation of soil
lEstot	W m-2	latent heat flux of soil
Hstot	W m-2	sensible heat of soil
Gtot	W m-2	soil heat flux
Resp	umol m-2 s-1	soil respiration rate
aPAR_Cab	umol m-2 s-1	absorbed PAR by chlorophylls a, b
aPAR	umol m-2 s-1	absorbed PAR by leaves
fPAR	-	fraction of absorbed PAR by canopy, excluding soil
aPAR_energyunits	W m-2	absorbed PAR
iPAR	W m-2	incident PAR
fluortot	W m-2	hemispherically and spectrally integrated chlorophyll fluorescence at the top
fluor_yield	W W-1	Fluortot / aPAR_energyunits

5.1.4 irradiance_spectra.dat

rows - time (simulation number)

columns - wl

variable	units	description
Rin * (fEsuno + fEskyo)	W m-2 um-1	spectrum of incoming radiation used in the simulation

5.1.5 pars_and_input.dat

rows - timestep

columns - all input parameters from `input_data.xlxs`

5.1.6 pars_and_input_short.dat

rows - timestep

columns - `Cab, Vcmo, LAI, hc, zo, d, z, Rin, Ta, Rli, p, ea, u, Ca, tts, SMC`

5.1.7 radiation.dat

rows - time (simulation number)

columns - variables

variable	units	description
timestep	-	time step counter
year	-	year
T	-	decimal day of year (DOY)
ShortIn (Rin)	W m-2	Incoming shortwave radiation (copy from input)
LongIn (Rli)	W m-2	Incoming longwave radiation (copy from input)
HemisOutShort (Eouto)	W m-2	hemispherical outgoing shortwave radiation
HemisOutLong (Eoutt + Eoutte)	W m-2	hemispherical outgoing longwave radiation
HemisOutTot (Eouto + Eoutt + Eoutte)	W m-2	total hemispherical outgoing radiation
Net (Rntot)	W m-2	total net radiation

5.1.8 reflectance.dat

rows - time (simulation number)

columns - wl

variable	units	description
Lo_ * pi / (Esun_ + Esky_)	-	fraction of radiation in observation direction * pi / irradiance

5.1.9 spectrum_hemis_optical.dat

rows - time (simulation number)

columns - wl number (2162)

variable	units	description
Eout_	W m-2 um-1	hemispherical outgoing radiation spectrum

5.1.10 spectrum_obsdir_optical.dat

rows - time (simulation number)

columns - wl number (2162)

variable	units	description
Lo_	W m-2 um-1 sr-1	radiance spectrum in observation direction

5.1.11 surftemp.dat

rows - time (simulation number)

columns - variables

variable	units	description
timestep	-	time step counter
year	-	year
T	-	decimal day of year (DOY)
Ta	ºC	Air temperature above the canopy
Tss(1)	ºC	Surface temperature of shaded soil
Tss(2)	ºC	Surface temperature of sunlit soil
Tcave	ºC	canopy weighted average temperature
Tsave	ºC	soil weighted average temperature

5.1.12 wl.dat

single row (2162)

variable	units	description
wl	nm	wavelengths of the spectral output files

5.2 options.calc_ebal & options.calc_planck

Contents

5.2.1 spectrum_hemis_thermal.dat

Note: options.calc_ebal & options.calc_planck

rows - time (simulation number)

columns - wl number (2162)

variable	units	description
Eoutte_	W m-2 um-1	hemispherical outgoing thermal radiation

5.2.2 spectrum_obsdir_BlackBody.dat

Note: options.calc_ebal

rows - time (simulation number)

columns - wl number (2162)

variable	units	description
LotBB_	W m-2 um-1 sr-1	thermal BlackBody emission spectrum in observation direction

5.2.3 spectrum_obsdir_thermal.dat

Note: `options.calc_ebal` & `options.calc_planck`

rows - time (simulation number)

columns - wl number (2162)

variable	units	description
Lot_	W m-2 um-1 sr-1	outgoing thermal radiation in observation direction

5.3 options.calc_vert_profiles

5.3.1 gap.dat

Note: `options.calc_vert_profiles`

rows - time (simulation number)

columns - [Ps Po Pso] => 61 * 3 columns

variable	units	description
Ps	-	fraction of sunlit leaves per layer
Po	-	fraction of observed leaves per layer
Pso	-	fraction of sunlit and (at the same time) observed per layer

5.3.2 layer_a.dat

Note: `options.calc_vert_profiles` & `options.calc_ebal`

rows - time (simulation number)

columns - photosynthesis per layer, total soil respiration (60 + 1)

variable	units	description
A1d	umol m-2 s-1	mean photosynthesis leaves, per layer
Resp	umol m-2 s-1	soil respiration rate

5.3.3 layer_aPAR.dat

Note: `options.calc_vert_profiles`

rows - time (simulation number)

columns - absorbed photosynthetically active radiation (aPAR)

variable	units	description
Pn1d	umol m-2 s-1	aPAR per leaf layer

5.3.4 layer_aPAR_Cab.dat

Note: `options.calc_vert_profiles`

rows - time (simulation number)

columns - absorbed photosynthetically active radiation (aPAR) by chlorophylls (Cab) per leaf layer

variable	units	description
Pn1d_Cab	umol m-2 s-1	aPAR by Cab per leaf layer

5.3.5 layer_fluorescence.dat

Note: `options.calc_vert_profiles` & `options.calc_fluor`

rows - time (simulation number)

columns - upward fluorescence per layer

variable	units	description
fluorescence	W m-2	upward fluorescence per layer

5.3.6 layer_h.dat

Note: `options.calc_vert_profiles` & `options.calc_ebal`

rows - time (simulation number)

columns - sensible heat flux per layer, total sensible heat of soil (60 + 1)

variable	units	description
Hc1d	W m-2	mean sensible heat leaves, per layer
Hstot	W m-2	sensible heat of soil

5.3.7 layer_le.dat

Note: `options.calc_vert_profiles` & `options.calc_ebal`

rows - time (simulation number)

columns - latent heat flux per layer, total latent heat of soil (60 + 1)

variable	units	description
lEc1d	W m-2	mean latent heat leaves, per layer
lEstot	W m-2	latent heat of soil

5.3.8 layer_NPQ.dat

Note: `options.calc_vert_profiles` & `options.calc_ebal`

rows - time (simulation number)

columns - average NPQ = 1-(fm-fo)/(fm0-fo0), per layer (60)

variable	units	description
qE		average NPQ = 1-(fm-fo)/(fm0-fo0), per layer

5.3.9 layer_rn.dat

Note: `options.calc_vert_profiles` & `options.calc_ebal`

rows - time (simulation number)

columns - net radiation per leaf layer, total net radiation of soil (60 + 1)

variable	units	description
Rn1d	W m-2	net radiation per leaf layer
Rnstot	W m-2	net radiation of soil

5.3.10 leaftemp.dat

Note: `options.calc_vert_profiles` & `options.calc_ebal`

rows - time (simulation number)

columns - leaf temperatures per layer (60 * 3) leaf temperature of sunlit leaves, shaded leaves, and weighted average leaf temperature per layer

variable	units	description
Tcu1d	ºC	leaf temperature of sunlit leaves, per layer
Tch	ºC	leaf temperature of shaded leaves, per layer
Tc1d	ºC	weighted average leaf temperature, per layer

5.4 options.calc_fluo

5.4.1 fluorescence.dat

Note: `options.calc_fluor`

rows - time (simulation number)

columns - fluorescence from both photosystems in observation direction

variable	units	description
LoF_	W m-2 um-1 sr-1	fluorescence per wavelength in observation direction

5.4.2 fluorescence_emitted_by_all_leaves.dat

Note: `options.calc_fluor`

rows - time (simulation number)

columns - total emitted fluorescence by all leaves. Within canopy scattering / re-absorption is omitted. Within leaf scattering / re-absorption is taken into account.

variable	units	description
Fem_	W m-2 um-1	hemispherical emitted fluorescence by all leaves

5.4.3 fluorescence_emitted_by_all_photosystems.dat

Note: `options.calc_fluor`

rows - time (simulation number)

columns - total emitted fluorescence by all photosystems for wavelengths Within canopy scattering / re-absorption is omitted. Within leaf scattering / re-absorption is omitted.

variable	units	description
Femtot	W m-2 um-1	hemispherical emitted fluorescence by all photosystems per wavelengths (excluding leaf and canopy re-absorption and scattering)

5.4.4 fluorescence_hemis.dat

Note: `options.calc_fluor`

rows - time (simulation number)

columns - top of canopy (TOC) hemispherical fluorescence

variable	units	description
Fhem_	W m-2 um-1	TOC hemispherical fluorescence

5.4.5 fluorescence_scattered.dat

Note: `options.calc_fluor`

rows - time (simulation number)

columns - top of canopy (TOC) fluorescence contribution from leaves and soil after scattering

variable	units	description
sum(LoF_scattered) + sum(LoF_soil)	W m-2 um-1 sr-1	TOC directional fluorescence from leaves and soil after scattering

5.4.6 fluorescence_shaded.dat

Note: `options.calc_fluor`

rows - time (simulation number)

columns - top of canopy (TOC) fluorescence contribution from shaded leaves in observer direction per wavelengths

variable	units	description
LoF_shaded	W m-2 um-1 sr-1	TOC fluorescence from shaded leaves in observer direction

5.4.7 fluorescence_sunlit.dat

Note: `options.calc_fluor`

rows - time (simulation number)

columns - top of canopy (TOC) fluorescence contribution from sunlit leaves in observer direction per wavelengths

variable	units	description
LoF_sunlit	W m-2 um-1 sr-1	TOC fluorescence from sunlit leaves in observer direction

5.4.8 fluorescencePSI.dat

Note: `options.calc_fluor` && `options.calc_PSI`

rows - time (simulation number)

columns - fluorescence of photosystem I (PSI) per wavelength in observation direction

variable	units	description
LoF1_	W m-2 um-1 sr-1	fluorescence of PSI per wavelength in observation direction

5.4.9 fluorescencePSII.dat

Note: `options.calc_fluor` && `options.calc_PSI`

rows - time (simulation number)

columns - fluorescence of photosystem II (PSII) per wavelength in observation direction

variable	units	description
LoF2_	W m-2 um-1 sr-1	fluorescence of PSII per wavelength in observation direction

5.5 `options.calc_directional` && `options.calc_ebal`

The output files are stored in folder Directions of your output

Note: This is an optional output that requires `options.calc_directional` & `options.calc_ebal` However, the folder will always be created

5.5.1 Directional/Angles (SunAngle x.xx degrees).dat

Contains the directions.

- The 1st row gives the observation zenith angles
- The 2nd row gives the observation azimuth angles
- The 3rd row gives the solar zenith angles (constant from `input_data.xlsx`)

columns - combination number (a set of direction used for simulation)

Columns in the output files correspond to the columns in Angles

5.5.2 Directional/BRDF (SunAngle x.xx degrees).dat

- The 1st column gives the wl values corresponding to the BRDF values
- Other columns give the BRDF values corresponding to the directions given by observation zenith angles (first column in the Angles file)

variable	units	description
wlS	nm	full wl range SCOPE
brdf_	-	bidirectional reflectance distribution function

5.5.3 Directional/Temperatures (SunAngle x.xx degrees).dat

- The 1st column gives the wl values corresponding to the brightness temperature values (except for broadband)
- Other columns give the brightness temperature (BT) values corresponding to the directions given by a column in the Angles file

variable	units	description
BrightnessT	K	brightness temperature
`options.calc_planck`		
wlT	nm	thermal wl range SCOPE
Lot_	W m-2 um-1 sr-1	outgoing thermal radiation in observation direction

5.5.4 Directional/Fluorescence (SunAngle x.xx degrees).dat

if `options.calc_fluor`

- The 1st column gives the wl values corresponding to the brightness temperature values (except for broadband)
- Other columns give the fluorescence corresponding to the directions given by a column in the Angles file

variable	units	description
wlF	nm	fluorescence wl range SCOPE
LoF_	W m-2 um-1 sr-1	outgoing fluorescence radiation in observation direction

5.5.5 Directional/read me.txt

The file with similar explanation

CHAPTER

SIX

BRIEF HISTORY OF THE MODEL

The SCOPE model has been developed between 2006 and 2009 by Wout Verhof, Joris Timmermans, Christiaan van der Tol, Anne Verhoef and Bob Su. The idea of the model was to develop a simulator for hyperspectral VNIR observations, the surface energy budget and photosynthesis. Chlorophyll fluorescence has been part of the model. It was originally the idea to develop a 3-D radiative transfer scheme, but this idea was (temporally) abandoned, and 1-D remained a 1-D vertical model. This had the advantage that the well-known SAIL model could be used as a basis, which is easily invertible, does not require many parameters, is computationally efficient and sufficient in many cases.

The key elements of the model have been the extension to the thermal domain (Joris Timmermans) and the radiative transfer of fluorescence (Wout Verhoef), the simulation of sensible, latent and ground heat flux, stomatal opening and photosynthesis (Christiaan van der Tol) and an aerodynamic resistance scheme (Anne Verhoef). Model inversion tools are not also available, see for example Van der Tol et al., (2016 [5]). There have been several updates since the published version of the model (version 1.21) in 2009. Other people have contributed to the model development as well, including Ari Kornfeld, Joe Berry, Federico Magnani (mainly the biochemical part, but also other parts), and many users provided useful feedback and suggestions (see *Acknowledgements*).

Model description Van der Tol et al., 2009 [11]

Biochemical routine Van der Tol et al., 2014 [10]

Leaf radiative transfer scheme Vilfan et al., 2016 [7]

Model inversion Van der Tol et al., 2016 [5]

CHAPTER

SEVEN

ACKNOWLEDGEMENTS

The development of SCOPE was supported by the Space program of the Netherlands Organization for Scientific Research (NWO), grants NWO-SRON-EO-071 and ALW-GO/13-32, and the European Space Agency (ESA ESTEC ITT AO/1-7088/12/NL/AF "Photosynthesis Study" and ESA RFP IPL-PEO/FF/lf/14.687 "FLEX Bridge Study").

