Package: rCTOOL (via r-universe)

July 5, 2026

Type Package

Title Soil Organic Carbon Turnover Modelling with 'C-TOOL'

Version 3.1.0

Description Provides an 'R' interface to the 'C-TOOL' soil carbon turnover model for simulating soil organic carbon dynamics in agricultural systems. The package supports the definition of carbon inputs, management schedules, soil parameters, and temperature forcing, and includes tools for scenario analysis and calibration of selected model parameters against observed soil organic carbon stocks. The 'C-TOOL' model and related modelling framework are described by Petersen et al. (2002) <doi:10.1016/S0304-3800(02)00034-0>, Petersen et al. (2005) <doi:10.1016/j.soilbio.2004.08.006>, Petersen et al. (2013) <doi:10.1016/j.jclepro.2013.03.007>, and Taghizadeh-Toosi et al. (2014) <doi:10.1016/j.ecolmodel.2014.08.016>. Further applications and developments are described by Taghizadeh-Toosi et al. (2016) <doi:10.1016/j.agsy.2016.03.004>, Keel et al. (2017) <doi:10.1111/ejss.12454>, Taghizadeh-Toosi et al. (2020) <doi:10.1007/s11104-020-04500-9>, and Taghizadeh-Toosi and Christensen (2021) <doi:10.1038/s41598-021-97744-z>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports data.table, utils

Suggests dplyr, ggplot2, knitr, rmarkdown, reshape2, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Evandro H. Figueiredo Moura da Silva [aut], João Serra [aut], Franca Giannini-Kurina [aut, cre]

Maintainer Franca Giannini-Kurina <francagk@agro.au.dk>

Repository https://francagiannini.r-universe.dev

Date/Publication 2026-07-04 08:50:02 UTC

RemoteUrl https://github.com/cran/rCTOOL

RemoteRef HEAD

RemoteSha 10cf1b8717ea3173a6979c25764af1598a22c747

Contents

.clean_monthly_allocations *.clean_monthly_allocations*

Description

Clean and harmonize monthly allocation inputs.

Usage

```
.clean_monthly_allocations(m_config)
```

Arguments

m_config	Management configuration list.

Details

Reads management configuration with monthly allocation vectors and returns a configuration containing 'plant_monthly_allocation'.

Depending on the user input: - if plant, grain and grass allocations are all provided, plant allocation is recalculated as grain plus grass - if only one allocation is provided, it is treated as plant allocation - if two allocations are provided, they must correspond to grain and grass; otherwise an error is returned

Value

A management configuration list containing 'plant_monthly_allocation'.

.decay *.decay*

Description

Calculate soil carbon loss by first-order decomposition.

Usage

```
.decay(CO_t, k, tempCoefficient)
```

Arguments

CO_t	Carbon amount in the pool at time t.
k	First-order decomposition rate constant.
tempCoefficient	Temperature response coefficient.

Value

Carbon change due to decomposition during the timestep.

`.soil_temp` *.soil_temp*

Description

.soil_temp

Usage

```
.soil_temp(depth, month, T_ave, Amplitude, th_diff)
```

Arguments

`depth`	Soil layer reference depth in cm. In the default C-TOOL configuration, the model operates with two fixed layers: topsoil (25 cm) and subsoil (100 cm).
`month`	Month index (1 to 12).
`T_ave`	Monthly mean air temperature.
`Amplitude`	Annual temperature amplitude used in the soil temperature equation.
`th_diff`	Thermal diffusivity.

Value

Soil temperature at the midpoint of the soil layer for the specified monthly timestep.

`.temp_coef` *.temp_coef*

Description

Calculate the temperature response coefficient for carbon decomposition.

Usage

```
.temp_coef(T_zt)
```

Arguments

`T_zt`	Soil temperature at depth z and time t.

Value

Temperature response coefficient for carbon decomposition.

basic_example *Basic rCTOOL example dataset*

Description

Example dataset used to demonstrate a basic rCTOOL simulation.

Usage

```
basic_example
```

Format

A data frame.

Source

Internal example dataset distributed with rCTOOL.

check_balance *check_balance*

Description

Check carbon balance consistency of a simulation.

