Microclimf: fast modelling of microclimate across real landscapes in R

Ilya M. D. Maclean[1]

[1]Environment and Sustainability Institute, University of Exeter Penryn Campus, Penryn TR10 9FE, United Kingdom. i.m.d.macleam@exeter.ac.uk

Abstract

1. Many ecological studies require climate data, but readily available datasets are poor surrogates for the conditions that organisms experience in nature. Understanding the climatic conditions experienced by organisms requires modelling microclimate rather than relying on coarse, station-based climate data.
2. I present microclimf, a mechanistic microclimate model designed for computationally efficient, gridded estimation of microclimate, within, and below vegetation canopies. The model is written in C++ with an R front end and requires only readily available spatial datasets as inputs. It incorporates a simplified Lagrangian canopy model, an optional snow model, and routines for efficient large-area processing at user-defined spatial and temporal resolutions. Outputs include temperature, humidity, wind speed and radiation fluxes.
3. Validation across diverse environments—including boreal and tropical forests—showed strong agreement with *in-situ* temperature measurements (RMSE 0.69–2.9 °C), demonstrating the model's utility for ecological applications requiring fine-scale climatic data.
4. The package addresses the need for improved estimation of regional and landscape--scale predictions of the conditions experienced by organisms, thereby facilitating more robust understanding and prediction of species responses to climatic changes.

Key words: biodiversity, biophysical ecology, climate change, modelling, species distribution model

Introduction

Many ecological studies require climate data, but readily available datasets are poor surrogates for the conditions that organisms experience in nature (Bramer et al., 2018; Potter et al, 2013). Global datasets are normally derived from, or representative of conditions measured by weather stations. However, organisms experience microclimatic conditions that sometimes differ substantially from those measured by weather stations, which shade direct sunlight and are located in open areas 1-2m above ground. It is thus increasingly recognised that estimates of microclimatic conditions are useful for the study of relationships between organisms and climate (De Frenne et al., 2021; Kemppinen et al., 2023; Lembrechts et al., 2019, Potter et al., 2013).

The means of deriving estimates of microclimate using biophysical principles has a long and well-established history: the basic equations for doing so were first developed for numerical weather prediction over a century ago (Richardson, 1922). By the 1950s, universal theory explaining near-ground temperature and vapour profiles were developed (Monin & Obukhov, 1954) and still form the basis of microclimatology today. Over the next few decades, these principles were applied widely in agricultural contexts, as the methods for quantifying energy and water balance that underpin biophysical microclimate modelling are the same as those used for calculating evapotranspiration (Monteith, 1965; Penman, 1948). They are also similar to those applied in land-surface models and have therefore been the subject of much study (Best et al., 2011; Chen et al., 2016, Flerchinger et al., 2015; Lawrence et al., 2019; Ogée et al., 2003).

Capitalising on advances in remote-sensing, ecologists have sought to quantify how real-world vegetation and terrain features influence microclimate conditions at increasingly high spatial resolutions (Lembrechts & Lenoir, 2020). Ecologically grounded attempts to model microclimate have generally proceeded using one of two approaches. On the one hand, driven by the need to model microclimate over large spatial extents, and drawing upon techniques with which ecologists are familiar, statistical approaches have been used (e.g. Haesen et al., 2021; Lembrechts et al., 2022). Here, the strategy has been to relate *in-situ* measurements climate to landscape and terrain features without explicitly considering the underlying processes. A significant advantage of so doing is that there is no need to run models at high temporal resolution. If monthly or yearly averaged estimates of microclimate are needed, then these variables can be related to directly to environmental predictors without the intermediary step of deriving hourly or daily microclimate, affording substantially computational advantages. However, there are also significant drawbacks. The most important determinant of microclimatic variation is solar radiation and its interaction with terrain and vegetation features (Campbell & Norman, 2012; Gates, 2012). These interactions are influenced strongly both by the position of the sun and by cloud cover, neither of which are constant in space nor time. Thus, statistical relationships established at one location or time-period, cannot be used to predict microclimate reliably at others.

In contrast, mechanistic models seek to estimate microclimatic conditions based explicitly on underlying processes (Briscoe et al., 2023; Kearney & Porter, 2009). Temperature and humidity are calculated from energy fluxes, which are modified by vegetation and terrain in universally predictable ways. Several such models have been developed for ecological applications. One of the earliest (Porter et al., 1973) has now been generalized and incorporated into the R package NicheMapR (Kearney & Porter, 2017). The package includes sophisticated routines for modelling of heat and mass exchange between organisms and their environments

and permits prediction of hourly above- and below-ground conditions from meteorological, terrain, vegetation and soil data. However, NicheMapR is a point model that produces time-series of microclimatic conditions for a specified location. If gridded microclimate outputs are required, they must be derived separately for each grid cell, which is computationally expensive and necessitates that predictor variables are provided as inputs rather than computed automatically from spatial datasets. To enable gridded microclimate outputs, Maclean et al. (2017) developed a series of functions for such to extend the model of (Bennie, et al. 2008), released as an R package `microclima` (Maclean et al., 2019). The package computes terrain variables directly from digital elevation datasets but required calibration with local observations of temperature and does not readily account for spatial variation in vegetation cover. The two packages were subsequently integrated (Kearney et al., 2020), but both treat vegetation as a homogeneous layer of phytomass without vertical structure. Conditions below canopy are simulated simply by applying a shade factor rather than by explicitly considering below-canopy microclimatic processes. To overcome this issue, and drawing on earlier work by Goudriaan (1977), the microclimc package was developed (Maclean & Klinges, 2021). Like NicheMapR, microclimc is a point model, and therefore computationally slow if applied over numerous grid cells, an issue compounded by the need to sub-divide the canopy into multiple layers to perform calculations. A further limitation is its adoption of a flux-gradient approach to modelling within canopy heat and exchange processes, whereas Langrangian models have been shown to provide more realistic representations of canopy heat and vapour transfer (Raupach 1989a;b).

What is hitherto lacking therefore is a model that (i) is driven with readily available spatial datasets so that it can be applied easily to variety of circumstances; (ii) provides gridded outputs at user-specified time-increments and is computationally efficient when applied over large areas; and (iii) models within canopy microclimate: the environment in which the majority of terrestrial organisms reside. Here we present the microclimf model that achieves this. Essential inputs are limited to those that can be readily derived from publicly available spatial datasets, and while the package permits flexibility in the way that model inputs are provided, those that are less likely to be known can instead be estimated by specifying a habitat or soil type. It is written in C++ with an R front-end.

Model description

In the main text I provide an accessible description of the main features of the model ─ its workings and how to run it. An expanded mathematical description and a tutorial vignette are included in Supporting Information.

Above canopy

Temperature and humidity above canopy are derived using the methods detailed in most standard textbooks on the subject (e.g. Campbell & Norman, 2012; Gates, 2012; Monteith & Unsworth, 2013), namely using Monin-Obukhov Similarity Theory (MOST; Foken, 2006). Thus, the temperature of a vegetated surface is derived by treating the canopy as a single vertically homogeneous layer of phytomass and computing the energy balance of this layer. Incoming radiation is either absorbed by the surface or reflected upwards. The surface also emits thermal radiation and the ground surface below the vegetation either stores or releases heat. Remaining energy is then partitioned between latent or sensible heat. Each of these terms has a temperature-dependence, and to derive the temperature of the canopy surface, the energy balance is equated to zero and solved for temperature using the Penman-Monteith equation

(Monteith 1965, Penman 1948). Thus, during the day, downward radiation heats the vegetated surface causing increases in emitted thermal radiation, evapotranspirative cooling and sensible heat loss to the air. Some of the energy is also used to heat the underlying ground surface. At night, when the net radiation balance is likely to be negative, the vegetated surface is assumed to cool. Warming or cooling is assumed to continues until the incoming radiation is balanced by outgoing sources and the temperature is derived by assuming these fluxes reach steady-state. This is not strictly true of the ground surface, which accumulates or dissipates heat over longer time-scales. However, by computing the rate of energy exchange with the underlying ground surface, the principles of energy balance can still be applied. Once the temperature of the vegetation surface has been derived, the temperature and vapour pressure at any given height above canopy follow a predictable log-linear height profile thereby enabling derivation of air temperature or relative humidity at any user-specified height above canopy. Each component of the energy balance is now described in more detail.

Absorbed radiation is given by the sum of its shortwave and longwave components (i.e. solar radiation and thermal radiation emitted downwards from the sky). The fractions of total incoming shortwave and longwave radiation that are absorbed are determined by the albedo and emissivity of the surface respectively. Emissivity is assumed constant and is set at 0.97. Albedo is a function of ground and vegetation reflectance, which are provided as user inputs, but is assumed to vary spatially as a function of canopy cover and temporally as a function of the direct to diffuse fraction of radiation and by inclination angle of leaves and the ground surface relative to the position of the sun. It is calculated using the Dickenson-Sellers two stream radiation model (Sellers, 1985), but with minor modifications proposed by Yuan et al (2017) and to accommodate terrain shading and inclined ground surfaces below the canopy. Emitted radiation is calculated using the Stefan-Boltzmann equation and thus scales with temperature of the surface measured in Kelvin to the power of 4.

The sensible heat flux is assumed to scale linearly with the difference in temperature between vegetated surface and air above it at reference height (provided as a user input). The gradient of this linear relationship is a measure of the thermal coupling of the surface to the atmosphere, itself varying as a function of both the structure of the vegetation and wind speed. Wind speed is provided as a user input but adjusted for terrain sheltering following Ryan (1977). The structural effects of vegetation manifest through their effects on the height above ground at which the wind profile extrapolates to zero and via their influence on surface friction and hence the shape of the vertical wind profile, which in turn affects how efficiently heat or vapour are transferred. The wind profile is calculated from vegetation height and the plant area index of vegetation following Raupach (1994). Additionally, corrections for atmospheric stability are made following Businger et al. (1971).

The latent heat exchange flux is assumed to scale linearly with the difference in vapour pressure between vegetated surface and air above it. The effective vapour pressure of the surface is inferred from surface temperature using Tetens equation (see e.g. Campbell & Norman, 2012). As with sensible heat, the vapour coupling of the surface to the atmosphere is function of both the structure of the vegetation and wind speed, but with an additional resistance to vapour loss caused by stomatal impedance. Bulk surface stomatal resistance is calculated based on a user-specified minimum stomatal resistance, representing conditions under optimal photosynthesis and ample light. Values for the major global vegetation types are provided by Körner (1995). Actual stomatal resistance is then reduced as a function of radiation following the approach of Kelliher et al. (1995).

Following De Vries & Van Wijk (1963) and Campbell & Norman (2012), it is assumed soil surface temperature follow approximately sinusoidal diurnal and annual cycles. The rate of heat storage and release by the soil is then derived from the amplitude of the diurnal or annual soil surface temperature cycle, and from the thermal conductivity and specific heat capacity of the soil. The latter are determined from soil water content and physical properties derived from soil type. Soil water content is either provided as a user input or is calculating using a simple two-layer soil model, applied for a point location at the centre of the study area and then spatially distributed within each time step using a topographic wetness index. The diurnal and annual cycles in ground surface temperature are calculated from the energy balance of the soil surface using the Penman-Monteith equation, but with absorbed and emitted radiation adjusted for canopy shading.

Because ground heat flux depends on ground surface temperature and vis-versa and owing also to interdependencies between the atmospheric stability and sensible heat exchange, the model must be run iteratively until convergence. This becomes computationally inefficient if applied to multiple grid cells of a study area as sometimes more than 50 iterations are required to achieve convergence. However, it can be shown numerically (see Supporting Information) that the offset error associated with deriving surface temperatures ignoring ground heat flux and diabatic correction, scales approximately linearly with the temperature difference between the surface the air above it computed without iteration. Thus, by solving the energy balance equations iteratively for one location, the temperature of other locations can be derived without iteration. The microclimf package thus implements a two-stage process. First, a point model is run iteratively for a hypothetical location within the study area with a flat ground surface and mean (or modal for categorical data) vegetation and soil properties. Outputs from this model are used to derive the temperatures of other grid cells within the study area without iteration. This affords the additional advantage that the grid model can then be run for specified time periods only: e.g. the day of the month with the warmest or median temperature rather than hourly over an entire year.

The model assumes that the weather variables provided as a user input represent conditions above canopy. This is problematic if seeking to model the exchange above tall canopies as weather datasets available for user input normally represent conditions 1.5-2m above ground. To circumvent this problem the microclimate model is used to height-adjust the weather data This is achieved by making the assumption that weather data are derived from a weather station located in a flat open area with short vegetation in accordance with World Meteorological Organization guidelines (WMO, 2021) and then extrapolating the log-linear height profiles of wind, temperature and vapour pressure upward.

Below ground

Relative to ground surface temperatures, both annual and diurnal temperature cycles are attenuated and occur later in the day or year with increasing depth. A simple empirical approach, in which the temperature at any given time or depth is determined from the rolling mean of ground surface temperatures over the last *n* hours, replicates well the results from the more complex multi-layer soil model in NicheMapR and is therefore used (Fig. S3). The number of hours over which the rolling mean is calculated is contingent on depth and soil properties. When running the grid model for selected hours only, outputs from the point model are used to construct the diurnal cycle for the grid model.

Below canopy

It is now widely recognised that the process that transfer heat and vapour below canopy cannot be predicted using conventional flux gradient theory inherent in the MOST approach (e.g. Bonan, et al. 2021). To replace MOST in this context, Raupach (1989a;b) developed an analytic Lagrangian theory, which assumes that temperature and humidity are determined from the concentration of heat and vapour emanating from foliage within the canopy. In this 'localized near-field' theory (LNZT), the mean concentrations of heat or vapour are expressed as the sum of a diffusive far-field contribution that obeys MOST, and a non-diffusive near-field contribution, which is determined from local sources by assuming the turbulence to be locally homogeneous. The diffusive far field component is essentially the sum of heat or vapour emanating from individual canopy elements weighted by wind speed-dependent convective conductance. The non-diffuse near-field contribution is determined primarily from local sources, and thus contingent primarily on foliage density at the location of interest.

LNZT models thus require that the canopy is divided into numerous layers each with known foliage density. Since this is both computationally intensive and relies on explicit knowledge of canopy structure, in microclimf, the following simplifying assumptions are made. First it is assumed that the sum of sensible and latent fluxes from individual canopy elements are equivalent to the fluxes for the entire canopy. This is reasonable and forms the basis of much-used MOST models, though see Bonan et al. (2021) for a detailed discussion of this point. Second, that the shape of the vertical heat and vapour transfer functions follows the original form proposed by Raupach (1989b) in the point model, but in the grid model can be approximated by a similar function that can be solved analytically without specific knowledge of the distribution of foliage and is therefore relative insensitive to canopy structure. This is a less reasonable assumption, but numerical analysis (Fig. S6) shows that doing so has only a minor influence on temperature and humidity. Lastly, it is assumed scaling of heat or vapour transfer with wind speed is consistent at the top of the canopy with what would be predicted by MOST models. This fails to account for transfer being partially contingent on periodic flushing of warm or moist air from the understory of the canopy by stronger wind gusts. It is nevertheless an implicant assumption of MOST models and is partially handled by adopting the unifying approach proposed by Harman & Finnigan (2007) who also provide detailed discussion of this point. Making these assumptions it is possible to derive and humidity at any given height below canopy as a function of the temperature and effective humidity of the ground surface and at the top of the canopy.

The mathematical formulation of both the LNZT model and the simplifying assumptions that are made are quite convoluted but described in detail in Supporting Information. The general pattern is that during the day, when the net radiation budget is positive, air temperature and vapour pressure gradually decrease with height in the top half of the canopy because the air is warmed and wetted by the canopy, but near the top, are closely coupled to cooler and drier air above canopy. In the third of the of the canopy temperatures increase with height, because near the ground, the air becomes more closely coupled to the ground surface, which usually cooler owing to the effects of canopy shading. At night, when the net radiation budget is negative, these temperature and humidity profiles are inverted with respect to height, but typically less pronounced.

Snow cover

When snow is present, the following adjustments are made to the energy balance model. To calculate absorbed radiation, the albedo of the canopy is replaced by snow albedo, which is assumed to vary with snow age. Snow is assumed to absorb radiation isotropically and is therefore not contingent on the angle of the sun relative to the surface. Foliage density, canopy height and leaf transmittance are adjusted to accommodate the presence of snow. When calculating the latent heat flux from the canopy, the surface is assumed to be freely evaporating or sublimating and therefore not contingent on stomatal resistance. When calculating the ground heat flux, snow surface rather than ground surface temperature is used. If the height for which model outputs are required lies below the snowpack but above the ground surface, microclimate is calculated as for below ground, but with the thermal properties of the ground replaced by the thermal properties for snow. If the requested output height lies above the snowpack, the effective transfer distance for heat and vapour is adjusted to account for snow depth. For example, if outputs are requested at 5 cm above ground and the snowpack is 4.5 cm deep, the model assumes a 5 mm height above the snow surface, with air temperatures closely coupled to the snowpack temperature

Snow depth is modelled using a mass balance model in which the change in mass per unit time is computed from snow-water equivalent precipitation, sublimation or fusion, and if air temperature is above 0°C, then from rain melt. Sublimation and fusion are computed from the latent heat term of the energy balance equation. Following Anderson (2006) and Kearney (2020) rain melt is assumed to scale linearly with the rate of rainfall and by the amount by which air temperatures is above freezing. Separate mass balances are computed for the ground and canopy, with snow interception by the canopy computed following Hedstrom and Pomeroy (1998). The snow mass budget is converted to snow depth using estimates of snow density derived from snow, age depth and climate classes following Sturm et al. (2010) and Kearney (2020). Snow is also redistributed spatially at regular intervals to simulate snow accumulation in hollows.

By default, users must provide an initial snow depth, and the snow model is run in hourly time increments for every grid cell and over the entire time-period for which input weather data are provided. This is necessary as snow depth at given point in time is contingent on snow depth in earlier time-steps. However, users also have the option to run a computationally faster approximation of the snow model if microclimate outputs for short, selected time-periods only are required. Here the snow mass balance is computed at hourly intervals for single location only. Local deviations from this are then calculated from spatially variable melt-factors inferred from landscape properties. Details and comparisons between the two approaches are shown in Supporting Information.

Model validation

Though hitherto unpublished, the model has already been used extensively and tested for a variety of applications. A summary overview is provided here, with further details given in Supporting Information. To validate `microclimf` regionally (see Kolstela et al., 2024), 150 TOMST loggers were deployed across three Finnish landscapes from May to August 2020, with temperatures recorded at 15 cm above ground. Loggers were stratified by terrain and canopy cover, capturing a wide range of slopes, aspects, and vegetation types. Modelled and measured temperatures showed reasonable agreement (RMSE 2.9 °C overall; 2.2–3.2 °C across sites), with smaller discrepancies under dense canopies. Global validation using data from 70 tropical forest sites (Trew et al. 2024) also reasonable agreement (RMSE 2.73 °C), with 'microclimf' outperforming measured *in-situ* differences from the macroclimate (RMSE 3.62 °C).

In the examples above, microclimate predictors were derived from gridded climate and geospatial datasets, with errors arising from both model structure and input data—sources that could not be separated. Additionally, the loggers used are prone to measurement error (Maclean et al., 2021), making it difficult to attribute the cause of discrepancies. To circumvent these problems, localised validation was conducted using more precise *in-situ* measurements. In northern Spain (43.5°N, 5.7°W), 8 SurveyTag loggers were deployed in a meadow from April to July 2024, recording hourly temperatures c. 5, 40 and 80 cm above ground. SurveyTag loggers offer improved accuracy in sunlight through the use of ultra-fine thermocouples and rapid burst averaging (Maclean et al., 2021). Weather inputs were taken from an on-site weather station, and vegetation structure was repeatedly surveyed via photogrammetry. A parallel study was conducted at a grassland site in Cornwall, UK (50.166°N, 5.273°W), with 16 loggers deployed between October and December 2023. In both cases, modelled and measured temperatures showed strong agreement (Spain RMSE: 1.105 °C, n = 16,936; Cornwall RMSE: 0.690 °C, n = 24,567; Fig. 1) with the model outperforming measured differences from macroclimate data (Spain RMSE: 2.364 °C; Cornwall RMSE: 0.734 °C).

As a final test, 36 loggers were deployed 1m above ground under forest canopy at five sites across the south of the UK between March and September 2024 (Cornwall: 50.179°N, 5.083°W and 50.335°N, 4.974°W; Devon: 50.577°N, 3.902°W; Surrey: 51.184°N, 0.857°W and West Sussex: 51.067°N, 0.105°W). Again, modelled and measured temperatures showed strong agreement (RMSE: 1.477 °C, n = 139,168; Fig. 1c) with the model outperforming measured differences from macroclimate data (RMSE: 1.765°C).

Example application

To illustrate the basic use of `microlimf`, the model is applied to derive microclimate surfaces for Caerthillean Cove in Cornwall, United Kingdom (49.968°N, 5.215°W). More detailed instructions and explanation of the optional ways in which the model can be set up are provided in the tutorial vignette. The model is coded as an R package on Github and so first needs to be installed using:

```
require(devtools)
install_github("ilyamaclean/microclimf")
```

The data required to drive the model for this example are built into the package, but normally the starting point of the workflow is to gather and prepare the model inputs and four datasets are required. (i) standard hourly meteorological climate-forcing variables representative of macroclimatic conditions across the study site, usually in the form of a data frame with single values for each hour (though the option to include an array of coarse-gridded values is also available). (ii) A high spatial variation digital elevation dataset. (iii) A time-invariant high-resolution dataset of soil properties: soil type and ground reflectance, or alternatively more detailed information on soil physical properties. (iv) high spatial resolution datasets of vegetation properties, of which the most important are vegetation height and plant area index. Other properties such as leaf inclination angles can be derived from habitat type if unknown. Vegetation variables can either be static or vary seasonally and the package handles this flexibly. The vignette describes the model inputs in more detail. The accompanying R package `microclimdata` (Maclean 2025) enables automated download and processing of these datasets globally.

The next stage is to run function `runpointmodel`, which runs the microclimate model for a single grid cell at the centre of the study area using mean or modal vegetation or soil properties, here demonstrated using the inbuilt datasets that accompany the package.

```
library(microclimf)
micropoint <- runpointmodel(climdata, reqhgt = 0.05, dtmcaerth,
     vegp, soilc)
attributes(micropoint)
```

The input `reqhgt` is the height above ground (m) for which the model is run. Assigning negative values means the model is run below ground. This function returns (1) a data frame ` of weather variables, but with temperature and wind speed height-adjusted to be above canopy if necessary and (2) a data frame of microclimate point model outputs required for running grid model. Other outputs, such as the latitude and longitude of the grid cell over which the model was run, the height to which weather data were adjusted and a POSIXlt object of observation times are also used by the grid model.

The following code plots the output shown in Fig. 2a, representing a time series of ground surface, canopy temperatures.

```
microp <- micropoint$dfo
tme <- as.POSIXct(micropoint$tmeorig)
par(mar=c(5,5,3,3))
plot(microp$Tg ~ tme, type="l", ylim = c(-5, 50), col =
rgb(1,0,0,0.5),
     xlab = "Month", ylab = "Temperature") # temperature of ground
surface
par(new = TRUE)
plot(microp$Tc ~ tme, type="l", ylim = c(-5, 50), col =
rgb(0,0.5,0.5,0.5),
     xlab = "", ylab = "") # temperature of canopy
```

At this point users have the option to subset the model, e.g. to return microclimate data for the hottest day in each month as in the example below.

```
# Subset point model outputs
micropoint_mx <- subsetpointmodel(micropoint, tstep = "month",
      what = "tmax")
```

The object returned from that function has the same format as that returned by `runpointmodel`, but only data for the relevant days have been selected. This is then passed to the grid modelling function as follows:

```
# Run grid model 5 cm above ground with inbuilt datasets
mout_mx <- runmicro(micropoint_mx, reqhgt = 0.05, vegp, soilc,
   dtmcaerth)
attributes(mout_mx)
```

This function returns a list of multi-layer rasters compatible with the `terra` package. Each raster comprised a different microclimate variable, with individual layers corresponding to values for each hour. If `reqhgt `> 0 (above ground) then by default the model returns the following: hourly air and leaf temperature (°C) with leaf temperature (set to the average

temperature of the canopy if `reqhgt `is above canopy), relative humidity (%), wind speed ($m \cdot s^{-1}$), the upward and downward direct, diffuse and longwave radiation fluxes ($W \cdot m^{-2}$) (all at 'reqhgt') and the fractional soil water content in the upper 10 cm of soil ($m^{-3} \cdot m^{-3}$). If reqhgt = 0, then wind speed, leaf temperature and relative humidity are not returned and the temperature values returned are for the soil surface. If reqhgt < 0, then only temperature and soil water content are returned. To save memory, users are optionally afforded flexibility with which variables to return.

Any of these variables can then be plotted for any hour using the `terra::plot` function as in the example below, which is used to generate Fig. 1b.

```
# Plot air temperatures on hottest hour in micropoint
# (2017-06-20 13:00:00 UTC)
require(terra) # required to plot SpatRasters
mypal <- colorRampPalette(c("darkblue", "blue", "green",
                            "yellow", "orange", "red"))(255)
plot(rast(mout_mx$Tz[,,134]), col = mypal, range = c(20, 48))
```

Because returned datasets are stored in internal memory, the model cannot be run over very large areas and extended time-periods. Function `runmicro_big` can thus be used to run the model in tiles. Here, terrain sheltering, topographic wetness and slope and aspect are first calculated across the entire study area to avoid edge tiling effects. As an additional guard against tiling effects users have the option to specify the extent of overlap between tiles and to mosaic individual tiled datasets together using a distance-weighting with function `mosaicblend`. An example of an output mosaiced in this way, maximum air temperature 5 cm above ground modelled at one metre grid resolution for a $1km^2$ region surrounding Caerthillean Cove is shown in Fig 3. Here the output as been rendered as a 3D visualisation using the 'plotly' package.

Additionally, because a common application of microclimate models is to run high-resolution species distribution models, functions `runbioclim` and `runbioclim_big` automatically derive microclimate equivalents of a set of 19 bioclimate variables commonly used in such analyses (Fick and Hijmans 2017). When doing so, seasonal and annual rainfall variables are instead replaced by soil moisture variables as the latter are likely more ecologically informative and vary much more at fine spatial scales.

While the model can be run in entirety using functions `runmicro`, individual steps of the model can be run in stages to improve model diagnoses and testing. Function `soilmdistribute` distributes soil moisture from the point model spatially by topographic wetness. Function `twostream` runs the Dickenson-Sellers two stream radiation model. Function `wind` applies terrain sheltering and calculates wind speed at the user specified height above ground. Function `soiltemp` calculates soil surface temperature. Then either the `aboveground` or `belowground` functions are used to derive microclimate above or below ground. The sequence in which model components are run is in the order listed here, but if a prior stage has not yet been run, functions handle this automatically and runs that stage.

In the workflows described above, the model is run without accounting for snow cover. To model snow depth, the function `runsnowmodel` is used after running and potentially sub-setting the point model. Then when using `runmicro` to derive microclimate variables, the

optional function input snow is set to TRUE and the snow model output passed to this function as follows:

```
smod <- runsnowmodel(climdata, micropoint, vegp, soilc, dtmcaerth)
mout <- runmicro(micropoint, reqhgt = 0.05, vegp, soilc, dtmcaerth,
      snow = TRUE, snowmod = smod)
```

Computational performance and limitations

Subject to memory constraints, the input weather datasets can be of unlimited length—allowing, for instance, the model to be run continuously over a 20-year period. Because the model calculates annual cycles in ground heat flux, it performs best when run over complete calendar years. Computation time depends on settings such as whether snow is modelled and whether the grid is run for all days of the year. On a standard desktop PC, and with snow modelling disabled, the model requires approximately 2.6 milliseconds per grid cell to generate monthly outputs for a full year. This translates to around 26 seconds for a 10 km × 10 km area at 100 m resolution. Generating hourly outputs for the same area takes approximately 360 seconds. By comparison, the equivalent analyses using the NicheMapR package would take 3.5 hours for monthly outputs and 9.6 hours for hourly outputs, albeit with some scope to reduce run times by coding bespoke R wrappers for the underlying FORTRAN code. Using 'microclimc' the equivalent analyses would take even longer: approximately 4.0 and 20.3 hours for monthly and hourly outputs respectively. `Microclimf` thus offers the utility of a full mechanistic model but remains computationally efficient even when applied to large spatial extents.

To date, validation has focused mainly on temperature; the ability of the model to estimate other microclimate variables requires further testing. Simplifying assumptions—such as the model's inability to account for periodic flushing of moist air from the forest understory—may affect humidity estimates, with implications for applications like fire risk modelling. Nonetheless, the package meets a key need for improved regional and landscape-scale estimates of the conditions experienced by organisms. This, in turn, enhances our ability to quantify species–environment relationships and improves understanding and prediction of how organisms respond to environmental change.

Acknowledgements

The work was supported by funding from the Natural Environment Research Council (NE/X015262/1 and NE/W006618/1) and from the Met Office Hadley Centre Climate Programme (HCCP) funded by UK government departments BEIS and Defra. Ruth Warfield's and Rolando Rodriguez-Munoz's help with sensor deployment and Jonathan Mosedale, Alexandra Gardner and Rounan Lee's help with photogrammetry is greatly appreciated.