Many users contributed with their feedback and suggestions. Particular thanks to: Ari Kornfeld, Albert Olioso, Jerome Démarty, Federico Magnani, Jose Moreno, Jochem Verrelst, Suvarna Punalekar, Yves Goulas, Marco Celesti, and Georg Wolfahrt.

The module `biochemical.m` is based on papers by Collatz et et. (1991, 1992 [1][2]), with significant contributions by Joe Berry and Ari Kornfeld. The module `biochemical_MD12.m` was built by Federico Magnani.

CHAPTER

EIGHT

ROADMAP

Documentation for functions will be rendered in a better way.

Research output: SCOPE use cases and papers will be added.

CHAPTER

NINE

REFERENCES

CHAPTER

TEN

VERSION HISTORY

Contents

10.1 1.73

2019

By Ari Kornfeld

- **Add "invalid CO2" error check to ebal**
  - Invalid complex-valued CO2 values generated by the energy balance routines were incorrectly attributed to fixed_brent (which is the only module that has its own error-checking). This change assigns "blame" closer to the source of the problem.

- **Fixes: An intercept termfor the Ball-Berry equation, *BallBerry0*, was added to the input files ("input_data.xls"x and "inpu**
  - Setting *BallBerry0* to 0 disables the iterative solver introduced in v1.7.
- **Fix bug because Ccu is not a vector (ebal.m)**
  - Add more input-checking to biochemical.m, to catch when initial input is bad.
- **pass leafbio.BallBerry0 to biochem_in**
  - Delete "null" code (assigning a value to biochem_in.A)
  - Allow active warnings when temperatures include NaN. (should be an error, but doesn't propagate to future time steps, so leave as a warning.
- Add gitignore to skip large, rapdily changing files. And gitattributes
- **Increase iter.maxit to 400, so ebal converges.**
  - 100 is too few for some realistic cases.
  - Note this does not affect Ball-Berry iteration.
  - Also remove clc, which can be a confusing side-effect.

10.2 1.72

2018

- **Bug with soil moisture content (SMC) for `BSM()` is solved.**
  - SMC range in input is from **0 to 1** (used in `calc_rssrbs()`, `Soil_Inertia1()`)
  - `BSM()` required SMC in the range from **0 to 100**
  - solution: scaling of SMC within `BSM()`: SMC * 100
  - now `BSM()` accepts SMC from **0 to 1**
  - this bug might effect the results if `options.soilspectrum == 1`
- **Misleading comments in `filenames` were corrected**
  - SMC is a **one-column** file
  - z-file is a **two-column** table
- `input_data_default.xlsx` was added with the verification run parameters to make it easier to check that SCOPE still works after you changed something in the code and do not remember the initial configuration of the `input_data.xlsx`

10.3 1.71

2018

- **No changes to output or calculations were done.**
- **Interactive documentation for ReadTheDocs was created (./docs):**
  - `code` folder was renamed to `src` for autodocumentation
  - all scripts were transformed to functions for autodocumentation

  - functions were grouped into matlab modules (directories starting with + sign), see *API*
  - **./SCOPE_v1.70/readme** was deleted

10.4 1.70

2017

- OPTIPAR of PROSPECT-D model used, complemented with Xanthophyll spectra for the Violaxanthin to Zeaxanthin conversion.
- The FLUSPECT model includes dynamic Xanthophyll reflectance due to the de-epoxydation state (the 'PRI effect') and Athocyanins
- A new radiative transfer model, RTMz, simulates the TOC reflectance as a function of the de-epoxydation state induced by light, water or temperature stress.
- The fluorescence emission spectra have been tuned to FluoWat leaf clip measurements. The option to use the fluorescence spectra of V1.62 and older remains.
- The biochemical routine has been updated, and now the internal CO2 concentration in the leaf is calculated iteratively (Ari Kornfeld)
- The BSM model for soil reflectance added as an option.
- SCOPE and SCOPE_mac_linux merged into a single script.
- The option to load the leaf inclination distribution from a file (besides the option to use the LIDFa and LIDFb parameters to simulate the distribution)
- New outputs: The total emitted fluorescence irradiance by all photosystems (i.e. before reabsorption within the leaf and canopy), the total emitted fluorescence irradiance by all leaves accumulated (i.e. before reabsorption by soil and canopy), and the fluorescence originating from sunlit and shaded leaves and the (multiple) scattered flux have been added as separate output files. The bottom of canopy irradiance flux (the flux on the soil) has been added to the output as a spectrum. Several outputs have been added to the 'fluxes' and 'radiation' files, including the incident PAR and the incident radiation.
- Two bugs in the RTMt_Planck have been fixed.

10.5 1.62

2016

- Photosynthesis is a function of aPAR absorbed by Chlorophyll (only) rather than total leaf aPAR as in earlier verions.

10.6 1.61

2015

- Bug in the saving of total evaporation data corrected (bug in versions 1.40 to 1.60). Bug in the loading of time series of roughness length for momentum (zo) and zero plane displacement height (d) calculated from LAI and canopy height was corrected.

10.7 1.60

2015

- Major revision of RTMf: computation speed improved (Ari Kornfeld), scattered fluorescence flux added to the directional flux (Christiaan van der Tol).
- Improved calculation speed of RTMt_sb (AK)
- Revision of Ball-Berry model in biochemical.m: now iterative calculation of Ci and stomatal conductance (AK)
- Minor improvements in the energy balance (soil heat flux computation, suggested by Georg Wolfahrt).
- Input spreadsheet in 'SCOPE' has changed from "input_data.xls" to "input_data.xlsx". Way of reading the sheets 'filenames' and 'options' has changed (AK and CvdT). 'SCOPE' should now also work for MAC and LINUX, but to be sure, SCOPE_mac_linux.m has been maintained.
- Default value of parameter 'fqe' in input spectrum has been tuned to FluoWat measurements

10.8 1.54

2014

- Fluspect replaced by Fluspect_bcar, an updated version of Fluspect with the absorption by carotenoids included, similar to PROSPECT 5

10.9 1.53

2014

- Correction of a bug in Fluspect, which caused the fluorescence spectra to be 2 × too low in version 1.52.

10.10 1.52

2013

- Additional fluorescence output, change in the input data of optipar, and some modification of biochemical_MD12.m. Saves also the path of the code (including SCOPE version) to the output. Bug fixed in Fluspect (a scattering coefficient). Correction for PSI fluorescence moved from RTMf to biochemical.m.

10.11 1.51

2013

- Addition of an alternative leaf level photosynthesis and fluorescence model according to Von Caemmerer (2000) and Magnani et al (2013). Correction of the bug in version 1.40

10.12 1.40

2014

- Major changes in the structure of the model. Coupling with MODTRAN-derived output files. The irradiance spectral input data are now calculated from MODTRAN atmospheric files. The input is specified in a spreadsheet. Variables are organized in structures which makes it easier to plug in new modules. This version has a bug in the unit of the CO2 concentration.

Version 1.40 is no longer available.

10.13 1.34

2012

- Update of FLUSPECT with separate fluorescence spectra for PSI and PSII. Replacing the TVR09 model for fluorescence with an empirical model. Hemispherically integrated fluorescence is added as an output. The photosynthesis model is made consistent with Collatz et al (1991 and 1992), also used in CLM and SiB models, includes C3 and C4 vegetation, and empirically calibrated fluorescence model according to Lee et al. (2013). The possibility to create Look-Up Tables has been introduced, as well as more options for running only parts of the model.

10.14 1.32

2012

- The leaf level optical model FLUSPECT was introduced, which produces leaf reflectance, transmittance and fluorescence spectra. Rather than using given fixed fluorescence matrices as inputs, SCOPE now uses FLUSPECT to calculate the excitation to fluorescence conversion matrices.

10.15 1.21

2009

- The SCOPE model as published in Biogeosciences (2009).

CHAPTER

ELEVEN

SUPPORT

For any questions, bugs and collaboration ideas please, create a topic in our SCOPE_model group.

CHAPTER

TWELVE

STRUCTS

12.1 input structs

12.1.1 F

Filenames from `filenames` sheet of `input_data.xlsx` or `filenames.m`

The files are located in ../*data*

Note: This is an array of 22 structs

Initialized

`SCOPE.m`

Used

variable	user
`soil_file, leaf_file, atmos_file` `LIDF_file` if provided	`SCOPE.m`
`Simulation_Name`	*`create_output_files()`*
other structs from this	*`load_timeseries()`*

Fields

Fields initialized in `SCOPE.m`. Each of 22 structs in this array has these fields.

variable	units	type	default	description
FileID	-	char	defined in `SCOPE.m`	SCOPE file identifiers
FileNames	-	[1 x 1] cell	-	filenames from `filenames`

12.1.2 angles

Solar and observation zenith and azimuth angles

Initialized

select_input()

Used

variable	user
tts tto, psi -> directional_angles	*calc_brdf()*
tto, psi	*RTMt_planck()* *RTMt_sb()*
tts, tto, psi	*RTMf()* *RTMo()* *RTMz()*
tts	*output_data()*

Fields

Fields initialized in *select_input()* (read from input_data.xlsx)

variable	units	type	default	description
tts	deg	double	30.0	solar zenith angle
tto	deg	double	0.0	observer zenith angle
psi	deg	double	90.0	azimuthal difference between solar and observation angle relative azimuth angle

12.1.3 atmo

Atmospheric transfer functions from standard FLEX atmospheres

Initialized

`SCOPE.m` loaded from ../data/input/*radiationdata* and aggregated by *`aggreg()`*.

Filename is specified on `filenames` sheet, `atmos_file` cell of `input_data.xlsx`

Used

variable	user
`M, Ta`	*`RTMo()`*

Fields

Fields initialized in `SCOPE.m`

variable	units	type	default	description
M	-	[2162 x 6] double	from `FLEX-S3_Tr12.atm`	atmospheric transmittance functions T1, T3, T4, T5, T12, T16
Ta	ºC	double	20.0 (== meteo.Ta)	air temperature

12.1.4 canopy

Canopy parameters, such as leaf area index and leaf inclination distribution function

Initialized

`SCOPE.m`

`select_input()`

Variations

canopy.lidf can be read from `LIDF_file` if its name is provided in the `filenames` sheet of `input_data.xlsx`

Note: `LIDF_file` must be located in `/data/input/leafangles` (*leafangles*) and have **3 header lines**.

canopy.zo, *canopy.d* may be calculated by *`zo_and_d()`* if `options.calc_zo` is selected

canopy.hc may be set in *`load_timeseries()`*

Warning: never change the angles in *canopy.litab* unless `leafangles()` ('ladgen') is also adapted

Used

variable	user
`nlayers, nlincl, nlazi, lidf`	*meanleaf()*
`CR, CD1, Psicor, LAI, hc`	*zo_and_d()*
`LAI, hc, zo, d`	*load_timeseries()*
`LIDFa, LIDFb`	*leafangles()*
`nlayers, kV, xl, LAI` `LAI, rwc, zo, d, hc, leafwidth, Cd` -> *Resist_in*	*ebal()*
`nlayers, lidf, litab, lazitab, LAI`	*RTMf()* *RTMo()* *RTMt_planck()* *RTMt_sb()* *RTMz()*
`nlincl, nlazi, x, hot`	*RTMo()*
`x, nlayers, LAI`	`SCOPE.m`

Fields

Fields initialized in `SCOPE.m`

variable	units	type	default	description
nlayers	-	int	60	the number of layers in a canopy
x	-	[60 x 1] double	(0 : -1] equally spaced vector	levels in canopy except for the top: `bottom = -1`, `top = -1/canopy.nlayers` in fact length == canopy.nlayers + 1
xl	-	[61 x 1] double	[0 : -1] equally spaced vector	levels in canopy and the top [0, canopy.x] in fact length == canopy.nlayers + 1
nlincl	-	int	13	number of leaf inclinations
nlazi	-	int	36	number of leaf azimuth angles
litab	deg	[13 x 1] double	[5 : 89] *non-equally* spaced vector	SAIL leaf inclination angles
lazitab	-	[1 x 36] double	[5 : 355] equally spaced vector	leaf azimuth angles relative to the sun
lidf	?	[13 x 1] double	*leafangles()*	leaf inclination distribution function

Fields initialized in *select_input()* (read from `input_data.xlsx`)

variable	units	type	default	description
LAI	m2 m-2	double	3.0	Leaf area index
hc	m	double	2.0	vegetation height
LIDFa	-	double	-0.35	leaf inclination
LIDFb	-	double	-0.15	variation in leaf inclination
leafwidth	m	double	0.1	leaf width
rb	s m-1	double	10.0	leaf boundary resistance
Cd	?	double	0.3	leaf drag coefficient
CR	?	double	0.35	Verhoef et al. (1997) Drag coefficient for isolated tree
CD1	?	double	20.6	Verhoef et al. (1997) fitting parameter
Psicor	?	double	0.2	Verhoef et al. (1997) Roughness layer correction
rwc	s m-1	double	0.0	within canopy layer resistance
kV	?	double	0.6396	extinction coefficient for `Vcmax` in the vertical (maximum at the top). 0 for uniform `Vcmax`
zo	m	double	0.246	roughness length for momentum of the canopy
d	m	double	1.34	displacement height
hot	?	double	0.05	hotspot parameter `canopy.leafwidth / canopy.hc`

12.1.5 iter

Numerical parameters, such as the number of iterations needed to reach energy balance closure

Initialized

`SCOPE.m`

Variations

`counter` is incremented in *`ebal()`*

`Wc` is set to 0.2 if `counter` > 50 in *`ebal()`*

Used

variable	user
`maxit, maxEBer, Wc`	*`ebal()`*
`counter`	*`initialize_output_structures()`* *`output_data()`*