Usage

```
check_balance(ctool_output, cin_config, s_config)
```

Arguments

ctool_output	Output from turnover model.
cin_config	Carbon input configuration.
s_config	Soil configuration.

Details

Computes the balance between initial soil carbon, cumulative carbon inputs, final carbon stocks and cumulative CO2 emissions. The balance should be zero or very close to zero.

Value

The input ‘ctool_output‘ data.frame, returned unchanged after checking carbon balance consistency.

`ctool_calibrate` *Calibrate C-TOOL using observed topsoil SOC*

Description

Calibrates two C-TOOL parameters against observed SOC stocks: ‘f_hum_top‘ and ‘k_hum‘.

Usage

```
ctool_calibrate(
  time_config,
  cinput_config,
  temperature_config,
  management_config,
  soil_config,
  observed,
  f_hum_top = c(min = 0.2, max = 0.6, by = 0.05),
  k_hum = c(min = 0.002, max = 0.004, by = 5e-04),
  cn_init = 10,
  years_col = "Year",
  obs_col = "SOC_obs",
  f_fom_top = 0.003,
  metric = "d_index",
  minimize = NULL,
  keep_simulations = FALSE,
  verbose = TRUE
)
```

Arguments

`time_config`	A C-TOOL time configuration object returned by [define_timeperiod()].
`cinput_config`	A C-TOOL carbon input configuration object returned by [define_Cinputs()].
`temperature_config`	A C-TOOL compatible temperature configuration used by [run_ctool()]. It must contain at least ‘month‘ and ‘Tavg‘. A single historical annual temperature amplitude is calculated internally from the monthly climatology of ‘Tavg‘.
`management_config`	A C-TOOL management configuration object returned by [management_config()].
`soil_config`	A C-TOOL soil configuration object returned by [soil_config()]. All parameters are preserved, except ‘f_hum_top‘, ‘k_hum‘, and the internally recalculated ‘f_rom_top‘.
`observed`	A data frame with observed SOC values. By default, it must contain columns ‘Year‘ and ‘SOC_obs‘.
`f_hum_top`	Named numeric vector defining the calibration range for ‘f_hum_top‘, with names ‘min‘, ‘max‘, and ‘by‘.

k_hum	Named numeric vector defining the calibration range for ‘k_hum‘, with names ‘min‘, ‘max‘, and ‘by‘.
cn_init	Initial C:N ratio used by [initialize_soil_pools()]. Default is ‘10‘.
years_col	Name of the year column in ‘observed‘. Default is ‘"Year"‘.
obs_col	Name of the observed SOC column in ‘observed‘. Default is ‘"SOC_obs"‘.
f_fom_top	Fixed topsoil FOM transfer fraction used to calculate ‘f_rom_top‘. Default is ‘0.003‘.
metric	Character. Metric used to select the best tested parameter set. Options are ‘"d_index"‘, ‘"RMSE"‘, ‘"R2"‘, ‘"MAE"‘, and ‘"Bias"‘. Default is ‘"d_index"‘. The legacy value ‘"d"‘ is also accepted and internally converted to ‘"d_index"‘.
minimize	Logical. Should the selected metric be minimized? If ‘NULL‘, sensible defaults are used: ‘FALSE‘ for ‘"d_index"‘ and ‘"R2"‘, and ‘TRUE‘ for ‘"RMSE"‘, ‘"MAE"‘, and absolute ‘"Bias"‘.
keep_simulations	Logical. If ‘TRUE‘, stores all calibrated simulation time series. Default is ‘FALSE‘.
verbose	Logical. If ‘TRUE‘, prints progress messages.

Details

This function assumes that all C-TOOL simulation inputs have already been prepared using the standard rCTOOL workflow. The only additional data required for calibration is a two-column data frame containing observed SOC stocks by year.

The observed data frame must contain, by default:

- ‘Year‘: observation year. - ‘SOC_obs‘: observed SOC stock.

For each tested value of ‘f_hum_top‘, ‘f_rom_top‘ is calculated internally as:

‘f_rom_top = 1 - f_hum_top - f_fom_top‘

Invalid combinations where ‘f_rom_top <= 0‘ are removed before model runs.