Conflict of Interest statement

The author declares no conflicts of interest

Data availability statement

Logger data used to test the model and data used to generate Fig. 2 are available from Zenodo (https://doi.org/10.5281/zenodo.15364781 & 10.5281/zenodo.8338611). Other datasets used are included with 'microclimf' R package available from https://github.com/ilyamaclean.

References

Anderson, E. A. (2006). National Weather Service river forecast system: Snow accumulation and ablation model. U. S. Department of Commerce, National Oceanic and Atmospheric Administration.

Bennie, J., Huntley, B., Wiltshire, A., Hill, M. O., Baxter, R. (2008). Slope, aspect and climate: spatially explicit and implicit models of topographic microclimate in chalk grassland. *Ecological Modelling*, 216, 47–59.

Best, M. J., Pryor, M., Clark, D., Rooney, G. G., Essery, R., Ménard, C., Edwards, J., Hendry, M., Porson, A., Gedney, N. (2011). The Joint UK Land Environment Simulator (JULES), model description–Part 1: energy and water fluxes. *Geoscientific Model Development*, 4, 677–699.

Bonan, G. B., Patton, E. G., Finnigan, J. J., Baldocchi, D. D., Harman, I. N. (2021). Moving beyond the incorrect but useful paradigm: reevaluating big-leaf and multilayer plant canopies to model biosphere-atmosphere fluxes–a review. *Agricultural and Forest Meteorology*, 306, 108435.

Bramer, I., Anderson, B. J., Bennie, J., Bladon, A. J., De Frenne, P., Hemming, D., Hill, R. A., Kearney, M. R., Körner, C., Korstjens, A. H. (2018). Advances in monitoring and modelling climate at ecologically relevant scales. In: Advances in Ecological Research, 58, 101–161. Elsevier.

Briscoe, N. J., Morris, S. D., Mathewson, P. D., Buckley, L. B., Jusup, M., Levy, O., Maclean, I. M., Pincebourde, S., Riddell, E. A., Roberts, J. A. (2023). Mechanistic forecasts of species responses to climate change: the promise of biophysical ecology. *Global Change Biology*, 29, 1451–1470.

Businger, J. A., Wyngaard, J. C., Izumi, Y., Bradley, E. F. (1971). Flux-profile relationships in the atmospheric surface layer. *Journal of the Atmospheric Sciences*, 28, 181–189.

Campbell, G. S., Norman, J. M. (2012). An Introduction to Environmental Biophysics. Springer Science, Business Media.

Chen, Y., Ryder, J., Bastrikov, V., McGrath, M. J., Naudts, K., Otto, J., Ottlé, C., Peylin, P., Polcher, J., Valade, A. (2016). Evaluating the performance of land surface model ORCHIDEE-CAN v1.0 on water and energy flux estimation with a single-and multi-layer energy budget scheme. *Geoscientific Model Development*, 9, 2951–2972.

De Frenne, P., Lenoir, J., Luoto, M., Scheffers, B. R., Zellweger, F., Aalto, J., Ashcroft, M. B., Christiansen, D. M., Decocq, G., De Pauw, K. (2021). Forest microclimates and climate change: Importance, drivers and future research agenda. *Global Change Biology*, 27, 2279–2297.

De Vries, D. A., Van Wijk, W. R. (1963). Physics of Plant Environment. North-Holland Publishing Company.

Fick, S. E., Hijmans, R. J. (2017). WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, 37, 4302–4315.

Flerchinger, G. N., Reba, M. L., Link, T. E., Marks, D. (2015). Modeling temperature and humidity profiles within forest canopies. *Agricultural and Forest Meteorology*, 213, 251–262.

Foken, T. (2006). 50 years of the Monin–Obukhov similarity theory. *Boundary-Layer Meteorology*, 119, 431–447.

Gates, D. M. (2012). Biophysical Ecology. Courier Corporation.

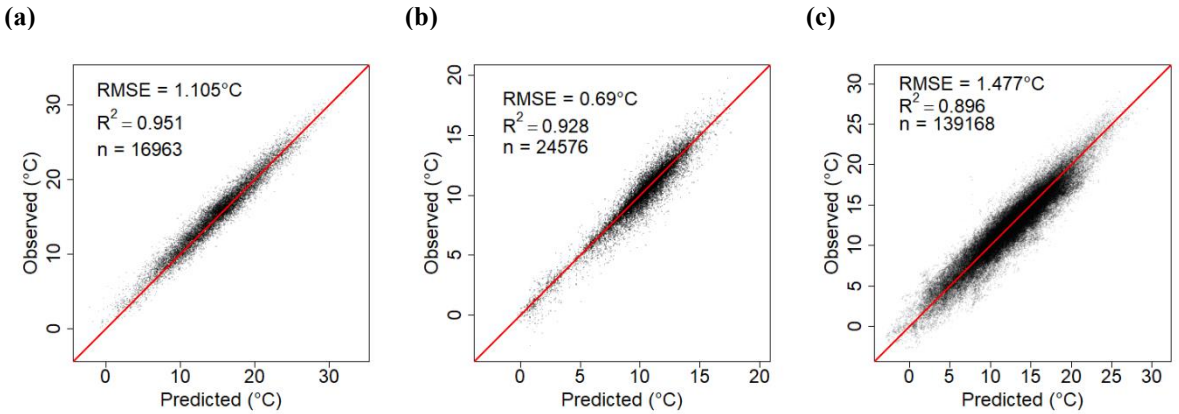
Goudriaan, J. (1977). Crop Micrometeorology: A. Simulation Study. Wageningen University and Research.

Haesen, S., Lembrechts, J. J., De Frenne, P., Lenoir, J., Aalto, J., Ashcroft, M. B., Kopecký, M., Luoto, M., Maclean, I. M. D., Nijs, I. (2021). ForestTemp–Sub-canopy microclimate temperatures of European forests. *Global Change Biology*, 27, 6307–6319.

Harman, I. N., Finnigan, J. J. (2007). A. simple unified theory for flow in the canopy and roughness sublayer. *Boundary-Layer Meteorology*, 123, 339–363.
Hedstrom, N., Pomeroy, J. W. (1998). Measurements and modelling of snow interception in the boreal forest. *Hydrological Processes*, 12, 1611–1625.
Kearney, M., Porter, W. P. (2009). Mechanistic niche modelling: combining physiological and spatial data to predict species' ranges. *Ecology Letters*, 12, 334–350.
Kearney, M. R., Porter, W. P. (2017). NicheMapR–an R. package for biophysical modelling: the microclimate model. *Ecography*, 40, 664–674.
Kearney, M. R. (2020). How will snow alter exposure of organisms to cold stress under climate warming? Global Ecology and Biogeography, 29, 1246–1256.
Kearney, M. R., Gillingham, P. K., Bramer, I., Duffy, J. P., Maclean, I. M. D. (2020). A. method for computing hourly, historical, terrain-corrected microclimate anywhere on earth. *Methods in Ecology and Evolution*, 11, 38–43.
Kelliher, F. M., Leuning, R., Raupach, M., Schulze, E.-D. (1995). Maximum conductances for evaporation from global vegetation types. *Agricultural and Forest Meteorology*, 73, 1–16.
Kemppinen, J., Lembrechts, J. J., Van Meerbeek, K., Carnicer, J., Chardon, N. I., Kardol, P., Lenoir, J., Liu, D., Maclean, I., Pergl, J. (2023). Microclimate, an inseparable part of ecology and biogeography. *Global Ecology and Biogeography*, 32, 31–45.
Kolstela, J., Aakala, T., Maclean, I., Niittynen, P., Kemppinen, J., Luoto, M., Rissanen, T., Tyystjärvi, V., Gregow, H., Vapalahti, O. (2024). Revealing fine-scale variability in boreal forest temperatures using a mechanistic microclimate model. *Agricultural and Forest Meteorology*, 350, 109995.
Körner, C. (1995). Leaf diffusive conductances in the major vegetation types of the globe. In: Ecophysiology of Photosynthesis (Eds E.-D. Schulze, M. M. Caldwell), pp. 463–490. Springer.
Lawrence, D. M., Fisher, R. A., Koven, C. D., Oleson, K. W., Swenson, S. C., Bonan, G., Collier, N., Ghimire, B., van Kampenhout, L., Kennedy, D. (2019). The Community Land Model version 5: Description of new features, benchmarking, and impact of forcing uncertainty. *Journal of Advances in Modeling Earth Systems*, 11, 4245–4287.
Lembrechts, J. J., Nijs, I., Lenoir, J. (2019). Incorporating microclimate into species distribution models. *Ecography*, 42, 1267–1279.
Lembrechts, J. J., Lenoir, J. (2020). Microclimatic conditions anywhere at any time! Global Change Biology, 26, 337–339.
Lembrechts, J. J., van den Hoogen, J., Aalto, J., Ashcroft, M. B., De Frenne, P., Kemppinen, J., Kopecký, M., Luoto, M., Maclean, I. M. D., Crowther, T. W. (2022). Global maps of soil temperature. *Global Change Biology*, 28, 3110–3144.
Maclean, I.M.D. (2025) *microclimdata: An R package for downloading data needed to run microclimate models*. Available: https://github.com/ilyamaclean/microclimdata.
Maclean, I. M. D., Suggitt, A. J., Wilson, R. J., Duffy, J. P., Bennie, J. J. (2017). Fine-scale climate change: modelling spatial variation in biologically meaningful rates of warming. *Global Change Biology*, 23, 256–268.
Maclean, I. M. D., Mosedale, J. R., Bennie, J. J. (2019). Microclima: An R. package for modelling meso-and microclimate. *Methods in Ecology and Evolution*, 10, 280–290.
Maclean, I. M. D., Duffy, J. P., Haesen, S., Govaert, S., De Frenne, P., Vanneste, T., Lenoir, J., Lembrechts, J. J., Rhodes, M. W., Van Meerbeek, K. (2021). On the measurement of microclimate. *Methods in Ecology and Evolution*, 12, 1397–1410.
Maclean, I. M. D., Klinges, D. H. (2021). Microclimc: A. mechanistic model of above, below and within-canopy microclimate. *Ecological Modelling*, 451, 109567.
Monin, A. S., Obukhov, A. M. (1954). Basic laws of turbulent mixing in the atmosphere near the ground. Trudy Geofiz. Inst. *AN SSSR*, 24, 163–187.

Monteith, J. L., Unsworth, M. H. (2013). Principles of Environmental Physics: Plants, Animals, and the Atmosphere. Academic Press.
Monteith, J. L. (1965). Evaporation and environment. In: Symposia of the Society for Experimental Biology, 19, 205–234. Cambridge University Press.
Ogée, J., Brunet, Y., Loustau, D., Berbigier, P., Delzon, S. (2003). MuSICA, a CO_2, water and energy multilayer, multileaf pine forest model: evaluation from hourly to yearly time scales and sensitivity analysis. *Global Change Biology*, 9, 697–717.
Penman, H. L. (1948). Natural evaporation from open water, bare soil and grass. Proceedings of the Royal Society of London. Series A. *Mathematical and Physical Sciences*, 193, 120–145.
Porter, W. P., Mitchell, J. W., Beckman, W. A., DeWitt, C. B. (1973). Behavioral implications of mechanistic ecology: thermal and behavioral modeling of desert ectotherms and their microenvironment. *Oecologia*, 13, 1–54.
Potter, K. A., Woods, H. A., Pincebourde, S. (2013). Microclimatic challenges in global change biology. *Global Change Biology*, 19, 2932–2939.
Raupach, M. R. (1989a). Applying Lagrangian fluid mechanics to infer scalar source distributions from concentration profiles in plant canopies. *Agricultural and Forest Meteorology*, 47, 85–108.
Raupach, M. R. (1989b). A. practical Lagrangian method for relating scalar concentrations to source distributions in vegetation canopies. *Quarterly Journal of the Royal Meteorological Society*, 115, 609–632.
Raupach, M. R. (1994). Simplified expressions for vegetation roughness length and zero-plane displacement as functions of canopy height and area index. *Boundary-Layer Meteorology*, 71, 211–216.
Richardson, L. F. (1922). Weather Prediction by Numerical Process. Cambridge University Press.
Ryan, B. C. (1977). A. mathematical model for diagnosis and prediction of surface winds in mountainous terrain. *Journal of Applied Meteorology and Climatology*, 16, 571–584.
Sellers, P. J. (1985). Canopy reflectance, photosynthesis and transpiration. *International Journal of Remote Sensing*, 6, 1335–1372.
Sturm, M., Taras, B., Liston, G. E., Derksen, C., Jonas, T., Lea, J. M. (2010). Estimating snow water equivalent using snow depth data and climate classes. *Journal of Hydrometeorology*, 11, 1380–1394.
Trew, B. T., Edwards, D. P., Lees, A. C., Klinges, D. H., Early, R., Svátek, M., Plichta, R., Matula, R., Okello, J., Niessner, A., Barthel, M., Six, J., Maeda, E., Barlow, J., Nascimento, R., Berenguer, E., Ferreira, J., Sallo-Bravo, J., Maclean, I. M. D. (2024). Novel temperatures are already widespread beneath the world's tropical forest canopies. *Nature Climate Change*, 14, 753–759.
WMO (2021). Guide to Instruments and Methods of Observation (WMO-No. 8). World Meteorological Organization.
Yuan, H., Dai, Y., Dickinson, R. E., Pinty, B., Shangguan, W., Zhang, S., Wang, L., Zhu, S. (2017). Reexamination and further development of two-stream canopy radiative transfer models for global land modeling. *Journal of Advances in Modeling Earth Systems*, 9, 113–129.

659



660 **Fig.1.** Comparison of observed and modelled temperatures in grassland sites in Asturias, Spain
661 (a) and Cornwall, UK (b), and beneath forest canopies in southern England (c). Root-mean-
662 square error (RMSE), number of observations (*n*), and Pearson's R^2 are shown for each site.
663 Further details are provided in supporting information.

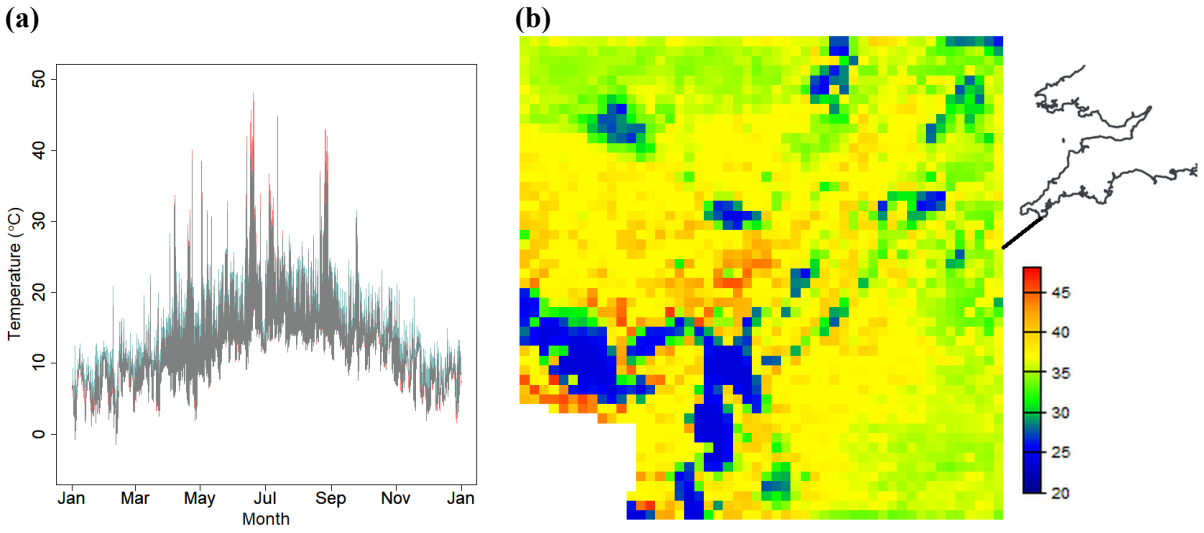


Fig. 2. Outputs from the microclimf model shown for Caerthillian Cove in Cornwall, United Kingdom (49.968°N, 5.215°W). In (a) hourly outputs in 2017 from the point model are shown as a time-series of hourly input air temperature (blue) and canopy temperatures (red). The plot colour is partially translucent and grey represents the overlap. In (b) one metre resolution air temperature 5 cm above ground is shown for 2pm on the 1st June 2017 for a 50m x 50m area. The areas shown in blue are shaded by vegetation.

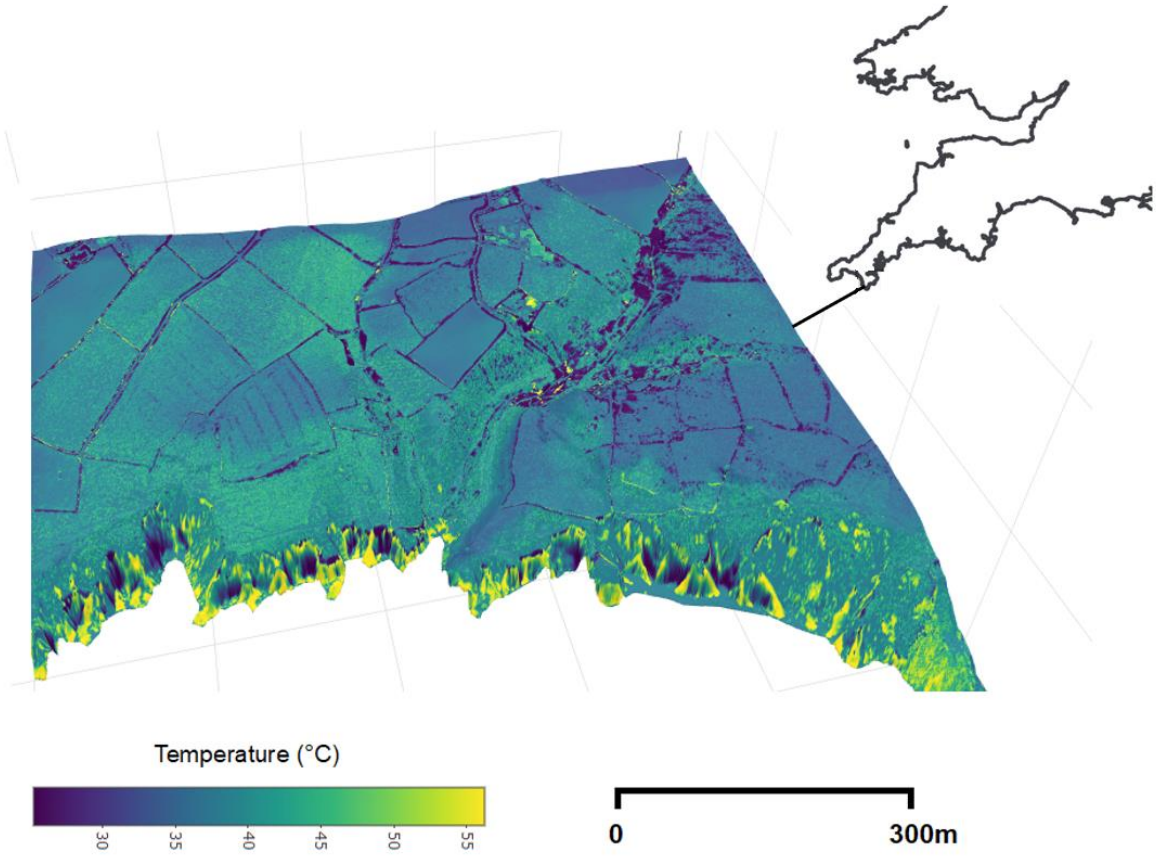


Fig. 3 *microclimf* model outputs generated with `runmicro_big` for a 1 × 1 km area around Caerthillian Cove, Cornwall, UK (49.968°N, 5.215°W). The map shows temperatures 5 cm above ground at 2 pm on 1 June 2017, modelled at 1 m resolution.

Supplementary information: Field testing

Spain: unimproved grass meadow

Logger deployment: In Asturias, Spain (43.5°N, 5.7°W, https://www.wildcrickets.org/), between 21st Apr and 5th July 2024, 8 SurveyTag loggers (ConceptShed, Falmouth, UK, https://surveytag.co.uk/) were mounted horizontally at a height of c. 40 cm above ground on four wooden poles, spaced approximately evenly over a 2.2 Ha unimproved meadow. Additionally, 3 SurveyTag loggers were mounted at heights of 5.7, 35.9 and 84.1 cm above ground at the centre of the study area. SurveyTag loggers are designed to operate without radiation shielding, allowing them to measure air temperature in natural conditions. Radiation-induced bias is minimised by using a 0.075 mm K-type ultra-fine thermocouple, which ensures the sensor is closely coupled to the surrounding air (Maclean *et al.* 2021). To further enhance accuracy, each logger records 1,024 rapid burst measurements over 1.9 seconds and stores the mean, reducing temperature fluctuations caused by air turbulence. Grass height was mown twice during this period, but was otherwise permitted to grow naturally, varying in height from 5.1 to 54 cm. A total of 16,963 hourly measurements were taken.

Weather data: at the centre of the study site, a Vantage Pro2 Automatic Weather Station (Davis Instruments, Hayward, CA) recorded air and dewpoint temperature, wind speed, wind direction, barometric pressure, downward shortwave radiation and rainfall at hourly intervals, over the same period over which loggers were deployed. Downward longwave radiation data were obtained from the ERA5 reanalysis dataset (Hersbach *et al.* 2020), produced by the European Centre for Medium-Range Weather Forecasts (ECMWF). Diffuse radiation was derived from total downward shortwave radiation following Skartveit *et al.* (1998)

Vegetation and ground data: Grass height, vertical variation in leaf area, and the leaf inclination angle coefficient were estimated by photographing a white board marked with height intervals. At intervals of between 7 and 10 days, photos were taken horizontally at ground level through the grass at distances of 10, 25, 50, and 100 cm from the board using a smartphone (Samsung Galaxy S22 Ultra). Images were ortho-corrected using the rubber-sheeting tool in ArcGIS, then processed to produce binary classifications distinguishing grass (white board obscured) from non-grass (white board visible) pixels. The fraction (F) of the image obscured by grass within each height band was then computed. Then by applying by rearranging following formula:

$$F = 1 - \exp\left(-KP_{AI}\right)$$

where P_{AI} represents the total one sided area of foliage per unit cross-sectional area within each height band between the camera lens and the white board. Then, by assuming P_{AI} per unit distance from the white board was approximately constant over the four distances, a value for K was estimated using linear-regression. From Campbell (1986)

$$K \approx \frac{\sqrt{x^2}}{x + 1.774(x + 1.182) - 0.733}$$

Permitting the leaf inclination angle coefficient x to be derived as:

$$x = \frac{1.361K}{2.774K - 1}$$

A simple allometric relationship was used to derived leaf width (l_w) vegetation height (h) as $l_w = 0.08h$ and the clumping coefficient was set at 0.25 following visual inspection of photographs. For the remainder of vegetation parameters, typical values for unimproved grassland were derived from literature (Kattge *et al.* 2011) and were thus set at 0.25 (leaf reflectance), 0.22 (leaf transmittance), 0.30

mol/m^2/s (maximum stomatal conductance) and 100 µmol/ m^2/s (value of PAR when stomatal conductance is at 50 percent of its maximum value). Slope and aspect were measured using a digital inclinometer and compass, and soil type, known for the study area, was set as clay-loam. Ground reflectance was estimated from colour-infrared (0.5m grid resolution) and red-green-blue (0.25m grid resolution) aerial images sourced from Edina Digimap (Morris, Medyckyj-Scott & Burnhill 2000; https://digimap.edina.ac.uk/) using the procedure coded into the microclimdata package (Maclean 2025).

Results: The root-mean-square error for each individual logger is shown in Table S3 (RMSE micro). Logger data have been uploaded to Zenodo (https://doi.org/10.5281/zenodo.15364781). The root-mean-square difference between modelled and air temperatures recorded by the weather station are also shown (RMSD macro). The RMSE of modelled air temperature was always lower than the RMSD from measured air temperature. Averaged across all loggers, the RMSE was 1.105°C in comparison to an RMSD of 2.364°C.

Table S3. Root-mean-square error for each individual logger (RMSE micro) and root-mean-square difference between modelled and weather station air temperatures (RMSD macro). Coordinates are rounded to one decimal place to protect the anonymity of the study site, which is privately owned and hosts sensitive, long-term research infrastructure.

Logger ID	Height above ground (cm)	Latitude (°N)	Longitude (°W)	RMSE micro (°C)	RMSD macro (°C)
SurveyTag_01	36.2	43.5	5.7	1.109	2.435
SurveyTag_02	36.8	43.5	5.7	1.102	2.474
SurveyTag_03	36.9	43.5	5.7	1.097	2.351
SurveyTag_04	37.1	43.5	5.7	1.149	2.246
SurveyTag_05	39.2	43.5	5.7	1.056	1.757
SurveyTag_06	39.3	43.5	5.7	1.119	2.16
SurveyTag_07	35.9	43.5	5.7	1.057	2.498
SurveyTag_08	43.2	43.5	5.7	1.117	2.099
SurveyTag_09	5.7	43.5	5.7	1.123	2.976
SurveyTag_10	39.8	43.5	5.7	1.122	2.343
SurveyTag_11	84.1	43.5	5.7	0.428	1.198

United Kingdom: unimproved grass meadow, above vegetation

Logger deployment: In Cornwall, United Kingdom (50.166°N, 5.273°W), 16 SurveyTag loggers were mounted horizontally on four wooden poles, spaced 45 m apart in a relatively flat and open unimproved grass field. Loggers were positioned at heights of 5, 10, 50, and 100 cm above ground and set to record hourly temperatures from 29 October to 31 December 2023. Grass height was maintained at 3 cm average height by regular mowing throughout the measurement period. Loggers were checked daily for failure two of the loggers initially deployed, which failed, replaced immediately. A total of 24,576 hourly measurements were taken.

Weather data: at the centre of the study site, a MiniMet Automatic Weather Station (Skye Instruments Ltd, Powys, UK) was programmed to record temperature, relative humidity, wind speed, wind direction and downward shortwave radiation at hourly intervals over the same period over which loggers were deployed. Prior to deployment, all instruments were factory recalibrated. Additional hourly precipitation, cloud cover and atmospheric pressure data were obtained from WMO weather station (Camborne, UK, MIDAS station ID: 1305) located 5.03 km from the field study site. Downard longwave radiation was derived from air temperature, relative humidity and cloud cover following Crawford & Duchon (1999) and diffuse radiation from total downward shortwave radiation following Skartveit *et al.* (1998).

Vegetation and ground data: Vegetation coefficients were derived using the same method described above. Slope and aspect were measured using a digital inclinometer and compass, and soil type, known

for the study area, was set as clay-loam. Ground reflectance was estimated from colour-infrared (0.5m grid resolution) and red-green-blue (0.25m grid resolution) aerial images sourced from Edina Digimap (Morris, Medyckyj-Scott & Burnhill 2000) using the procedure coded into the microclimdata package (Maclean 2025).

Results: The root-mean-square error for each individual logger is shown in Table S4 (RMSE micro). Observed and modelled data and the vegetation, ground and other input data used to drive the model are available on Zenodo (https://doi.org/10.5281/zenodo.15364781). Since models were deployed in winter when microclimate temperatures are not so different from reference air temperature, the root-mean-square difference between modelled and air temperatures recorded by the weather station are also shown (RMSD macro). Though both modelled and measured microclimate air temperatures were quite close to air temperature at reference height, the RMSE of modelled air temperature was always lower than the RMSD from measured air temperature. Averaged across all loggers, the RMSE was 0.597°C in comparison to an RMSD of 0.734°C.

Table S4. Root-mean-square error for each individual logger (RMSE micro) and root-mean-square difference between modelled and weather station air temperatures (RMSD macro).

Logger ID	Height above ground (cm)	Latitude (°N)	Longitude (°W)	RMSE micro (°C)	RMSD macro (°C)
SurveyTag_01	5	50.1662	5.27247	1.015	1.262
SurveyTag_02	10	50.1662	5.27247	0.818	1.013
SurveyTag_03	50	50.1662	5.27247	0.347	0.464
SurveyTag_04	100	50.1662	5.27247	0.178	0.232
SurveyTag_05	5	50.16644	5.27258	1.011	1.255
SurveyTag_06	10	50.16644	5.27258	0.845	1.010
SurveyTag_07	50	50.16644	5.27258	0.356	0.463
SurveyTag_08	100	50.16644	5.27258	0.178	0.231
SurveyTag_09	5	50.16608	5.27257	1.083	1.281
SurveyTag_10	10	50.16608	5.27257	0.802	1.032
SurveyTag_11	50	50.16608	5.27257	0.360	0.474
SurveyTag_12	100	50.16608	5.27257	0.182	0.236
SurveyTag_13	5	50.16618	5.27292	1.045	1.191
SurveyTag_14	10	50.16618	5.27292	0.805	0.950
SurveyTag_15	50	50.16618	5.27292	0.363	0.433
SurveyTag_16	100	50.16618	5.27292	0.171	0.216

United Kingdom: woodland, below vegetation

Logger deployment: a total of 36 SurveyTag loggers were deployed between 11th Feb and 12th Oct 2024 across four sites in southern England. Each logger was mounted horizontally one metre above ground below canopy in woodlands dominated either by Norway Spruce (*Picea abies*) or English Oak (*Quercus robur*). The sites were (from west to east) Mylor Bridge near Falmouth in Cornwall (50.18°N, 5.08°W), Trelassick Wood near Truro in Cornwall (50.33°N, 4.97°W), Bellever Forest within the Dartmoor National Park in Devon (50.57°N, 3.90°W), Alice Holt, near Farnham in Surrey (51.18°N, 0.86°W) and Wakehurst Park near Crawley in West Sussex (51.06°N, 0.10°W). The precise locations of logger and their associated forest type are shown in Table S5.