Fields

Fields initialized in `SCOPE.m`

variable	units	type	default	description
maxit	-	int	100	maximum number of iterations
maxEBer	W m-2	double	1.0	maximum accepted error in energy balance
Wc	-	double	1.0	weight coefficient for iterative calculation of Tc
counter	-	int	0	counter, changed in *`ebal()`*

12.1.6 leafbio

Leaf biochemical parameters

Initialized

`select_input()`

`SCOPE.m`

Variations

leafbio.Cca may be calculated as 25% of Cab: `options.Cca_function_of_Cab`

leafbio.fqe may be double (PSII only) or [2 x 1] double (PSI = 0.2 * PSII, PSII) if `options.calc_PSI`

leafbio.V2Z can be set to 0 with `options.calc_PSI`

Used

variable	user
`Cab, Cca, V2Z, Cw, Cdm, Cs, Cant, N, fqe`	*`fluspect_B_CX()`* if `options.calc_PSI` *`fluspect_B_CX_PSI_PSII_combined`*
`Type, m, Rdparam, Tyear, beta, qLs, kNPQs, stressfactor, Tparam, Vcmo` -> *biochem_in*	*`ebal()`*
`Vcmo, Cab`	*`load_timeseries()`*
`rho_thermal, tau_thermal, fqe`	`SCOPE.m`

Fields

Fields initialized in *select_input()* (read from input_data.xlsx)

variable	units	type	default	description
Cab	ug cm-2	double	80.0	Chlorophyll AB content
Cca	ug cm-2	double	20.0	Carotenoid content. Usually 25% of Cab if options.Cca_function_of_Cab
Cdm	g cm-2	double	0.012	Dry matter content
Cw	cm	double	0.009	leaf water equivalent layer
Cs	-	double	0.0	senescent material fraction
Cant	ug cm-2	double	0.0	Anthocyanins
N	-	double	1.4	leaf thickness parameters
Vcmo	umol m-2 s-1	double	60.0	maximum carboxylation capacity (at optimum temperature)
m	?	double	8.0	Ball-Berry stomatal conductance parameter
Type	-	int => char	0 ('C3')	Photochemical pathway: 0 => 'C3', 1 => 'C4'
Tparam	ºK	[5 x 1] double	[0.2, 0.3, 281, 308, 328]	See PFT.xls. These are five parameters specifying the temperature response.
fqe	-	[1 x 1] | [2 x 1] double	0.01	fluorescence quantum yield efficiency at photosystem level
Rdparam	-	double	0.015	Respiration = Rdparam * Vcmcax
rho_thermal		double	0.01	broadband thermal reflectance
tau_thermal		double	0.01	broadband thermal transmittance
Tyear	ºC	double	15.0	mean annual temperature
beta	-	double	0.507	fraction of photons partitioned to PSII (**0.507** for C3, **0.4** for C4; Yin et al. 2006 [8]; Yin and Struik 2012 [9])
kNPQs	s-1	double	0.0	rate constant of sustained thermal dissipation (Porcar-Castell 2011 [3])
qLs	-	double	1.0	fraction of functional reaction centres (Porcar-Castell 2011 [3])
stressfactor	-	double	1.0	optional input: stress factor to reduce Vcmax (for example soil moisture, leaf age)

Fields initialized in SCOPE.m

variable	units	type	default	description
V2Z	-	double	1.0	violaxantine to zeaxantine ratio. 0 if options.calc_PSI

12.1.7 meteo

Meteorological variables

Initialized

select_input()

Variations

ebal() uses `max(meteo.u, 0.2)`

Used

variable	user
`z`	*load_timeseries()*
`Ta, ea, Ca, p` `u, z` -> *Resist_in* `Oa` -> *biochem_in*	*ebal()*
`Rin, Rli`	*RTMo()*
`Rin, Rli`	*output_data()*

Fields

Fields initialized in *select_input()* (read from `input_data.xlsx`)

variable	units	type	default	description
z	m	double	10.0	measurement height of meteorological data
Rin	W m-2	double	600.0	broadband incoming shortwave radiation (0.4-2.5 um)
Ta	ºC	double	20.0	air temperature
Rli	W m-2	double	300.0	broadband incoming longwave radiation (2.5-50 um)
p	hPa	double	970.0	air pressure
ea	hPa	double	15.0	atmospheric vapour pressure
u	m s-1	double	2.0	wind speed at height z
Ca	ppm	double	380.0	atmospheric CO2 concentration
Oa	per mille	double	209.0	atmospheric O2 concentration

12.1.8 soil

Soil properties (such as soil moisture, emissivity and the reflectance spectrum)

Initialized

select_input()

SCOPE.m

Variations

soil.Tsold may be changed by *ebal()* if `options.soil_heat_method < 2` (default case)

soil.rss, *soil.rbs* may be calculated by *calc_rssrbs()* if `options.calc_rss_rbs` is selected

soil.GAM produced by *Soil_Inertia0()* or *Soil_Inertia1()* if `options.soil_heat_method`

Used

variable	user
spectrum, rs_thermal	SCOPE.m
cs, rhos, lambdas	*Soil_Inertia0()*
SMC	*Soil_Inertia1()* *BSM()* *calc_rssrbs()*
CSSOIL	*zo_and_d()*
refl	*RTMf()* *RTMo()* *RTMt_planck()* *RTMt_sb()* *RTMz()*
Ts, Tsold, GAM, rss rbs, rss -> *Resist_in*	*ebal()*

Fields

Fields initialized in *select_input()* (read from `input_data.xlsx`)

variable	units	type	default	description
spectrum	-	int	1	spectrum number (column in the database soil_file)
rss	s m-1	double	500.0	soil resistance for evaporation from the pore space
rs_thermal	-	double	0.06	broadband soil reflectance in the thermal range `1 - emissivity`
cs	J kg-1 K-1	double	1180.0	specific heat capacity of the soil
rhos	kg m-3	double	1800.0	specific mass of the soil
CSSOIL	?	double	0.01	Drag coefficient for soil Verhoef et al. (1997) *(from Aerodynamic)*
lambdas	J m-1 K-1	double	1.55	heat conductivity of the soil
rbs	s m-1	double	10.0	soil boundary layer resistance *(from Aerodynamic)*
SMC	-	double	0.25	volumetric soil moisture content in the root zone
BSMBrightness	?	double	0.5	BSM model parameter for soil brightness
BSMlat	?	double	25.0	BSM model parameter 'lat'
BSMlon	?	double	45.0	BSM model parameter 'long'

Derived variables

variable	units	type	default	description
GAM	?	double	~1814.4 *Soil_Inertia0()*	soil thermal inertia

Fields initialized in `SCOPE.m`

variable	units	type	default	description
Tsold	ºC?	[12 x 2] double	20.0	only if `options.soil_heat_method < 2`
refl	-	[2162 x 1] double	[2162 x 1] double	soil reflectance in fact length == nwl
Ts	ºC?	[2 x 1] double	[~15; ~15]	initial soil surface temperature

12.1.9 xyt

Geographical location and time of the project

Initialized

select_input() load_timeseries()

Variations

SCOPE overwrites *xyt.t*, *xyt.year* if options.simulation != 1

Used

variable	user
t, timezn, LON, LAT	*load_timeseries()*
t	*ebal()*
t, startDOY, endDOY	SCOPE
year, t	*output_data()*

Fields

Fields initialized in *select_input()* (read from input_data.xlsx)

variable	units	type	default	description
startDOY	-	double	169.0	Julian day (decimal) of start of simulations
endDOY	-	double	170.0	Julian day (decimal) of end of simulations
LAT	decimal deg	double	52.25	Latitude
LON	decimal deg	double	5.69	Longitude
timezn	hours	double	1.0	east of Greenwich

Fields initialized in *load_timeseries()*

variable	units	type	default	description
t	-	[n x 1] double	[214 x 1]	Julian day (decimal)
year	-	[m x 1] int	[216 x 1]	Calendar year

12.2 constant structs

12.2.1 constants

Physical constants

Initialized

define_constants()

Variations

Used

variable	user
sigmaSB	*calc_brdf()*
MH2O, Mair, rhoa, cp, g, kappa, sigmaSB	*ebal()*
kappa	*zo_and_d()* *resistances()*
rhoa, cp, MH2O, R	*heatfluxes()*
deg2rad, Mair, MCO2, rhoa	*load_timeseries()*
sigmaSB, C2K	*Brightness_T()* *RTMt_sb()*
deg2rad	*RTMf()* *RTMz()* *RTMt_planck()* *RTMt_sb()*
A, h, c	*RTMo()*

Fields

Fields initialized in *define_constants()*

variable	units	type	default	description
A	mol-1	double	6.02214E23	Constant of Avogadro
h	J s	double	6.6262E-34	Planck's constant
c	m s-1	double	299792458	Speed of light
cp	J kg-1 K-1	double	1004	Specific heat of dry air
R	J mol-1K-1	double	8.314	Molar gas constant
rhoa	kg m-3	double	1.2047	Specific mass of air
g	m s-2	double	9.81	Gravity acceleration
kappa	-	double	0.4	Von Karman constant
MH2O	g mol-1	double	18	Molecular mass of water
Mair	g mol-1	double	28.96	Molecular mass of dry air
MCO2	g mol-1	double	44	Molecular mass of carbon dioxide
sigmaSB	W m-2 K-4	double	5.67E-8	Stefan Boltzman constant
deg2rad	rad	double	pi/180	Conversion from deg to rad
C2K	K	double	273.15	Melting point of water

12.2.2 optipar

Leaf optical parameters: specific absorption coefficients (SAC) of leaf chemical components

Concentration * SAC

Initialized

`SCOPE.m`: read from *fluspect_parameters*

Variations

Different files (corresponding to PROSPECT versions) from *fluspect_parameters* can be selected on `filenames` sheet, `leaf_files` cell in `input_data.xlsx` or uncomment lines directly in `SCOPE.m` (~ 170)

Used

variable	user
`nr, Kdm, Kab, Kw, Ks, Kant` `Kca (or KcaV, KcaZ)`	*fluspect_B_CX()* *fluspect_B_CX_PSI_PSII_combined*
`phiI, phiII` (if `options.calc_PSI`) `phi`	`SCOPE.m` *fluspect_B_CX()* if `options.calc_PSI` `SCOPE.m` *fluspect_B_CX_PSI_PSII_combined*

Fields

Fields loaded in `SCOPE.m`

variable	units	type	description
wl	nm	[2001 x 1] double	SAC wavelength range 400-2400
nr	nm-1	[2001 x 1] double	refractive index
Kab	nm-1	[2001 x 1] double	SAC of chlorophylls a + b
Kca	nm-1	[2001 x 1] double	SAC of carotenoids (violaxanthin + zeaxanthin)
Ks	nm-1	[2001 x 1] double	SAC of senescent material
Kw	nm-1	[2001 x 1] double	SAC of water
Kdm	nm-1	[2001 x 1] double	SAC of dry matter
phiI	nm-1	[2001 x 1] double	SAC of PSI
phiII	nm-1	[2001 x 1] double	SAC of PSI
phi	nm-1	[2001 x 1] double	SAC of PSI + PSII
KcaV	nm-1	[2001 x 1] double	SAC of violaxanthin
KcaZ	nm-1	[2001 x 1] double	SAC of zeaxanthin
Kant	nm-1	[2001 x 1] double	SAC of anthocyanins
KcaV2	nm-1	[2001 x 1] double	former SAC of violaxanthin
GSV	nm-1	[2001 x 3] double	Global Soil Vectors spectra
nw	nm-1	[2001 x 1] double	water refraction index spectrum

12.2.3 soilemp

Note: This is an optional struct created for *BSM()* with `options.soilspectrum == 1`

Initialized

`SCOPE.m`

Used

variable	user
`SMC, film`	*`BSM()`*

Fields

Fields initialized in `SCOPE.m`

variable	units	type	default	description
SMC		double	25	soil moisture capacity parameter
film		double	0.015	single water film optical thickness

12.2.4 spectral

Wavelength ranges of MODTRAN, SCOPE, PAR, fluorescence

Initialized

`define_bands()`

`SCOPE.m`

Variations

Used

variable	user
wlS, wlF, wlT as index IwlT	calc_brdf()
wlS	ebal()
wlE, wlF, wlP	fluspect_B_CX() fluspect_B_CX_PSI_PSII_combined
wlS	create_output_files()
wlS, wlF	initialize_output_structures()
wlS, wlT, wlF	output_data()
wlS, wlF, wlE, IwlF	RTMf()
wlS, wlP, wlT, wlPAR, IwlP, IwlT	RTMo()
wlS, wlT, IwlT	RTMt_planck()
wlS	RTMt_sb()
wlS, wlZ	RTMz()
SCOPEspec	aggreg()
wlS, wlP, wlT, wlF, IwlP, IwlT, IwlF	SCOPE.m

Fields

Fields initialized in define_bands()

variable	units	type	default	description
wlS	nm	[1 x 2162] int	400 : 1 : 2400 2500 : 100 : 15000 16000 : 1000 : 50000	SCOPE ranges
wlP	nm	[1 x 2001] int	400 : 1 : 2400	PROSPECT data range
wlE	nm	[1 x 351] int	400 : 1 : 750	excitation in E-F matrix
wlF	nm	[1 x 211] int	640 : 1 : 850	chlorophyll fluorescence in E-F matrix
wlO	nm	[1 x 2001] int	400 : 1 : 2400	optical part (== wlP)
wlT	nm	[1 x 161] int	2500 : 100 : 15000 16000 : 1000 : 50000	thermal part
wlZ	nm	[1 x 101] int	500 : 1 : 600	xanthophyll region
wlPAR	nm	[1 x 301] int	400 : 1 : 700	PAR range