The function first evaluates the current C-TOOL parameter set supplied through ‘soil_config‘. It then evaluates the tested calibration grid. If the current C-TOOL parameter set performs as well as or better than the best tested calibration, the function recommends keeping the current parameters.

The main goodness-of-fit statistics returned by the function are RMSE, MAE, mean bias, R2, and the Willmott index of agreement, here reported as ‘d_index‘.

Value

An object of class ‘"ctool_calibration"‘, a list containing:

best_params Best tested calibration parameter set and its metrics.

recommended_params Recommended parameter set. This may be either the current C-TOOL parameter set or the best tested calibration.

recommendation Text explaining whether to keep current parameters or use the best tested calibration.

metrics Metrics for the current C-TOOL parameters and best tested calibration.

all_results Metrics for all tested parameter combinations.

observed Observed SOC data used for calibration.

default_simulation Simulation using the supplied soil configuration.

best_simulation Simulation using the best tested calibration.

recommended_simulation Simulation using the recommended parameter set.

parameter_grid Parameter grid used for calibration.

all_simulations Optional list of all tested calibrated simulations.

settings Calibration settings.

References

Willmott, C. J. (1981). On the validation of models. Physical Geography, 2(2), 184-194.

Examples

```
# Example workflow:
#
# observed <- data.frame(
#   Year = c(1923, 1932, 1942, 1950),
#   SOC_obs = c(54.2, 53.8, 52.1, 51.4)
# )
#
# calib <- ctool_calibrate(
#   time_config = time_cfg,
#   cinput_config = cin_cfg,
#   temperature_config = t_cfg,
#   management_config = m_cfg,
#   soil_config = soil_cfg,
#   observed = observed,
#   f_hum_top = c(min = 0.20, max = 0.60, by = 0.05),
#   k_hum = c(min = 0.0020, max = 0.0040, by = 0.0005)
# )
```

`ctool_calibration_metrics`

Calculate C-TOOL calibration metrics

Description

Calculates performance metrics between observed and simulated SOC values.

Usage

```
ctool_calibration_metrics(observed, simulated)
```

Arguments

observed	Numeric vector with observed values.
simulated	Numeric vector with simulated values.

Details

The returned ‘d_index‘ is the Willmott index of agreement.

Value

A data frame with ‘d_index‘, ‘RMSE‘, ‘R2‘, ‘Bias‘, ‘MAE‘, and ‘n‘.

References

Willmott, C. J. (1981). On the validation of models. Physical Geography, 2(2), 184-194.

Examples

```
ctool_calibration_metrics(
  observed = c(50, 52, 55),
  simulated = c(49, 53, 54)
)
```

---

define_Cinputs *define_Cinputs*

Description

Prepare annual carbon input (from plants and/or manure) configuration.

Usage

```
define_Cinputs(
  management_filepath = NULL,
  Cin_top = NULL,
  Cin_sub = NULL,
  Cin_man = NULL,
  time_config = NULL
)
```

Arguments

management_filepath	Either a filepath to a management template or a data.frame containing 'Cin_top', 'Cin_sub' and 'Cin_man'.
Cin_top	Annual carbon input from plant residues in the topsoil.
Cin_sub	Annual carbon input from plant residues in the subsoil.
Cin_man	Annual carbon input from manure.
time_config	Time configuration object returned by 'define_timeperiod()'

Value

A list containing 'Cin_top', 'Cin_sub' and 'Cin_man'.

Examples

```
time_config <- define_timeperiod(yr_start = 2006, yr_end = 2008)
define_Cinputs(
  Cin_top = c(2, 2, 2),
  Cin_sub = c(0.5, 0.5, 0.5),
  Cin_man = c(1, 1, 1),
  time_config = time_config
)
```

---

define_physical_boundaries *define_physical_boundaries*

Description

Check whether a parameter value lies within specified physical bounds.

Usage

```
define_physical_boundaries(value, min_limit, max_limit)
```

Arguments

value	parameter value
min_limit	Minimum allowed value.
max_limit	Maximum allowed value.

Value

The input value if it lies within the specified physical bounds.