Weather data: climate data for each survey location were sourced using the mcera5 R package (Klinges *et al.* 2022), which automatically downloads a freely accessible climate reanalysis product provided by the European Centre for Medium-Range Weather Forecasts. These climate data are available at hourly temporal and 25 km grid resolutions The 25 km grid resolution data were then downscaled to 1km spatial resolution using the microclimdata package (Maclean 2025). This package automatically sources one km grid resolution daily precipitation and minimum and maximum data from Hadley UK Gridded Climate Observations dataset (Hollis *et al.* 2019). The function 'haduk_blend', then locally adjusts the diurnal temperature range and daily precipitation totals in the 25 km data, preserving the intra-diurnal patterns in these data. The remaining climate datasets are downscaled using bilinear interpolation.

Vegetation and ground data: Leaf area index (LAI) was derived from hemispherical photographs taken using a Yarrashop mobile phone fisheye lens mounted on a smartphone, processed using the *Hemisfer* software (Schleppi & Paquette 2017). At each site, photographs were taken twice monthly during the leaf-on and leaf-off periods (February to mid-April and mid-September to October), and monthly at other times. Lens distortion parameters were estimated prior to LAI calculation using the calibration procedure described in the *Hemisfer* manual. Canopy height was obtained from the 10 m global canopy height model produced by Lang *et al.* (2023), downloaded and processed using the microclimdata R package. Other vegetation parameters were assigned based on published values for the dominant vegetation type at each site. To account for uncertainty in vegetation inputs, limited model tuning was carried out. A Morris sensitivity analysis was used to identify the six most influential vegetation-related parameters affecting microclimate outputs. These parameters were then optimised using the bayesOpt() function from the ParBayesianOptimization R package, with root mean square error (RMSE) as the performance criterion. Maximum and minimum values for each parameter were set at double and half the initial predicted value, respectively, to define plausible bounds for the optimisation. The optimiser began by running the point version of the microclimate model at a set of initial parameter combinations. These results were used to fit a Gaussian Process (GP) emulator, which approximates the relationship between parameter values and model performance across the tested space. The emulator was then used to guide the selection of new parameter combinations, balancing exploration and exploitation via the expected improvement criterion. As additional forward model runs were performed, the GP emulator was iteratively updated, allowing efficient convergence on a parameter set that minimised RMSE. This approach enabled efficient convergence on an optimal parameter set while avoiding the computational cost of brute-force sampling. Derivation of ground parameters followed the procedure outlined for above canopy.

Results: The root-mean-square error for each individual logger is shown in Table S5 (RMSE micro). Observed and modelled data and the vegetation, ground and other input data used to drive the model are available on Zenodo (https://doi.org/10.5281/zenodo.15364781). Root-mean-square difference between modelled and air temperatures recorded by the weather station are also shown (RMSD macro). Though both modelled and measured microclimate air temperatures were quite close to air temperature at reference height, the RMSE of modelled air temperature was always lower than the RMSD from measured air temperature. Averaged across all loggers, the RMSE was 1.477°C in comparison to an RMSD of 1.765°C.

Table S5. Root-mean-square error for each individual logger (RMSE micro) and root-mean-square difference between modelled and weather station air temperatures (RMSD macro) for loggers deployed below canopy.

Logger ID	Forest type	Site	Latitude (°N)	Longitude (°W)	RMSE micro (°C)	RMSD macro (°C)
ah_nsp_1	Norway Spruce	Alice Holt	51.18394	0.85707	1.538	2.214
ah_nsp_2	Norway Spruce	Alice Holt	51.18393	0.85715	1.505	2.134
ah_nsp_3	Norway Spruce	Alice Holt	51.18427	0.85705	1.530	2.309
ah_nsp_4	Norway Spruce	Alice Holt	51.18429	0.85759	1.515	2.267
ah_oak_1	Oak	Alice Holt	51.17737	0.85242	1.553	1.653
ah_oak_2	Oak	Alice Holt	51.18004	0.85402	1.524	2.015
ah_oak_3	Oak	Alice Holt	51.17999	0.85440	1.557	1.850
ah_oak_4	Oak	Alice Holt	51.17978	0.85509	1.560	1.958
ah_oak_5	Oak	Alice Holt	51.18059	0.85361	1.506	2.084
be_nsp_1	Norway Spruce	Bellever Forest	50.57701	3.90211	1.422	1.752
be_nsp_2	Norway Spruce	Bellever Forest	50.57697	3.90234	1.367	1.775
be_nsp_3	Norway Spruce	Bellever Forest	50.57270	3.90339	1.444	1.388
be_nsp_4	Norway Spruce	Bellever Forest	50.57274	3.90340	1.488	1.464
be_nsp_5	Norway Spruce	Bellever Forest	50.57278	3.90328	1.320	1.549
be_nsp_6	Norway Spruce	Bellever Forest	50.57283	3.90296	1.283	1.741
be_oak_1	Oak	Bellever Forest	50.56677	3.89258	1.353	1.568
be_oak_2	Oak	Bellever Forest	50.56742	3.89306	1.396	1.673
be_oak_3	Oak	Bellever Forest	50.56768	3.89278	1.289	1.722
cw_nsp_1	Norway Spruce	Cornwall	50.33543	4.973782	1.447	1.503
cw_nsp_2	Norway Spruce	Cornwall	50.33538	4.973955	1.501	1.566
cw_nsp_3	Norway Spruce	Cornwall	50.33516	4.97320	1.437	1.468
cw_nsp_4	Norway Spruce	Cornwall	50.33518	4.974113	1.501	1.608
cw_nsp_5	Norway Spruce	Cornwall	50.33521	4.973614	1.459	1.588
cw_nsp_6	Norway Spruce	Cornwall	50.33518	4.973637	1.492	1.535
cw_oak_1	Oak	Cornwall	50.17850	5.082619	1.491	1.631
cw_oak_2	Oak	Cornwall	50.17929	5.078606	1.386	1.376
cw_oak_3	Oak	Cornwall	50.17922	5.079226	1.437	1.538
wh_nsp_1	Norway Spruce	Wakehurst Park	51.06707	0.104812	1.472	1.718
wh_nsp_2	Norway Spruce	Wakehurst Park	51.06707	0.104812	1.430	1.854
wh_nsp_3	Norway Spruce	Wakehurst Park	51.06707	0.104812	1.467	1.344
wh_nsp_4	Norway Spruce	Wakehurst Park	51.06707	0.104812	1.515	1.671
wh_oak_1	Oak	Wakehurst Park	51.05979	0.101671	1.443	1.685
wh_oak_2	Oak	Wakehurst Park	51.05979	0.101870	1.427	1.843
wh_oak_3	Oak	Wakehurst Park	51.05979	0.101870	1.577	1.681
wh_oak_4	Oak	Wakehurst Park	51.05979	0.101870	1.576	1.495

Campbell, G.S. (1986) Extinction coefficients for radiation in plant canopies calculated using an ellipsoidal inclination angle distribution. *Agricultural and Forest Meteorology,* **36,** 317-321.

Crawford, T.M. & Duchon, C.E. (1999) An improved parameterization for estimating effective atmospheric emissivity for use in calculating daytime downwelling longwave radiation. *Journal of Applied Meteorology,* **38,** 474-480.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R. & Schepers, D. (2020) The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society,* **146,** 1999-2049.

Hollis, D., McCarthy, M., Kendon, M., Legg, T. & Simpson, I. (2019) HadUK-Grid—A new UK dataset of gridded climate observations. *Geoscience Data Journal,* **6,** 151-159.

Kattge, J., Diaz, S., Lavorel, S., Prentice, I.C., Leadley, P., Bönisch, G., Garnier, E., Westoby, M., Reich, P.B. & Wright, I.J. (2011) TRY–a global database of plant traits. *Global change biology,* **17,** 2905-2935.

Klinges, D.H., Duffy, J.P., Kearney, M.R. & Maclean, I.M. (2022) mcera5: Driving microclimate models with ERA5 global gridded climate data. *Methods in Ecology and Evolution,* **13,** 1402-1411.

Lang, N., Jetz, W., Schindler, K. & Wegner, J.D. (2023) A high-resolution canopy height model of the Earth. *Nature Ecology & Evolution,* **7,** 1778-1789.

Maclean, I. (2025) microclimdata: An R package for downloading data needed to run microclimate models. Available: https://github.com/ilyamaclean/microclimdata.

Maclean, I.M., Duffy, J.P., Haesen, S., Govaert, S., De Frenne, P., Vanneste, T., Lenoir, J., Lembrechts, J.J., Rhodes, M.W. & Van Meerbeek, K. (2021) On the measurement of microclimate. *Methods in Ecology and Evolution,* **12,** 1397-1410.

Morris, B., Medyckyj-Scott, D. & Burnhill, P. (2000) EDINA Digimap: new developments in the Internet Mapping and Data Service for the UK Higher Education community. *LIBER Quarterly: The Journal of the Association of European Research Libraries,* **10,** 445-453.

Schleppi, P. & Paquette, A. (2017) Solar radiation in forests: theory for hemispherical photography. *Hemispherical photography in forest science: theory, methods, applications***,** 15-52.

Skartveit, A., Olseth, J.A. & Tuft, M.E. (1998) An hourly diffuse fraction model with correction for variability and surface albedo. *Solar energy,* **63,** 173-183.

Model Equations

Overview

The starting point for modelling is use to standard Monin-Obukhov Similarity Theory (MOST, Foken 2006) to derive the energy balance of a vegetated surface. Thus, vegetation and the underlying ground surface are treated as single vertically homogeneous layer of phytomass – essentially a 'big leaf'. Components of the energy balance have a temperature dependence, and the temperature of the surface is then derived by assuming the energy fluxes, all expressed in $W \cdot m^2$, reach steady state such that

$$R_{abs} - R_{lw}^{\uparrow} = H + L + G$$

where R_{abs} is absorbed radiation, $R_{lw}^{\uparrow}$ is emitted radiation, H sensible heat, L latent heat and G the rate of heat storage by the ground. Radiation absorbed by the canopy R_{abs} is given by

$$R_{abs} = (1 - \alpha)R_{sw}^{\downarrow} + \varepsilon R_{lw}^{\downarrow}$$

where $R_{sw}^{\downarrow}$ is downward shortwave radiation, $R_{lw}^{\downarrow}$ is downward longwave radiation, α is the combined canopy and ground albedo, calculated following Sellers (1985) – see below, and ε the emissivity of the canopy surface, set at 0.97. Emitted radiation is given by

$$R_{lw}^{\uparrow} = \varepsilon_v \sigma T_C^4$$

where T_C the temperature of the canopy surface (K). Sensible heat is a function of the difference in temperature between the canopy heat exchange surface (T_c) and the air (T_A) at reference height (z_R):

$$H = \frac{\tilde{\rho} c_p}{r_{Ha}}(T_C - T_A)$$

where $\tilde{\rho}$ and c_p are the density and specific heat of air respectively ($J \cdot m^{-3} \cdot K^{-1}$) and r_{Ha} is the resistance to heat transfer ($s \cdot m^{-1}$) given by

$$r_{Ha} = \frac{\ln \frac{z_R - d}{z_H} + \psi_H(z_R)}{\kappa u_*}$$

where u_* is the friction velocity of wind ($m \cdot s^{-1}$) given by

$$u_* = \frac{\kappa u_{z_R}}{\ln \frac{z_R - d}{z_M} + \psi_M(z_R)}$$

where u_{z_R} ($m \cdot s^{-1}$) is the wind speed at reference height. Here κ is the Von Kármán constant (taken as 0.4) and $\psi_H(z_R)$ and $\psi_M(z_R)$ the diabatic correction coefficient for heat and momentum calculated following Businger *et al* (1971). The zero plane displacement height d and the roughness lengths for momentum (z_M) and heat ($z_H = 0.2 z_M$) are calculated from vegetation height and the total plant area index for the canopy following Raupach (1994). Latent heat is given by

$$L = \frac{\lambda \tilde{\rho}}{r_V p_A}(e_C - e_A)$$

where λ is the latent heat of vapourization ($J \cdot mol^{-1}$), p_A atmospheric pressure (kPa), e_C and e_a are the effective vapour pressure of the canopy exchange surface and air respectively (kPa). The resistance to vapour exchange (r_v) is given by $r_v = r_H a + r_c$ where r_c is bulk surface stomatal resistance ($s \cdot m^{-1}$), calculating following Kelliher *et al* (1995) from solar radiation and a user-specified minimum stomatal resistance (see below).

Following van Wijk & de Vries (1963) and Campbell and Norman (2012), a closed-form equation for the ground heat flux is derived by assuming that the diurnal and annual cycles in ground surface temperature are approximately sinusoidal and that soil properties, including heat conductance are moderately uniform within the soil profile. The flux is then proportional to the amplitude of these cycles in manner dictated by soil physical properties and water content, but exhibits a phase shift, with the maximum heat flux density occurring 1/8 cycle before the maximum temperature. This means that soil surface temperatures peak later in the day or year than the peak in solar radiation. The method is elaborated on below.

The energy balance of the canopy is then solved for temperature by linearising the emitted radiation and latent heat terms using the Penman-Monteith equation

$$T_c = T_A + \frac{R_{abs} - \varepsilon_C \sigma T_A^4 - \tilde{\rho} \lambda \theta_C \frac{D}{p_A r_V} - G}{\tilde{\rho} \left(c_p K_{H_R} + \frac{\lambda}{r_V} s \theta_C \right)}$$

where D is the vapour pressure deficit of the atmosphere (kPa), $s = \Delta / p_A$ where Δ is the slope of the saturated vapour pressure curve calculated using Tetens equation, θ_C is an effective relative humidity of the canopy surface to accomodate drought conditions and

$$K_{Hr} = \frac{4 \varepsilon_C \sigma T_A^4}{\tilde{\rho} c_p + 1 / r_{Ha}}$$

is radiative conductance. To derive the ground heat flux, the temperature (T_G) of the ground surface is also derived using the Penman-Monteith equation

$$T_G = T_A + \frac{R_{abs}^G - \varepsilon_G \sigma T_A^4 - \tilde{\rho} \lambda \theta_G \frac{D}{p_A r_{Ha}} - G}{\tilde{\rho} \left(c_p K_{H_R} + \frac{\lambda}{r_{Ha}} s \theta_G \right)}$$

where R_{abs}^G is radiation absorbed by the ground surface, ε_G the emissivity of the ground surface and θ_G is the effective relative humidity of the ground surface calculated from soil moisture.

Temperature (T_Z), vapour pressure (e_Z) and wind speed (u_Z) at any given height z above canopy are then computed as

$$T_z = T_A + (T_C - T_A)\left(1 - \frac{\log_e \frac{z-d}{z_H}}{\log_e \frac{z_R-d}{z_H}}\right)$$

$$e_z = T_A + (\theta_C e_C - e_A)\left(1 - \frac{\log_e \frac{z-d}{z_H}}{\log_e \frac{z_R-d}{z_H}}\right)$$

$$u_z = \frac{u_*}{\kappa}\left(\ln \frac{z-d}{z_M} + \psi_M(z)\right)$$

where

$$\psi_M(z) = \psi_M(Z_R)\frac{\ln \frac{z-d}{z_M}}{\ln \frac{z_R-d}{z_M}}$$

Following Raupach (1989a,b), the air temperature T_z at height z below canopy is given $\tilde{\rho} c_p T_z = C_f(z) + C_n(z)$ where $C_f(z)$ is a far-field component and $C_n(z)$ a near-field component. Assuming the canopy to comprise i layers each at height z_i, the far field component at height z is given by

$$C_f(z) = C(h) - C_n(h) + \int_z^h \frac{H(z_i)}{K_H(z_t)} dz$$

where $C(h) = \tilde{\rho} c_p T_H$ is the heat concentration at the top of the canopy at height h, with T_h being temperature at the top of the canopy, $C_n(h)$ is the near field-component at the top of the canopy (see below), $K_H(z_i)$ is the thermal diffusivity at height z_i given by $K_H = \sigma_\omega^2 T_L$ and $H(z_i)$ is the sensible heat flux at height z_i approximated by $H(z_i) = \tilde{\rho} c_p / r_L (T_{c_i} - T(z_i)) + F_G$ where F_G is the heat flux from the ground. Here T_{c_i} is the temperature of canopy elements such as leaves derived using the Penman-Monteith equation by considering the energy balance of the leaf and $r_L =$ is the boundary layer resistance of the leaf given by $318\sqrt{0.71 L_w / u_z}$, where L_w is leaf width (m) and u_z is wind speed given, from Harman & Finnigen (2008) as

$$u_z = u_h \exp\left(\frac{\beta}{L_M}(z-h)\right)$$

Here u_h is wind speed at the top of the canopy at height h, $\beta = u_*/u_h$ and L_M is a mixing length given by $L_M = 2\beta^3 L_C$ where $L_C = (c_d a)^{-1}$ with $a = P_{AI}/h$ and $c_d = 0.25$

The near field component at height z is given by

$$C_n(z) = \int_0^\infty \frac{S(z_i)}{\sigma_\omega(z)} k_n \left[\frac{z - z_i}{\sigma_\omega(z_i) T_L} + \frac{z + z_i}{\sigma_\omega(z_i) T_L}\right] dz_i$$

where $S(z_i)$ is the source concentration given by $f_d(z_i) H(z_i)$ where $f_d(z_i)$ is foliage density for layer z_i (m^3/m^3) and k_n is a kernal function given by $k_n(\zeta) = -0.39894 \ln(1 - e^{|\zeta|}) - 0.152623 e^{|\zeta|}$ where $\zeta = (z - z_i)/\sigma_\omega(z_i) T_L$.

No precise formulations for T_L and $\sigma_\omega(z_i)$ are given by Raupach, but a plausible profile is proposed as

$$T_L = \frac{a_2 h}{u_*}$$

$$\sigma_\omega(z_i) = u_* \left[0.75 + 0.5 \cos\left(\pi\left(1 - \frac{z}{h}\right)\right)\right]$$

Following Ogée *et al* (2003), a value for a_2 can be derived by making thermal diffusivity equivalent to its above canopy formulation at height $z = h$ such that

$$a_2 = \frac{\kappa\left(1 - \frac{d}{h}\right)}{1.5625}\phi_H$$

where ϕ_H is a diabatic correction for stability.

Owing to several inter-dependances in the model – e.g. between G and T_G, between leaf and air temperature and between the diabatic coefficients and sensible heat fluxes, the model needs to be run iteratively to convergence. This is computationally intensive when seeking to model microclimate over large areas at high resolution. Thus when running the grid model, a point model is run iteratively first and outputs from the point model used as basis for running the grid model without iteration. Individual components of the model and methods for avoiding iteration when applying the grid model, are now described in more detail.

Radiation and albedo

The albedo of the vegetated surface can be defined as the proportion of shortwave radiation that is reflected – i.e. total downward flux less that absorbed. It is dependent upon both the fraction absorbed by the canopy and by underlying ground surface. Whereas the reflectance of a surface - the ratio of the radiant flux reflected from a material to the incident radiant flux is an innate property of the surface, the albedo of a vegetated surface depends both the properties of the vegetation and the underlying ground surface, but also changes in relation to the transmission of radiation through the vegetation and hence also on the angle of the reflecting surfaces relative to the solar beam. It therefore changes as a function of both the sun's position and the fraction of direct radiation relative to diffuse radiation.

From Sellers (1985) and Yuan *et al* (2017), but with adaptation allowing for radiation to pass through larger gaps in the canopy, albedos are derived using a two-stream radiative transfer model. The two-stream model is also used to derive the radiative flux within the canopy and the energy balance of foliage elements within the canopy. We therefore describe the model in full.

The underpinning assumption of a two-stream model is that solar radiation arrives at the top of the canopy in the form of either diffuse or direct radiation. Because the leaves are partially reflective and translucent, a portion of the direct radiation at any given point within the canopy is scattered in either an upward or downward direction.The a portion of the diffuse radiation is likewise scattered both upward and downward. Consequently there are several radiation streams that need to be derived: (i) Downward direct, (ii) downward diffuse, (iii) upward diffuse, (iv) the contribution of downward direct to downward diffuse and (v), the contribution of downward direct to upward diffuse. There is is no upward direct, as any radiation upward is assumed to be scattered isotropically and therefore diffuse. The albedo is determined from the upward and downward streams at the top of the canopy, but it is also possible to derive upward and downward streams within the canopy.

Downward direct radiation

It is convenient to envisage the canopy as being divided into multiple layers, each with a series of nodes i situation at the top of each layer and numbered sequentially downward from the top of the canopy. Let us combine the effects of leafs and other canopy elements such as branches and tree trunks and let P_i be the cumulative area of these canopy constituents per unit ground area (from the top of the canopy) to node i and let P_{i+1} be the cumulative area at the node below. From e.g. Campbell (1985), the attenuation of radiation at node i is then described as $R_i^{\downarrow b} = R_0^{\downarrow b}\exp(-KP_i)$ where $R_i^{\downarrow b}$ is the downward flux of direct radiation that would be intercepted by a surface perpendicular to the solar beam at node i, $\downarrow R_0^{\downarrow b}$ the is the downward flux of direct radiation intercepted perpendicular to the solar beam at the top of the canopy, and K is a radiation extinction coefficient, which depends on the inclination angle of canopy elements relative to the solar beam (vertically orientated leaves will block out more sunlight when the sun is low above the horizon).

However, rather than defining each canopy element individually, it is more convenient to characterise the canopy as comprising surfaces that have a continuous distribution of inclinations. From Campbell (1986) and Campbell (1990), it is reasonable to assume that the real distribution of inclinations can be approximated by assuming they conform to a prolate or oblate spheroid distribution. By adjusting the ratio of horizontal to vertical axes of the spheroid, canopy element angle distributions of any canopy from erectophile to planophile can be simulated. Defining x as the ratio of average projected elements on horizontal surfaces, such that $x = 0$ for a vertical distribution, $x = \infty$ for a horizontal distribution and $x = 1$ for a spherical distribution, K is approximated as

$$K \approx \frac{\sqrt{x^2 + \tan^2 Z}}{x + 1.774(x + 1.182)^{-}0.733}$$

where Z is the solar zenith angle (see below). In the special case where $x = 1$, the equation can be simplified to $1/(2\cos Z)$ and when $x = 0$, $K = 2/(\pi \tan(0.5\pi - Z))$. When $x \to \infty$, $K \to 1$ and therefore does not depend on the sun's position.

This assumes canopy elements are small and randomly distributed. In reality, canopy elements are non-randomly distributed and radiation can penetrate through larger gaps in the canopy unimpeded. We thus define a gap fraction (F_G) representing the fraction of the canopy that is unobscured by vegetation when the sun is at its zenith. The probability of the sunbeam's path through the canopy being unobscured diminishes as the optical path length increases, but the extent to which it does depends on the shape of the vegetation clumps. It is assumed that the shape of the clumps is dictated by the inclination angles of leaves such that canopies with planophile leafs will typically have planophile clumps of vegetation.

A correction must also be applied when the canopy overlies an inclined ground surface. While it can be assumed, for a given habitat type, that the inclination of canopy elements is unaffected by ground surface inclination, the optical path length through the canopy will be shorter for ground surfaces facing in the direction of the sun. A full equation for the transmission of direct radiation through the canopy is thus given by

$$R_i^{\downarrow b} = R_0^{\downarrow b}\left[{\tau_i}^b + (1 - {\tau_i}^b)\exp(-\tilde{K}\tilde{P}_i)\right]$$

where ${\tau_i}^b = F_G^{m_z K_C}$ is the fraction of direct radiation transmitted through larger canopy gaps, $K_C = \tilde{K}/K(0)$ and $\tilde{K} = K(Z)\cos Z/s_c$, where $K(Z)$ and $K(0)$ are the values of K derived using the equation above, when the solar zenith angles are Z and 0 respectively, $m_z = (h - z_i)/h$ adjust the gap transmission for height above ground z_i relative to total canopy height h and $\tilde{P}_i$ is the cumulative canopy element area now concentrated into 'non-gaps' given by $\tilde{P}_i = P_i/(1 - F_G)$ and s_c is the fraction of direct beam radiation intercepted by an inclined surface given by

$$s_c = \cos Z \cos s + \sin Z \sin s \cos(\Lambda_s - \Lambda)$$

where s the slope angle of the surface measured from horizontal, Λ_s the solar azimuth (direction from north) and Λ the aspect of the surface (relative to north). Slopes and aspects are calculated from digital elevation data, but set to zero when running the point model.

The position of the sun

To calculate radiation absorption in real environments, it is necessary to calculate the solar zenith and azimuth. The solar zenith (Z) depends on latitude ϕ_y and time as follows

$$\cos Z = \sin\delta \sin\phi_y + \cos\delta \cos\phi_y \cos h_s$$

where h_s is the hour angle in solar time (t_s) given by

$$h_s = (\pi \,/\, 12)(t_s - 12)$$

and δ is the current declination angle of the sun calculated from the Astronomical Julian day (j) as

$$\delta = \frac{\pi 23.5}{180}\cos\left(2\pi\frac{j - 159.5}{365.25}\right)$$

The Astronomical Julian day is the continuous count of days since the beginning of the Julian period. The date 1st Jan 2022 has an Astronomical Julian day of 2459581. The solar time is calculated from longitude (ϕ_x) and local time (t_l) as

$$t_s = t_l + \frac{4(\phi_x - \phi_m) - \Delta_t}{60}$$

where ϕ_m is the longitude of the local time zone meridian (e.g. 0 for Greenwich Mean Time) and Δ_t is the equation of time – a correction applied to account obliquity due to tilt of the Earth's rotational axis and for the east-west component of the analemma, namely the angular offset of the Sun from its mean position on the

celestial sphere due the eccentricity of the Earth's orbit. These two factors have different wavelengths, amplitudes and phases that vary over geological timescales. From Milne (1921)

$$\Delta_t = -7.659 \sin(M + 9.683) \sin(2M - 3.5932)$$

where M is the mean anomaly given by $6.24004077 + 0.01720197(j - 2451545)$.

The solar azimuth (Λ_s) is calculated as

$$\sin \Lambda_s = -\frac{\sin h_s \cos \delta}{\sin Z}$$

Diffuse radiation

The diffuse radiation streams can be written using two sets of differential equations:

$$-\frac{dR_i^{d\uparrow}}{dP} = -(a+\gamma)R_i^{d\downarrow} + sR_i^{b\downarrow}$$

$$\frac{dR_i^{d\downarrow}}{dP} = -(a+\gamma)R_i^{d\downarrow} + \gamma R_i^{d\uparrow} + s' R_i^{b\downarrow}$$

Here $R_i^{d\uparrow}$ is upward diffuse radiation (i.e. that reflected upward by canopy elements and the ground surface, $R_i^{d\downarrow}$ is downward diffuse radiation (analogous to the processes described for direct radiation above) and $R_i^{b\downarrow}$ is downward direct radiation.

Each of the equation's components describes a different physical process. In both equations, the left hand term describes the overall attenuation of upward or downward radiation through the canopy. The first right hand term in the top equation describes the fraction of upward diffuse radiation that is re-scattered in upward direction and the equivalent in the bottom equation describes the fraction of downward diffuse radiation that is re-scattered downward. a is the absorption coefficient for incoming diffuse radiation per unit leaf area and γ is the backward scattering coefficient. The second term on the right hand side in the top equation describes the fraction of the downward diffuse radiation that is converted into an upward diffuse flux by back-scattering and the equivalent in the bottom equation describes the fraction of the upward diffuse radiation that is converted into an downward diffuse flux by back-scattering.The final term on the right hand side of the top equation refers to the contribution to the upward diffuse flux by the scattering of direct radiation penetrating into the canopy to i. The equivalent in the bottom equation is the contribution from scattering of direct radiation to the downward diffuse flux. s and s' are thus the backward and forward scattering coefficient for direct radiation respectively.

The term a is given by $a = (1 - \omega)$, where ω is the single scattering albedo of individual canopy elements given by $\omega = \alpha_P + \tau_P$ where α_P is canopy element reflectance and τ_P is canopy element transmittance, both provided as user inputs.