Fields used by `aggreg()` to read MODTRAN data `spectral.SCOPEspec`

variable	units	type	default	description
SCOPEspec.nreg		int	3	number of regions
SCOPEspec.start	nm	[1 x SCOPE-spec.nreg] int	[400, 2500, 16000]	number of regions
SCOPEspec.end	nm	[1 x SCOPE-spec.nreg] int	[2400, 15000, 50000]	number of regions
SCOPEspec.res	nm	[1 x SCOPE-spec.nreg] int	[1, 100, 1000]	number of regions

Fields initialized in `SCOPE.m`

variable	units	type	default	description
IwlP	-	[1 x 2001] int	1 : 2001	index of wlP in wlS
IwlT	-	[1 x 161] int	2002 : 2162	index of wlT in wlS
IwlF	-	[1 x 211] int	241 : 451	index of wlF in wlS

12.3 output structs

12.3.1 directional

Note: This is an optional struct that requires `options.calc_directional` & `options.calc_ebal` to be output and calculated

Calculated

`calc_brdf()`

Variations

`psi, tto` initialized in `SCOPE.m` are extended in *`calc_brdf()`*

`options.calc_planck` enables `Lot_` calculation (disables of `Lot` and `BrightnessT`)

`options.calc_fluor` enables `LoF_` calculation

Output file

- *Directional/Angles (SunAngle x.xx degrees).dat*

Used

variable	user
`noa`	*`calc_brdf()`*

Fields

Fields initialized in `SCOPE.m` (read from *directional*)

variable	units	type	description
noa	-	double	number of observation angles
psi	deg	[333 x 1] double	observation relative azimuth angles
tto	deg	[333 x 1] double	observation zenith angles

Fields calculated in *`calc_brdf()`*

variable	units	type	description
psi	deg	[364 x 1] double	observation relative azimuth angles
tto	deg	[364 x 1] double	observation zenith angles
brdf_	-	[2162 x 364] double	bidirectional reflectance distribution function
Eoutte		[1 x 364] double	
Lot		[161 x 364] double	
BrightnessT	K	[1 x 364] double	brightness temperature
`options.calc_planck`			
Lot_	W m-2 um-1 sr-1	[161 x 364] double	outgoing thermal radiation in observation direction
`options.calc_fluor`			
LoF_	W m-2 um-1 sr-1	[211 x 364] double	outgoing fluorescence radiation in observation direction

12.3.2 fluxes

Fluxes calculated by the model (turbulent heat exchange, radiation, CO2)

Fields are initialized by `initialize_output_structures()`

Calculated

`ebal()`

Output file

- *fluxes.dat*

if `options.calc_vert_profiles` & `options.calc_ebal` soil fluxes will also be recorded, because soil is the 61st layer.

- *leaftemp.dat*
- *layer_h.dat*
- *layer_le.dat*
- *layer_a.dat*
- *layer_rn.dat*

Variations

Various absorbed photosynthetically active radiations (aPAR) can be calculated by *ebal()* (if `options.clc_ebal`) or by `SCOPE.m`

Used

variable	user
`aPAR_Cab_eta` -> `rad.Femtot`	`SCOPE` (if `options.calc_fluor`)
`aPAR_Wm2` -> fPAR	*output_data()*

Fields

Fields calculated in *ebal()*

variable	units	type	description
Rntot	W m-2	double	total net radiation
lEtot	W m-2	double	total latent heat flux
Htot	W m-2	double	total sensible heat
Atot	umol m-2 s-1	double	total net CO2 uptake (canopy + soil)
Rnctot	W m-2	double	net radiation of canopy
lEctot	W m-2	double	latent heat flux of canopy
Hctot	W m-2	double	sensible heat of canopy
Actot	umol m-2 s-1	double	net photosynthesis of canopy
Rnstot	W m-2	double	net radiation of soil
lEstot	W m-2	double	latent heat flux of soil
Hstot	W m-2	double	sensible heat of soil
Gtot	W m-2	double	soil heat flux
Resp	umol m-2 s-1	double	soil respiration rate
Au	umol m-2 s-1	[13 x 36 x 60] double	sunlit leaves net CO2 assimilation
Ah	umol m-2 s-1	[60 x 1] double	shaded leaves net CO2 assimilation

Fields added by *ebal()* (if `options.calc_ebal == 1`) or by `SCOPE.m`

variable	units	type	description
aPAR	umol m-2 s-1	double	absorbed PAR by leaves
aPAR_Cab	umol m-2 s-1	double	absorbed PAR by chlorophylls a, b
aPAR_Wm2	W m-2	double	absorbed PAR
aPAR_Cab_eta	umol m-2 s-1	double	green ePAR * relative fluorescence emission efficiency

12.3.3 gap

Probabilities of levels being observed, illuminated per layer

Note: This is an optional struct that requires `options.calc_vert_profiles` to be output, however it will be calculated anyway

Calculated

RTMo()

Output file

if `options.calc_vert_profiles`

gap.dat

Used

variable	user
`Ps`	*ebal()* `SCOPE.m`
`Pso`	*RTMo()*
`Ps, Po, Pso`	*RTMf()* *RTMz()* *RTMt_planck()* *RTMt_sb()*
`K`	*RTMt_planck()* *RTMt_sb()*

Fields

Fields initialized in *RTMo()*

variable	units	type	description
Ps	-	[61 x 1] double	fraction of sunlit leaves per layer
Po	-	[61 x 1] double	fraction of observed leaves per layer
Pso	-	[61 x 1] double	fraction of sunlit and (at the same time) observed per layer
K	-	double	extinction coefficient in direction of observer integrated over leaf angles
k	-	double	extinction coefficient in direction of sun integrated over leaf angles

12.3.4 leafopt

Leaf optical properties

Note:

- This output is used, available in the workspace after a SCOPE run but is **not** written to any output file
- **leafoptZ** is exactly the same struct calculated by selected version of Fluspect but when all violaxanthin is transformed to zeaxanthin (`V2Z == 1`)

Calculated

`fluspect_B_CX_PSI_PSII_combined()`

Variations

`fluspect_B_CX()` **if `options.calc_PSI` =>**

`Mb` is partitioned to `MbI, MbII`
`Mf` is partitioned to `MfI, MfII`

Output file

- not saved

Used

variable	user
`refl, tran`	*RTMf()* *RTMo()* *RTMt_planck()* *RTMt_sb()* *RTMz()*
`Mb, Mf`	*RTMf()*
`kChlrel`	*RTMo()*
`reflZ, tranZ`	*RTMz()*

Fields

Fields calculated in *fluspect_B_CX_PSI_PSII_combined()* or *fluspect_B_CX()* if `options.calc_PSI`

variable	units	type	description
refl	-	[2162 x 1] double	leaf reflectance
tran	-	[2162 x 1] double	leaf transmittance
kChlrel	-	[2001 x 1] double	
Mb	-	[211 x 351] double	backward scattering fluorescence matrix
Mf	-	[211 x 351] double	forward scattering fluorescence matrix
Mbl_rc	-	[211 x 351] double	backward scattering fluorescence matrix without re-absorption
Mfl_rc	-	[211 x 351] double	forward scattering fluorescence matrix without re-absorption

These values are added from **leafoptZ**, calculated with *leafbio*.V2Z == 1.

variable	units	type	description
reflZ	-	[2162 x 1] double	leaf reflectance with only zeaxanthin
tranZ	-	[2162 x 1] double	leaf transmittance with only zeaxanthin

12.3.5 profiles

Vertical profiles of temperatures and fluxes

Note: This is an optional struct that requires `options.calc_vert_profiles` to be calculated and output

Calculated

RTMo()

Variations

RTMf() if `options.calc_fluor`

ebal() if `options.calc_ebal`

Output file

if `options.calc_vert_profiles`

- *layer_aPAR.dat*
- *layer_aPAR_Cab.dat*

if `options.calc_fluor` & `options.calc_vert_profiles`

- *layer_fluorescence.dat*

if `options.calc_ebal` & `options.calc_vert_profiles`

- *leaftemp.dat*
- *layer_h.dat*
- *layer_le.dat*
- *layer_a.dat*
- *layer_NPQ.dat*
- *layer_rn.dat*

Used

variable	user
`etah, etau`	*RTMf()*
`Knh, Khu`	*RTMz()*

Fields

Fields calculated in *RTMo()* if `options.calc_vert_profiles`

variable	units	type	description
Pn1d	umol m-2 s-1	[60 x 1] double	absorbed photosynthetically active radiation (aPAR) per leaf layer
Pn1d_Cab	umol m-2 s-1	[60 x 1] double	aPAR per leaf layer

Fields added in *RTMf()* if `options.calc_fluor`

variable	units	type	description
fluorescence	W m-2	[60 x 1] double	upward fluorescence per layer

Fields added in *ebal()* if `options.clc_ebal`

> **Warning:** Averaging temperatures is not physically accurate

variable	units	type	description
etah	-	[60 x 1] double	Fs / Fo ratio for shaded leaves
etau	-	[13 x 36 x 60] double	Fs / Fo ratio for sunlit leaves
Tchave	℃	double	mean temp shaded leaves
Tch	℃	[60 x 1] double	leaf temperature of shaded leaves, per layer
Tcu1d	℃	[60 x 1] double	leaf temperature of sunlit leaves, per layer
Tc1d	℃	[60 x 1] double	weighted average leaf temperature, per layer
Hc1d	W m-2	[60 x 1] double	mean sensible heat leaves, per layer
lEc1d	W m-2	[60 x 1] double	mean latent heat leaves, per layer
A1d	umol m-2 s-1	[60 x 1] double	mean photosynthesis leaves, per layer
Rn1d	W m-2	[60 x 1] double	net radiation per leaf layer
F_Pn1d		[60 x 1] double	mean fluorescence leaves, per layer
qE		[60 x 1] double	average NPQ = 1-(fm-fo)/(fm0-fo0), per layer
Knu		[13 x 36 x 60] double	NPQ of sunlit leaves
Knh		[60 x 1] double	NPQ of shaded leaves

12.3.6 rad

Radiation fluxes: both input (MODTRAN) and output

A large number of radiative fluxes: spectrally distributed and integrated, and canopy radiative transfer coefficients.

Fields are initialized by *initialize_output_structures()*

Calculated

RTMo()

RTMf()

RTMt_planck()

RTMt_sb()

ebal()

`SCOPE.m`

Output file

- *radiation.dat*
- *spectrum_obsdir_optical.dat*
- *spectrum_hemis_optical.dat*
- *irradiance_spectra.dat*
- *reflectance.dat*
- *BOC_irradiance.dat*

if `options.calc_ebal`

- *spectrum_obsdir_BlackBody.dat*

if `options.calc_planck`

- *spectrum_hemis_thermal.dat*
- *spectrum_obsdir_thermal.dat*

if `options.calc_fluor`

- *fluorescence.dat*
- *fluorescencePSI.dat*
- *fluorescencePSII.dat*
- *fluorescence_hemis.dat*
- *fluorescence_emitted_by_all_leaves.dat*
- *fluorescence_emitted_by_all_photosystems.dat*
- *fluorescence_sunlit.dat*
- *fluorescence_shaded.dat*
- *fluorescence_scattered.dat*

Variations

if `options.calc_PSI` fluorescence (`LoF_`) is partitioned between photosystems `LoF1_`, `LoF2_`

Used

variable	user
Lot LoF_	calc_brdf()
Rnuc, Rnhct, Rnuct, Rnhst, Rnust, Rnhc, Rnuc, Rnhs, Rnus Pnh_Cab, Pnu_Cab -> *biochem_in* Pnh, Pnu, Pnh_PAR, Pnu_PAR Eoutte	ebal()
vb, vf, Esun_, Emin_, Eplu	RTMf() RTMz()
Pnh, Pnu, Pnh_Cab, Pnu_Cab, Rnh_PAR, Rnu_PAR	SCOPE.m

Fields

Fields initialized in RTMo()

variable	units	type	description
rsd	-	[2162 x 1] double	conical-hemispherical reflectance factor (specular in -> diffuse out)
rdd	-	[2162 x 1] double	bihemispherical reflectance factor (diffuse in -> diffuse out)
rdo	-	[2162 x 1] double	hemispherical-conical reflectance factor (diffuse in -> specular out)
rso	-	[2162 x 1] double	biconical reflectance factor (specular in -> specular out)
vb	-	[2162 x 1] double	directional back scattering coefficient for diffuse incidence
vf	-	[2162 x 1] double	directional forward scattering coefficient for diffuse incidence
Esun_	mW m-2 um-1	[2162 x 1] double	incident solar spectrum
Esky_	mW m-2 um-1	[2162 x 1] double	incident sky spectrum
PAR	mol m-2 s-1	double	incident spectrally integrated PAR
fEsuno	-	[2162 x 1] double	fraction of direct light (optical)
fEskyo	-	[2162 x 1] double	fraction of diffuse light (optical)
fEsunt	-	[2162 x 1] double	fraction of direct light (thermal)
fEskyt	-	[2162 x 1] double	fraction of diffuse light (thermal)
Eplu_	mW m-2 um-1	[61 x 2162] double	upward diffuse radiation in the canopy
Emin_	mW m-2 um-1	[61 x 2162] double	downward diffuse radiation in the canopy
Lo_	mW m-2 um-1 sr-1	[2162 x 1] double	top of canopy (TOC) radiance in observation direction
Eout_	mW m-2 um-1	[2162 x 1] double	top of canopy (TOC) upward radiation
Eouto	W m-2	double	spectrally integrated upward optical radiation
Eoutt	W m-2	double	spectrally integrated upward thermal radiation

continues on next page

Table 1 – continued from previous page

Rnhs	W m-2	double	net radiation of shaded soil
Rnus	W m-2	double	net radiation of sunlit soil
Rnhc	W m-2	[60 x 1] double	net radiation of shaded leaves
Rnuc	W m-2	[13 x 36x 60] double	net radiation of sunlit leaves
Pnh	mol n-2 s-1	[60 x 1] double	net PAR of shaded leaves
Pnu	mol n-2 s-1	[13 x 36x 60] double	net PAR of sunlit leaves
Pnh_Cab	mol n-2 s-1	[60 x 1] double	net PAR absorbed by Cab of shaded leaves
Pnu_Cab	mol n-2 s-1	[13 x 36x 60] double	net PAR absorbed by Cab of sunlit leaves
Pnh_PAR	W m-2	[60 x 1] double	net PAR of shaded leaves (W m-2)
Pnu_PAR	W m-2	[13 x 36x 60] double	net PAR of sunlit leaves (W m-2)
Etoto		double	