Examples

```
define_physical_boundaries(0.3, 0, 1)
```

define_timeperiod *define_timeperiod*

Description

Create the monthly simulation time structure.

Usage

```
define_timeperiod(yr_start, yr_end)
```

Arguments

yr_start	Initial simulation year.
yr_end	Final nd simulation year.

Value

A list with two elements: 'timeperiod', a data.frame containing monthly timesteps, years and annual timestep identifiers; and 'steps', the total number of simulated months.

Examples

```
define_timeperiod(yr_start=2006, yr_end=2010)
```

export_management_template *export_management_template*

Description

Exports a template that can be used to provide management inputs for the model. The exported file includes monthly allocation variables and carbon input columns initialized with zeros, so they can be filled by the user before being read by the model.

Usage

```
export_management_template(yr_start, yr_end, filepath)
```

Arguments

yr_start	Initial simulation year.
yr_end	Final simulation year.
filepath	Filepath where management template will be exported.

Details

Users should keep only the columns relevant to their workflow and avoid double accounting among allocation variables. For example, when 'plant_monthly_allocation' is used, grain and grass allocation columns should not be used simultaneously.

If monthly carbon inputs are provided directly, additional allocation columns may not be needed. These consistency checks are not enforced automatically and must be handled by the user when preparing the input file.

Value

The exported template as a data.frame.

Examples

```
path <- tempfile(fileext = ".csv")
export_management_template(2006, 2010, path)
```

`FOM_sub_calculations` *FOM_sub_calculations*

Description

Calculate FOM decomposition and carbon fluxes in the subsoil layer.

Usage

```
FOM_sub_calculations(FOM_sub_t, month, t_avg, amplitude, s_config)
```

Arguments

`FOM_sub_t`	Numeric. Carbon content in FOM subsoil pool (Mg C ha-1).
`month`	Integer. Month index (1–12).
`t_avg`	Numeric. Monthly mean air temperature (°C).
`amplitude`	Numeric. Annual historical temperature amplitude used in the soil temperature function.
`s_config`	List. Soil configuration parameters

Value

A list with updated pool values and fluxes for FOM subsoil layer.

Examples

```
FOM_sub_calculations(
  FOM_sub_t = 10,
  month = 7,
  t_avg = 15,
  amplitude = 10,
  s_config = soil_config()
)
```

`FOM_top_calculations` *FOM_top_calculations*

Description

Calculate FOM decomposition and carbon fluxes in the topsoil layer.

Usage

```
FOM_top_calculations(
  FOM_top_t,
  month,
  t_avg = t_avg,
  amplitude = amplitude,
  s_config
)
```

Arguments

`FOM_top_t`	Numeric. Carbon content in FOM top pool (Mg C ha-1).
`month`	Integer. Month index (1–12).
`t_avg`	Numeric. Monthly mean air temperature (°C).
`amplitude`	Numeric. Annual historical temperature amplitude used in the soil temperature function.
`s_config`	List. Soil configuration parameters.

Value

A list with updated pool values and fluxes for FOM topsoil layer.

Examples

```
FOM_top_calculations(
  FOM_top_t = 10,
  month = 7,
  t_avg = 15,
  amplitude = 10,
  s_config = soil_config()
)
```

HUM_sub_calculations *HUM_sub_calculations*

Description

Calculate HUM decomposition and carbon fluxes in the subsoil layer.

Usage

```
HUM_sub_calculations(HUM_sub_t, month, t_avg, amplitude, s_config)
```

Arguments

HUM_sub_t	Numeric. Carbon content in the HUM subsoil pool (Mg C ha-1).
month	Integer. Month index (1-12).
t_avg	Numeric. Monthly mean air temperature (degrees C).
amplitude	Numeric. Annual historical temperature amplitude used in the soil temperature function.
s_config	List. Soil configuration parameters.

Value

A list with updated HUM subsoil pool values and associated fluxes.

Examples

```
HUM_sub_calculations(
  HUM_sub_t = 20,
  month = 7,
  t_avg = 15,
  amplitude = 10,
  s_config = soil_config()
)
```

HUM_top_calculations *HUM_top_calculations*

Description

Calculate HUM decomposition and carbon fluxes in the topsoil layer.