The term γ is given by $\gamma = 0.5(\omega + J\delta)$ where δ is given by $\delta = \alpha_P - \tau_P$ and J is an integral function of the inclination distribution of canopy elements approximated by $J = \cos^2 \hat{\Theta}_P$ where $\hat{\Theta}_P$ is the mean inclination angle of canopy elements in the zenith direction. From Campbell (1990) $\hat{\Theta}_P = 9.65(3 + x)^{1.65}$ and from Verhoef (1984) and Pinty ***et al*** (2006) $s = 0.5(\omega + \frac{J\delta}{K})K$. The forward scattering coefficient for incident direct radiation, s' is given by $s' = \omega K - s$.

Solving the differential equations for a two-stream model is not entirely straightforward, but is achieved by setting boundary conditions at the bottom and the top of the canopy. A full explanation of their derivation is given in Sellers (1985) and Yuan ***et al*** (2017). Only the final equations are presented here. These are as follows:

$$R_i^{d\uparrow} = R_0^{d\downarrow}\left[(1-\tau_i^d)(p_1 \exp(-h\tilde{P}_i) + p_2 \exp(h\tilde{P}_i)) + \alpha_G \tau_i^d\right] + R_b^{d\uparrow}$$

$$R_i^{d\downarrow} = R_0^{d\downarrow}\left[(1-\tau_i^d)(p_3 \exp(-h\tilde{P}_i) + p_4 \exp(h\tilde{P}_i)) + \tau_i^d\right] + R_b^{d\downarrow}$$

where $R_i^{d\uparrow}$ is upward diffuse radiation and $R_i^{d\downarrow}$ is downward diffuse at i, $R_0^{d\downarrow}$ is downward diffuse at the top of the canopy, α_G is ground reflectance. Here $\tau_d = {F_G}^{2m_z}$ is transmission of diffuse radiation through larger gaps in the

canopy and $R_i^{db\uparrow}$ and $R_i^{db\downarrow}$ are the contribution of direct radiation to the diffuse upward and downward flux respectively given by

$$R_i^{db\uparrow} = R_0^{b\downarrow}\left[(1-\tau_i^b)\left(\frac{p_5}{\sigma_p}\exp(-K\tilde{P}_i) + p_6\exp(-h\tilde{P}_i) + p_7\exp(h\tilde{P}_i)\right) + \alpha_G\tau_i^b\right]$$

and

$$R_i^{db\downarrow} = R_0^{b\downarrow}\left[\left(1-\tau_i^b\right)\left(\frac{p_8}{\sigma_p}\exp(-K\tilde{P}_i) + p_9\exp(-h\tilde{P}_i) + p_{10}\exp(h\tilde{P}_i)\right)\right]$$

The equations for calculating albedos are given by

$$\alpha_d = \left(1-\tau_0^d\right)\left(p_1+p_2\right) + \tau_0^b$$

and

$$\alpha_b = \left(1-\tau_0^d\right)\left(\frac{p_5}{\sigma_p} + p_6 + p_7\right) + \tau_0^d$$

where α_d is white-sky albedo (i.e. Bi-Hemispherical Reflectance: that reflecting diffuse radiation), α_b is black-sky albedo (i.e. Directional Hemispherical Reflectance - that reflecting direct radiation) and τ_0^d and τ_0^b are the values of τ_i^d and τ_i^b with $m_z = 1$. The total albedo is given by

$$\alpha = f_d\alpha_d + (1-f_d)\alpha_b$$

where f_d is the diffuse radiation fraction given by $R_0^{d\downarrow}/R_{sw}^{\downarrow}$.

The derivation of the terms $p_{1..10}$ is given below.

Diffuse components:

$$p_1 = \frac{\gamma}{D_1 S_1}(u_1 - h)$$

$$p_2 = \frac{-\gamma S_1}{D_1}(u_1 + h)$$

$$p_3 = \frac{1}{D_2 S_1}(u_2 + h)$$

$$p_4 = -\frac{S_1}{D_2}(u_2 - h)$$

$$D_1 = \frac{1}{S_1}(a+\gamma+h)(u_1-h) - S_1(a+\gamma-h)(u_1-h)$$

$$D_2 = \frac{1}{S1}(u_2+h) - S_1(u_2-h)$$

$$S_1 = \exp(-hP_{AI})$$

$$S_2 = \exp(-KP_{AI})$$

$$u_1 = a + \gamma(1 - \frac{1}{\alpha_G})$$

$$u_2 = a + \gamma(1-\alpha_G)$$

$$h = \sqrt{a^2 + 2a\gamma}$$

Direct components:

$$p_5 = -s(a+\gamma-K) - \gamma s\text{'}$$

$$p_6 = \frac{1}{D_1}[\frac{v_1}{S_1}(u_1-h) - S_2 v_2(a+\gamma-h)]$$

$$p_7 = -\frac{1}{D_1}[v_1 S_1(u_1+h) - S_2 v_2(a+\gamma+h)]$$

$$p_8 = s\text{'}(a+\gamma+K) - \gamma s$$

$$p_9 = -\frac{1}{D_2}[\frac{p_8}{\sigma_p S_1}(u_2+h) + v_3]$$

$$p_{10} = -\frac{1}{D_2}\left[\frac{p_8 S_1}{\sigma_p}(u_2 - h) + v_3\right]$$

$$v_1 = s - \frac{p_5(a + \gamma + K)}{\sigma_p}$$

$$v_2 = s - \gamma - \frac{p_5}{\sigma_p}(u_1 + K)$$

$$v_3 = S_2(s' + \gamma\alpha_G^* - \frac{p_8}{\sigma_p}(u_2 - K))$$

$$\sigma_p = K^2 + a^2 - (a + \gamma)^2$$

Here α_G^* is the blue sky albedo of the ground surface, effectively the the ground reflectance α_G adjusted for inclination given by $\alpha_G \cos Z / s_c$

Longwave and emitted radiation

Leaves and other elements of the canopy typically have very low reflectance to longwave radiation so there is limited scattering of longwave radiation. In consequence, the longwave radiation streams within the canopy can be thought of as originating from three sources. Firstly from the ground surface, where incident longwave radiation at any given height below canopy is determined by ground temperature and the tranmission of radiation from the ground surface. The second is from all the individual canopy elements, which in turn is determined by the temperature of those surfaces and tranmission from those surfaces. The third is downward from the sky, which is determined by the effective sky temperature and tranmission from the sky.

Dividing the canopy into n layers and defining ΔP_i as the total one sided plant area within a layer i, the downward flux of radiation $R_j^{\downarrow lw}$ at j is given by

$$R_j^{\downarrow lw} = \tau_{j,h}^d R_{lw}^{\downarrow} + \sum_{i=j}^{i=n}\left(\Delta\tilde{P}_i \tau_{i,j} \varepsilon_i \sigma T_i^4\right)$$

and the upward flux $R_j^{\uparrow lw}$ is given by

$$R_j^{\uparrow lw} = \tau_{0,j}^d \varepsilon_G \sigma T_G^4 + \sum_{i=0}^{i=j}\left(\Delta\tilde{P}_i \tau_{i,j}^d \varepsilon_i \sigma T_i^4\right)$$

where $\tau_{i,j}^d$ is the transmission of longwave radiation between i and j given by

$$\tau_{i,j}^d = F_G^{2m_{i,j}} + \left(1 - F_G^{2m_{i,j}}\right)\exp\left(-\tilde{P}_{i,j}\right)$$

where $m_{i,j} = |z_i - z_j|/h$ adjust the grap transmission for distance from i to j and $\tilde{P}_{i,j} = |\tilde{P}_i - \tilde{P}_j|$ is the is the total one sided plant area between i and j. Here ε_G and ε_i are the emissivities of the ground surface and canopy elements respectively (both set at 0.97), σ the Stefan–Boltzmann constant and T_G and T_i are absolute temperatures of the ground surface and canopy elements respectively in Kelvin. Here plant areas $\tilde{P}$ refer to the plant areas concentrated into non-gaps.

In order to calculate the longwave radiation incident on a canopy element one must know the vertical temperature profile of canopy elements, but in order to calculate this, one must know the longwave radiation

absorbed by that element. Resultantly the model is run iteratively.

Emitted radiation is given by

$$R_{lw}^{\uparrow} = \varepsilon_C \sigma T_C^4$$

The linearisation of emitted radiation, required to solve for temperature in the Penman-Monteith equation assumes

$$\varepsilon_C \sigma T_C^4 \approx \varepsilon_C \sigma \left(T_A^4 + 4T_A^3(T_C - T_A)\right) = \varepsilon_C \sigma T_A^4 + \frac{\tilde{\rho} c_p}{r_R}(T_L - T_A)$$

where the radiative resistance r_R is given by

$$r_R = \frac{\tilde{\rho} c_p}{4\varepsilon_C \sigma T_A^3}$$

Leaf and ground radiation absorption

The two-stream model also used to calculate radiation absorbed by the ground surface ($R_{abs}(G)$), given by:

$$R_{abs}(G) = (1 - \alpha_G)\left(R_{i=0}^{d\downarrow} + s_c R_{i=0}^{b\downarrow}\right) + \varepsilon_G R_{j=0}^{\downarrow lw}$$

where ε_G is the emissivity of the ground surface.

Radiation absorption by canopy elements at height i, averaged over the upward and downward-facing surfaces, is given by

$$R_{abs}(L_i) = (1 - \alpha_L)\left(0.5(R_i^{d\downarrow} + R_i^{d\uparrow}) + 0.5 s_L (R_i^{\downarrow b} + R_i^{\uparrow b}) + 0.5\varepsilon_L (R_{j=i}^{\downarrow lw} + R_{j=i}^{\uparrow lw})\right)$$

where ε_L is the emissivity of canopy elements and $s_L = K / \cos Z$

Radiation adjustments in the grid model

In the grid model, downward radiation is adjusted for terrain such that $R_0^{\downarrow b}$ is set to zero if solar altitude is less than the horizon angle ($\theta_h(\Lambda_s)$) in azimuth direction Λ_s, and $R_0^{\downarrow d}$ and $R_0^{\downarrow lw}$ are adjusted by sky view factor s_v

The horizon angle is computed for each grid cell as

$$\theta_h(\Lambda_s) = \max_i \left(\tan^{-1}\left(\frac{z_i - z_0}{d_i}\right)\right)$$

where z_0 is the elevation of the focal grid cell z_i is the elevation of the grid cell in direction i and d_i the horizontal distance from the focal grid cell to grid cell i. The maximum is taken over 10 points of increasing distance from the focal grid cell along a given solar azimuth direction Λ_s.

The sky view factor is computed as

$$s_v = 0.5 \cos(\tilde{h})$$

where

$$\tilde{h} = \frac{1}{24}\sum_{i=1}^{24} \theta_h\left(\frac{2\pi i}{24}\right)$$

Sensible Heat

From Fourier's and Fick's laws, the diffusive eddy transport of heat is described by $H = \tilde{\rho} c_p (dT/dz)$ where $\tilde{\rho}$ is the molar density of air ($mol \cdot m^{-3}$), $dT\ /\ dz$ the rate of change in temperature T (K) with height z (m) and K_H is eddy thermal diffusivity ($m^2 \cdot s^{-1}$). However, in practical terms, as reference air temperature (i.e. that recorded by

a weather station) is measured at some height above canopy, it is more convenient to describe the sensible flux in the form $H = (\tilde{\rho} c_p \,/\, r_{Ha})(T_H - T_A)$ where r_{Ha} is the resistance to heat transfer ($s \cdot m^{-1}$), the average of the inverse of thermal diffusivity derived by integration between two heights.

Treating the vegetated surface as a single vertically homogeneous layer of phytomass, eddy thermal diffusivity is given by $K_H = \kappa u_* (z - d) \phi_H(z)$ where κ is the Von Kármán constant (~0.4), $\phi_H(z)$ is a diabatic influencing factor for heat and u_* is the friction velocity of wind ($m \cdot s^{-1}$) given by

$$u_* = \frac{\kappa u_{z_R}}{\ln \frac{z_R - d}{z_M} + \psi_M(z_R)}$$

where u_{z_R} is the wind speed ($m \cdot s^{-1}$) at height z_R (that at which provided wind speed is measured), d is the zero plane displacement height (m) – the height at which the wind profile above canopy extrapolates to zero, z_M is a roughness length (m) for momentum and $\phi_M(z_R)$ is a diabatic correction coefficient for momentum (see below). From Raupach (1994)

$$d = h \frac{1 - (1 - \exp(-\sqrt{7.5 P_{AI}}))}{\sqrt{7.5 P_{AI}}}$$

and

$$z_M = (h - d) \exp\left(\frac{-\kappa}{\beta} - \psi_H(z_R) \right)$$

where h is the height of the canopy (m), β is given by $\sqrt{0.003 + 0.1 P_{AI}}$, where P_{AI} is the total one side plant area (living + dead vegetation) per unit ground area and $\psi_H(z_R)$ is a diabatic correction factor for heat.

Substituting the equation for thermal diffusivity into that for eddy transport of heat, and integrating the resulting equation from the canopy heat exchange surface to the height at which air temperature is measured, gives the equation for the bulk surface aerodynamic resistance:

$$r_{Ha} = \frac{\ln \frac{z_R - d}{z_H} + \psi_H(z_R)}{\kappa u_*}$$

where $z_H = 0.2 z_M$ is a roughness length for heat.

Diabatic effects

From Harman & Finnigen (2008) the diabatic influencing factors and coefficients feature when the canopy surface is strongly heated or cooled and are given by

$$\psi_M(z) = \psi_M \frac{z_M}{L_O} - \psi_M \frac{z - d}{L_O}$$

$$\psi_H(z) = \psi_H \frac{z_H}{L_O} - \psi_H \frac{z - d}{L_O}$$

where from Businger *et al* (1971)

$$\psi_M(\zeta) = \ln \left[\left(\frac{1 + x}{2} \right)^2 \left(\frac{1 + x^2}{2} \right) \right] - 2 \arctan x + \frac{\pi}{2}$$

with $x = (1 - 15\zeta)^{1/4}$ under unstable conditions ($H > 0$) and

$$\psi_M(\zeta) = -4.7\zeta$$

under stable conditions ($H < 0$). The equivalent formulae for heat are given by

$$\psi_H(\zeta) = \ln \left[\left(\frac{1 + y}{2} \right)^2 \right]$$

With $y = 1 \,/\, \phi_H = \sqrt{1 - 9\zeta}$ under unstable conditions and $\psi_H(\zeta) = -4.7\zeta / 0.74$ under stable conditions. Here L_0 is the Obukhov length given by

$$L_O = -\frac{\tilde{\rho} c_p u_*^3 \overline{T}}{\kappa g H}$$

where g is the gravitational constant (~9.81) and $\overline{T}$ is the average temperature of the height profile (K), taken as the average of the canopy and the air.

From Yasuda (1988)

$$\sqrt{\phi_H} = \frac{1}{(1 - 16\zeta)^{1/4}}$$

under unstable conditions and

$$\phi_H = 1 + \frac{6\zeta}{1 + \zeta}$$

under stable conditions. To avoid unrealistic reversals of the temperature profiles caused by an extreme diabatic influence, the diabatic coeffcients are capped such that

$$|\psi_M| < 0.9 \ln \frac{z_R - d}{z_M}, \quad |\psi_H| < 0.9 \ln \frac{z_R - d}{z_H}$$

Additionally, to avoid the diabatic influencing factor for heat resulting in a negative roughness length for momentum, it is capped to

$$\psi_H < 0.9 \frac{\kappa}{\beta}$$

Grid model simplifications

The issue with deriving the sensible heat flux in a computationally efficient manner is that the wind friction velocity (u_*), needed to derive H, depends on the diabatic coefficients, but these in turn depend on H and u_*. It is therefore necessary to run the model iteratively until convergence. This is computationally intensive if seeking to do so over numerous grid cells. The approach in `microclimf` is therefore to run point model iteratively to convergence for a flat reference location R at the centre of the study area with mean (or modal) soil and vegetation properties. The wind friction velocity of this reference location (u_*^R) can then be directly related to the friction velocity of any other location (u_*) such that

$$u_* = \mu_u \mu_*^R$$

where μu is given by

$$\mu_u = \frac{u_*}{\mu_*^R} = \frac{(\kappa u_{z_r}) \,/\, \left(\ln \frac{z_r - d}{z_M} + \psi_M(z_r)\right)}{(\kappa u_{z_r}) \,/\, \left(\ln \frac{z_r - d_R}{z_{M_R}(z)} + \psi_{M_R}(z_r)\right)} w_c = \frac{\ln \frac{z_r - d_R}{z_{M_R}} + \psi_{M_R}(z_r)}{\ln \frac{z_r - d}{z_M} + \psi_M(z_r)} w_c$$

where here u_{z_r} is wind speed at some reference height z_r such that it is near identical at all locations, d and d_T the zero-plane displacement heights for the target and reference locations respectively, z_M and z_M^R the roughness lengths for the target and reference and target locations respectively $\psi_M(z_r)$ and $\psi_{M_R}(z_r)$ the respective diabatic coefficients.

It can be shown empirically (Fig. S1) that effects of the diabatic coefficients on this ratio are relatively minor and therefore

$$\mu_u \approx \frac{\ln(z_r - d_R) - \ln z_{M_R}}{\ln(z_r - d) - \ln z_M} w_c$$

Thus, once u_*^R has been derived iteratively, the friction velocity for any other location can be derived from the roughness lengths and zero-plane displacement heights (and hence from plant area and vegetation height),

without the need to run the model iteratively across the entire landscape. The final term, w_c is a terrain wind shelter coefficient that is applied to target locations computed, following Ryan (1977) as

$$w_c = 1 - \frac{\arctan 0.17 H_W}{1.65}$$

where H_W is the horizon angle in an upwind direction computed using digital elevation data.

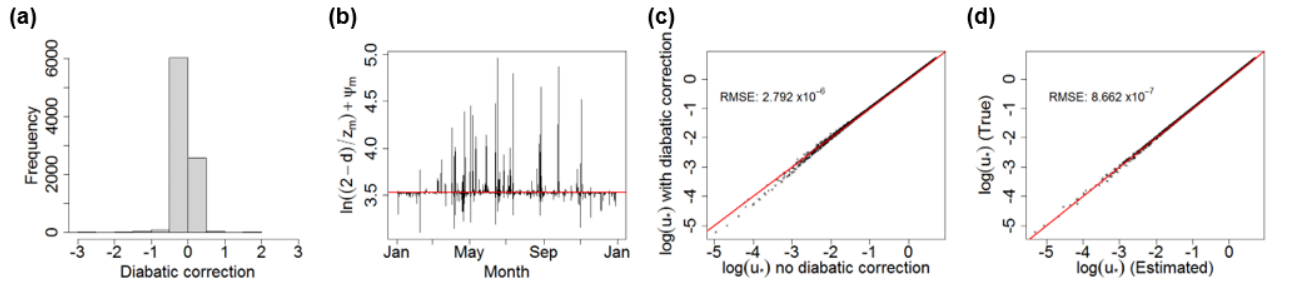


Fig. S1. Effects of diabatic coefficients on the estimation of friction velocity. Here the iterative point microclimate model is run for a reference location with a plant area index of 3 and vegetation height of 1m and for a target location with a plant area index of 1 and a vegetation height of 0.5 over the course of an entire year using the inbuilt weather datasets for 2017 from Caerthillian Cove in Cornwall. In (a) a frequency histogram of the diabatic coefficients for the target location are shown. In (b) the vegetation-driven determinant of friction velocity $\ln((2-d)/z_M) + \Psi_M$ for the target location is plotted over the course of the year, with the red line depicting $\Psi_M = 0$. In (c) the log of friction velocity for the target location with and without a diabetic correction applied are plotted against one another, demonstrating that friction velocity is under-estimated when wind speeds are low. In (d) the friction velocity estimated using the ratio approach described above (x-axis) is plotted against the true friction velocity calculated by running the point microclimate iteratively (y-axis). The ratio approach significantly improves estimates of friction velocity, particularly when wind speeds are low.

A small issue is that the diabatic coefficient for momentum (ψ_M) is needed to compute wind speeds at user-specified above canopy (z). A similar ratio approach is used for doing so such that

$$\psi_M(z) = \psi_{M_R}(z) \frac{\ln(z) - \ln(d - z_M)}{\ln(z) - \ln(d_R - z_M^R)}$$

where $\psi_{M_R}(z)$ is the diabatic correction coefficient for the reference location. This eliminates the need to derive ψ_M iteratively.

Latent Heat

A full expression for the latent heat flux from the canopy is given by

$$L = \frac{\lambda \tilde{\rho}}{p_a r_v}(\theta_C e_C - e_a)$$

where λ, the latent heat of vaporisation is given by

$$\lambda = 45068.7 - 42.8428T,\ T \geq 0;\ \ \lambda = 51078.69 - 4.338T - 0.06367T^2,\ T < 0$$

where here T is the average of the surface and air temperature measured in degrees: $T = (T_a + T_c) + 273.15$

p_A is atmospheric pressure (kPa), θ_C is the effective relative humidity of the surface and e_c the temperature-dependent effective relative humidity of the surface (kPa) given from Tetens equation by

$$e_c = 0.61078 \exp\left(\frac{17.27(T_c - 273.15)}{T_C - 35.85}\right)$$

where T_C is measured in Kelvin or

$$e_c = 0.61078 \exp\left(\frac{17.27 T_c}{T_C + 237.3}\right)$$

where T_C is measured in degrees. When T_c measured in degrees is below zero

$$e_c = 0.61078 \exp\left(\frac{21.875 T_c}{t_c + 265.5}\right)$$

In `microclimf`, θ_C is set to 1.0 immediately following rain and 0.8 at other times. In application of this formula in the Penman-Monteith equation, air temperatures are capped to not drop below dewpoint. The resistance to vapour exchange is given by $r_v = r_{Ha} + r_c$ where r_c is bulk surface stomatal resistance.

To a reasonable approximation (Raupach 1994; Kelliher *et al* 1995), r_c is the inverse parallel sum of the stomatal resistances of individual leaves and the soil combined, the former, which under ample root water supply, non-extreme temperatures and low humidity deficit, varies only in response to variation in photosynthetically active radiation. Self-shading by leaves largely compensates for the presence of additional foliage contributing to the parallel sum under densely foliated canopies and under sparse canopies, evaporation from soil surface largely compensates for lack of evapotranspiration from foliage. Thus, following (Kelliher *et al* 1995), a simple relationship between bulk surface stomatal conductance and photosynthetically active radiation (PAR) can be derived as

$$\frac{\tilde{\rho}}{r_c} = 3 g_{mx} \frac{Q_a}{Q_a + 3 Q_{50}}$$

where $Q_a = 4.6 R_{sw}^{\downarrow}$ is an estimate of PAR absorption in $\mu mol \cdot m^{-2} \cdot s^{-1}$, $R_{sw}^{\downarrow}$ is downward shortwave radiation ($W \cdot m^{-2}$), $Q_{50} = 100$ is the value of Q, when stomatal conductance is at 50 percent of its maximum and g_{mx} is a user provided maximum stomatal conductance, provided as molar conductance ($mol \cdot m^{-2} \cdot s$) to ensure conformance with Körner (1994) who gives values for major vegetation types of the globe. The formulation $r_c = \tilde{\rho} / g_c$ converts between stomatal resistance ($s \cdot m^{-1}$) and stomatal conductance.

To derive stomatal resistance for individual canopy elements (r_L), the formulation is

$$\frac{\tilde{\rho}}{r_L} = g_{mx} \frac{Q_a}{Q_a + Q_{50}}$$

For the soil, there is no stomatal resistance, and conductance for vapour is given by $r_v - r_H a$. The effective relative humidity of the soil is approximated by $\theta_G = (S_M - S_{mn})/(S_{mx} - S_{mn})$ where S_M is the volumetric soil moisture fraction, S_{mx} the saturated water fraction and S_{mn} the residual soil moisture fraction. The later to are provided as user inputs, though an inbuilt dataset allows one to estimate them from soil type.

The linearisation of the latent heat term to solve for temperature in the Penman-Monteith equation assumes that

$$\frac{\lambda \tilde{\rho}}{r_V p_A}(e_c - e_a) = \frac{\lambda \tilde{\rho}}{r_V p_A}(e_c - e_s) + \frac{\lambda \tilde{\rho}}{r_V p_A}(e_s - e_A) \approx \frac{\lambda \tilde{\rho}}{r_V p_A}(\Delta(T_C - T_A) + D)$$

where $D = e_s - e_a$ is the vapour deficit of the atmosphere where e_s is saturated vapour pressure and Δ is the slope of the saturated vapour pressure curve given by $\Delta = de_s/dT$

The soil model

Ground heat flux

Following van Wijk & de Vries (1963) and Campbell and Norman (2012), the ground heat flux G at time t (in seconds) is deriving by assuming that the diurnal and annual cycles in ground surface temperature are approximately sinusoidal such that

$$G = \frac{\sqrt{2} A_D(0) k_s \sin[\omega_D (t - t_0(D)) + \pi/4]}{D_D} + \frac{\sqrt{2} A_A(0) k_s \sin[\omega_A (t - t_0(A)) + \pi/4]}{D_A}$$

where A_D and A_A are the amplitude in the diurnal and annual temperature cycles, k_s thermal conductivity ($W \cdot m^{-1} \cdot K^{-1}$), ω_D and ω_A are the angular frequencies, given by $\omega_D = 2\pi/(24 \times 3600)$ and $\omega_D = 2\pi/(24 \times 3600 \times n)$ where n is the number of days in the year. Here D_D and D_A are the diurnal and annual damping depths given by $D_D = \sqrt{2\kappa/\omega_D}$ and $D_A = \sqrt{2\kappa/\omega_A}$ where κ is thermal diffusivity given by $\kappa = k_s/\rho_s c_s$ where ρ_s and c_s are the volumetric density and specific heat of the soil respectively.

From Campbell (1986), thermal conductivity (k_s) is given by

$$k_s = c_1 + c_2 S_M - (c_1 - c_4)\exp(-c_3 S_M^4)$$

where

$$c_1 = \frac{0.57 + 1.73Q_v + 0.93M_v}{1 - 0.74Q_v - 0.49M_v} - 2.8(Q_v + M_v)(1 - (Q_v - M_v))$$

$$c_2 = 1.06\rho_D S_M,\ \ c_3 = 1 + 2.6C_M^{-1/2},\ \ c_4 = 0.03 + 0.7(M_v + Q_V)^2$$

where ρ_D is the bulk density of soil ($Mg \cdot m^{-3}$), Q_v and M_v the volumetric quartz and mineral content of the soil respectively, C_M the mass fraction of clay and S_M the volumetric water fraction of the soil.

The volumetric density (ρ_s) and specific heat (c_s) of the soil are then given as follows

$$\rho_s = 1000(\rho_D(1 - S_M) + S_M),\ \ c_s = \frac{2400\rho_D}{2.64} + 4180S_M$$

Typical values of ρ_D, Q_v, M_V and C_M for given soil types are shown in Table 1.

Table 1. Soil properties for given soil types. This table is stored as an internal dataset in micro. Here ρ is the bulk density of the soil, Q_V and M_V volumetric quartz and mineral content respectively, C_M the mass fraction of clay, $\theta_{S_{mx}}$ and $\theta_{S_{mn}}$ the saturated and residual soil moisture fraction, K_{sat} the saturated hydraulic conductivity and $s_{1..4}$ coefficients for the hydrological model.

Soil type	ρ_D	Q_V	M_V	C_M	θ_{mx}	θ_{mn}	K_{sat}	s_1	s_2	s_3	s_4
Sand	1.59778	0.3	0.3	0.01	0.399	0.049	501.12	0.037449	0.0002939	0.00302	0.9631
Loamy sand	1.58708	0.24	0.355	0.035	0.402	0.054	146.88	0.03875	0.0003021	0.00832	1.03755
Sandy loam	1.57898	0.18	0.41	0.06	0.403	0.058	62.208	0.024034	0.0001907	0.00654	0.66748
Loam	1.51351	0.12	0.44	0.0844	0.422	0.074	31.968	0.023019	0.0001898	0.00903	0.80757
Silt loam	1.35864	0	0.47	0.124	0.447	0.067	16.416	0.020029	0.0001721	0.02607	1.57229
Sandy clay loam	1.61751	0.14	0.464	0.3648	0.388	0.089	10.368	0.021304	0.0001847	1.19126	4.07721
Clay loam	1.52964	0.06	0.509	0.5422	0.419	0.091	5.5296	0.021303	0.0001912	0.05977	1.13477
Silty clay loam	1.47251	0.04	0.508	0.3948	0.441	0.089	3.6288	0.019098	0.0001758	0.13255	1.90744
Silty clay	1.64224	0.15	0.4655	0.505	0.381	0.103	2.8512	0.016682	0.0001737	0.09953	0.82008
Clay	1.67068	0	0.624	0.55	0.368	0.073	2.16	0.014268	0.0001494	0.14717	1.15604

Grid model simplifications

When running the model as a point model, because the ground heat flux depends on ground surface temperature and vis-versa, a solution is found by iteration. To avoid the need to iterate the model over all grid cells, it assumed that the ground heat flux (G) at any given target location scales to the ground heat flux (G_R) at a reference location by a parameter μ_G such that

$$G = \mu_G G_R,\ \mu_G = \frac{G}{G_R}$$

Further, it is assumed that because the ground heat flux associated with daily temperature cycles is much greater than that associated with annual cycles

$$\mu_G \approx \frac{\sqrt{2}A_D(0)k_s \sin[\omega_D(t - t_0(D)) + \pi/4]/D_D}{\sqrt{2}A_D^R(0)k_s^R \sin[\omega_D(t - t_0(D)) + \pi/4]/D_D^R} \approx \frac{A_D(0)' k_s D_D^R}{A_D^R(0)' k_s^R D_D}$$

where k_s, k_s^R, D_D and D_D^R are the thermal conductivities and damping depths for the target and reference location respectively. The remaining challenge is then to calculate the approximate diurnal temperature amplitudes $A_D^R(0)'$ and $A_D(0)'$. This is achieved by using the Penman-Monteith equation, but with G and G_R set to zero and diabatic corrections ignored. This avoids inter-dependencies and hence the need to derive values iteratively. The accuracy of this approximation method is shown in Fig S2.