Fields initialized in `RTMf()`

Note: Model simulated fluorescence at 3 levels:

- level of photosystems individually (PSI, PSII) or together
- level of leaves
- **level of canopy**
  - in observation direction (reaching sensor) (typically starts with **Lo**)
  - hemispherically integrated

variable	units	type	description
Fem_	W m-2 um-1	[211 x 1] double	total emitted fluorescence by all leaves, excluding within canopy scattering / re-absorption
Fhem_	W m-2 um-1	[211 x 1] double	TOC hemispherically integrated fluorescence
LoF_	W m-2 um-1 sr-1	[211 x 1] double	fluorescence per wavelength
LoF1_	W m-2 um-1 sr-1	[211 x 1] double	fluorescence from photosystem I (PSI) per wavelength
LoF2_	W m-2 um-1 sr-1	[211 x 1] double	fluorescence from photosystem II (PSII) per wavelength
Fhem_	W m-2 um-1	[211 x 1] double	
Fmin_	W m-2 um-1	[211 x 61] double	downward fluorescence flux profile
Fplu_	W m-2 um-1	[211 x 61] double	upward fluorescence flux profile
LoF_sunlit	W m-2 um-1 sr-1	[211 x 2] double	TOC fluorescence contribution from sunlit leaves in observer direction per wavelengths
LoF_shaded	W m-2 um-1 sr-1	[211 x 2] double	TOC fluorescence contribution from shaded leaves in observer direction per wavelengths
LoF_scattered	W m-2 um-1 sr-1	[211 x 2] double	TOC fluorescence contribution after scattering from leaves
LoF_soil	W m-2 um-1 sr-1	[211 x 2] double	TOC fluorescence contribution after scattering from soil
Eoutf	W m-2	double	hemispherically and spectrally integrated TOC fluorescence
Eminf_	W m-2 sr-1	[61 x 21] double	
Epluf_	W m-2 sr-1	[61 x 21] double	

Fields initialized in `RTMt_planck()`

variable	units	type	description
Lot_		double	
Eoutte_		double	
Eplut_		[61 x 1] double	
Emint_		[61 x 1] double	

Fields initialized in `RTMt_sb()`

variable	units	type	description
Lot		double	
Eoutte		double	
Eplut		[61 x 1] double	
Emint		[61 x 1] double	
Rnuct		[13 x 36 x 60] double	
Rnhct		[60 x 1] double	
Rnust		double	
Rnhst		double	

Fields added in `ebal()`

variable	units	type	description
LotBB_	W m-2 sr-1	[2162 x 1] double	blackbody radiance

Fields added in `SCOPE.m`

variable	units	type	description
Femtot	W m-2 um-1	[211 x 1] double	total emitted fluorescence by all photosystems per wavelengths (excluding leaf and canopy re-absorption and scattering)

12.3.7 thermal

Leaf, soil and air temperatures

Fields are initialized by `initialize_output_structures()`

Calculated

`ebal()`

Output file

- *surftemp.dat*
- *aerodyn.dat*

Used

variable	user
`Tcu, Tch, Ts(1)`	*calc_brdf()* (if `options.calc_planck`) -> *RTMt_planck()* *calc_brdf()* (else) -> *RTMt_sb()*
`Tcu, Tch, Ts(1), Ts(2)`	`SCOPE.m` (if `options.calc_planck`) -> *RTMt_planck()*

Fields

Fields added in *ebal()*

variable	units	type	description
Tcave	ºC	double	canopy weighted average temperature
Tsave	ºC	double	soil weighted average temperature
raa	s m-1	double	total aerodynamic resistance above canopy
rawc	s m-1	double	canopy total aerodynamic resistance below canopy
raws	s m-1	double	soil total aerodynamic resistance below canopy
ustar	m s-1	double	friction velocity
Ts	ºC	[2 x 1] double	sunlit and shaded soil temperature
Ta	ºC	double	air temperature as in input
Tcu	ºC	[13 x 36 x 60] double	sunlit leaves canopy temperature
Tch	ºC	[60 x 1] double	shaded leaves canopy temperature

12.4 internal structs

This structures are created within functions and can be available in debug mode.

12.4.1 V

Variable names and values.

Note:

- This is an array of 63 structs
- This is a transitional data structure between `input_data.xlsx` file and *input structs*

Initialized

assignvarnames() -> `Name`

`SCOPE.m` or *load_timeseries()* -> `Val`

Used

variable	user
all	*select_input()*

Fields

Fields initialized in `SCOPE.m`. Each of 63 structs in this array has these fields.

variable	units	type	default	description
Name	-	char	defined in `SCOPE.m`	SCOPE file identifiers
Val	corresponding	char	-	values from `input_data.xlsx` or files in timeseries

12.4.2 biochem_in

Note: This is an internal struct of *ebal()*. It is only available within *ebal()* in debug mode.

Input of the biochemical routine *biochemical()* or *biochemical_MD12()* for **leaf** photosynthesis and fluorescence.

Initialized

ebal()

Variations

For sunlit leaves size is [13 x 36 x 60] for shaded [60 x 1]

Fields

Fields initialized in *ebal()*

variable	units	type	description
Fluorescence_model		bool	`options.Fluorescence_model`
Type	-	char	photosynthesis type C3 or C4
p	hPa	double	atmospheric pressure
m		double	Ball-Berry stomatal conductance parameter
O	per mille	double	atmospheric O2 concentration
Rdparam	-	double	fraction of respiration
T	ºC	[13 x 36 x 60] double [60 x 1] double	leaf temperature per canopy layer
eb	hPa	[13 x 36 x 60] double [60 x 1] double	leaf water (ea) per canopy layer
Cs	ppm	[13 x 36 x 60] double [60 x 1] double	leaf CO2 concentration per canopy layer
Vcmo	umol m-2 s-1	[13 x 36 x 60] double [60 x 1] double	maximum carboxylation rate per canopy layer
Q	W m-2	[13 x 36 x 60] double [60 x 1] double	absorbed photosynthetically active radiation (PAR) by chlorophylls per canopy layer
A	umol m-2 s-1	[13 x 36 x 60] double [60 x 1] double	photosynthesis (CO2 assimilation rate) per canopy layer

These parameters are specific for *biochemical()*

variable	units	type	description
tempcor	-	double	`options.apply_T_corr`
Tparams	K	[5 x 1] double	the temperature response of fluorescence
stressfactor	-	double	stress factor to reduce Vcmax

These parameters are specific for *biochemical_MD12()*

variable	units	type	description
Tyear	ºC	double	mean annual temperature
beta	-	double	fraction of photons partitioned to PSII
kNPQs	s-1	double	rate constant of sustained thermal dissipation
qLs	-	double	fraction of functional reaction centres

12.4.3 Biochem_out

Note: This is an internal struct of `ebal()`. It is only available within `ebal()` in debug mode.

Output of the biochemical routine `biochemical()` or `biochemical_MD12()` for **leaf** photosynthesis and fluorescence.

Initialized

`ebal()`

Variations

For sunlit leaves size is [13 x 36 x 60] for shaded [60 x 1]

Fields

Output specific for `biochemical()`

variable	units	type	description
A	umol m-2 s-1	[13 x 36 x 60] double [60 x 1] double	photosynthesis (CO2 assimilation rate) per canopy layer
Ci	ppm	[13 x 36 x 60] double [60 x 1] double	within leaf CO2 concentration per canopy layer
eta	-	[13 x 36 x 60] double [60 x 1] double	Fs / Fo
rcw	-	[13 x 36 x 60] double [60 x 1] double	stomatal resistance
qE	-	[13 x 36 x 60] double [60 x 1] double	non-photochemical quenching 1 - (Fm - Fo) / (Fm0 -Fo0)
Kn	-	[13 x 36 x 60] double [60 x 1] double	NPQ = (Fm - Fm')/Fm' = Kn/(Kf + Kd)

Output specific specific for `biochemical_MD12()`

variable	units	type	description

12.4.4 Resist_in

Aerodynamic resistance parameters

12.4.5 Resist_out

Aerodynamic resistance state variables

CHAPTER

THIRTEEN

API

13.1 Core

main script `SCOPE.m`

`biochemical` (*biochem_in*, *Ci_input*)

Date: 21 Sep 2012 Update: 20 Feb 2013 Update: Aug 2013: correction of L171: Ci = Ci*1e6 ./ p .* 1E3; Update: 2016-10 - (JAK) major rewrite to accomodate an iterative solution to the Ball-Berry equation

- also allows for g_m to be specified for C3 plants, but only if Ci_input is provided.

Authors: Joe Berry and Christiaan van der Tol, Ari Kornfeld, contributions of others. Sources:

Farquhar et al. 1980, Collatz et al (1991, 1992).

This function calculates:

- stomatal resistance of a leaf or needle (s m-1)
- photosynthesis of a leaf or needle (umol m-2 s-1)
- fluorescence of a leaf or needle (fraction of fluor. in the dark)

Usage: biochem_out = biochemical(biochem_in) the function was tested for Matlab R2013b

Calculates net assimilation rate A, fluorescence F using biochemical model

Input (units are important): structure 'biochem_in' with the following elements: Knparams % [], [], [] parameters for empirical Kn (NPQ) model: Kn = Kno * (1+beta).*x.^alpha./(beta + x.^alpha);

[Kno, Kn_alpha, Kn_beta]

or, better, as individual fields: Kno Kno - the maximum Kn value ("high light") Kn_alpha, Kn_beta alpha, beta: curvature parameters

Cs % [ppmV or umol mol] initial estimate of conc. of CO2 in the ... bounary layer of the leaf

Q % [umol photons m-2 s-1]net radiation, PAR fPAR % [0-1] fraction of incident light that is absorbed by the leaf (default = 1, for compatibility) T % [oC or K] leaf temperature eb % [hPa = mbar] intial estimate of the vapour pressure in leaf boundary layer O % [mmol/mol] concentration of O2 (in the boundary

... layer, but no problem to use ambient)

p % [hPa] air pressure Vcmax25 (Vcmo) % [umol/m2/s] maximum carboxylation capacity @ 25 degC BallBerrySlope (m) % [] Ball-Berry coefficient 'm' for stomatal regulation BallBerry0 % [] (OPTIONAL) Ball-Berry intercept term 'b' (if present, an iterative solution is used)

setting this to zeo disables iteration. Default = 0.01

Type % ['C3', 'C4'] text parameter, either 'C3' for C3 or any ... other text for C4

tempcor % [0, 1] boolean (0 or 1) whether or not ... temperature correction to Vcmax has to be applied.

Tparams % [],[],[K],[K],[K] vector of 5 temperature correction parameters, look in spreadsheet of PFTs.
Only if tempcor=1, otherwise use dummy values

... Or replace w/ individual values: slti [] slope of cold temperature decline (C4 only) shti [] slope of high temperature decline in photosynthesis Thl [K] T below which C4 photosynthesis is <= half that predicted by Q10 Thh [K] T above which photosynthesis is <= half that predicted by Q10 Trdm [K] T at which respiration is <= half that predicted by Q10

biochemical_MD12 (*biochem_in*)

[A,Ci,eta] = biochemical_VCM(Cs,Q,T,eb,O,p,Vcmo,m,Type,Rdparam,stress,Tyear,beta,qLs,NPQs) Date: 21 Sep 2012 Update: 28 Jun 2013 Adaptation for use of Farquhar model of C3 photosynthesis (Farquhar et al 1980)

18 Jul 2013 Inclusion of von Caemmerer model of C4 photosynthesis (von Caemmerer 2000, 2013) 15 Aug 2013 Modified computation of CO2-limited electron transport in C4 species for consistency with light-limited value 22 Oct 2013 Included effect of qLs on Jmax and electron transport; value of kNPQs re-scaled in input as NPQs

Authors: Federico Magnani, with contributions from Christiaan van der Tol

This function calculates:

- CO2 concentration in intercellular spaces (umol/mol == ppmv)
- leaf net photosynthesis (umol/m2/s) of C3 or C4 species
- fluorescence yield of a leaf (fraction of reference fluorescence yield in dark-adapted and un-stressed leaf)

Usage: function [A,Cs,eb,f,rcw] = biochemical(C,Cs,Q,T,ea,eb,O,p,Vcmo,gcparam,Type,tempcor,ra,Tparams,Rdparam,stressfacto
the function was tested for Matlab 7.2.0.232 (R2006a)

Input (units are important; when not otherwise specified, mol refers to mol C): Cs % [umol/mol] CO2 concentration at leaf surface Q % [uE/m2/s] photochemically active radiation absorbed by the leaf T % [oC or K] leaf temperature eb % [hPa] vapour pressure in leaf boundary layer O % [mmol/mol] ambient O2 concentration p % [Pa] air pressure Vcmo % [umol/m2/s] maximum carboxylation capacity m % [mol/mol] Ball-Berry coefficient 'm' for stomatal regulation Type % [] text parameter, either 'C3' for C3 or any other text for C4 Rdparam % [mol/mol] respiration at reference temperature as fraction of Vcmax stress % [] optional input: stress factor to reduce Vcmax (for example soil moisture, leaf age). Default value = 1 (no stress). Tyear % [oC] mean annual temperature beta % [] fraction of photons partitioned to PSII (0.507 for C3, 0.4 for C4; Yin et al. 2006; Yin and Struik 2012) qLs % [] fraction of functional reaction centres (Porcar-Castell 2011) NPQs % [s-1] rate constant of sustained thermal dissipation, normalized to (kf+kD) (=kNPQs'; Porcar-Castell 2011)