Usage

```
HUM_top_calculations(
  HUM_top_t,
  month,
  t_avg = t_avg,
  amplitude = amplitude,
  s_config
)
```

Arguments

HUM_top_t	Numeric. Carbon content in the HUM topsoil pool (Mg C ha-1).
month	Integer. Month index (1-12).
t_avg	Numeric. Monthly mean air temperature (degrees C).
amplitude	Numeric. Annual historical temperature amplitude used in the soil temperature function.
s_config	List. Soil configuration parameters.

Value

A list with updated HUM topsoil pool values and associated fluxes.

Examples

```
HUM_top_calculations(
  HUM_top_t = 20,
  month = 7,
  t_avg = 15,
  amplitude = 10,
  s_config = soil_config()
)
```

---

initialize_soil_pools *initialize_soil_pools*

Description

Initialize topsoil and subsoil carbon pools.

Usage

```
initialize_soil_pools(cn, soil_config)
```

Arguments

cn	Soil carbon:nitrogen ratio.
soil_config	Soil configuration list.

Value

A list containing initialized topsoil and subsoil pools.

Examples

```
s_config <- soil_config()
initialize_soil_pools(cn=15, soil_config = s_config)
```

`management_config` *management_config*

Description

Prepares management configuration for monthly carbon input allocation.

Usage

```
management_config(
  management_filepath = NULL,
  plant_monthly_allocation = NULL,
  grain_monthly_allocation = NULL,
  grass_monthly_allocation = NULL,
  manure_monthly_allocation = NULL,
  f_man_humification = 0.12
)
```

Arguments

`management_filepath`
: Either a filepath to a management template or a data.frame containing management allocation variables.

`plant_monthly_allocation`
: Monthly distribution of plant carbon inputs.

`grain_monthly_allocation`
: Monthly distribution of grain carbon inputs.

`grass_monthly_allocation`
: Monthly distribution of grass carbon inputs.

`manure_monthly_allocation`
: Monthly distribution of manure carbon inputs.

`f_man_humification`
: Fraction of manure already humidified.

Details

This function can be used in two ways: 1. from a management template file or data.frame, such as one exported with ‘export_management_template()‘ 2. by directly providing fixed monthly allocation vectors of length 12.

In the first approach, the user can specify monthly allocation fractions directly in the input file. When no crop rotation is considered, ‘plant_monthly_allocation‘ should be used. When crop rotation is considered, grain and grass allocation fractions can be specified separately.

In the second approach, the user can directly provide monthly allocation vectors of length 12.

Value

A list containing management allocation settings.

Examples

```
management_config(
f_man_humification=0.192,
plant_monthly_allocation = c(0,0,0,.08,.12,.16,.64,0,0,0,0,0),
 manure_monthly_allocation = c(0,0,1,0,0,0,0,0,0,0,0,0)
 )
```

ROM_sub_calculations *ROM_sub_calculations*

Description

Calculate ROM decomposition and carbon fluxes in the subsoil layer.

Usage

```
ROM_sub_calculations(ROM_sub_t, month, t_avg, amplitude, s_config)
```

Arguments

ROM_sub_t	Numeric. Carbon content in the ROM subsoil pool (Mg C ha-1).
month	Integer. Month index (1-12).
t_avg	Numeric. Monthly mean air temperature (degrees C).
amplitude	Numeric. Annual historical temperature amplitude used in the soil temperature function.
s_config	List. Soil configuration parameters.

Value

A list with updated ROM subsoil pool values and associated fluxes.

Examples

```
ROM_sub_calculations(
  ROM_sub_t = 10,
  month = 7,
  t_avg = 15,
  amplitude = 10,
  s_config = soil_config()
)
```

`ROM_top_calculations` *ROM_top_calculations*

Description

Calculate ROM decomposition and carbon fluxes in the topsoil layer.

Usage

```
ROM_top_calculations(ROM_top_t, month, t_avg, amplitude, s_config)
```

Arguments

`ROM_top_t`	Numeric. Carbon content in the ROM topsoil pool (Mg C ha-1).
`month`	Integer. Month index (1-12).
`t_avg`	Numeric. Monthly mean air temperature (degrees C).
`amplitude`	Numeric. Annual historical temperature amplitude used in the soil temperature function.
`s_config`	List. Soil configuration parameters.