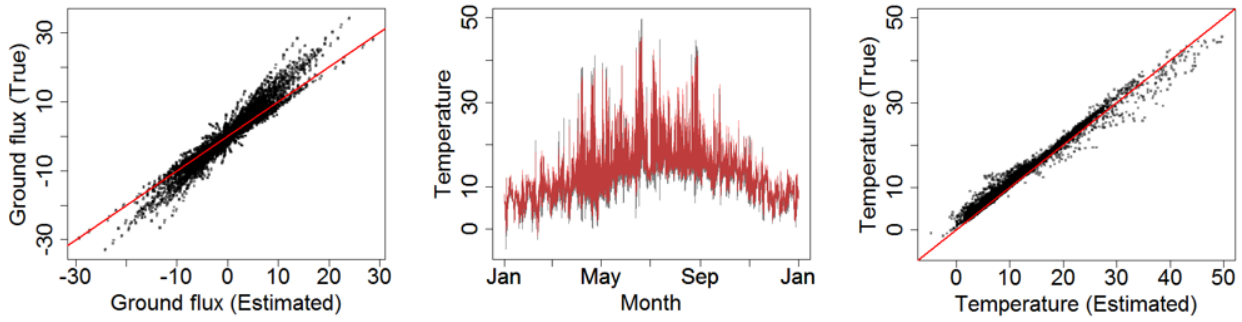


Fig S2. Comparisons of ground heat flux (a) and soil surface temperature (b, c) estimates derived using the full model described for a reference point location (true) and using the simpler approximation estimate applied when deriving across the study area (estimated). In (b) over-laid time-series are shown, with the full method depicted by red lines and simpler method shown as grey lines. In (c) estimates derived using the two methods are plotted against one another. In both (a) and (c), the red lines are the 1:1 relationship.

Temperatures below ground

In the soil, heat storage is significant, and both the annual and diurnal temperature cycles are attenuated and occur later in the day or year. Assuming that the annual and diurnal cycles are approximately sinusoidal, that the soil layer is infinitely deep and with uniform thermal properties, from Campbell & Norman (2012) and de Vries (1963), the temperature at depth z and time t is given by

$$T_z(t) = \overline{T_G} + A_D(0)\exp\left(\frac{z}{D_D}\right)\sin\left[\omega(t - t_0(D)) + \frac{z}{D_D}\right] + A_A(0)\exp\left(\frac{z}{D_A}\right)\sin\left[\omega(t - t_0(A)) + \frac{z}{D_A}\right.$$

where $\overline{T_G}$ is mean annual temperature of the ground surface (K), z is depth below ground (m) and negative, A_D and A_A are the amplitude in the diurnal and annual temperature cycles and D_D and D_A are the diurnal and annual damping depths. It can be shown empirically, that the above relationship is approximated well by the following relationship:

$$T_z(t) = \frac{1}{n}\sum_{i=t-n+1}^{i=t} T_G(i)$$

where $\sum_{i=t-n+1}^{i=t} T_G(i)$ is the rolling mean of the ground surface temperature over the last n hours. The number of hours over which the rolling mean is calculated is contingent on depth and is given by

$$n = -118.35\frac{z}{D_D}$$

The assumption of vertically uniform soil properties and an approximately sinusoidal diurnal cycle in ground surface temperature is necessary in order to get a computationally efficient closed-form solution to the ground heat flux equation and to the estimation of below-ground temperatures. An alternative approach, following e.g. Campbell (1985) is to divide the soil into multiple layers and to compute fluxes and temperatures seperately for each layer. This is the approach adopted by the NicheMapR package as described by Kearney & Porter (2017). Comparisons of the simplified approach adopted by `microclimf` with those obtained by the multi-layer soil model in NicheMapR are shown in Fig. 1. The two methods produce remarkably similar results.

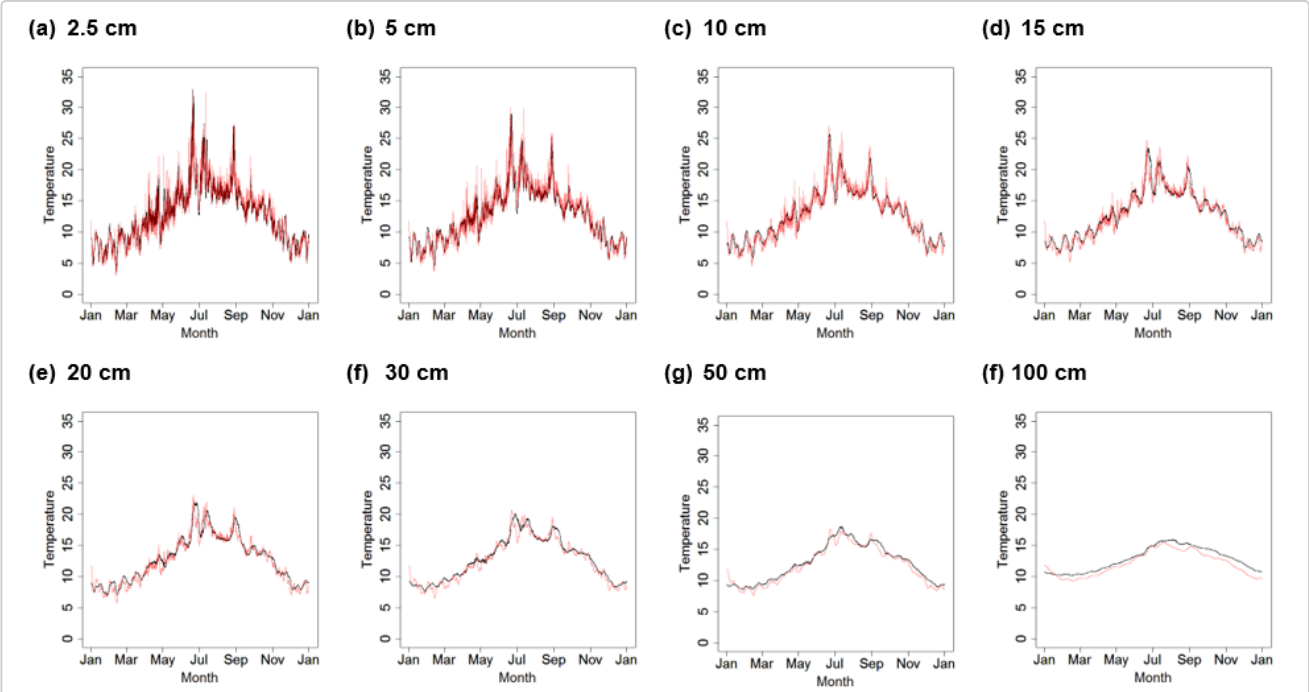


Fig. S3 Predictions of soil temperature made using the multi-layer soil model included in NicheMapR (red) and the simpler rolling-mean approach implemented in microclimf (black)

Where the model input data encompass a full year, but don't match a given calander year, the annual cycle in temperatures is simply recycled to accommodate $i < 1$. I.e. to compute temperatures below ground in early January, data from late December are used. However, the `microclimf` package allows for the microclimate model to be run for periods of time that extend for less than a year or for select days only, making it impossible to calculate a rolling mean of ground surface temperatures than extends back further than the period for which ground surface temperatures have been calculated. It is recommended that a complete time sequence for a year or more is modelled when seeking to derive temperatures below ground. However, when this is not the case, the following approximation method is used:

$$T_z(t) = T_G,\ n \leq 1$$

$$T_z(t) = \omega_1 T_G + (1 - \omega_1)\tilde{T}_z(D),\ 1 < n \leq 24$$

$$T_z(t) = \omega_2 \tilde{T}_z(D) + (1 - \omega_2)\tilde{T}_z(A),\ 24 < n < h_y$$

$$T_z(t) = \tilde{T}_z(A),\ n \geq h_y$$

where where h_y is the number of hours in a year, $\tilde{T}_z(A)$ is mean annual air temperature computed if data for an entire year are available or alternatively provided as a user input, and

$$\omega_1 = \frac{1}{1 + n^2/24},\ \omega_2 = \frac{24}{24 + n^2/h_y}$$

and $\tilde{T}_z(D)$ is an estimate of the mean daily temperature given by

$$\tilde{T}_z(D) = \omega_3(T_z^R - \tilde{T_z^R}(D)) + \tilde{T_G}(D) - \tilde{T_G^R}(D)$$

where $\tilde{T_z}^R(D)$ and $\tilde{T_G}^R(D)$ are daily mean temperature at depth $-z$ and the gorund surface respectively derived for a reference location using the point model and

$$\omega_3 = \frac{A_D(0)}{A_D^R(0)}$$

where $A_D(0)$ and $A_D^R(0)$ are the amplitude of the diurnal cycles in ground surface temperature derived from the grid model and point model respectively.

Soil water point model

An estimate of soil water content is required to estimate the thermal heat capacity and conductivity of the ground. In the point version of the model, `microclimf` implements a modified version of the Mahrt and Pan (1984) two-layer model of soil hydrology. The volumetric soil moisture fraction at each daily time-increment j in a top shallow soil layer is computed as

$$\theta_j = \theta_{j-1} - s_1 P - s_2 R_{netp}$$

where θ_{j-1} is soil moisture in the previous time increment, P is daily precipitation, R_{netp} is the daily mean of hourly positive net radiation. Soil moistures are capped so that θ_j cannot exceed the volumetric water fraction at saturation (S_{mx}) or drop below the residual volumetric water fraction for a given soil type (S_{mn}).The assumption is thus that daily evaportransporation is linearly related to R_{netp} and thus that the Bowen ratio remains relative constant during the day when net radiation budget is positive.

At each daily time increment, water is then exchanged with an underlying deeper soil layer of 10 times the volume as follows:

$$\theta_j = \theta_{j-1} + s_3 k \left(\theta_j^d - \theta_j \right)$$

$$\theta_j^d = \theta_{j-1}^d + 0.1 s_3 k \left(\theta_j - \theta_j^d \right)$$

where θ_j^d is the soil moisture in underlying deeper soil layer at time j and k is the hydraulic conductivity given by

$$k = k_{sat} \left(\frac{\overline{\theta_j}}{\theta_{mx}} \right)^{s_4}$$

where k_{sat} is saturated hydraulic conductivity, $\overline{\theta_j}$ is the mean volumetric water content of the two soil layers given by $(\theta_j + \theta_j^d)/2$. The terms $s_{1..4}$ are coefficients estimated by iteration for given soil types using the outputs of the more complex multi-layer hydrological model given in Campbell (1986) and included in the NicheMapR package (Kearney & Porter 2017). Examples of the model output in comparison to a more complex multi-layer model in NicheMapR are shown in Fig. S4.

Typical values of S_{mx}, S_{mn} and k_{sat} and estimated values for $s_{1..4}$ for given soil types are shown in Table 1.

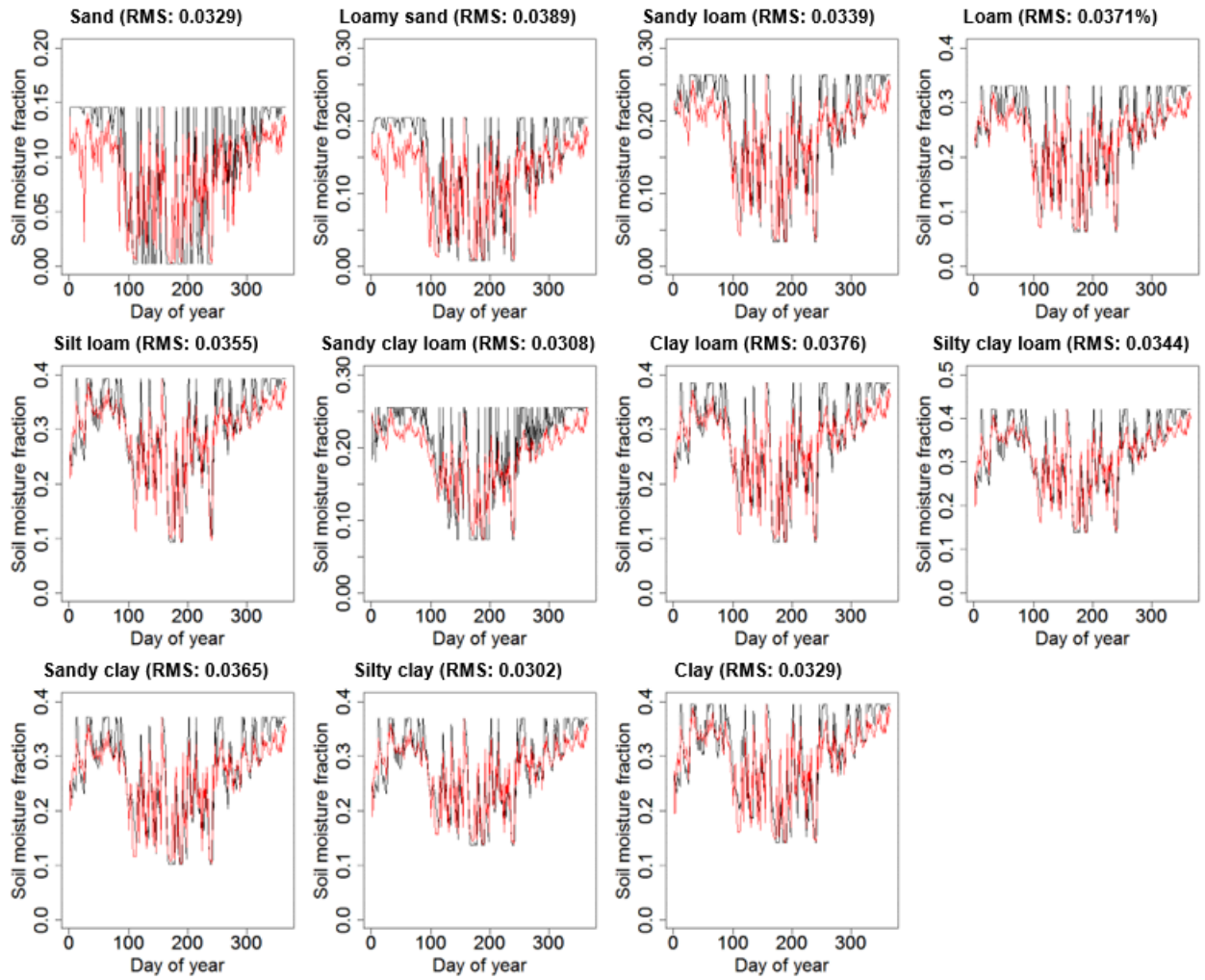


Fig S4. Comparison of soil moisture (0 – 5 cm depth) estimates derived using the simplified method in microclimf (black) with those obtained using NicheMapR (red). Estimates are for 2017 from Caerthillian Cove in Cornwall, U.K, (49.968°N, 5.215°W) with an assumption of no canopy shading. Data are shown for 11 soil types. Root-Mean-Square (RMS) differences are also indicated.

Soil water spatial model

To avoid the need to evoke a spatially explicit hydrological model, soil moisture derived from the point model is simply distributed spatially using the Bevan & Kirkby (1979) topographic wetness index when running the grid model. Thus soil moisture $\theta_{i,j}$ in each time increment i and grid cell j is thus computed as

$$\theta_{i,j} = \left(\frac{1}{1 + \exp\left(- \ln \frac{\theta'_{i,j}}{1-\theta'_{i,j}} + w_j - \tilde{w} \right)} + \theta_{mn} \right) (\theta_{mx} - \theta_{mn})$$

where $\theta'_{i,j} = (\theta_{i,j} - \theta_{mn})/(\theta_{mx} - \theta_{mn})$ rescales $\theta_{i,j}$ by the soil water fraction at saturation (θ_{mx}) and wilting point (θ_{mn}), $\tilde{w}$ is mean topographic wetness across the study area and w_j is the topographic wetness of grid cell j given by

$$w_j = \ln \frac{a}{\tan S} s$$

where, following Bevan and Kirkby (1979), a is the contributing area, essentially accumulated flow, S is the slope angle and s is a user-defined scaling factor controlling the sensitivity of spatial variation in soil moisture to topographic wetness, by default set to 2/3. An example of the output obtained using this approach, is shown in Fig. S5.

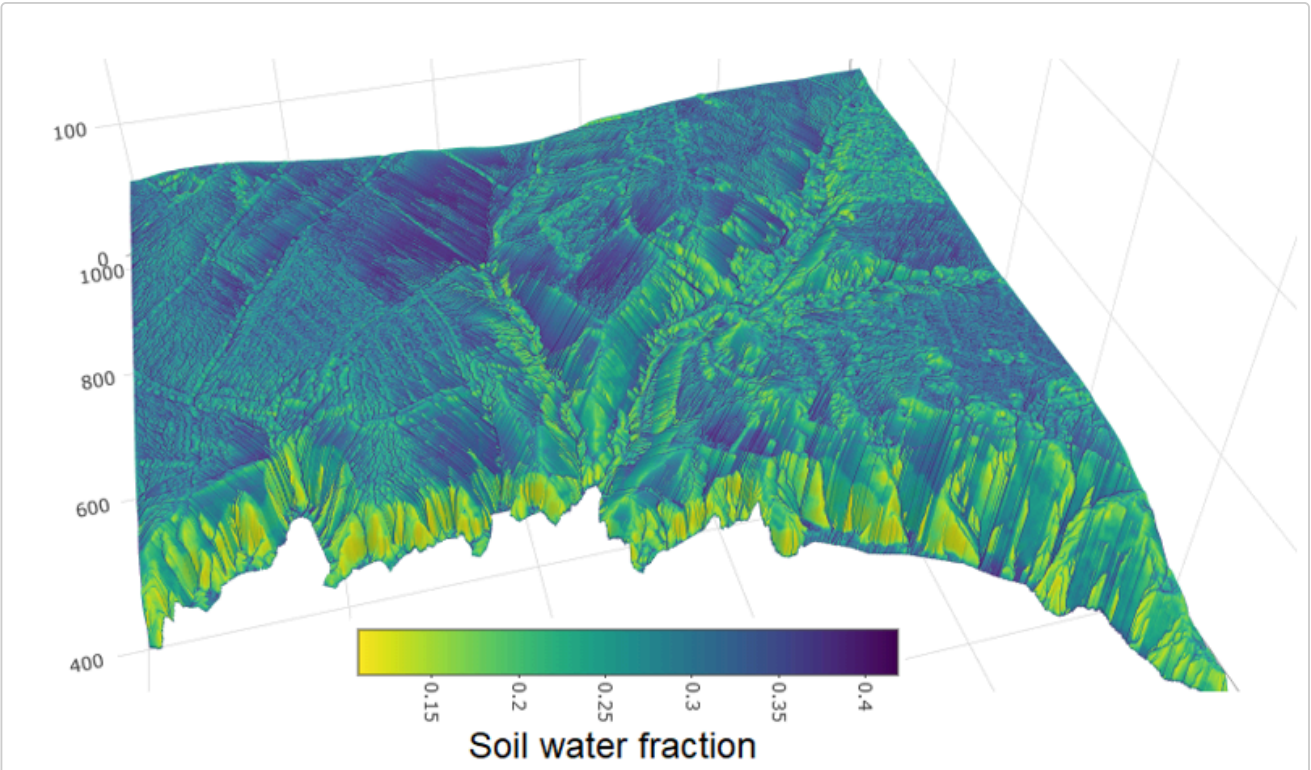


Fig S5. Example output obtained by spatially distributing soil moisture by topographic wetness with θ from the point model set to 0.3 and s = 2/3. Here shown for a 1 km by 1 km area at one m grid resolution, centred on Caerthillian Cove, Lizard, Cornwall (49.968N, 5.216W).

Below canopy model

Conceptual overview

In contrast to above canopy, it is now widely recognised that the transfer of heat and vapour below canopy cannot be predicted using a MOST approach (e.g. Bonan, et al. 2021). To replace MOST in this context, Raupach (1989a,b) developed an analytic Lagrangian theory, which predicts the concentration heat and vapour emanating from a spatially extensive source downwind of the point of interest. In this 'localized near-field' theory (LNZT), the mean concentrations of heat or vapour are expressed as the sum of a diffusive far-field contribution that obeys MOST, and a non-diffusive near-field contribution, which is determined from local sources by assuming the turbulence to be locally homogeneous.

To recap formulae presented above:

$$\tilde{\rho} c_p T_z = C_f(z) + C_n(z)$$

$$C_f(z) = \tilde{\rho} c_p T_h - C_n(h) + \int_z^h \frac{H(z_i)}{K_H(z_i)} dz$$

$$H(z_i) = \frac{\tilde{\rho} c_p}{r_L} (T_{c_i} - T(z_i)) + F_G$$

where $C_f(z)$ and $C_n(z)$ are the far-field and near-field contributions, T_h is air temperature at the top of the canopy ($C_f(h) = \tilde{\rho} c_p T_h$ scales the source concentration to temperature), K_H is thermal diffusivity, r_K resistance to heat loss by canopy elements such as leaves with temperature T_{c_i} and F_G is the ground heat flux.

The formulae can seem quite confusing, but essentially capture four processes. First, near the top of the canopy temperatures will closely match temperatures at the top of the canopy because $\sum_z^h \frac{H(z_i)}{K_H(z_t)} dz \to 0$ as $z_i \to h$ (i.e. there is nothing to sum over). Second, as one descends through the canopy, temperatures will typically increase if the sensible heat flux from canopy elements is positive as would typically be the case when the net radiation budget is strongly positive. This is because $\sum_z^h \frac{H(z_i)}{K_H(z_t)} dz$ increases as $z \to 0$. This seems a bit counter-

intuitive as one typically imagines increased shading to result in cooler temperatures, but the key point is that heat arriving at height z_i is emanating from canopy elements at all heights. However, near the top of the canopy, temperatures are more closely coupled to the air above it. The cooling effect of canopies is the result of two processes: lower down in the canopy, negative sensible heat flux from individual canopy elements that are shaded is possible even if the total flux from the canopy is positive. More importantly though, the ground heat flux comes into play: if the ground surface is cooler than the leaves at the top of the canopy there is a strong cooling effect simply because $F_g \to \infty$ as $z \to 0$ (see below). The last effect arises due the near-field contribution given by

$$C_n(z) = \int_0^\infty \frac{S(z_i)}{\sigma_\omega(z)} k_n \left[\frac{z - z_i}{\sigma_\omega(z_i) T_L} + \frac{z + z_i}{\sigma_\omega(z_i) T_L} \right] dz_i$$

where $S(z_i)$ is the source concentration given by $f_d(z_i)H(z_i)$ where $f_d(z_i)$ is foliage density for layer z_i and k_n is a kernal function, approximated by

$$k_n = -0.39894 \ln(1 - e^\zeta) - 0.15623 e^\zeta$$

where

$$\zeta = \frac{|h - z|}{\sigma_\omega(z_i) T_L}$$

The important take homes from these formulae is canopy elements heated by the sun will warm the air immediately surrounding the elements with the effect influenced strongly by the density of canopy foliage, and diminishing very sharply with distance (as dictated by the shape of the kernal function).

The Langrangian timescale problem

The term Langrangian essential means that the heat or vapour in a canopy can be considered as being emitted from (or absorbed by) a large number of point sources (or sinks), namely the individual leaves and other canopy elements. The spread of the plume from each canopy element is determined by the motion of heat or vapour 'particles' emanating from the canopy element in question. Of key importance is the effect of the persistence of the motions within the canopy on the dispersion properties of these particles, ultimately determined by two parameters: the particle position variance (σ_ω^2) and the Lagrangian time scale T_L, both of which appear in the formulae above, but neither of which are actually known.

However, plausable vertical profiles are proposed by Raupach (1989b) as:

$$T_L = \frac{a_2 h}{u_*}$$

$$\sigma_\omega(z_i) = u_* \left[0.5(a_1 + a_0) + 0.5(a_1 - a_0) \cos\left(\pi \left(1 - \frac{z_i}{h} \right) \right) \right]$$

with $a_1 = 1.25$ (the value of $\sigma_\omega(z_i)/u_*$ at the top of the canopy at height h) and $a_0 = 0.25$. The parameter a_2, though generally taken to be vertically constant within the canopy is effectively unknown. Although it is questionable whether conventional MOST and LNZT theory can be reconciled in this way, following several authors (e.g. Ogée *et al* 2003) a solution to the problem is found by making thermal diffusivity equivalent to its above canopy formulation at height $z = h$. Since above canopy

$$K_H = \kappa u_* (z - d)/\phi_H$$

and below canopy

$$K_H = \sigma_\omega(z_i)^2 T_L$$

Setting $a_1 = 1.25$ and $a_0 = 0.25$, it follows that

$$\sigma_\omega(z_i) = \frac{a_2 h}{u_*} [u_*(0.75 + 0.5\cos(0))]^2$$

Thus

$$a_2 = \frac{\kappa(1-\frac{d}{h})}{1.25^2}\phi_H$$

Thus, to derive below-canopy microclimate, the above canopy component of the model, in which the canopy is treated as a single homogeneous layer of phytomass, is run first. This allows T_L and $\sigma_\omega(z_i)$ and the temperature and humidity at the top of the canopy to be calculated, which then enables derivation of below canopy microclimate.

Ground heat flux below canopy

Ground surface temperature T_G is inferred prior to running the below canopy model and thus, from Fick's law:

$$F_G = \tilde{\rho} c_p \bar{K}_{z,o}(T_G - T_z)$$

where T_z is air temperature at height z and $\bar{K}_{z,o}$ is the mean thermal diffusivity between the ground and height z given by

$$\frac{1}{\bar{K}_{z,0}} = z\int_0^z \frac{1}{K_H}dz$$

Assumptions in grid model

The method described above requires that the canopy is divided into numerous layers each with known foliage density. Since this is both computationally intensive and relies on explicit knowledge of canopy structure, when applying the grid model, the following simplifying assumptions are made.

First, it is assumed that the effects of vertical variation in H and K_H on hte far-field contribution are small, relative to the effects of thermal (or hydric) coupling to the air above canopy. Thus, the canopy contribution to the far-field component can be approximated as

$$\int_z^h \frac{H(z_i)}{K_H(z_i)}dz \approx \frac{H_c}{\bar{K}_{h,0}}\left(1-\frac{z}{h}\right)$$

where H_c is the total combined heat flux from canopy elements and $\bar{K}_{h,0}$ is the average thermal diffusivity within the canopy given by

$$\frac{1}{\bar{K}_{h,0}} = h\int_0^h \frac{1}{K_H}dz =$$

The effects of this assumption are shown in Fig. S6b (dotted lines).

Next, recalling that the vegetated surface is initially treating as a 'big leaf' – effectively a single vertically homogeneous layer of phytomass, it should be noted that *sensu stricto*, this layer comprises both the vegetated canopy *and* the underlying ground surface and that therefore the sensible heat flux H from the entire surface can be broken down into the component fluxes from the canopy (H_C) and underlying ground surface H_G such that $H = H_C + H_G$. To a first approximation it can be assumed that the fraction H_G/H is approximately equal the portion of the ground surface unobscured by canopy elements such that

$$H_G = H\exp(-P_{AI}),\ H_C = H(1-\exp(-P_{AI}))$$

An energy balance equation for air temperature below canopy T_z at height z can then be written as

$$\tilde{\rho} c_p \bar{K}_{z,0}\left(T_G - T_z\right) + \tilde{\rho} c_p \bar{K}_{h,z}\left(T_H - T_z\right) + \tilde{\rho} c_p \bar{K}_{h,0}\left(T_C - T_z\right) = 0$$

yielding

$$T_z = \frac{\bar{K}_{z,0}T_G + \bar{K}_{h,z}T_H + \bar{K}_{h,0}T_C}{\bar{K}_{z,0} + \bar{K}_{h,z} + \bar{K}_{h,0}}$$

where T_H is air temperature at the top of the canopy and T_C is given by

$$T_C = T_H + \frac{H_c}{\tilde{\rho} c_p \bar{K}_{h,0}}$$

The mean thermal diffusivities $\bar{K}_{h,z}$, $\bar{K}_{h,0}$ and $\bar{K}_{h,z}$ are derived by integration:

$$\frac{1}{\bar{K}_{z,0}} = z \int_0^z \frac{1}{K_H(z)} dz = z \frac{2}{a_2 \pi u_*} (p_1 + p_2)$$

where

$$p_1 = \frac{48 \tan \frac{\sqrt{5} f_1}{f_2}}{5^{\frac{3}{2}}}, \; p_2 = \frac{32 f_1}{f_2 \left(\frac{25 f_1}{f_2^2} + 5 \right)}$$

with

$$f_1 = \sin x, \; f_2 = \cos x + 1, x = \frac{\pi z}{h}$$

In the case where $z = h$, this simplifies to

$$\int_0^h \frac{1}{K_H(h)} dz = \frac{4.293251}{a_2 u_*}$$

The effects of these assumptions on the temperatures are shown in Fig S6b (dashed lines)

To compute the near-field contribution, it is recognised that the effects of k_n are such that the near-field concentration at height z are far more influenced by the local source concentration $S(z)$, and the contribution at heights z_i are negligible when $|z - z_i| >> 0$. Thus $C_n(z)$ is determined solely from $S(z)$ with a minor empirically-derived adjustment made to scale $C_n(z)$ to $S(z)$ based on total leaf area such that

$$C_n(z) = S(z)(3.05 + 0.129 \ln P_{AI})$$

The accuracy of all of these approximation methods is shown in Fig S6.