Note: always use the prescribed units. Temperature can be either oC or K Note: input can be single numbers, vectors, or n-dimensional matrices Note: For consistency reasons, in C4 photosynthesis electron transport rates under CO2-limited conditions are computed by inverting the equation

applied for light-limited conditions(Ubierna et al 2013). A discontinuity would result when computing J from ATP requirements of Vp and Vco, as a fixed electron transport partitioning is assumed for light-limited conditions

BSM (*soilpar*, *spec*, *emp*)

Spectral parameters

calc_brdf (*options*, *directional*, *spectral*, *angles*, *rad*, *atmo*, *soil*, *leafopt*, *canopy*, *meteo*, *profiles*, *thermal*)

ebal (*iter*, *options*, *spectral*, *rad*, *gap*, *leafopt*, *angles*, *meteo*, *soil*, *canopy*, *leafbio*, *xyt*, *k*, *profiles*)

function ebal.m calculates the energy balance of a vegetated surface

authors: Christiaan van der Tol (tol@itc.nl) Joris Timmermans (j_timmermans@itc.nl)

date 26 Nov 2007 (CvdT) updates 29 Jan 2008 (JT & CvdT) converted into a function

11 Feb 2008 (JT & CvdT) improved soil heat flux and temperature calculation 14 Feb 2008 (JT) changed h in to hc (as h=Avogadro`s constant) 31 Jul 2008 (CvdT) Included Pntot in output 19 Sep 2008 (CvdT) Converted F0 and F1 from units per aPAR into units per iPAR 07 Nov 2008 (CvdT) Changed layout 18 Sep 2012 (CvdT) Changed Oc, Cc, ec

Feb 2012 (WV) introduced structures for variables Sep 2013 (JV, CvT) introduced additional biochemical model

parent: master.m (script) uses:

RTMt_sb.m, RTMt_planck.m (optional), RTMf.m (optional) resistances.m heatfluxes.m biochemical.m soil_respiration.m

Table of contents of the function

1. **Initialisations for the iteration loop** intial values are attributed to variables
2. **Energy balance iteration loop** iteration between thermal RTM and surface fluxes
3. Write warnings whenever the energy balance did not close
4. Calculate vertical profiles (optional)
5. Calculate spectrally integrated energy, water and CO2 fluxes

The energy balance iteration loop works as follows:

RTMo More or less the classic SAIL model for Radiative Transfer of sun and sky light (no emission by the vegetation)

While continue Here an iteration loop starts to close the energy

balance, i.e. to match the micro-meteorological model and the radiative transfer model

RTMt_sb A numerical Radiative Transfer Model for thermal radiation emitted by the vegetation

resistances Calculates aerodynamic and boundary layer resistances of vegetation and soil (the micro-meteorological model)

biochemical Calculates photosynthesis, fluorescence and stomatal resistance of leaves (or biochemical_MD12: alternative)

heatfluxes Calculates sensible and latent heat flux of soil and vegetation Next soil heat flux is calculated, the energy balance is evaluated, and soil and leaf temperatures adjusted to force energy balance closure

end {while continue}

meanleaf Integrates the fluxes over all leaf inclinations azimuth angles and layers, integrates over the spectrum

usage:

[iter,fluxes,rad,profiles,thermal] . . .

= ebal(iter,options,spectral,rad,gap,leafopt, . . . angles,meteo,soil,canopy,leafbio)

The input and output are structures. These structures are further specified in a readme file.

Input:

iter numerical parameters used in the iteration for energy balance closure options calculation options spectral spectral resolutions and wavelengths rad incident radiation gap probabilities of direct light penetration and viewing leafopt leaf optical properties angles viewing and observation angles soil soil properties canopy canopy properties leafbio leaf biochemical parameters

Output:

iter numerical parameters used in the iteration for energy balance closure fluxes energy balance, turbulent, and CO2 fluxes rad radiation spectra profiles vertical profiles of fluxes thermal temperatures, aerodynamic resistances and friction velocity

fluspect_B_CX(*spectral*, *leafbio*, *optipar*)

function [leafopt] = fluspect(spectral,leafbio,optipar) calculates reflectance and transmittance spectra of a leaf using FLUSPECT, plus four excitation-fluorescence matrices

Authors: Wout Verhoef, Christiaan van der Tol (tol@itc.nl), Joris Timmermans, Date: 2007 Update from PROSPECT to FLUSPECT: January 2011 (CvdT)

Nov 2012 (CvdT) Output EF-matrices separately for PSI and PSII

31 Jan 2013 (WV) Adapt to SCOPE v_1.40, using structures for I/O 30 May 2013 (WV) Repair bug in s for non-conservative scattering 24 Nov 2013 (WV) Simplified doubling routine 25 Nov 2013 (WV) Restored piece of code that takes final refl and

tran outputs as a basis for the doubling routine

03 Dec 2013 (WV) Major upgrade. Border interfaces are removed before the fluorescence calculation and later added again

23 Dec 2013 (WV) Correct a problem with N = 1 when calculating k and s; a test on a = Inf was included

01 Apr 2014 (WV) Add carotenoid concentration (Cca and Kca) 19 Jan 2015 (WV) First beta version for simulation of PRI effect 17 Mar 2017 (CT) Added Anthocyanins (following Prospect-D)

usage: [leafopt] = fluspect_b(spectral,leafbio,optipar)

inputs: Cab = leafbio.Cab; Cca = leafbio.Cca; V2Z = leafbio.V2Z; % Violaxanthin - Zeaxanthin transition status

[0-1]

Cw = leafbio.Cw; Cdm = leafbio.Cdm; Cs = leafbio.Cs; Cant = leafbio.Cant; N = leafbio.N; fqe = leafbio.fqe;

nr = optipar.nr; Kdm = optipar.Kdm; Kab = optipar.Kab; Kca = optipar.Kca; KcaV = optipar.KcaV; KcaZ = optipar.KcaZ; Kw = optipar.Kw; Ks = optipar.Ks; phiI = optipar.phiI; phiII = optipar.phiII;

outputs: refl reflectance tran transmittance Mb backward scattering fluorescence matrix, I for PSI and II for PSII Mf forward scattering fluorescence matrix, I for PSI and II for PSII

fluspect_B_CX_PSI_PSII_combined(*spectral*, *leafbio*, *optipar*)

function [leafopt] = fluspect(spectral,leafbio,optipar) calculates reflectance and transmittance spectra of a leaf using FLUSPECT, plus four excitation-fluorescence matrices

Authors: Wout Verhoef, Christiaan van der Tol (tol@itc.nl), Joris Timmermans, Date: 2007 Update from PROSPECT to FLUSPECT: January 2011 (CvdT)

Nov 2012 (CvdT) Output EF-matrices separately for PSI and PSII

31 Jan 2013 (WV) Adapt to SCOPE v_1.40, using structures for I/O 30 May 2013 (WV) Repair bug in s for non-conservative scattering 24 Nov 2013 (WV) Simplified doubling routine 25 Nov 2013 (WV) Restored piece of code that takes final refl and

tran outputs as a basis for the doubling routine

03 Dec 2013 (WV) Major upgrade. Border interfaces are removed before the fluorescence calculation and later added again

23 Dec 2013 (WV) Correct a problem with N = 1 when calculating k and s; a test on a = Inf was included

01 Apr 2014 (WV) Add carotenoid concentration (Cca and Kca) 19 Jan 2015 (WV) First beta version for simulation of PRI effect 17 Mar 2017 (CT) Added Anthocyanins according to Prospect-D

usage: [leafopt] = fluspect_b(spectral,leafbio,optipar)

inputs: Cab = leafbio.Cab; Cca = leafbio.Cca; V2Z = leafbio.V2Z; % Violaxanthin - Zeaxanthin transition status

[0-1]

Cw = leafbio.Cw; Cdm = leafbio.Cdm; Cs = leafbio.Cs; Cant = leafbio.Cant; N = leafbio.N; fqe = leafbio.fqe;

nr = optipar.nr; Kdm = optipar.Kdm; Kab = optipar.Kab; Kca = optipar.Kca; KcaV = optipar.KcaV; KcaZ = optipar.KcaZ; Kw = optipar.Kw; Ks = optipar.Ks; phi = optipar.phi; outputs: refl reflectance tran transmittance Mb backward scattering fluorescence matrix, I for PSI and II for PSII Mf forward scattering fluorescence matrix, I for PSI and II for PSII

heatfluxes(*ra*, *rs*, *Tc*, *ea*, *Ta*, *e_to_q*, *PSI*, *Ca*, *Ci*)

resistances(*resist_in*)

function resistances calculates aerodynamic and boundary resistances for soil and vegetation

Date: 01 Feb 2008 Authors: Anne Verhoef (a.verhoef@reading.ac.uk)

Christiaan van der Tol (tol@itc.nl) Joris Timmermans (j_timmermans@itc.nl)

Source: Wallace and Verhoef (2000) 'Modelling interactions in mixed-plant communities: light, water and carbon dioxide', in: Bruce Marshall, Jeremy A. Roberts (ed), 'Leaf Development and Canopy Growth', Sheffield Academic Press, UK. ISBN 0849397693

ustar: Tennekes, H. (1973) 'The logaritmic wind profile', J. Atmospheric Science, 30, 234-238 Psih: Paulson, C.A. (1970), The mathematical representation of wind speed and temperature in the unstable atmospheric surface layer. J. Applied Meteorol. 9, 857-861

Note: Equation numbers refer to equation numbers in Wallace and Verhoef (2000)

Usage: [resist_out] = resistances(resist_in)

The input and output are structures. These structures are further specified in a readme file.

Input: resist_in aerodynamic resistance parameters and wind speed

The strucutre resist_in contains the following elements: u = windspeed L = stability LAI = Leaf Area Index

RTMf(*spectral*, *rad*, *soil*, *leafopt*, *canopy*, *gap*, *angles*, *profiles*)

function 'RTMf' calculates the spectrum of fluorescent radiance in the observer's direction in addition to the total TOC spectral hemispherical upward Fs flux

Authors: Wout Verhoef and Christiaan van der Tol (c.vandertol@utwente.nl) Date: 12 Dec 2007 Update: 26 Aug 2008 CvdT Small correction to matrices

07 Nov 2008 CvdT Changed layout

Update: 19 Mar 2009 CvdT Major corrections: lines 95-96, 101-107, and 119-120.

Update: 7 Apr 2009 WV & CvdT Major correction: lines 89-90, azimuth dependence was not there in previous verions (implicit assumption of azimuth(solar-viewing) = 0). This has been corrected

Update: May-June 2012 WV & CvdT Add calculation of hemispherical Fs fluxes

Update: Jan-Feb 2013 WV Inputs and outputs via structures for SCOPE Version 1.40

Update: Jan 2015 CvdT Added two contributions to SIF radiance cuased by rescattering of hemispherical SIF fluxes Update: Jan 2015 JAK (from SCOPE 1.53): Improved speed by factor of 9+! (by vectorizing the summation over the 60 layers) Update: Jan 2015 WV Rearranged some arrays to smoothen the vectorizations; adjusted some internal names

The inputs and outputs are structures. These structures are further specified in a readme file.

Input: spectral information about wavelengths and resolutions rad a large number of radiative fluxes: spectrally distributed

and integrated, and canopy radiative transfer coefficients.

soil soil properties leafopt leaf optical properties canopy canopy properties (such as LAI and height) gap probabilities of direct light penetration and viewing angles viewing and observation angles profiles vertical profiles of fluxes

Output:

rad a large number of radiative fluxes: spectrally distributed and integrated, and canopy radiative transfer coefficients. Here, fluorescence fluxes are added

0.0 globals

RTMo (*spectral*, *atmo*, *soil*, *leafopt*, *canopy*, *angles*, *meteo*, *rad*, *options*)
function RTMo

calculates the spectra of hemisperical and directional observed visible and thermal radiation (fluxes E and radiances L), as well as the single and bi-directional gap probabilities

the function does not require any non-standard Matlab functions. No changes to the code have to be made to operate the function for a particular canopy. All necessary parameters and variables are input or global and need to be specified elsewhere.

Authors: Wout Verhoef (verhoef@nlr.nl) Christiaan van der Tol (tol@itc.nl) Joris Timmermans (j_timmermans@itc.nl)

updates: 10 Sep 2007 (CvdT) - calculation of Rn

5 Nov 2007 - included observation direction

12 Nov 2007 - included abs. PAR spectrum output

- improved calculation efficiency

13 Nov 2007 - written readme lines 11 Feb 2008 (WV&JT) - changed Volscat

(JT) - small change in calculation Po,Ps,Pso

- introduced parameter 'lazitab'
- changed nomenclature
- Appendix IV: cosine rule

04 Aug 2008 (JT) - Corrections for Hotspot effect in the probabilities 05 Nov 2008 (CvdT) - Changed layout 04 Jan 2011 (JT & CvdT) - Included Pso function (Appendix IV)

- removed the analytical function (for checking)

02 Oct 2012 (CvdT) - included incident PAR in output

Jan/Feb 2013 (WV) - Major revision towards SCOPE version 1.40:

- Parameters passed using structures
- Improved interface with MODTRAN atmospheric data
- Now also calculates 4-stream reflectances rso, rdo, rsd and rdd analytically

Apri 2013 (CvT) - improvements in variable names and descriptions

Table of contents of the function

0. Preparations

   0.1 parameters 0.2 initialisations

1. Geometric quantities

   1.1 general geometric quantities 1.2 geometric factors associated with extinction and scattering 1.3 geometric factors to be used later with rho and tau 1.4 solar irradiance factor for all leaf orientations 1.5 probabilities Ps, Po, Pso

2. Calculation of upward and downward fluxes
3. Outgoing fluxes, hemispherical and in viewing direction, spectrum

4. Net fluxes, spectral and total, and incoming fluxes A1 functions J1 and J2 (introduced for stable solutions) A2 function volscat A3 function e2phot A4 function Pso

references:

{1} Verhoef (1998), 'Theory of radiative transfer models applied in optical remote sensing of vegetation canopies'. PhD Thesis Univ. Wageninegn

{2} Verhoef, W., Jia, L., Xiao, Q. and Su, Z. (2007) Unified optical - thermal four - stream radiative transfer theory for homogeneous vegetation canopies. IEEE Transactions on geoscience and remote sensing, 45,6.