Value

A list with updated ROM topsoil pool values and associated fluxes.

Examples

```
ROM_top_calculations(
  ROM_top_t = 10,
  month = 7,
  t_avg = 15,
  amplitude = 10,
  s_config = soil_config()
)
```

run_ctool *run_ctool*

Description

Run C-TOOL over the full simulation period.

Usage

```
run_ctool(
  time_config,
  cin_config,
  m_config,
  t_config,
  s_config,
  soil_pools,
  verbose = FALSE
)
```

Arguments

time_config	Time configuration object created by ‘define_timeperiod()‘.
cin_config	Carbon input configuration.
m_config	Management configuration with monthly allocation patterns.
t_config	Monthly temperature configuration containing at least ‘month‘ and ‘Tavg‘.
s_config	Soil parameter configuration.
soil_pools	Initial soil pool configuration.
verbose	Logical; if ‘TRUE‘, run balance checking.

Details

Iteratively applies ‘turnover()‘ over all timesteps defined in ‘time_config‘ and returns monthly carbon pool sizes, soil carbon stocks, transport fluxes and CO2 emissions.

The temperature configuration must provide monthly mean air temperature. A single historical annual temperature amplitude is calculated internally from the monthly climatology of ‘Tavg‘ and is used in the soil temperature response function.

Value

A data.frame containing the monthly simulation output across the full simulation period.

scenario *rCTOOL scenario example dataset*

Description

Example dataset used to demonstrate scenario simulations in rCTOOL.

Usage

```
scenario
```

Format

A data frame.

Source

Internal example dataset distributed with rCTOOL.

scenario_temperature *rCTOOL scenario temperature dataset*

Description

Monthly temperature dataset used in rCTOOL scenario simulations.

Usage

```
scenario_temperature
```

Format

A data frame.

Source

Internal example dataset distributed with rCTOOL.

set_monthly_temperature_data

set_monthly_temperature_data

Description

Prepare monthly temperature input data for rCTOOL.

Usage

```
set_monthly_temperature_data(file = NULL, Tavg = NULL)
```

Arguments

file	Either a filepath to a csv file or a data.frame containing monthly temperature data.
Tavg	Numeric vector of average monthly temperature values.

Details

Monthly temperature data can be provided in two ways: 1. directly through the argument 'file', as a filepath or data.frame containing monthly temperature data 2. directly through the argument 'Tavg', as a numeric vector of average monthly temperature values

Value

A data.frame containing monthly temperature input data. The output always contains 'Tavg'.

Examples

```
df_temp <- data.frame(Tavg = c(5, 6, 7, 8, 9, 15, 14, 11, 10, 9, 5, 1))
set_monthly_temperature_data(file = df_temp)

set_monthly_temperature_data(
  Tavg = c(5, 6, 7, 8, 9, 15, 14, 11, 10, 9, 5, 1)
)
```

set_monthly_temperature_data_historical_amplitude

set_monthly_temperature_data_historical_amplitude

Description

Prepare monthly temperature input with a single historical amplitude.

Usage

```
set_monthly_temperature_data_historical_amplitude(file = NULL, Tavg = NULL)
```

Arguments

file	Either a filepath to a csv file or a data.frame containing monthly temperature data.
Tavg	Numeric vector of average monthly temperature values.

Details

Monthly 'Tavg' data can be provided either as a filepath/data.frame through 'file' or directly as a numeric vector through 'Tavg'.

A single historical amplitude is calculated from the full monthly 'Tavg' series as:

$$(max(Tavg) - min(Tavg))/2$$

This historical amplitude is then assigned to the full series in the 'Amplitude' column.

Value

A data.frame containing 'Tavg' and a single historical 'Amplitude' repeated over the full series.

Examples

```
set_monthly_temperature_data_historical_amplitude(
  Tavg = c(1, 2, 4, 7, 11, 15, 17, 16, 13, 9, 5, 2)
)
```

---

soil_config *soil_config*

Description

Prepare soil configuration parameters.