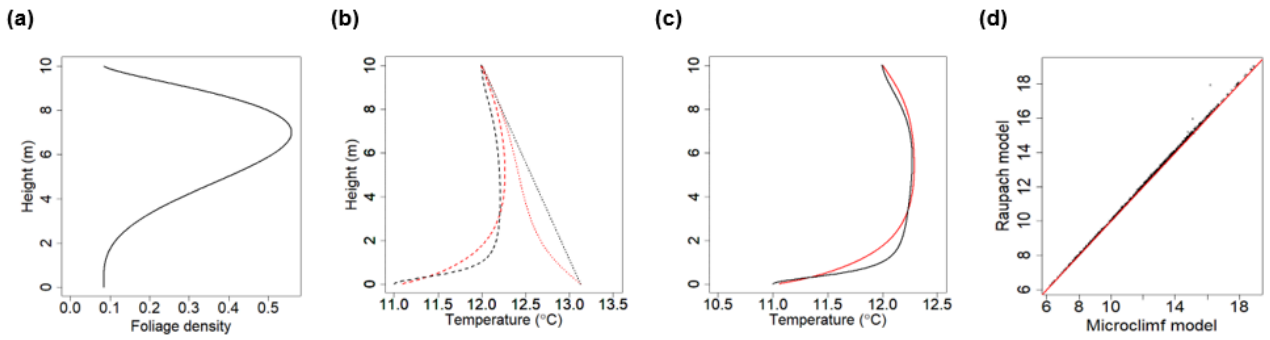


Fig. S6. Comparison between vertical temperature profiles estimated using Raupach's localised near-field model (red) and the simpler model emulator approach in microclimf (black) here shown for a 10m tall canopy with a foliage distribution as shown in (a) and calculated using an hourly weather dataset from Caerthillian Cove, Lizard, UK (49.968°N, 5.216°W) here for a sunny day in May. In (b) the modelled and emulated canopy contribution (dotted lines) and combined canopy and ground contributions (dashed lines) to the far-field component are shown. In (b) the combined near- and far-field contributions are shown. In (d) estimates 5 m above ground are shown for all hours in May (red line = 1:1 relationship)

A near-identical procedure is used for estimated vapour pressure profiles, but with sensible heat fluxes and temperatures substituted for latent heat fluxes and vapour pressure such that

$$e_z = \frac{\bar{K}_{z,0} e_G + \bar{K}_{h,z} e_H + \bar{K}_{h,0} e_C}{\bar{K}_{z,0} + \bar{K}_{h,z} + \bar{K}_{h,0}}$$

where e_z, e_G, e_C and e_H and are the effective vapour pressures at height z, of the ground, of the canopy and at the top of the canopy respectively, and

$$e_c = e_H + \frac{p_A L c}{\tilde{\rho} \lambda}$$

where L_C is the total latent heat flux for the canopy.

The snow model

The modelling of snow broadly follows Kearney (2020) except that the energy and mass budget of the snowpack are computed seperately for the canopy and underlying ground surface.

Snowpack energy balance

The energy balance of both the ground and canopy ground layer are given by

$$R_{abs} - R_{lw}^{\uparrow} - H_S - L_S - G_s - F_M = 0$$

where R_{abs} is radiation absorbed by the snowpack, $R_{lw}^{\uparrow}$ is radiation emitted by the snowpack, H_S is the sensible heat flux, L_S is the latent heat flux due to sublimation, G_S is the rate of heat storage by the snowpack and F_M is the energy flux removed from the pack during melt.

Radiation budget

Radiation absorbed by the canopy is given by

$$R_{abs} = (1 - \alpha_s)R_{sw}^{\downarrow} + 0.97R_{lw}^{\downarrow}$$

where a_s is snow surface albedo calculated as

$$\alpha_s = -9.874 \ln(d_s + 78.3434)$$

where d_s is the number of decimal days since he last snowfall.

To calculate the temperature of the snow-covered ground surface, the radiation budget must accommodate the presence of the canopy. The shortwave component of absorption is calculated using the two-stream model described above, but with the following adjustments made. First, snow is assumed to absorb direct radiation isotropically and in consequence is assumed independent of ground slope and aspect. The coefficient x is also set to one irrespective of leaf inclination angles. Second, both ground (α_G) and leaf (α_P) reflectance are set to snow albedo (α_S). Third, canopy element transmittance (τ_P) is adjusted to accommodate the additional presence of snow such that

$$\tau = \tau_P \tau_S, \;\; \tau_s = \exp(-k_S z_{i_s})$$

where from Warren *et al* (2006) $k_s = 0.1$ in an optical extinction coefficient for radiation transmitted through snow and z_{i_s} is the average snow water equivalent thickness of snow within each canopy layer given by

$$z_{i_s} = \frac{h_{sc}\rho_s}{(h - h_{sg})\rho_w}$$

where h_sG is the depth of ground snow (m) and h_sC is the depth of canopy snow (m).

Lastly, the plant area index is adjusted to accommodate both the presence of both snow lying on the ground and distributed through the canopy as follows

$$P_{AI}^S = \frac{h - g_F h_s}{h} P_{AI} + \frac{(1 - g_F)h_s}{h - g_F h_s}$$

The absorption of longwave radiation is given by

$$R_0^{\downarrow lw} = \tau_c 0.97 R_{lw}^{\downarrow} + (1 - \tau_c)0.97^2 \sigma {T_s}^4$$

where τ_c is calculated as for the plant canopy, but with P_{AI} and h adjusted for snow. Transmission of longwave radiation through the canopy given by

$$R_0^{\downarrow lw} = \tau_0^d 0.97 R_{lw}^{\downarrow} + (1 - \tau_0^d) 0.97^2 \sigma T_s^4$$

where τ_0^d is calculated as for the canopy without snow, but with P_{AI} and h adjusted for snow.

Where $h_s > h$, i.e. snow height exceed vegetation height, canopy attenuation is ignored.

Radiation emitted by the snowpacks is given by

$$R_{lw}^{\uparrow} = 0.97 \sigma {T_{S_C}}^4$$

and

$$R_{lw}^{\uparrow} = 0.97 \sigma {T_{S_G}}^4$$

where T_{S_C} and T_{S_G} are snow surface temperatures for the ground and canopy derived by solving the energy balance equation using the Penman-Monteith equation.

Sensible heat fluxes

The sensible heat fluxes are given by

$$H_S = \frac{\tilde{\rho} c_p}{r_{Ha}} (T_S - T_A)$$

where, as for the canopy $T_s = T_{S_c}$ and for the ground $T_S = T_{S_G}$ and r_{Ha} is the bulk surface aerodynamic resistance of the snow pack given by

$$r_{Ha} = \frac{\ln \frac{z_R - d}{z_H} + \psi_H(z_R)}{\kappa u_*}$$

where u_* is the friction velocity of wind given by $u_* = \kappa u_{z_R} / [\ln \frac{z_R - d}{z_M} + \psi_M(z_R)]$. The key distinctions are that the zero plane displacement height and roughness lengths accommodate the presence of snow and are thus given by

$$d = h \frac{1 - \left[1 - \exp\left(-\sqrt{7.5 P_{AI}^S}\right)\right]}{\sqrt{7.5 P_{AI}^S}}, \quad h_s < h$$

$$d = h_s, \quad , h_s \geq h$$

where h is vegetation height (m), h_s is snow depth given by $h_s = S \rho_w / \rho_s$ where S is snow water equivalent (m), $\rho_w = 1000\ kg \cdot m^{-3}$ is the density of water and ρ_s is the density of snow. Here P_{AI}^S is the snow-covered plant area index above the snow calculated as described above. Following Sturm *et al* (2010) and Kearney (2020), snow density is computed as

$$\rho_s = (\rho_{max} - \rho_0)(1 - \exp(-k_1 h_s - k_2 a_s) + \rho_0$$

where ρ_{max} is the maximum allowable density, ρ_0 is the initial density, a_s is snow age (in decimal days) and k_1 and k_2 are fitting parameters, derived fom the snow environment (Table 2).

Snow environment	$\boldsymbol{\rho_{max}}$	$\boldsymbol{\rho_0}$	$\boldsymbol{k_1}$	$\boldsymbol{k_2}$
Alpine	0.5975	0.2237	0.0012	0.0038
Maritime	0.5979	0.2578	0.0010	0.0038
Prairie	0.5940	0.2332	0.0016	0.0031
Tundra	0.3630	0.2425	0.0029	0.0049
Taiga	0.2170	0.2170	0.0000	0.0000

The roughness length for momentum is computed as

$$z_M = (h - d) \exp\left(\frac{-\kappa}{\beta} - \psi_H(z_R)\right), \quad h > h_s$$

$$z_M = 0.0005, \;\; h_s \geq h$$

where $Be = \sqrt{0.003 + 0.1 P_{AI}^S}$

Latent heat fluxes

Latent heat fluxes are given by

$$L_S = \frac{\lambda \tilde{\rho}}{r_{Ha} p_A}(e_S - e_A)$$

where e_S is the effective vapour pressure of either the canopy or ground surfaces calculated from snow temperatures using Tetens equation and λ is the latent heat of sublimation of the snow pack.

Rate of heat storage by snowpack

As for the diurnal ground heat flux cycle, the rate of heat storage by the canopy and ground snow packs are given by

$$G_s = \frac{\sqrt{2} A_{T_S}(0) k_s \sin[\omega(t - t_0) + \pi/4]}{D_S}$$

where A_{T_S} is the amplitude of the diurnal snow surface temperature cycles, k_s is snow thermal conductivity ($W \cdot m^{-1} \cdot K^{-1}$), ω is the angular frequency given by $\omega_D = 2\pi/(24 \times 3600)$ and D_S is the damping depth given by $D_S = \sqrt{2\kappa/\omega}$ where κ is thermal diffusivity given by $\kappa = k/\rho_s c_s$ where ρ_s and c_s are the volumetric density and specific heat of the snow pack. Snow thermal conductivity is estimated from Djachkova's formula, following Anderson (2006) and Kearney (2020)

$$k_s = 0.0442 \exp(5.181 \rho_s)$$

The Energy Balance equation, excluding F_M is then solved for temperature using the Penman-Monteith equation. If snow surface temperature exceeds zero, the snow pack is then assumed to melt until temperatures attain zero, so snow pack temperature is set to zero. The energy flux removed from the pack during melt is thus given by:

$$F_M = \frac{(T_s - 273.15) c_s \rho_w S}{\Delta_t}$$

where $c_s \approx 2100 J \cdot kg^{-1} \cdot K^{-1}$ is the specific heat of ice and $\Delta_t = 3600 \; s$ is the time-step of the model. Since ρ_w is also approximately constant, the equation simplifies to

$$F_M = 583.3(T_s - 273.15) S$$

The relevance of explicit calculation of F_M is in the calculation of the mass balance.

Snowpack mass balance

The mass balance of the canopy and ground snow models are given by

$$\Delta S_C = I_C - (1 - f_G) M_R - ((1 - f_G) \frac{L_s}{\rho_w \lambda_s} + \frac{F_M}{\rho_w \lambda_f}) \Delta_t$$

$$\Delta S_G = (P_s - I_C) - f_G M_R - (f_G \frac{L_s}{\rho_w \lambda_s} + \frac{F_M}{\rho_w \lambda_f}) \Delta_t$$

where ΔS_C and ΔS_G are the change in canopy and ground snow balance (m snow-water equivelent), P_S is precipitation (m) falling as snow (assumed to do so when air temperature is below freezing), I_C is canopy interception of snow, λ_s and λ_f are the latent heat of sublimation and fusion respectively here measured in $J \cdot kg^{-1}$ where the conversion is given by $\lambda(J \cdot kg^{-1}) = \lambda(J \cdot mol^{-1})/M_w$ where $M_w = 18.015 \times 10^{-3} kg \cdot mol$ is the molar mass of water. Here M_R is rain melt calculated following Anderson (2006) and Kearney (2020) as

$$R_M = 0.0125(T_A - 273.15)P, \;\; T_s > 273.15$$

$$R_M = 0, \;\; T_A \leq 273.15$$

$f_G \approx \exp(-P_{AI}^S)$ partitions sublimation and rain melt between the canopy and the underlying round layer (the portioning of melt is handled explicitely as it is contingent on mass balance of the snowpacks)

Canopy interception

Following Hedstrom & Pomeroy (1998) snow interception by the canopy is given by

$$I_c = 0.678(L^* - L_{t-1}(1 - \exp(-\frac{C_f(0)}{L^*}P))$$

where L^* is the maximum canopy snow load (mm snow water equivalent), L_{t-1} the snow load in the previous time-step and $C_f(0)$ is effective canopy cover perpendicular to the direction of snow fall given by $C_p(0) = 1 - \exp(-KP_{AI}^A)$ where $K = 1/2\cos(Z_s)$ where Z_s is the zenith angle of the direction of snow fall given by $\arctan(\overline{u_c}/0.8)$ where 0.8 m/s is the terminal velocity of snow fall and $\overline{u_c}$ is mean canopy wind speed derived by integrating the wind speed profile below canopy such that

$$\overline{u_c} = \frac{8u_h\beta^2}{P_{AI}^A}(1 - \exp(-\frac{P_{AI}^A}{8\beta^2}))$$

where u_h the wind speed at the top of the canopy is given by $u_h = u_*/\beta$.

Modelling snow depth spatially

By default, the snow model is run in hourly time intervals over every grid cell. To accommodate wind-driven snow accumulation into hollows and crevices in the landscape, every five days snow depth is restributed as follows

$$\tilde{S_G}(z) = \omega_p S_G(z)$$

where $\tilde{S_G}(z)$ is redistributed ground snow depth, $S_G(z)$ is modelled snow depth prior to redistribution and

$$\omega_p = \frac{\omega_T}{\tilde{\omega_T}}$$

where

$$\omega_T = \tilde{z_e}(r) - z_e(p)$$

and $\tilde{\omega_T}$ is mean ω_T across the study area. Here $z_e(p)$ is the elevation of any given grid cell p and $\tilde{z_e}(r)$ is the mean elevation of the area surrounding p within a radius p. The radius is adjusted by mean wind speed across the five days $\tilde{u_z}$ such that

$$r = 10\sqrt{\tilde{u_z}}$$

where r and $z_e(p)$ are in metres and $\tilde{u_z}$ in metres per second.

Approximating snow depth with snapshot data

The microclimf package also has the option to model snow quickly. Here snow melt in any given focal grid cell (M) and over any given time period (Δt) is given by

$$M(\Delta t) = \omega_M M_R(\Delta t)$$

where M_R is snow melt estimated over the equivelent time-period using the point model run in hourly time-steps for a reference location at the centre of the study area and ω_M is a melt factor given by

$$\omega_m = \frac{\sum\left(T_S\right),\ T_S > 273.15}{\sum\left(T_S^R\right),\ T_S^R > 273.15}$$

where $\sum\left(T_S\right),\ T_S > 273.15$ is the sum of snow pack temperatures exceeding freezing over the period Δt, with T_S^R representing the temperature of the snow pack at the reference location and T_S estimated as

$$T_S = s_v\left(T_S^R - T_A\right) + T_A$$

where s_v is the sky view factor and T_A is air temperature. This essentially makes makes two assumptions. First that $s_v = (R_{net})/(R_{net}^R)$ where R_{net} is net radiation at the focal location and R_{net}^R, which is approximately true if snow albedos are similar at the two locations and any azimuthal biases in the reduction of direct radiation by sky view are minimal. Second it assumes that the sensible heat fluxes is directly proportional to radiation, which is approximately true if latent if the rate of heat storage and latent heat are small and increase in a linear manner with net radiation. This is faultier assumption, but the approach is adequate for deriving microclimate estimates (Fig S7).

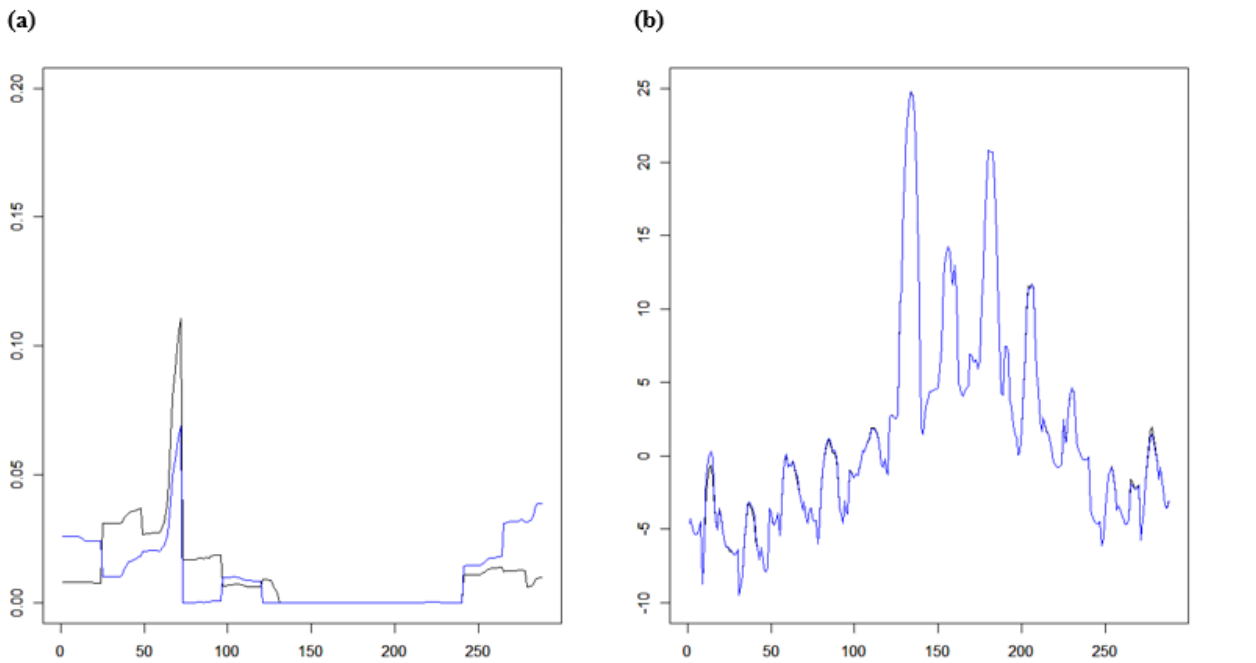


Fig. S7. Comparisons between the (approximate) blue and full (black) snow model run for Caerthillean Cove, Lizard, Cornwall (49.968N, 5.216W), but with air temperatures decreased by 8°C to ensure that temperatures drop below freezing. Values shown are hourly estimates for the coldest day in each month. In (**a**) snow depth (cm) is shown and in (**b**) air temperatures 5 cm above ground. Though the two approaches deviate in their estimation of snow depth, the estimates of air temperature match closely.

Microclimate model with snow

The snow microclimate model forms an extension of the non-snow model and its formulation is nearly identical to that used when snow is absent, except that in hours with snow present, the following adjustments are made.

Firstly, as for calculation of the snow energy balance, ground and leaf reflectance are replaced by snow albedo, the coefficient x is set to 1 and foliage density, canopy height and leaf transmittance adjusted to accommodate the presence of snow.

Secondly, when calculating the latent heat fluxes, vegetation is assumed to be freely evaporating or sublimating. Resistance to heat and vapour loss are thus identical and the additional stomatal resistance is ignored.

Thirdly, when calculating the ground heat flux below canopy, ground snow surface temperature rather than ground surface temperature is used and z_i is adjusted to $z_i - S_G(z)$ where z_i is the user-specified height above ground for which microclimate estimates are produced and $S_G(z)$ is the snow depth of the ground layer.

Fourthly, when modelling microclimate below canopy, heat fluxes are summed only for those canopy layers that are above the height of ground snow.

Lastly, if the height for which model outputs are required lies below the snow pack but above the ground surface, temperatures are calculated as if there were below ground but with the thermal properties of the ground replaced by the thermal properties for snow. Thus

$$T_z(t) = \frac{1}{n} \sum_{i=t-n+1}^{i=t} T_S(i)$$

where

$$n = -118.35 \frac{S_G(z) - z}{D_D}$$

References

Anderson E (2006) *National weather service river forecast system: Snow accumulation and ablation model - SNOW-17*. NOAA Technical Memorandum NWS Hydro-17.

Beven KJ, Kirkby MJ (1979) A physically based, variable contributing area model of basin hydrology. *Hydrological Sciences Journal* 24: 43-69.

Businger JA, Wyngaard JC, Izumi Y, Bradley EF (1971) Flux-profile relationships in the atmospheric surface layer.*Journal of Atmospheric Sciences* 28: 181-9.

Campbell GS (1985) *Soil physics with BASIC: transport models for soil-plant systems*. Elsevier.

Campbell GS (1986) Extinction coefficients for radiation in plant canopies calculated using an ellipsoidal inclination angle distribution. *Agricultural and Forest Meteorology* 36: 317-21.

Campbell GS (1990) Derivation of an angle density function for canopies with ellipsoidal leaf angle distributions. *Agricultural and Forest Meteorology* 49: 173-6.

Campbell GS, Norman JM (2012) *An introduction to environmental biophysics*. 2nd edition. Springer, New York.

de Vries DA (1963) *Thermal properties of soils*. pp 210-235 in Physics of plant environment (eds de Vries DA, van Wijk WR). North-Holland Publishing Company, Amsterdam.

Foken T (2006) 50 years of the Monin–Obukhov similarity theory. *Boundary-Layer Meteorology* 119: 431-47.

Harman IN, Finnigan JJ (2008) Scalar concentration profiles in the canopy and roughness sublayer. *Boundary-Layer Meteorology* 129: 323-51.

Hedstrom NR, Pomeroy JW (1998) Measurements and modelling of snow interception in the boreal forest. *Hydrological Processes* 12: 1611-25.

Kearney MR (2020) How will snow alter exposure of organisms to cold stress under climate warming? *Global Ecology and Biogeography* 29: 1246-56.

Kearney MR, Porter WP (2017) NicheMapR–an R package for biophysical modelling: the microclimate model. *Ecography* 40: 664-74.

Kelliher FM, Leuning R, Raupach MR, Schulze ED (1995) Maximum conductances for evaporation from global vegetation types. *Agricultural and Forest Meteorology*, 73: 1-16.

Körner C (1995) *Leaf diffusive conductances in the major vegetation types of the globe*, pp 463-90 in Ecophysiology of Photosynthesis (eds Schulze ED, Caldwell MM). Springer, Berlin.

Mahrt L, Pan H (1984) A two-layer model of soil hydrology. *Boundary-Layer Meteorology* 29: 1-20.

Ogée J, Brunet Y, Loustau D, Berbigier P, Delzon S (2003) MuSICA, a CO2, water and energy multilayer, multileaf pine forest model: evaluation from hourly to yearly time scales and sensitivity analysis.*Global Change Biology* 9: 697-717.

Pinty B, Lavergne T, Dickinson RE, Widlowski J-L, Gobron N, Verstraete MM (2006) Simplifying the interaction of land surfaces with radiation for relating remote sensing products to climate models. *Journal of Geophysical Research: Atmospheres* 111: D2.

Raupach MR (1989a) Applying Lagrangian fluid mechanics to infer scalar source distributions from concentration profiles in plant canopies. *Agricultural and Forest Meteorology* 47: 85-108.

Raupach MR (1989b) A practical Lagrangian method for relating scalar concentrations to source distributions in vegetation canopies.*Quarterly Journal of the Royal Meteorological Society* 115: 609-32.

Raupach MR (1994) Simplified expressions for vegetation roughness length and zero-plane displacement as functions of canopy height and area index. *Boundary-Layer Meteorology* 71: 211-6.

Ryan BC (1977) A mathematical model for diagnosis and prediction of surface winds in mountainous terrain. *Journal of Applied Meteorology and Climatology* 16: 571-84.

Sellers PJ (1985) Canopy reflectance, photosynthesis and transpiration. *International Journal of Remote Sensing* 6: 1335-72.

Sturm M, Taras B, Liston GE, Derksen C, Jonas T, Lea J (2010) Estimating snow water equivalent using snow depth data and climate classes. *Journal of Hydrometeorology* 11: 1380–94.

van Wijk WR, de Vries DA (1963) *Periodic temperature variations in a homogeneous soil*, pp 102-43 in Physics of plant environment (eds de Vries DA, van Wijk WR). North-Holland Publishing Company, Amsterdam.

Verhoef W (1984) Light scattering by leaf layers with application to canopy reflectance modeling: The SAIL model. *Remote Sensing of Environment* 16: 125–41.

Warren SG, Brandt RE, Grenfell TC (2006) Visible and near-ultraviolet absorption spectrum of ice from transmission of solar radiation into snow. *Applied Optics* 45: 5320-34.

Yasuda N (1988) Turbulent diffusivity and diurnal variations in the atmospheric boundary layer. *Boundary-Layer meteorology* 43: 209-21.

Yuan H, Dai Y, Dickinson RE, Pinty B, Shangguan W, Zhang S, Wang L, Zhu S (2017) Reexamination and further development of two-stream canopy radiative transfer models for global land modeling. *Journal of Advances in Modeling Earth Systems* 9: 113-29.

Running microclimf

Ilya Maclean

2025-04-17

Overview

This vignette describes the R package 'microclimf'. The package contains a series of functions for modelling above and below canopy or below-ground microclimatic conditions across real landscapes, providing gridded outputs. In line with standard approaches for mechanistic microclimate modelling, the model is founded on the principles of energy balance, with the temperature of a surface or the air above it being contingent on how much energy is received or lost. Opaque surfaces in the environment, namely the canopy and the ground, absorb radiation from the sun, but also emit radiation as thermal energy. These surfaces also exchange sensible heat with the surrounding air and undergo latent heat fluxes, namely evaporative and evapotranspirative cooling. Some of the energy is also stored or released by the ground. Because the various components of the energy budget have a dependence on temperature, the temperature of the environment is calculated by assuming that energy budget always remain in balance and then re-arranging the energy balance equations to solve for temperature. However, because of various interdependencies, e.g. between the degree of surface heating and the exchange of sensible heat, and the temperature of the ground surface and the rate of storage by the ground, a closed-form mathematical solution to the energy budget equations cannot be derived. Rather the model must be solved iteratively, which is computational expensive if modelling over multiple grid cells.

A key aim if microclimf is to ensure computational efficiency, which is achieved in four ways. First, it is assumed that the energy budget can be solved mathematically using the Penman-Monteith method (Penman 1948; Monteith 1965) if these interdependencies are ignored, resulting in only modest errors. If doing so for a single point location, the ratio of the temperature offset from ambient air temperature for that location relative to that for any other location is preserved when solving the model iteratively. Thus, running the model iteratively for a single point location and solving the model mathematically for all grid cells, provides a route to estimating the iterative solution for the entire landscape in a computationally efficient manner. Second, and a further advantage of running a point model separately is that one can subset outputs from the point model. While the point model must be run in hourly time-increments there is no need to do so for the grid model: one can instead opt to select from the point model, only those hours that correspond to e.g. the monthly maximum, minimum or median temperature. Thirdly, some simplifying assumptions are about the nature of vegetated canopies to avoid the need to describe vertical variation in leaf foliage density in detail when characterising below-canopy microclimates. This eliminates the need to evoke a multi-layer canopy model. Lastly, most of the heavy lifting is done by c++ code, which typically runs much more quickly than R code. The R functions are essentially wrappers for the underlying c++ code meaning users get the high-level expressiveness of R, but the computational performance of compiled c++.

Package install

Start by installing the package form Github as follows:

```
require(devtools)
install_github("ilyamaclean/microclimf")
```

The package has a few dependencies, which may also need to be installed or updated. If there are any install issue, a useful starting point for troubleshooting is to install the dependencies first. These are `abind`, `ncdf4`, `Rcpp`, `sf`, `stats`, `terra`, `utils` and `zoo` all of which are on CRAN.