{3} Verhoef (1985), 'Earth Observation Modeling based on Layer Scattering

Matrices', Remote sensing of Environment, 17:167-175

Usage: function [rad,gap,profiles] = RTMo(spectral,atmo,soil,leafopt,canopy,angles,meteo,rad,options)

The input and output are structures. These structures are further specified in a readme file.

Input: spectral information about wavelengths and resolutions atmo MODTRAN atmospheric parameters soil soil properties leafopt leaf optical properties canopy canopy properties (such as LAI and height) angles viewing and observation angles meteo has the meteorological variables. Is only used to correct

the total irradiance if a specific value is provided instead of the usual Modtran output.

rad initialization of the structure of the output 'rad' options simulation options. Here, the option

'calc_vert_profiles' is used, a boolean that tells whether or not to output data of 60 layers separately.

Output: gap probabilities of direct light penetration and viewing rad a large number of radiative fluxes: spectrally distributed

and integrated, and canopy radiative transfer coefficients.

profiles vertical profiles of radiation variables such as absorbed PAR.

RTMt_planck (*spectral*, *rad*, *soil*, *leafopt*, *canopy*, *gap*, *angles*, *Tcu*, *Tch*, *Tsu*, *Tsh*, *obsdir*)

function 'RTMt_planck' calculates the spectrum of outgoing thermal radiation in hemispherical and viewing direction

Authors: Wout Verhoef and Christiaan van der Tol (tol@itc.nl) Date: 5 November 2007 Update: 14 Nov 2007

16 Nov 2007 CvdT improved calculation of net radiation 17 Dec 2007 JT simplified, removed net radiation 07 Nov 2008 CvdT changed layout 16 Mar 2009 CvdT removed calculation of Tbright 12 Apr 2013 CvdT introduced structures

Table of contents of the function:

0. preparations

   0.0 globals 0.1 initialisations 0.2 parameters 0.3 geometric factors of Observer 0.4 geometric factors associated with extinction and scattering 0.5 geometric factors to be used later with rho and tau 0.6 fo for all leaf angle/azumith classes

1 calculation of upward and downward fluxes 2 outgoing fluxes, hemispherical and in viewing direction A1 function planck (external function is now used)

Usage: function rad = RTMt_planck(spectral,rad,soil,leafopt,canopy,gap,angles,Tcu,Tch,Tsu,Tsh,obsdir)

Input: Symbol Description Unit Dimension —— ——— —- ——— Tcu temperature sunlit leaves C [13,36,nl] Tch temperature shaded leaves C [nl] Tsu temperature sunlit soil C [1] Tsu temperature shaded soil C [1] rad a structure containing soil a structure containing soil reflectance canopy a structure containing LAI and leaf inclination

RTMz (*spectral*, *rad*, *soil*, *leafopt*, *canopy*, *gap*, *angles*, *profiles*)

function 'RTMz' calculates the small modification of TOC outgoing radiance due to the conversion of Violaxanthin into Zeaxanthin in leaves

Author: Christiaan van der Tol (c.vandertol@utwente.nl) Date: 08 Dec 2016

The inputs and outputs are structures. These structures are further specified in a readme file.

Input: spectral information about wavelengths and resolutions rad a large number of radiative fluxes: spectrally distributed

and integrated, and canopy radiative transfer coefficients.

soil soil properties leafopt leaf optical properties canopy canopy properties (such as LAI and height) gap probabilities of direct light penetration and viewing angles viewing and observation angles profiles vertical profiles of fluxes

Output:

rad a large number of radiative fluxes: spectrally distributed and integrated, and canopy radiative transfer coefficients. Here, fluorescence fluxes are added

0.0 globals

RTMt_sb (*spectral*, *rad*, *soil*, *leafopt*, *canopy*, *gap*, *angles*, *Tcu*, *Tch*, *Tsu*, *Tsh*, *obsdir*)

function 'RTMt_sb' calculates total outgoing radiation in hemispherical direction and total absorbed radiation per leaf and soil component. Radiation is integrated over the whole thermal spectrum with Stefan-Boltzman's equation. This function is a simplified version of 'RTMt_planck', and is less time consuming since it does not do the calculation for each wavelength separately.

Authors: Wout Verhoef and Christiaan van der Tol (tol@itc.nl) date: 5 Nov 2007 update: 13 Nov 2007

16 Nov 2007 CvdT improved calculation of net radiation 27 Mar 2008 JT added directional calculation of radiation 24 Apr 2008 JT Introduced dx as thickness of layer (see parameters) 31 Oct 2008 JT introduced optional directional calculation 31 Oct 2008 JT changed initialisation of F1 and F2 -> zeros 07 Nov 2008 CvdT changed layout 16 Mar 2009 CvdT removed Tbright calculation

Feb 2013 WV introduces structures for version 1.40

Table of contents of the function

0 preparations 0.0 globals 0.1 initialisations 0.2 parameters 0.3 geometric factors of Observer 0.4 geometric factors associated with extinction and scattering 0.5 geometric factors to be used later with rho and tau 0.6 fo for all leaf angle/azumith classes

1 calculation of upward and downward fluxes 2 total net fluxes Appendix A. Stefan-Boltzmann

usage: [rad] = RTMt_sb(options,spectral,rad,soil,leafopt,canopy,gap,angles,Tcu,Tch,Tsu,Tsh)

Most input and output are structures. These structures are further specified in a readme file. The temperatures Tcu, Tch, Tsu and Tsh are variables.

Input: options calculation options spectral information about wavelengths and resolutions rad a large number of radiative fluxes: spectrally distributed

and integrated, and canopy radiative transfer coefficients

soil soil properties leafopt leaf optical properties canopy canopy properties (such as LAI and height) gap probabilities of direct light penetration and viewing angles viewing and observation angles Tcu Temperature of sunlit leaves (oC), [13x36x60] Tch Temperature of shaded leaves (oC), [13x36x60] Tsu Temperature of sunlit soil (oC), [1] Tsh Temperature of shaded soil (oC), [1]

Output:

rad a large number of radiative fluxes: spectrally distributed and integrated, and canopy radiative transfer coefficients. Here, thermal fluxes are added

0.0 globals

13.2 +equations

The following modules contain physical or empirical equations used in the model

calc_rssrbs (*SMC*, *LAI*, *rbs*)

calczenithangle (*Doy*, *t*, *Omega_g*, *Fi_gm*, *Long*, *Lat*)

author: Christiaan van der Tol (c.vandertol@utwente.nl) date: Jan 2003 update: Oct 2008 by Joris Timmermans (j_timmermans@itc.nl):

- corrected equation of time

Oct 2012 (CvdT) comment: input time is GMT, not local time!

function [Fi_s,Fi_gs,Fi_g]= calczenithangle(Doy,t,Omega_g,Fi_gm,Long,Lat)

calculates pi/2-the angle of the sun with the slope of the surface.

input: Doy day of the year t time of the day (hours, GMT) Omega_g slope azimuth angle (deg) Fi_gm slope of the surface (deg) Long Longitude (decimal) Lat Latitude (decimal)

output: Fi_s 'classic' zenith angle: perpendicular to horizontal plane Fi_gs solar angle perpendicular to surface slope Fi_g projected slope of the surface in the plane through the solar beam and the vertical

fixedp_brent_ari (*func*, *x0*, *corner*, *tolFn*, *verbose*)

Find a fixed point of func(x) using Brent's method, as described by Brent 1971

func is a single-argument function, f(x) that returns a value the same size as x: The goal is to find f(x) = x (or for Brent, f(x) - x = 0).

x0 is the initial guess (or 2 x n matrix if we want to generalize) tol is the tolerance in x (or if two-valued, x, f(x)?) corner (optional) is a known "edge" in the function that could slow down the algorithm

if specified and the first two points include the corner, the corner will be substituted as a starting point.

Written by: Ari Kornfeld, 2016-10

leafangles (*a*, *b*)

Subroutine FluorSail_dladgen Version 2.3 For more information look to page 128 of "theory of radiative transfer models applied in optical remote sensing of vegetation canopies"

FluorSail for Matlab FluorSail is created by Wout Verhoef, National Aerospace Laboratory (NLR) Present e-mail: w.verhoef@utwente.nl

This code was created by Joris Timmermans, International institute for Geo-Information Science and Earth Observation. (ITC) Email: j.timmermans@utwente.nl

main function

meanleaf (*canopy*, *F*, *choice*, *Ps*)

Planck (*wl*, *Tb*, *em*)

satvap (*T*)

function [es,s]= satvap(T) Author: Dr. ir. Christiaan van der Tol Date: 2003

calculates the saturated vapour pressure at temperature T (degrees C) and the derivative of es to temperature s (kPa/C) the output is in mbar or hPa. The approximation formula that is used is: es(T) = es(0)*10^(aT/(b+T)); where es(0) = 6.107 mb, a = 7.5 and b = 237.3 degrees C and s(T) = es(T)*ln(10)*a*b/(b+T)^2

Soil_Inertia0 (*cs*, *rhos*, *lambdas*)

soil thermal inertia

Soil_Inertia1 (*SMC*)

soil inertia method by Murray and Verhoef (

soil_respiration (*Ts*)

> **Warning:** function soil_respiration always returns 0 no matter what the input is

tav (*alfa*, *nr*)

zo_and_d (*soil*, *canopy*)

function zom_and_d calculates roughness length for momentum and zero plane displacement from vegetation height and LAI

Date: 17 November 2008 17 April 2013 (structures)

Author: A. Verhoef implemented into Matlab by C. van der Tol (c.vandertol@utwente.nl)

Source: Verhoef, McNaughton & Jacobs (1997), HESS 1, 81-91

usage: zo_and_d (soil,canopy)

canopy fields used as inpuyt: LAI one sided leaf area index hc vegetation height (m)

soil fields used: Cd Averaged drag coefficient for the vegetation CR Drag coefficient for isolated tree CSSOIL Drag coefficient for soil CD1 Fitting parameter Psicor Roughness layer correction

constants used (as global) kappa Von Karman's constant

output: zom roughness lenght for momentum (m) d zero plane displacement (m)

13.3 +helpers

Functions that are based on well-known equations: integration etc.

aggreg (*atmfile*, *SCOPEspec*)
Aggregate MODTRAN data over SCOPE bands by averaging (over rectangular band passes)

count (*nvars*, *v*, *vmax*, *id*)
nvars = number of digits v = current vector of digits vmax = maximum values of digits id = starting digit vnew = new vector of digits

Sint (*y*, *x*)
Simpson integration x and y must be any vectors (rows, columns), but of the same length x must be a monotonically increasing series

13.4 +io (input output)

This is a Matlab module => the folder starts with the '+' sign and it should not be changed.

define_constants ()

readStructFromExcel (*filename*, *sheetName*, *headerIdx*, *dataIdx*, *data_is_char*, *data_in_rows*)
Read data into a struct with names matching those found in the first column/row default is for data to be in columns (A and B); if data_in_rows = true, data are in rows 1 & 2 example:

readStructFromExcel('../input_data.xlsx', 'options', 3, 1) readStructFromExcel('../input_data.xlsx', 'filenames', 1, 2, true)

assignvarnames ()

define_bands ()
Define spectral regions for SCOPE v_1.40 All spectral regions are defined here as row vectors WV Jan. 2013

select_input (*V*, *vi*, *canopy*, *options*, *xyt*, *soil*)

load_timeseries (*V*, *leafbio*, *soil*, *canopy*, *meteo*, *constants*, *F*, *xyt*, *path_input*, *options*)

initialize_output_structures (*spectral*)

create_output_files (*parameter_file*, *F*, *path_of_code*, *options*, *V*, *vmax*, *spectral*)
Create DATA files author J.timmermans last modified 4 Aug 2008: Added the creation of log file (file with input parameters)

4 Aug 2008: j.timmermans: included variable output directories

31 Jul 2008: (CvdT) added layer_pn.dat 19 Sep 2008: (CvdT) added spectrum.dat 16 Apr 2009: (CvdT) added layer_rn.dat 18 Nov 2013: (CvdT) several updates.

output_data (*Output_dir*, *options*, *k*, *iter*, *xyt*, *fluxes*, *rad*, *thermal*, *gap*, *meteo*, *spectral*, *V*, *vi*, *vmax*, *profiles*, *directional*, *angles*)
OUTPUT DATA author C. Van der Tol modified: 31 Jun 2008: (CvdT) included Pntot in output fluxes.dat last modified: 04 Aug 2008: (JT) included variable output directories

31 Jul 2008: (CvdT) added layer_pn.dat 19 Sep 2008: (CvdT) spectrum of outgoing radiation 19 Sep 2008: (CvdT) Pntot added to fluxes.dat 15 Apr 2009: (CvdT) Rn added to vertical profiles 03 Oct 2012: (CvdT) included boolean variabel calcebal 04 Oct 2012: (CvdT) included reflectance and fPAR 10 Mar 2013: (CvdT) major revision: introduced structures 22 Nov 2013: (CvdT) added additional outputs

Standard output

output_verification(*Output_dir*)

Date: 07 August 2012 Author: Christiaan van der Tol (tol@itc.nl) output_verification.m (script) checks if the output of the latest run with SCOPE_v1.51 matches with a 'standard' output located in a directory called 'verificationdata'. If it does not, warnings will appear in the Matlab command window. The following is tested:

- does the number of output files match?
- does the size of the files match (number of bytes)?
- are all files that are in the verification dataset present with the

same file names? - is the content of the files exactly the same?