Usage

```
soil_config(
  Csoil_init = 70.4,
  f_hum_top = 0.48,
  f_rom_top = 0.49,
  f_hum_sub = 0.312,
  f_rom_sub = 0.6847,
  Cproptop = 0.47,
  clay_top = 0.1,
  clay_sub = 0.15,
```

```
  phi = 0.035,
  f_co2 = 0.628,
  f_romi = 0.012,
  k_fom = 0.12,
  k_hum = 0.0028,
  k_rom = 3.85e-05,
  ftr = 0.0025,
  temp_method = "rctool",
  temp_amplitude_hist = NA_real_,
  temp_offset = 0,
  temp_th_diff = 3.5e-07
)
```

Arguments

`Csoil_init`	Initial carbon stock at depth 1m (Mg C ha-1).
`f_hum_top`	Initial HUM fraction in the topsoil layer.
`f_rom_top`	Initial ROM fraction in the topsoil layer.
`f_hum_sub`	Initial HUM fraction in the bottom layer.
`f_rom_sub`	initial ROM fraction in the bottom layer.
`Cproptop`	Proportion of the total carbon allocated to the topsoil.
`clay_top`	Clay fraction in the top soil.
`clay_sub`	Clay fraction in the subsoil.
`phi`	Legacy diffusion parameter used in the original rCTOOL temperature formulation.
`f_co2`	Respiration fraction.
`f_romi`	Romification fraction.
`k_fom`	FOM decomposition rate constant.
`k_hum`	HUM decomposition rate constant.
`k_rom`	ROM decomposition rate constant.
`ftr`	Vertical transport rate.
`temp_method`	Temperature method identifier.
`temp_amplitude_hist`	Optional historical annual amplitude to be used in the soil temperature calculation.
`temp_offset`	Phase offset used in the soil temperature calculation.
`temp_th_diff`	Thermal diffusivity used in the physical soil temperature formulation.

Details

Exports a template that can be used to provide management inputs for the model. The exported file includes monthly allocation variables and carbon input columns initialized with zeros, so they can be filled by the user before being read by the model.

Users should keep only the columns relevant to their workflow and avoid double accounting among allocation variables. For example, when 'plant_monthly_allocation' is used, grain and grass allocation columns should not be used simultaneously.

If monthly carbon inputs are provided directly, additional allocation columns may not be needed. These consistency checks are not enforced automatically and must be handled by the user when preparing the input file.

Value

A list containing soil configuration parameters.

Examples

```
soil_config(Csoil_init=72, f_hum_top=0.5, clay_sub = 0.35, clay_top=0.25)
```

`soil_pool_decomposition` *soil_pool_decomposition*

Description

Calculate temperature-modified decomposition of a soil carbon pool.

Usage

```
soil_pool_decomposition(
  soil_pool,
  k,
  soil_depth,
  month,
  t_avg,
  amplitude,
  s_config
)
```

Arguments

`soil_pool`	Carbon amount in the soil pool.
`k`	First-order decomposition rate constant.
`soil_depth`	Soil layer thickness in cm. In the default C-TOOL configuration, the model operates with two fixed layers: topsoil (25 cm) and subsoil (100 cm).
`month`	Month index from 1 to 12.
`t_avg`	Monthly mean air temperature.
`amplitude`	Historical annual temperature amplitude previously calculate from the monthly temperature series.
`s_config`	Soil parameter configuration.

Value

Carbon change in the pool due to temperature-modified first-order decomposition.

`turnover` *turnover*

Description

Perform one monthly turnover step of C-TOOL.

Usage

```
turnover(
  timestep,
  time_config,
  cin_config,
  m_config,
  t_config,
  s_config,
  out,
  amplitude_hist
)
```

Arguments

`timestep`	Integer index of the simulation timestep.
`time_config`	Time configuration object created by ‘define_timeperiod()‘.
`cin_config`	Carbon input configuration.
`m_config`	Management configuration with monthly allocation patterns.
`t_config`	Monthly temperature configuration containing at least ‘month‘ and ‘Tavg‘.
`s_config`	Soil parameter configuration.
`out`	Data frame containing pool values from the previous timestep.
`amplitude_hist`	Numeric. Historical annual air temperature amplitude calculated from the monthly climatology of ‘Tavg‘.