Quick start

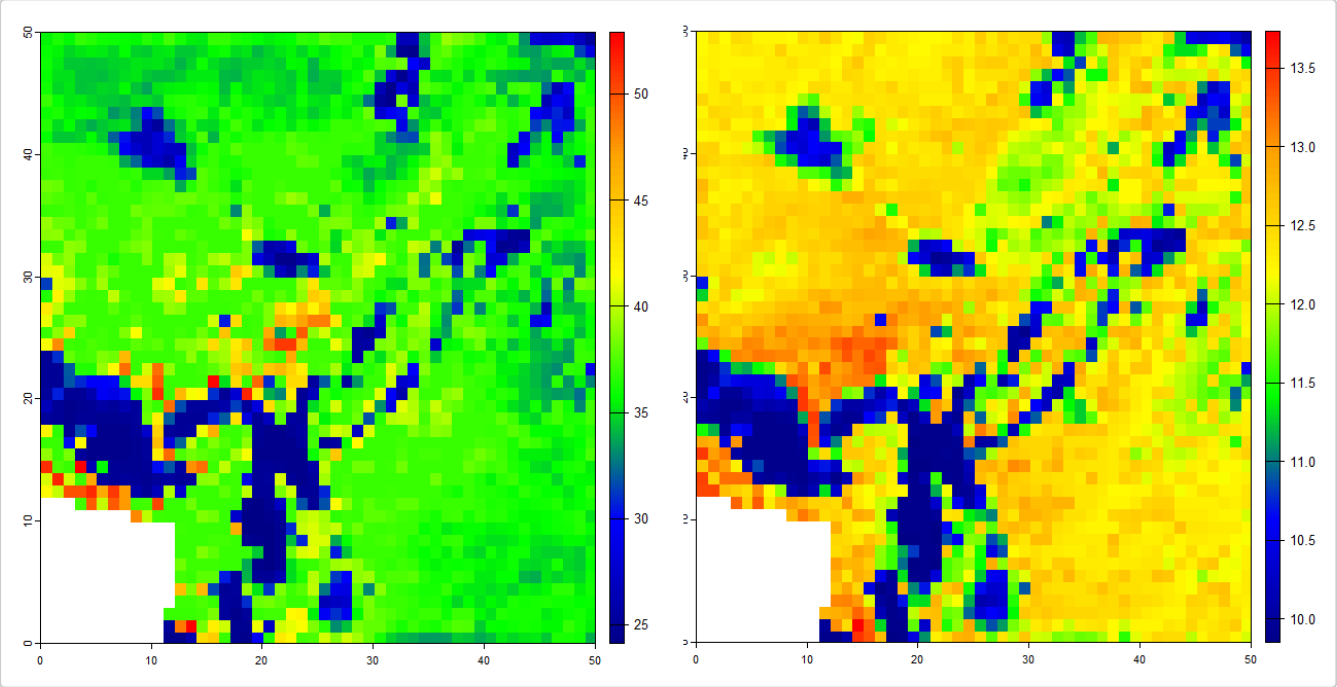
Here I provide brief instructions for how to run `microclimf`. More in-depth instructions are provided below. Four sets of input variables are needed: (1) a dataset of hourly weather, (2) a digital elevation dataset, (3) a dataset of vegetation parameters and (4) a dataset of soil properties. The datasets should have exactly the same format and units as the example datasets included with the package. The spatial resolution and extent of outputs is determined by the spatial resolution of the digital elevation dataset, and the spatial datasets of vegetation parameters and soil properties should also match the digital elevation dataset in terms of resolution and extent. It is important also that the x, y and z dimensions of the digital elevation dataset are equivalent – i.e. an equal area projection is used rather than say latitude and longitude, with units of x, y and height all identical and typically in metres.

The first step to run the point model in hourly time increments using function `runpointmodel`. One then has the option to subset the point model to say return monthly values and pass these as inputs to the grid model. Both the point model and grid model can be run in two modes: either with hourly weather data provided as a data frame in which case the point model is run once only and the weather is assumed identical across the study area. Alternatively the weather data can be provided as multi-layer SpatRasters, in which the point model is run for each grid cell of the SpatRaster and the weather is assumed to vary across the study region.

In the code below, weather data are provided as a data frame. The point microclimate model is run and then subset to return only those hours corresponding to the day in each month with the hottest and coldest temperature (as determined by the point model). These are then passed to the grid model. The model returns temperatures (leaf, ground and air), relative humidity, wind speed, and components of the radiation

budget, all as 3D arrays, representing values for each pixel and time-period. In the final lines of codes, selected outputs are plotted

```
library(microclimf)
library(terra)
# Runs point microclimate model with inbuilt datasets
micropoint <- runpointmodel(climdata, reqhgt = 0.05, dtmcaerth, vegp, soilc)
# Subset point model outputs
micropoint_mx <- subsetpointmodel(micropoint, tstep = "month", what = "tmax")
micropoint_mn <- subsetpointmodel(micropoint, tstep = "month", what = "tmin")
# Run grid model 5 cm above ground with subset values and inbuilt datasets (takes ~20
seconds)
mout_mx <- runmicro(micropoint_mx, reqhgt = 0.05, vegp, soilc, dtmcaerth)
mout_mn <- runmicro(micropoint_mn, reqhgt = 0.05, vegp, soilc, dtmcaerth)
attributes(mout_mx)
# Plot air temperatures on hottest hour in micropoint (2017-06-20 13:00:00 UTC)
mypal <- colorRampPalette(c("darkblue", "blue", "green", "yellow", "orange", "red"))
(255)
plot(rast(mout_mx$Tz[,,134]), col = mypal)
# Plot mean of monthly max and min
mairt<-apply((mout_mn$Tz + mout_mx$Tz) / 2, c(1,2), mean)
plot(rast(mairt), col = mypal)
```



Maximum (right) and mean (left) temperature 5 cm above ground. On warm sunny days, the temperature immediately above sunward facing slopes gets pretty hot. The colder areas are those that have shade cover.

Model inputs

Three sets of parameters are needed to run the model: (i) standard hourly meterological climate-forcing variables representative of macroclimatic conditions across the study site, usually in the form of a data.frame with single values for each hour (though the option to include an array of coarse-gridded values is also available - see below). (ii) A suite of parameters describing properties of the canopy in the form of high-resolution gridded values. (iii) A suite of parameters describing properties of the soil in the form of

high-resolution gridded values. A raster of digital elevation data is also required. Optionally, some additional parameters can be set when running the models, as detailed below.

Each set of parameters is described in turn. Obtaining the right data to drive the microclimate model is often one of the most challenging aspects of modelling microclimate. Users may wish to explore the `microclimdata` package for automated downloading and processing of various datasets available globally or regionally for doing so. The package is available on Github: https://github.com/ilyamaclean/microclimdata

Meteorological data

The inbuilt data,frame `climdata` gives an example of the hourly meteorological variables needed to run the model:

```
library(microclimf)
head(climdata)
#>              obs_time  temp   relhum    pres swdown difrad   lwdown windspeed
#> 1 2017-01-01 00:00:00 7.483 92.02474 101.852      0      0 310.6365     5.895
#> 2 2017-01-01 01:00:00 7.456 94.36533 101.801      0      0 310.8483     5.603
#> 3 2017-01-01 02:00:00 7.244 96.83536 101.765      0      0 309.2949     5.470
#> 4 2017-01-01 03:00:00 7.071 98.12848 101.739      0      0 308.5616     5.540
#> 5 2017-01-01 04:00:00 6.988 98.18165 101.721      0      0 307.3412     5.816
#> 6 2017-01-01 05:00:00 6.948 97.45834 101.710      0      0 307.0171     6.253
#>   winddir     precip
#> 1     283 0.04870449
#> 2     294 0.03049236
#> 3     307 0.01623711
#> 4     321 0.01246609
#> 5     334 0.02634117
#> 6     345 0.03545065
```

The data frame contains the following columns: `obs_time` – POSIXlt object of observation times for each climate variable, `temp` – temperatures (deg C), `relhum` - relative humidity (percentage), `pres` - atmospheric pressure (kPa), `swdown` - total downward shortwave radiation received by a horizontal surface (W/m^2), `difrad` - diffuse radiation (W/m^2), `lwdown` total downward longward radiation (W/m^2), `windspeed` - wind speed at reference height (m/s), `winddir` - wind direction in degrees and `precip` - hourly precipitation (mm). Precipitation is used to compute soil moisture and sub-model for this actually runs in daily time-steps. Thus, if only daily precipitation data are available, hourly data can be provided as daily values / 24.

Importantly, the entries of obs_time must all be in UTC (Coordinated Universal Time).

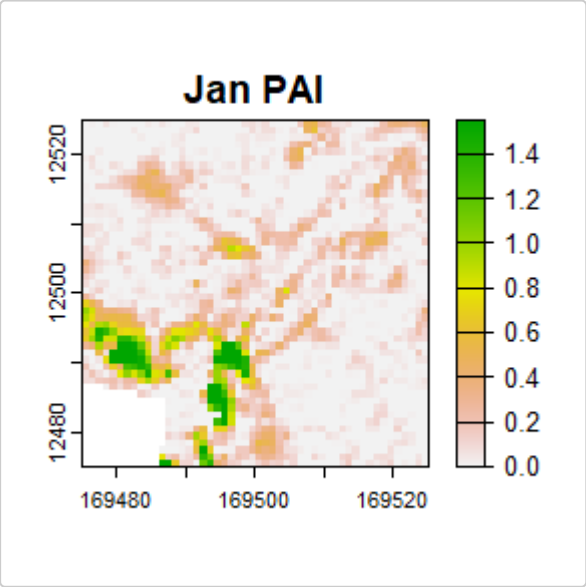
Any input weather dataset provided must use the same format, column names and units as in this example dataset. Most of these are standard meteorology variables that are readily available globally. If unknown for the study area, users may wish to explore the `mcera5` package on github (https://github.com/dklinges9/mcera5) or the `micorclimdata` package detailed above. Diffuse radiation, is sometimes harder to come by as many standard weather stations only estimate total radiation. If unknown, it can be estimated using the `difprop` function in the microctools package (https://github.com/ilyamaclean/microctools). The microctools package, also contains a function `converthumidity`, for converting absolute or specific humidity or vapour pressure to relative humidity.

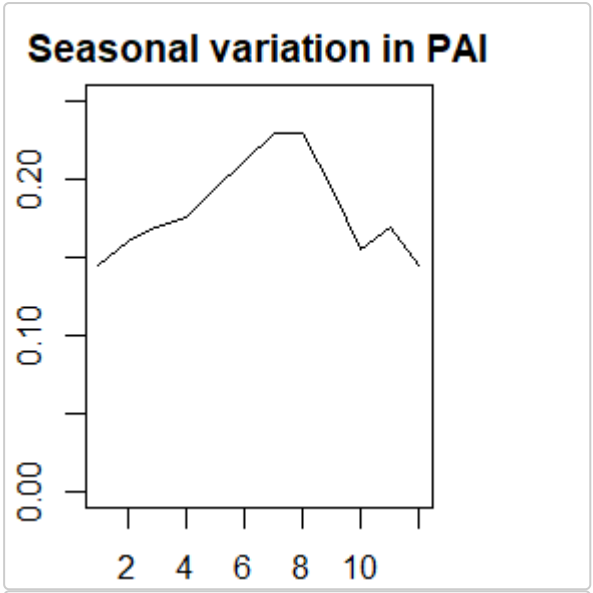
Vegetation parameters

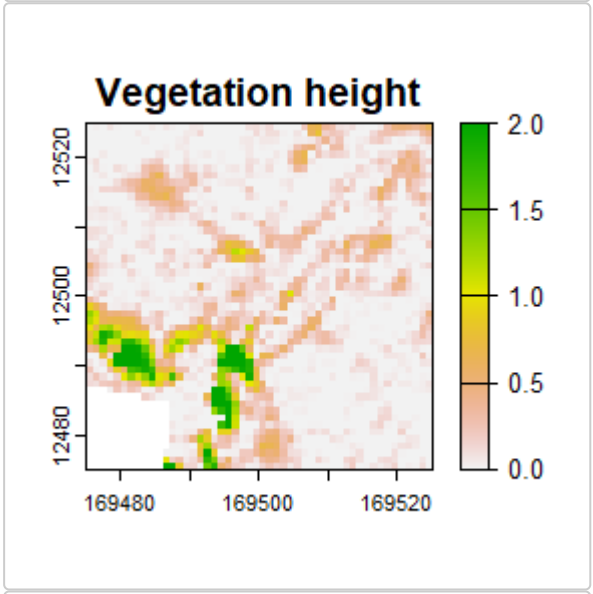
The inbuilt dataset `vegp` gives an example of the vegetation parameters needed to run the model. Here the attributes are shown and individual parameters plotted. Data are all stored as SpatRasters though in the inbuilt dataset these are `PackedSpatRasters` (see `terra::wrap`).

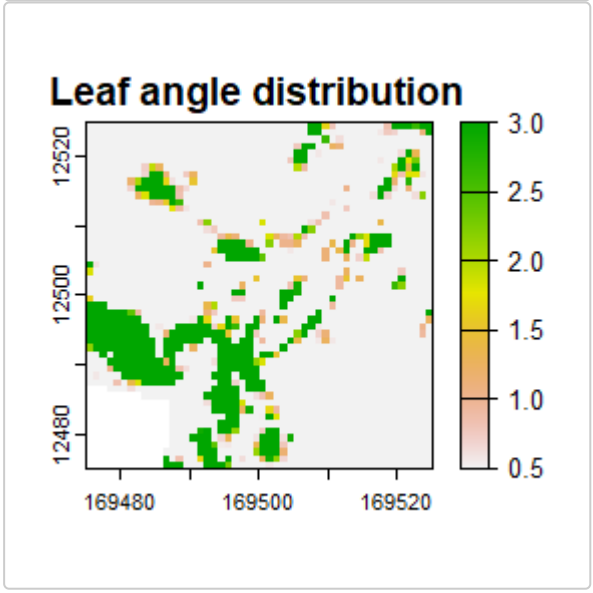
All vegetated data can be provided as either single layer SpatRasters, in which case they are assumed time-invariant, or as multi-layer SpatRasters, in which case they are assumed to vary seasonally. For example, in the inbuilt dataset `vegp$pai` is a 12 layer SpatRaster corresponding to approximately monthly values when the model is run over an entire year. Had the SpatRaster contained 365 values, vegetation would be assummed to vary daily. No more than one value per day can be provided. A mixture of single layer and multi-layer SpatRasters can be provided - the model takes care of things. Note, however, that the model does not `know` which layers correspond to which time-period. It is simply that if e.g. 12 layers are provided, the entire time-sequence over which the model is run is divided into 12 approximately equal sized chunks and separate vegetation data used for each chunk. Note that the data are matched to the time-sequence of the original climate date used to run the point model, not the values returned by sub-setting the point model.

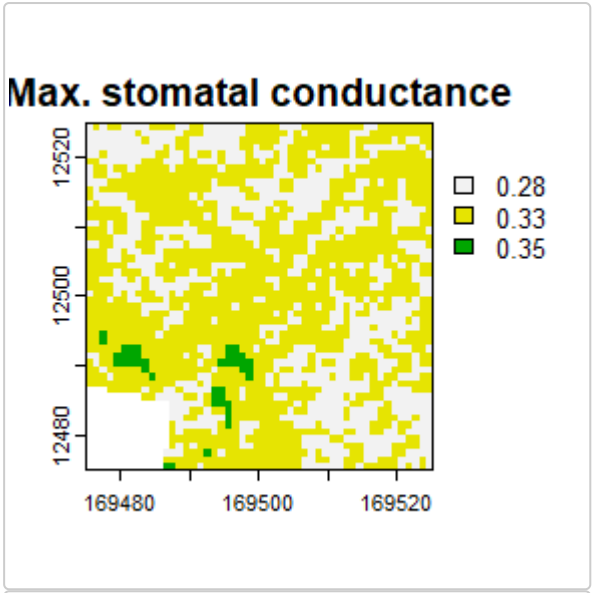
```
library(terra)
#> terra 1.7.29
attributes(vegp)
#> $names
#> [1] "pai"   "hgt"   "x"     "gsmax" "leafr" "clump" "leafd" "leaft"
#>
#> $class
#> [1] "vegparams"
# Plot spatial and temporal variation in pai
plot(rast(vegp$pai)[[1]], main = "Jan PAI")
paiarray <- as.array(rast(vegp$pai))
vegpmean<-apply(paiarray, 3, mean, na.rm = TRUE)
plot(vegpmean, type="l", ylim = c(0, 0.25), main = "Seasonal variation in PAI")
# Plot other variables
plot(rast(vegp$hgt), main="Vegetation height")
plot(rast(vegp$x), main = "Leaf angle distribution")
plot(rast(vegp$gsmax), main="Max. stomatal conductance")
plot(rast(vegp$clump)[[1]], main = "Jan Canopy clumping factor") # set to 0
plot(rast(vegp$leafr),col=gray(0:255/255), main = "Leaf reflectance")
plot(rast(vegp$leafd), main = "Mean leaf diameter") # set to 0.05
plot(rast(vegp$leaft),col=gray(0:255/255), main = "Leaf transmittance") # set equal to
leafr
```

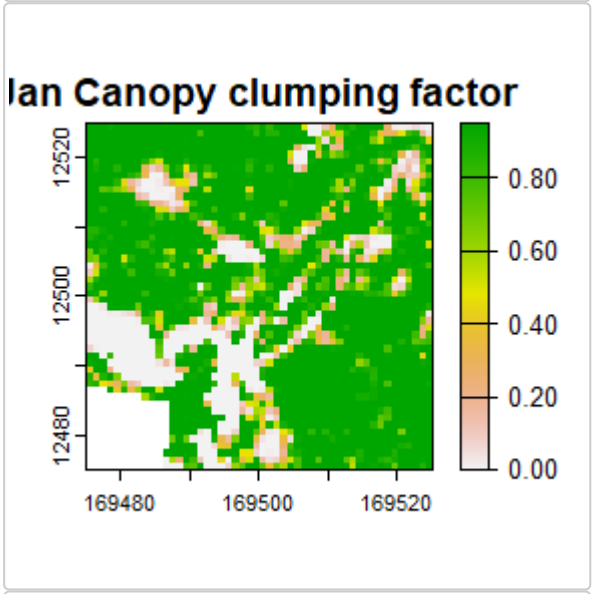


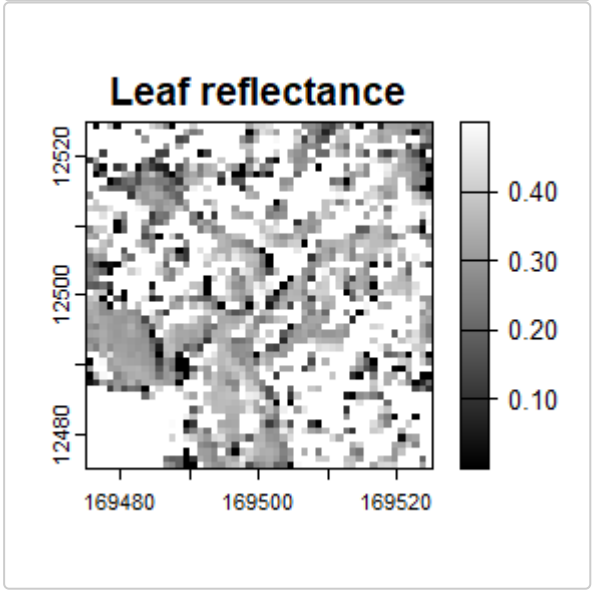

Seasonal variation in PAI
0.00
0.10
0.20
2
4
6
8
10

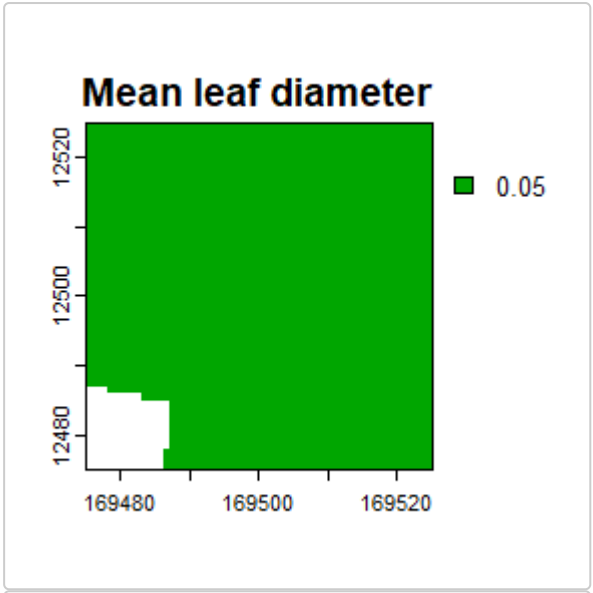

Vegetation height
12480
12500
12520
169480
169500
169520
0.0
0.5
1.0
1.5
2.0

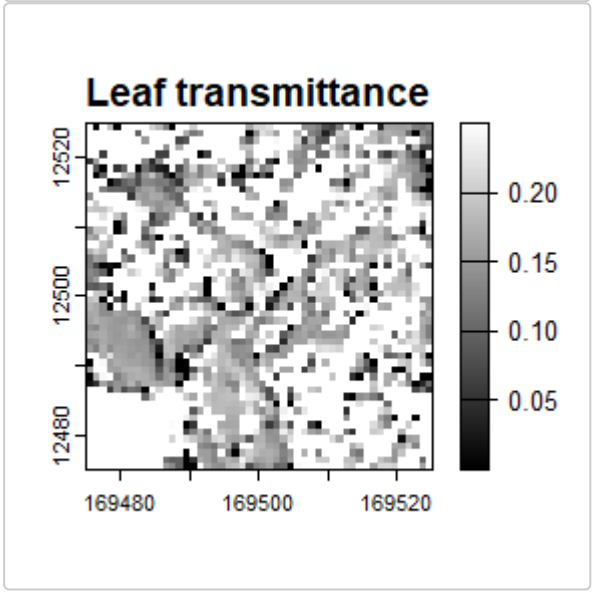

Leaf angle distribution
12480
12500
12520
169480
169500
169520
0.5
1.0
1.5
2.0
2.5
3.0


Max. stomatal conductance
12520
12500
12480
169480
169500
169520
0.28
0.33
0.35


lan Canopy clumping factor
12520
12500
12480
169480
169500
169520
0.80
0.60
0.40
0.20
0.00


Leaf reflectance
12520
12500
12480
169480
169500
169520
0.40
0.30
0.20
0.10



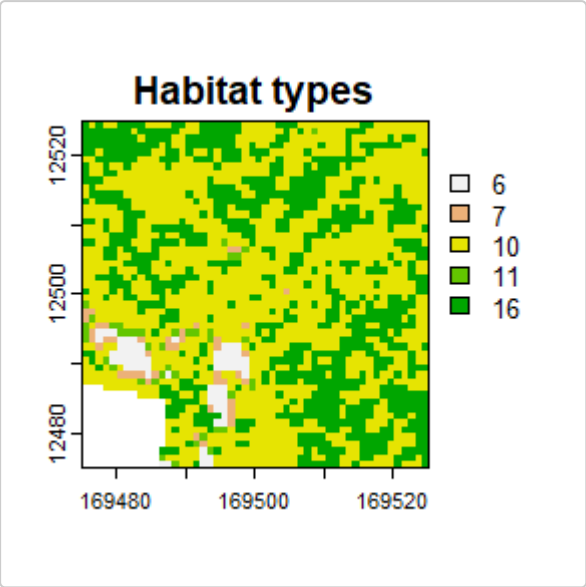


If users do know values of these vegetation parameters across their study area, they can be approximated from habitat type using the `vegpfromhab` function. This function takes as an input, a raster of habitat types numerically coded as follows:

1. for Evergreen needleleaf forest,
2. for Evergreen broadleaf forest,
3. for Deciduous needleleaf forest,
4. for Deciduous broadleaf forest,
5. for Mixed forest,
6. for Closed shrubland,
7. for Open shrubland,
8. for Woody savanna,
9. for Savanna,
10. for Short grassland,
11. for Tall grassland,
12. for Permanent wetland,
13. for Cropland,
14. for Urban and built-up,
15. for Cropland / Natural vegetation mosaic and
16. for Barren or sparsely vegetated

It returns an object of class vegparams as required by the model. Here this is illustrated using he inbuilt habitat SpatRast layer

```
plot(rast(habitats), main = "Habitat types") # inbuilt habitat SpatRast layer
tme<-as.POSIXlt(c(0:8783)*3600,origin="2000-01-01 00:00", tz = "GMT")
# Create an object of class vegparams:
veg<-vegpfromhab(habitats,tme=tme)
```



Note however, that be doing so, all values for a habitat type will be identical, when in reality this is unlikely to be the case. If one is unable to quantify the main determinants of microclimatic variation then there is little prospect of being able to model microclimatic conditions accurately and any outputs form the model should be treated with a high-degree of skepticism.

The model is most sensitive to `pai` (the total one sided area of both leaves and woody and dead vegetation per unit ground area) and `hgt` (vegetation height in metres). The former is needed primarily so that canopy cover can be estimated, but even for temperatures above canopy, `pai` partially determines the temperature profile. `hgt` is important as determines whether `reqhgt` is above or below vegetation and also dictates the shape of the temperature profile above vegetation.

The most sensible use-case for the `vegpfromhab` function is thus when one has alternative data that could be used to estimate `pai` and `hgt` that can be slotted into the output returned by this function. This is straightforward as the vegetation inputs to the model as returned by `vegpfromhab` is just a list of SpatRasters.

The model is less sensitive to other parameters. The parameter `x` represents how vertically or horizontally the leaves of the canopy are orientated and controls how much direct radiation is transmitted through the canopy at a given solar angle (when the sun is low above the horizon, less radiation is transmitted through vertically orientated leaves). Users may refer to Campbell (1986) for a detailed explanation. Values for deciduous woodland are typically around 1, but for grassland may be closer to 0.2. The parameter `gsmax` is the maximum stomatal conductance (mol / m^2 / s) of leaves and is needed for evapotranspiration calculations. Values typically range from 0.23 for deciduous broadleaf forest to 0.55 for wetland vegetation. Körner (1995) gives values for major vegetation types of the world. The parameter 'leafr' is the leaf reflectance to shortwave radiation, with typical values around 0.4.

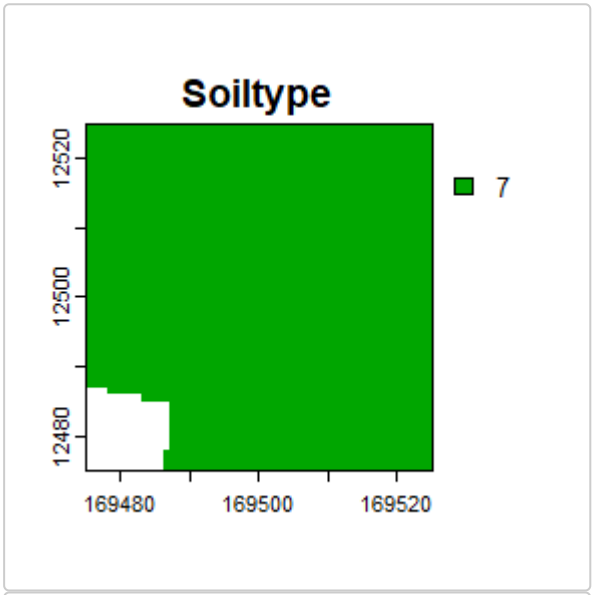
The parameter `clump` dictates how much radiation passes through gaps in the canopy, and therefore represents the sub-pixel canopy clumpiness, with values ranging from 0 (uniform) to 1 (highly clumped). In general, it varies with vegetation height and plant area index. The function `clumpestimate` can be used to derive an approximate estimate. The parameter `leafd` is the mean diameter of leaves.

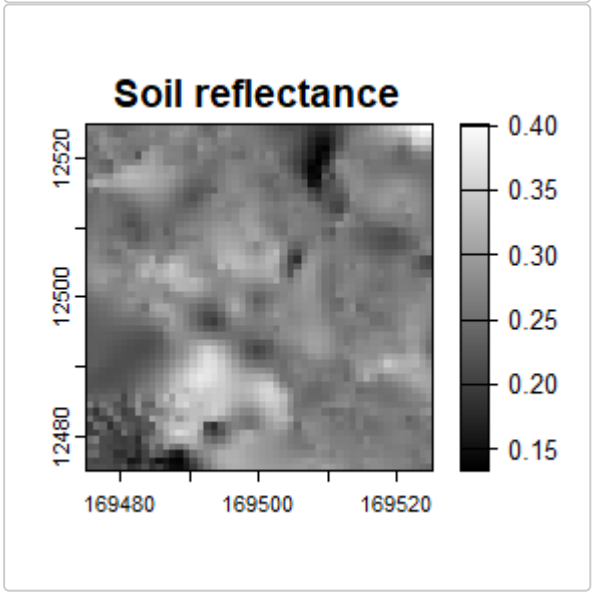
The `vegpfromhab` function assigns approximate values for leaf reflectance accordance to habitat type. However, it can also be estimated from surface albedo using function `leafrfromalb`. In applying this function, leaf transmittance is assumed proportional to leaf reflectance and a proportionality coefficient can be specified. In general, model outputs are not sensitive to this coefficient,

Soil parameters

The inbuilt dataset `soilc` gives an example of the soil parameters needed to run the model. Here the attributes are shown and plotted:

```
attributes(soilc)
#> $names
#> [1] "soiltype" "groundr"
#>
#> $class
#> [1] "soilcharac"
plot(rast(soilc$soiltype), main = "Soiltype") # Clay loam throughout
plot(rast(soilc$groundr), col=gray(0:255/255), main = "Soil reflectance")
```





- This is a list of the following:
  - `soiltype` - a PackedSpatRast object of integer soil types
  - `groundr` - a PackedSpatRast object of soil reflectance values for shortwave radiation (0 - 1)

Again, users in creating such a dataset, can store `soiltype` and `groundr` as either a PackedSpatRast or a SpatRast object.