If the output is different, for example because different parameter values have been used in the simulations, then the variables that are different will be plotted: the verification data in blue, and the latest run in red. In this way the differences can be visually inspected.

13.5 not_used

This files are not used anymore but might be handy for plotting or calculations.

Brightness_T(*H*)

vangenuchten(*input*, *thetares*, *thetasat*, *alpha*, *n*, *option*)

h = vangenuchten(input,thetares, thetasat, alpha,n,option); if option not specified, or option <>1, h = input, and theta is calculated, otherwise theta = input, and h is calculated

calculate_vert_profiles(*profiles*, *canopy*)

this function is incomplete and apparently never called

These are useful for the visualization of results

plot_directional_figure4_function(*directory*)

Use: plot_directional_figure4(directory) makes BRDF, BFDF and bidirectional temperature polar plots from a SCOPE output directory (string 'directory') of directional data.

resizefigure(*spfig*, *nx*, *ny*, *xo*, *yo*, *xi*, *yi*, *xend*, *yend*)

progressbar(*varargin*)

Description: progressbar() provides an indication of the progress of some task using

graphics and text. Calling progressbar repeatedly will update the figure and automatically estimate the amount of time remaining.

This implementation of progressbar is intended to be extremely simple to use

while providing a high quality user experience.

Features:

- Can add progressbar to existing m-files with a single line of code.
- Supports multiple bars in one figure to show progress of nested loops.

- Optional labels on bars.
- Figure closes automatically when task is complete.
- Only one figure can exist so old figures don't clutter the desktop.
- Remaining time estimate is accurate even if the figure gets closed.
- Minimal execution time. Won't slow down code.
- Randomized color. When a programmer gets bored…

Example Function Calls For Single Bar Usage: progressbar % Initialize/reset progressbar(0) % Initialize/reset progressbar('Label') % Initialize/reset and label the bar progressbar(0.5) % Update progressbar(1) % Close

Example Function Calls For Multi Bar Usage: progressbar(0, 0) % Initialize/reset two bars progressbar('A', '') % Initialize/reset two bars with one label progressbar('', 'B') % Initialize/reset two bars with one label progressbar('A', 'B') % Initialize/reset two bars with two labels progressbar(0.3) % Update 1st bar progressbar(0.3, []) % Update 1st bar progressbar([], 0.3) % Update 2nd bar progressbar(0.7, 0.9) % Update both bars progressbar(1) % Close progressbar(1, []) % Close progressbar(1, 0.4) % Close

Notes: For best results, call progressbar with all zero (or all string) inputs

before any processing. This sets the proper starting time reference to calculate time remaining.

Bar color is choosen randomly when the figure is created or reset. Clicking

the bar will cause a random color change.

Demos: % Single bar m = 500; progressbar % Init single bar for i = 1:m

pause(0.01) % Do something important progressbar(i/m) % Update progress bar

end

% Simple multi bar (update one bar at a time) m = 4; n = 3; p = 100; progressbar(0,0,0) % Init 3 bars for i = 1:m

progressbar([],0) % Reset 2nd bar for j = 1:n

progressbar([],[],0) % Reset 3rd bar for k = 1:p

pause(0.01) % Do something important progressbar([],[],k/p) % Update 3rd bar

end progressbar([],j/n) % Update 2nd bar

end progressbar(i/m) % Update 1st bar

end

% Fancy multi bar (use labels and update all bars at once) m = 4; n = 3; p = 100; progressbar('Monte Carlo Trials','Simulation','Component') % Init 3 bars for i = 1:m

for j = 1:n

for k = 1:p pause(0.01) % Do something important % Update all bars frac3 = k/p; frac2 = ((j-1) + frac3) / n; frac1 = ((i-1) + frac2) / m; progressbar(frac1, frac2, frac3)

end

end

end

Author: Steve Hoelzer

Revisions: 2002-Feb-27 Created function 2002-Mar-19 Updated title text order 2002-Apr-11 Use floor instead of round for percentdone 2002-Jun-06 Updated for speed using patch (Thanks to waitbar.m) 2002-Jun-19 Choose random patch color when a new figure is created 2002-Jun-24 Click on bar or axes to choose new random color 2002-Jun-27 Calc time left, reset progress bar when fractiondone == 0 2002-Jun-28 Remove extraText var, add position var 2002-Jul-18 fractiondone input is optional 2002-Jul-19 Allow position to specify screen coordinates 2002-Jul-22 Clear vars used in color change callback routine 2002-Jul-29 Position input is always specified in pixels 2002-Sep-09 Change order of title bar text 2003-Jun-13 Change 'min' to 'm' because of built in function 'min' 2003-Sep-08 Use callback for changing color instead of string 2003-Sep-10 Use persistent vars for speed, modify titlebarstr 2003-Sep-25 Correct titlebarstr for 0% case 2003-Nov-25 Clear all persistent vars when percentdone = 100 2004-Jan-22 Cleaner reset process, don't create figure if percentdone = 100 2004-Jan-27 Handle incorrect position input 2004-Feb-16 Minimum time interval between updates 2004-Apr-01 Cleaner process of enforcing minimum time interval 2004-Oct-08 Seperate function for timeleftstr, expand to include days 2004-Oct-20 Efficient if-else structure for sec2timestr 2006-Sep-11 Width is a multiple of height (don't stretch on widescreens) 2010-Sep-21 Major overhaul to support multiple bars and add labels

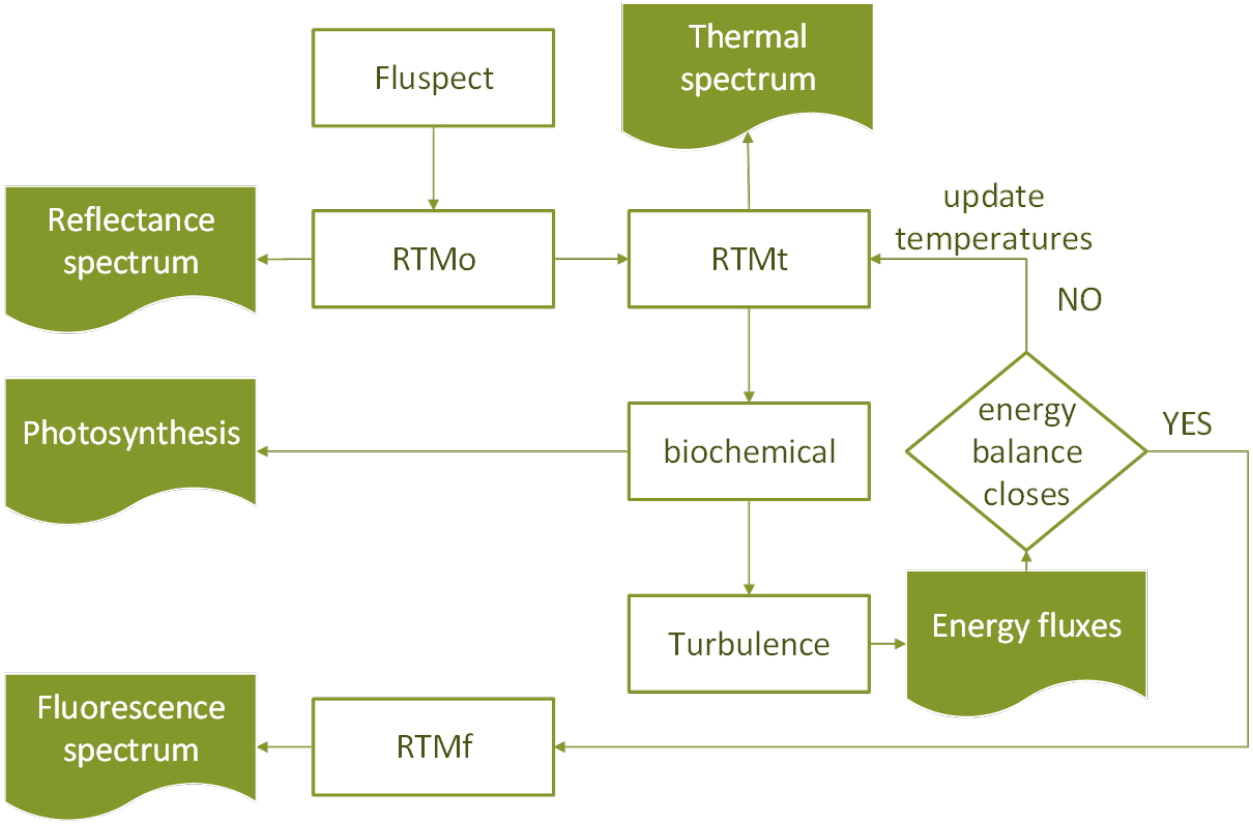
This functions are present inside RTMo as nested functions

e2phot (*lambda*, *E*)
molphotons = e2phot(lambda,E) calculates the number of moles of photons corresponding to E Joules of energy of wavelength lambda (m)

ephoton (*lambda*)
E = phot2e(lambda) calculates the energy content (J) of 1 photon of wavelength lambda (m)

13.6 +plot

plots (*Output_dir*)
plots.m (script) makes plots the output of SCOPE_v1.51 of the latest run.


Fluspect
Thermal spectrum
Reflectance spectrum
RTMo
RTMt
update temperatures
NO
Photosynthesis
biochemical
energy balance closes
YES
Turbulence
Energy fluxes
Fluorescence spectrum
RTMf

BIBLIOGRAPHY

[1] G.James Collatz, J.Timothy Ball, Cyril Grivet, and Joseph A Berry. Physiological and environmental regulation of stomatal conductance, photosynthesis and transpiration: a model that includes a laminar boundary layer. *Agric. For. Meteorol.*, 54(2-4):107–136, apr 1991. URL: https://www.sciencedirect.com/science/article/pii/0168192391900028, doi:10.1016/0168-1923(91)90002-8.

[2] GJ Collatz, M Ribas-Carbo, and JA Berry. Coupled Photosynthesis-Stomatal Conductance Model for Leaves of C \textless sub\textgreater 4\textless /sub\textgreater Plants. *Aust. J. Plant Physiol.*, 19(5):519, 1992. URL: http://www.publish.csiro.au/?paper=PP9920519, doi:10.1071/PP9920519.

[3] Albert Porcar-Castell. A high-resolution portrait of the annual dynamics of photochemical and non-photochemical quenching in needles of Pinus sylvestris. *Physiol. Plant.*, 143(2):139–153, oct 2011. URL: http://www.ncbi.nlm.nih.gov/pubmed/21615415http://doi.wiley.com/10.1111/j.1399-3054.2011.01488.x, doi:10.1111/j.1399-3054.2011.01488.x.

[4] G. Schaepman-Strub, M. E. Schaepman, T. H. Painter, S. Dangel, and J. V. Martonchik. Reflectance quantities in optical remote sensing-definitions and case studies. *Remote Sens. Environ.*, 103(1):27–42, 2006. doi:10.1016/j.rse.2006.03.002.

[5] Christiaan van der Tol, Micol Rossini, Sergio Cogliati, Wouter Verhoef, Roberto Colombo, Uwe Rascher, and Gina Mohammed. A model and measurement comparison of diurnal cycles of sun-induced chlorophyll fluorescence of crops. *Remote Sens. Environ.*, 186:663–677, dec 2016. URL: https://www.sciencedirect.com/science/article/pii/S0034425716303649, doi:10.1016/j.rse.2016.09.021.

[6] Wout. Verhoef and Nationaal Lucht- en Ruimtevaartlaboratorium (Netherlands). *Theory of radiative transfer models applied in optical remote sensing of vegetation canopies*. [publisher not identified], 1998. ISBN 9054858044. URL: https://library.wur.nl/WebQuery/wda/945481.

[7] Nastassia Vilfan, Christiaan van der Tol, Onno Muller, Uwe Rascher, and Wouter Verhoef. Fluspect-B: A model for leaf fluorescence, reflectance and transmittance spectra. *Remote Sens. Environ.*, 186:596–615, 2016. URL: http://dx.doi.org/10.1016/j.rse.2016.09.017, doi:10.1016/j.rse.2016.09.017.

[8] XINYOU YIN, JEREMY HARBINSON, and PAUL C. STRUIK. Mathematical review of literature to assess alternative electron transports and interphotosystem excitation partitioning of steady-state C3 photosynthesis under limiting light. *Plant, Cell Environ.*, 29(9):1771–1782, sep 2006. URL: http://doi.wiley.com/10.1111/j.1365-3040.2006.01554.x, doi:10.1111/j.1365-3040.2006.01554.x.

[9] Xinyou Yin and Paul C. Struik. Crop systems biology as an avenue to bridge applied crop science and fundamental plant biology. In *Proc. - 2012 IEEE 4th Int. Symp. Plant Growth Model. Simulation, Vis. Appl. PMA 2012*, 15–17. IEEE, oct 2012. URL: http://ieeexplore.ieee.org/document/6524806/, doi:10.1109/PMA.2012.6524806.

[10] C Van der Tol, J A Berry, P K E Campbell, and U Rascher. Models of fluorescence and photosynthesis for interpreting measurements of solar-induced chlorophyll fluorescence. *J. Geophys. Res. Biogeosciences*, 119(12):2312–2327, 2014.

[11] C. Van der Tol, W. Verhoef, J Timmermans, A Verhoef, and Z Su. An integrated model of soil-canopy spectral radiances, photosynthesis, fluorescence, temperature and energy balance. *Biogeosciences*, 6(12):3109–3129, dec 2009. URL: www.biogeosciences.net/6/3109/2009/, doi:10.5194/bg-6-3109-2009.

MATLAB MODULE INDEX

S

INDEX

T

V

Z