Details

Updates FOM, HUM and ROM pools for one monthly timestep using carbon inputs, management allocations, monthly mean air temperature, soil parameters, and the previous pool state.

Value

A data.frame containing updated monthly pool sizes, carbon stocks, transport fluxes and CO2 emissions for the current timestep.

turnover_beta *turnover_beta*

Description

Legacy monthly turnover routine retained for backward compatibility.

Usage

```
turnover_beta(i)
```

Arguments

i	Integer index of the simulation timestep.

Details

This function updates FOM, HUM and ROM pools for one monthly timestep using the historical C-TOOL workflow. It relies on objects defined in the calling environment and is retained only for backward compatibility.

Value

A one-row data.frame containing updated pool sizes, carbon stocks, transport fluxes and CO2 emissions for the current timestep.

update_monthly_FOM_sub *update_monthly_FOM_sub*

Description

update_monthly_FOM_sub

Usage

```
update_monthly_FOM_sub(
  FOM_sub_t1,
  FOM_transport,
  C_in_plant_sub,
  month,
  m_config
)
```

Arguments

FOM_sub_t1	FOM content in the subsoil layer from the previous timestep.
FOM_transport	FOM transported from the topsoil.
C_in_plant_sub	Plant carbon input to the subsoil.
month	Month index from 1 to 12.
m_config	Management configuration list.

Value

Updated FOM content in the subsoil layer after plant inputs and transport from the topsoil.

---

update_monthly_FOM_top

update_monthly_FOM_top

Description

update_monthly_FOM_top

Usage

```
update_monthly_FOM_top(FOM_top_t1, Cin_plant_top, Cin_manure, month, m_config)
```

Arguments

FOM_top_t1	FOM content in the topsoil layer from the previous timestep.
Cin_plant_top	Plant carbon input to the topsoil.
Cin_manure	Manure carbon input.
month	Month index from 1 to 12.
m_config	Management configuration list.

Value

Updated FOM content in the topsoil layer after monthly carbon inputs.

update_monthly_HUM_sub *update_monthly_HUM_sub*

Description

update_monthly_HUM_sub

Usage

```
update_monthly_HUM_sub(HUM_sub_t1, HUM_transport, FOM_humified_sub)
```

Arguments

HUM_sub_t1	HUM content in the subsoil layer from the previous timestep.
HUM_transport	HUM transported from the topsoil.
FOM_humified_sub	Humified FOM added to the subsoil HUM pool.

Value

Updated HUM content in the subsoil layer after transport and humified FOM additions.

---

update_monthly_HUM_top *update_monthly_HUM_top*

Description

update_monthly_HUM_top

Usage

```
update_monthly_HUM_top(HUM_top_t1, C_in_man, FOM_humified_top, month, m_config)
```

Arguments

HUM_top_t1	HUM content in the topsoil layer from the previous timestep.
C_in_man	Manure carbon input.
FOM_humified_top	Humified FOM added to the topsoil HUM pool.
month	Month index from 1 to 12.
m_config	Management configuration list.

Value

Updated HUM content in the topsoil layer after manure inputs and humified FOM additions.

update_monthly_ROM_sub *update_monthly_ROM_sub*

Description

update_monthly_ROM_sub

Usage

```
update_monthly_ROM_sub(ROM_sub_t1, HUM_romified_sub, ROM_transport)
```

Arguments

ROM_sub_t1	ROM content in the subsoil layer from the previous timestep.
HUM_romified_sub	Romified HUM added to the subsoil ROM pool.
ROM_transport	ROM transported from the topsoil.

Value

Updated ROM content in the subsoil layer after romified HUM additions and transport from the topsoil.

update_monthly_ROM_top *update_monthly_ROM_top*

Description

update_monthly_ROM_top

Usage

```
update_monthly_ROM_top(ROM_top_t1, HUM_romified_top)
```

Arguments

ROM_top_t1	ROM content in the topsoil layer from the previous timestep.
HUM_romified_top	Romified HUM added to the topsoil ROM pool.

Value

Updated ROM content in the topsoil layer after romified HUM additions.

Index