Soil type 7 corresponds to Clay loam. A full list of which numeric values correspond to which soil types, along with parameters associated with these soil types is shown in the `soilparameters` table:

```
soilparameters
#>          Soil.type Number  Smax  Smin     Ksat   b psi_e   Vq     Vm     Vo
#> 1             Sand      1 0.399 0.049 501.1200 1.7   0.7 0.30 0.3000 0.0010
#> 2       Loamy sand      2 0.402 0.054 146.8800 2.1   0.9 0.24 0.3550 0.0030
#> 3       Sandy loam      3 0.403 0.058  62.2080 3.1   1.5 0.18 0.4100 0.0070
#> 4             Loam      4 0.422 0.074  31.9680 4.5   1.1 0.12 0.4400 0.0180
#> 5        Silt loam      5 0.447 0.067  16.4160 4.7   2.1 0.00 0.4700 0.0830
#> 6  Sandy clay loam      6 0.388 0.089  10.3680 4.0   2.8 0.14 0.4640 0.0080
#> 7        Clay loam      7 0.419 0.091   5.5296 5.2   2.6 0.06 0.5090 0.0120
#> 8  Silty clay loam      8 0.441 0.089   3.6288 6.6   3.3 0.04 0.5080 0.0110
#> 9       Sandy clay      9 0.381 0.103   2.8512 6.0   2.9 0.15 0.4655 0.0035
#> 10      Silty clay     10 0.368 0.073   2.1600 7.9   3.4 0.00 0.6240 0.0080
#> 11            Clay     11 0.394 0.073   1.4688 7.6   3.7 0.00 0.6000 0.0060
#>        Mc      rho        mult      rmu        a      pwr
#> 1  0.0100 1.597779 0.000293942 0.037449 0.003018 0.963099
#> 2  0.0350 1.587082 0.000302099 0.038750 0.008320 1.037554
#> 3  0.0600 1.578984 0.000190685 0.024034 0.006535 0.667482
#> 4  0.0844 1.513506 0.000189847 0.023019 0.009031 0.807574
#> 5  0.1240 1.358636 0.000172124 0.020029 0.026067 1.572287
#> 6  0.3648 1.617506 0.000184663 0.021304 1.191259 4.077206
#> 7  0.5422 1.529643 0.000191202 0.021303 0.059765 1.134773
#> 8  0.3948 1.472509 0.000175762 0.019098 0.132554 1.907443
#> 9  0.5050 1.642237 0.000173624 0.016682 0.099531 0.820078
#> 10 0.5500 1.670682 0.000149402 0.014268 0.147172 1.156038
#> 11 1.0000 1.604273 0.000164801 0.016936 0.239373 1.514030
```

The model also copes with these values being provided as individual data layers if these are added to the list of SpatRasters contained in `soilc`. Note that in all instances soil properties (apart from soil moisture, which is modelled explicitly) are assumed time-invariant.

Additional optional parameters

In addition to specifying `reqhgt` the height (m) above or below ground for which microclimate estimates are required, there are also a set of optional parameters that can be provided to the run functions that control model behaviour. For the point model, these are as follows: * `runchecks` - logical indicating whether to call function `checkinputs` to run checks on format and units of input data (see details under model input functions). * `windhgt` - height above ground of wind speed data in weather (see details under wind) * `soilm` - a vector of hourly soil moisture values in upper 10 cm of the soil (calculated using a simple soil model if not supplied - see details under running the point model) * `dTmx` - maximum amount by which canopy or ground surface temperatures can exceed air temperatures when running the point model (see details under running the point model). * `maxiter` - integer indicating the maximum number of iterations to use when running the point model (see details under running the point model) Additionally, there are a number of options for internal use, which can generally be ignored by the user as they are calculated if not supplied, but in brief, these are: `yearG` - an option dictating whether or not to account for annual cycles in the ground flux when a year or more of data are provided, `lat` and `long` - the latitude and longitude of the location for which the point model is run, `vegp_p`, `ground_p` the vegetation and ground parameters used for running the point model, `soiltype` - the assumed soiltype at the location for which the model is run and `mxhgt` - the height to which weather data are adjusted, if not supplied calculated from vegetation height across the study area.

For the grid model, the following additional parameters can be supplied

- `dtmc` - a coarse-resolution digital elevation dataset matching the resolution of climate data and used to perform elevation adjustments only if climate data are provided as multi-layer SpatRasters.

- `altcorrect` - a single numeric value indicating whether to apply an elevation lapse rate correction to temperatures and pressures (0 = no correction, 1 = fixed lapse rate correction, 2 = humidity-dependent variable lapse rate correction)
- `runchecks` - as for the point model.
- `pai_a` an array of plant area index values above `reqhgt`. Estimated by assuming a plausible vertical distribution of leaf foliage density if left as 'NA` (see details under radiation).
- `tfact` an optional coefficient determining the sensitivity of spatial variation in soil moisture to variation in topographic wetness (see details under soil moisture).
- `out` an optional vector of logicals indicating which variables to return ordered as for the listed outputs when `rehgt > 0'` (e.g. `out[1] = TRUE` indicates that `Tz` is returned, `out[2] = TRUE` that `tleaf` is returned etc). By default all variables are returned, but if, for example, only temperature or humidity are required as outputs, setting relevant values to `FALSE` can save a lot of memory.
- `slr`, `apr` and `twi` - optional SpatRaster objects of slope, aspect, and topographic wetness. If not supplied, these are calculated from the provided `dtm`, but users may wish to provide their own values to avoid edge effects.
- `hor`, `svfa` and `wsa` - optional array of the tangent of the angle to the horizon in 24 directions (used for calculating terrain shading), skyview factors (used for adjusting diffuse and downward longwave radiation) and wind shelter coefficients in 8 directions (used for determining wind speed). If not supplied, these are calculated from the provided `dtm`, but users may wish to provide their own values to avoid edge effects.
- `method` - set either as `R` or by default as `Cpp`. If set to `Cpp` the entire model is run using c++ code, and is therefore optimized for speed and memory allocation. If set to `R` individual components of the model coded as R wrappers are run as described below. This is marginally slower and much more memory hungry, but affords users greater flexibility to e.g. interrogate individual model components or to swap their own functions in and out of the model for individual components.

Running the point microclimate model

To ensure the grid model can be run without iteration, the first stage of modelling is to run a point microclimate model iteratively for a flat surface at the centre of the study area using as inputs to the model, vegetation and soil characteristics that are broadly representative of the study area. This is achieved automatically using function `runpointmodel`, using as inputs to the model, the same inputs that are supplied to the grid model as follows:

```
# Run the point model
micropoint <- runpointmodel(climdata, reqhgt = 0.05, dtmcaerth, vegp, soilc)
```

Interrogating the attributes of `micropoint` allows us to see what the model returns:

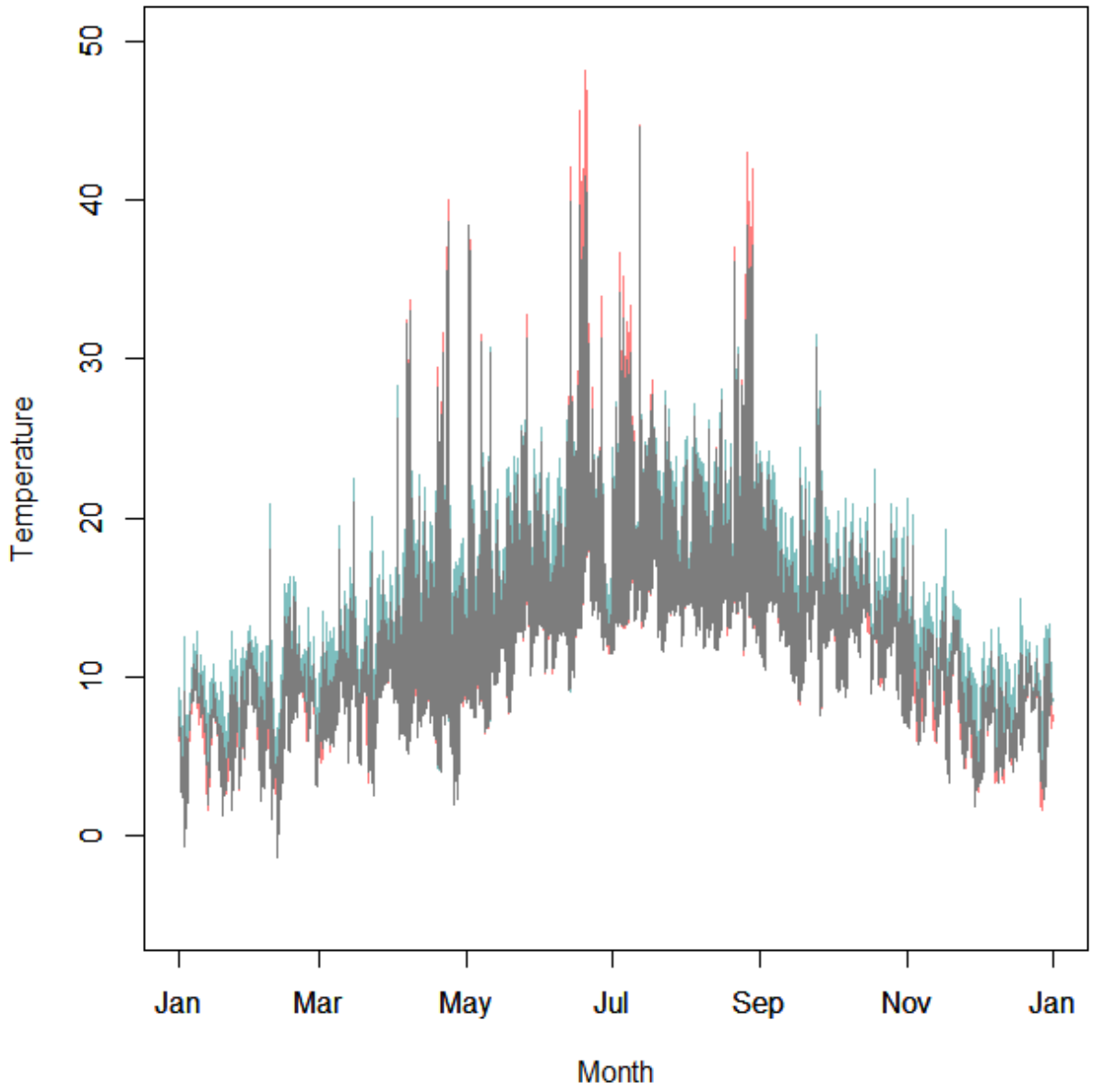
```
attributes(micropoint)
```

The function essentially gathers various things into a single list object. These are as follows:

- `weather` - the original supplied weather data, but height adjusted if vegetation within the study area exceeds the height of temperature or wind speed measurements.
- `dfo` - this is a data.frame that stores all the useful stuff that the grid model needs. Most are not worth remaking on as they are essentially used to handle adjustments made to the grid model outputs to avoid the need to run it iteratively (e.g. umu scales wind speeds without and without diabatic correction coefficients included), but four variables are worth remaking on. These are `G` the rate of heat storage by the soil (W/m^2), `soilm` - soil water content expressed as a volumetric fraction, `Tg` - ground surface temperature and `Tc` - the average temperature of vegetation. Plots of `Tg` and `Tc` are shown below.

- `Tbz` if the model is run above ground this is just set to NA, but if run below ground, this is a vector of soil temperatures at depth `reqhgt`.
- `lat` and `long` are the latitude and longitude of the centre of the study area, which is the location for which the point model was run.
- `zref` represents the height above ground to which weather data have been adjusted. If none of the vegetation is greater than two meters in height the output `zref` is set to 2 m and `windhgt` is also two meters, the returned weather dataset is identical to that provided the function. However, for the grid microclimate model to derive below-canopy wind and temperature profiles, the reference height must be higher than the tallest vegetation in the study area. For that reason, if the tallest vegetation exceeds two meters, `zref` is set to the maximum height of the vegetation and the wind speeds and temperatures are adjusted for height in the weather dataset.
- `subs` and `tmeorig` help the grid model handle sub-setting of the point model (see below). Prior to sub-setting, `tmeorig` is a POSIXlt object of dates and corresponding to dates and times in `weather` and `dfo` and `subs` is just a vector indicating which values have been returned (all prior to any sub-setting). Below, ground and vegetated surface temperatures are plotted to show what the outputs of the point model look like.

```
micropoint <- runpointmodel(climdata, reqhgt = 0.05, dtmcaerth, vegp, soilc)
microp <- micropoint$dfo
tme <- as.POSIXct(micropoint$tmeorig)
par(mar=c(5,5,3,3))
plot(microp$Tg ~ tme, type="l", ylim = c(-5, 50), col = rgb(1,0,0,0.5), xlab = "Month",
ylab = "Temperature") # temperature of ground surface
par(new = TRUE)
plot(microp$Tc ~ tme, type="l", ylim = c(-5, 50), col = rgb(0,0.5,0.5,0.5), xlab = "",
ylab = "")
```



The equivalent to the `runpointmodel` function when climate data are provided as arrays is `runpointmodela`. An example of its use is shown below.

Subsetting the microclimate model

The point microclimate model is usually run in hourly time-increments using complete time sequences of weather data to fully allow for the diurnal cycles in ground heat fluxes to be accounted for, but if desired, the grid model can be run for just say the hottest days in each month to derive maximum temperatures. This is achieved using function `subsetpointmodel`. This function takes an object of class 'pointmicro as an input and also returns an object of class 'pointmicro, but with the request hours extracted from 'pointmicro In the example below, the model is subset to return only those hours corresponding to the day in each month with the hottest temperature (as determined by the point model). The function has several inputs that control its behaviour. If 'tstep' is set toyearthe day in each year with the e.g. the hottest or coldest hourly temperature is identified, and if 'tstep' is set tomonththe days in each month in each year with e.g. the hottest or coldest hourly temperatures are returned. Ifwhatis set totmaxortminthe hottest or coldest hour within each month or year are identified. Ifwhatis set totmedianhourly temperatures within the month or year are ranked and the median hour identified. The

final option is to provide a vector of the days in the time sequence to return data for using inputdays. If providedtstep` is ignored. It is necessary that all hours of a given day are returned for two reasons. First, it ensures that the ground heat flux in the grid microclimate model can be estimated as it depends on the full diurnal cycle. Second because the hottest hour on a flat surface may not be the hottest hour on e.g. a steeply south-westerly facing slope - temperatures will typically peak later in the day when the slope is directly facing the sun. Returning hourly values for an entire day ensures that these terrain effects can be properly handled by the grid model.

```
micropoint <- subsetpointmodel(micropoint, tstep = "month", what = "tmax")
```

Preparing model inputs

The entire grid model is run using runmicro, but to illustrate its working, we here run each component of the model in stages. If running it in stages, the first stage is to gather the input variables and reformat them ready to run the model. There are two options for preparing the data for running the model. Firstly where the climate data are in the form of a data frame of hourly weather for a point location. Second, where the climate data are in the form of course-gridded SpatRasters of values. Both cases are handled flexibly by the function modelin.

In the examples that follow, the inbuilt datasets of parameter values and a dtm for the study area, dtmcaerth are used and the model is run using a data.frame of weather data. Subsequently, the equivalent workflow for when weather data are in the form of course-gridded SpatRasters is shown.

```
micro <- modelin(micropoint, vegp, soilc, dtmcaerth)
```

By default modelin calls function checkinputs. This performs some basic checks on the vegetation and soil parameters data to check for consistency in extent to the dtm. It also ensure values in the climate datasets are typical of what would be expected, thereby helping to ensure the correct units are used.

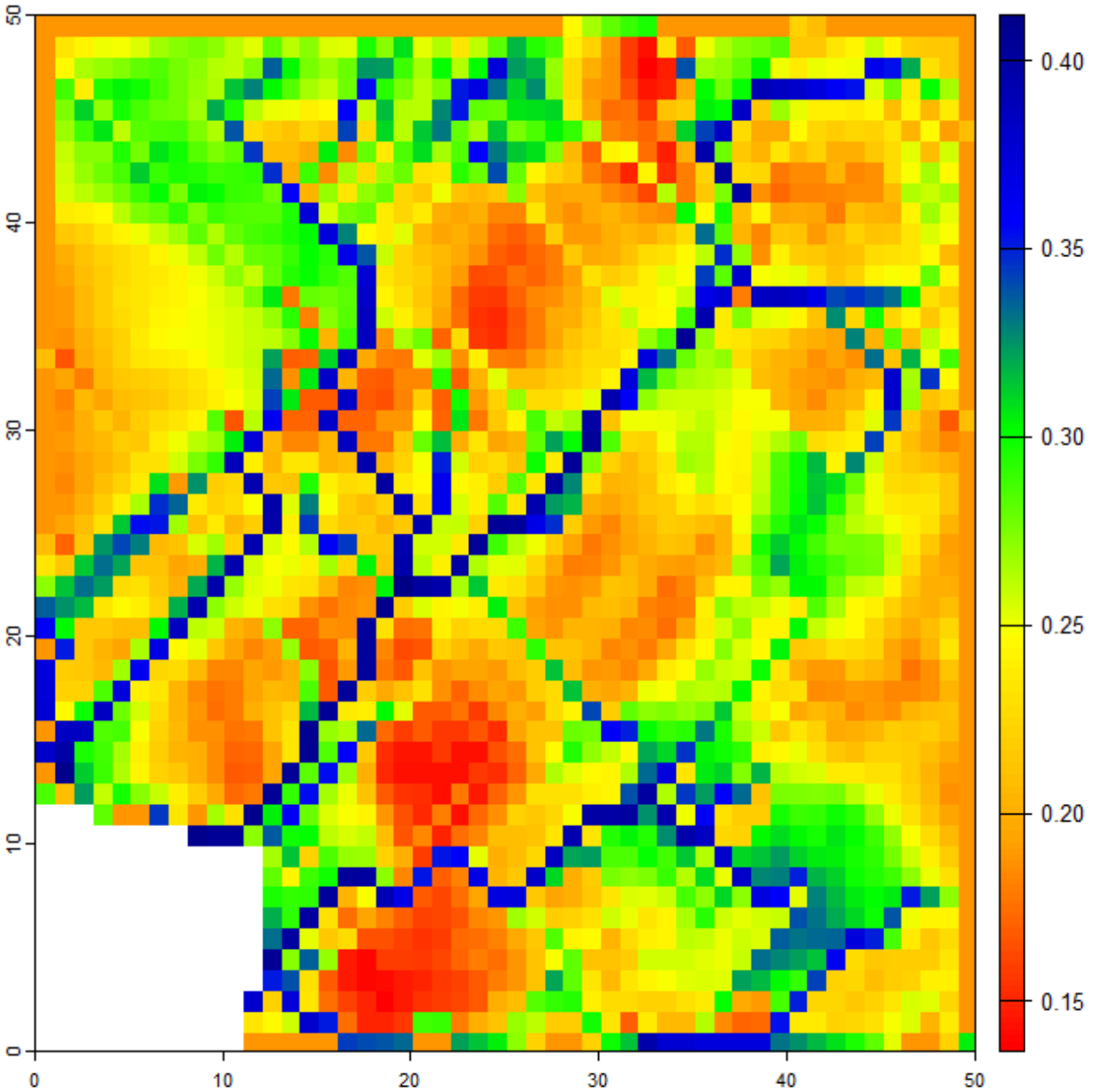
In subsequent downscaling of wind, the drag effects of vegetation, determined by vegetation height and foliage area are accounted for and calculated at this stage. In so doing, it is necessary to accommodate the possibility that the wind speed is not just affected by the surface roughness in each pixel, but also by vegetation surrounding the location. This is accommodated for by applying xyf which effectively smooths the surface roughness coefficients using terra::aggregate where xyf is the aggregation factor. If xyf is set to NA, the roughness coefficients are averaged across the entire study area. In the example above, we do not specify a value for xyf, and the default of one is therefore applied, which means that no smoothing is performed.

Running the model

Soil moisture

The first step of the microclimate model is to estimate soil moisture. This is handled in microclimf, by spatially distributing the soil moisture values returned by the point microclimate model for each time increment using the the Bevan and Kirkby (1979) topographic wetness index, such that valleys and flat areas are assumed to have higher water content. Values are adjusted such that the average for the study area in each time step is equivalent to the value obtained by running the point model. Users have the option to control the sensitivity of this topographic adjustment. Irrespective of whether the model is run in hourly or daily time-increments, these calculations are performed using the soilmdistribute as in the example below.

```
micro<-soilmdistribute(micro)
par(mfrow = c(1,1)) # make sure output is a single panel figure
plot(rast(micro$soilm[,,134]), col = rev(mypal))
```



Radiation

The next stage, needed to calculate soil surface temperature, is to estimate radiation absorbed by the ground. Because the ground lies below canopy in some instances, it is necessary also to consider the transmission of radiation through the canopy. The radiation fluxes are modelled using function `twostream`, which implements a variant of the Dickenson-Sellers two-stream radiation model described in Yuan et al. (2017) J Adv Model Earth Sy 9: 113–129 to model radiation interception by the canopy. It also varies from the Dickenson-Sellers model in more explicitly handing sloped ground surfaces beneath a canopy.

Absorbed radiation is the total incoming radiation received by a surface less that transmitted or reflected. The total incoming radiation can be partitioned into three sources, each of which is modified by the environment in slightly different ways. The first is direct radiation from the sun. Here, absorption depends on the angle of the surface relative to perpendicular. This is the reason why equatorward-facing slopes are

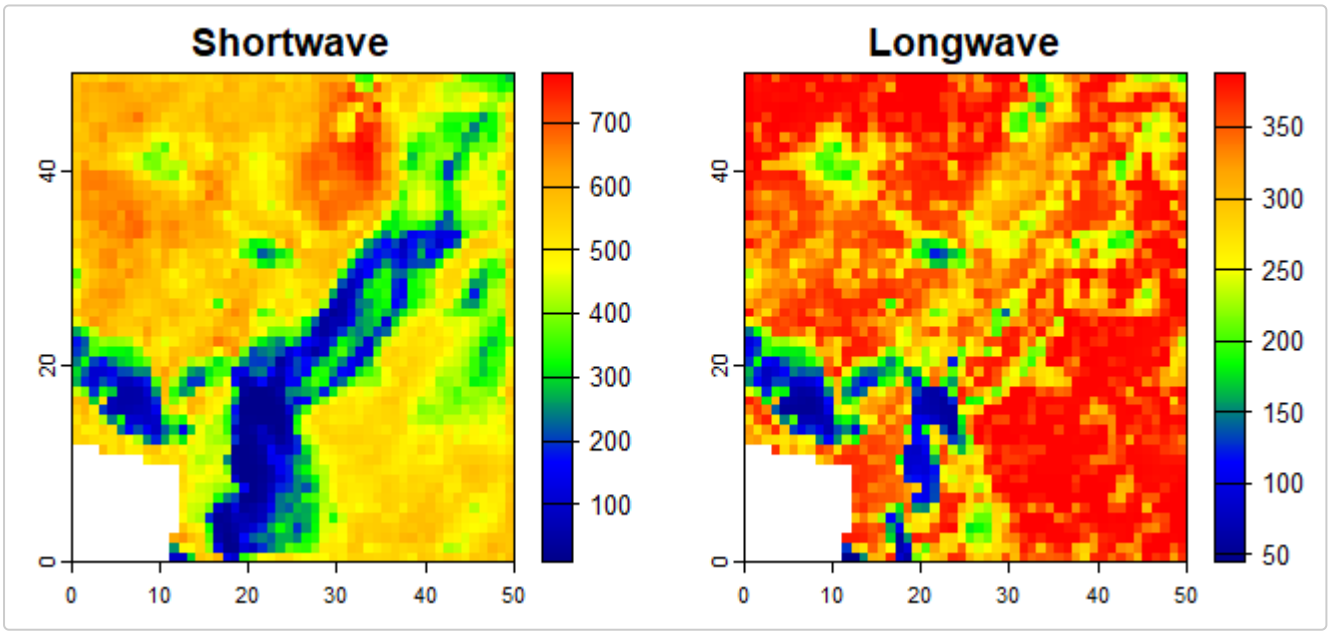
warmer than those that face poleward and is indeed the main reason why temperature increases with latitude. Here the solar beam is more concentrated, rather like shining a torch directly on a surface as opposed to obliquely. The second source is diffuse solar radiation: that scattered by particles and clouds in the atmosphere. The final source is longwave radiation emitted from surrounding surfaces and the sky. The latter two are isotropic (i.e. having the same value when measured in any direction). In consequence, for these sources, the direction of the surface is unimportant, and radiation interception is instead influenced by sky-view.

Function `twostream` calculates all of these fluxes. If `reqhgt > 0` it also calculates the fluxes at the height of interest, including the upward fluxes resulting from reflection by the ground surface and scattering by leaves within the canopy. Additionally, to aid with modelling of air temperatures, it also calculates the flux density of radiation absorbed by both the canopy and the ground surface.

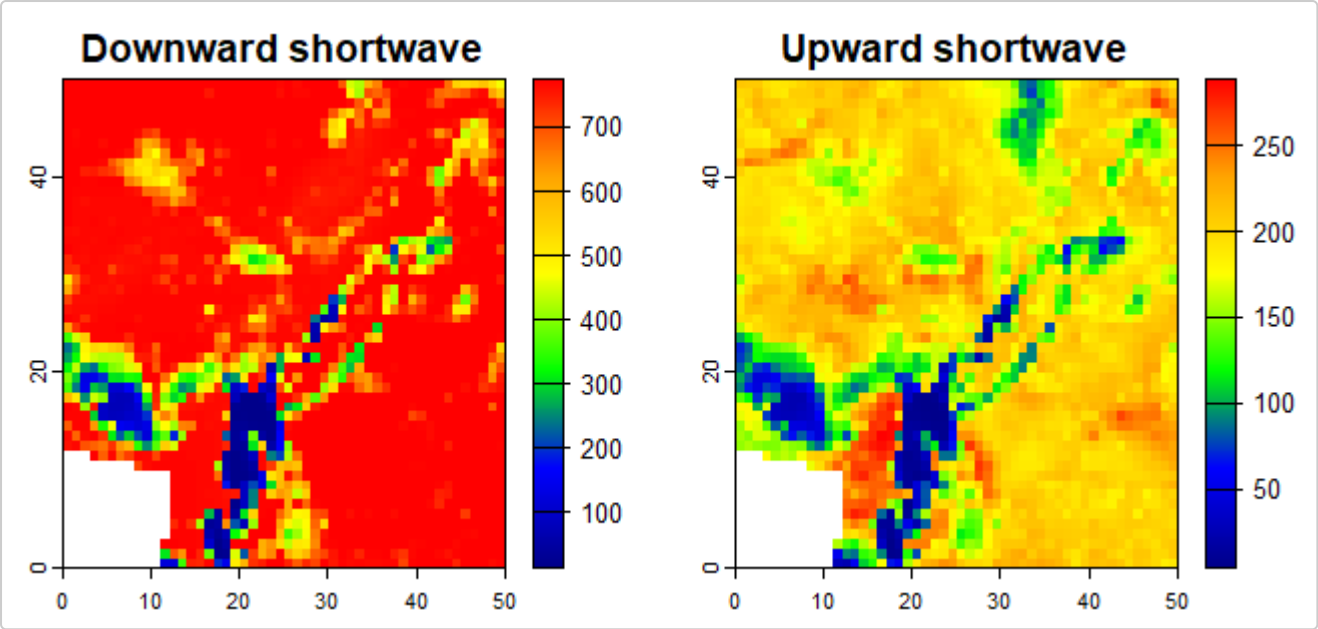
To model radiation, the canopy as a turbid medium and the transmission of radiation by vegetation is thus described using an equation similar to Beer's law, in which flux density of radiation is assumed dependent on the total one-side lead area per unit ground area and by an extinction coefficient for the canopy. For direct radiation, the extinction coefficient is assumed to depend on the distribution of leaf angles (with more vertically orientated leaves transmitting less radiation at lower solar altitudes). This is where the model input `vegp$x` comes into play. For isotropic sources of radiation (i.e. diffuse and longwave), leaf angle is assumed unimportant.

In the example below the flux density of shortwave and longwave radiation absorbed by the ground surface at 10:00 hours on 20th Jun 2017 is shown. Below that, the flux density of the upward and downward radiation streams 5 cm above the ground are shown (the downward flux comprises both direct and diffuse radiation, the upward flux is assumed entirely diffuse).

```
micro <- twostream(micro, reqhgt = 0.05)
par(mfrow=c(1,2))
plot(rast(micro$radGsw[,,131]), col = mypal, main = "Shortwave")
plot(rast(micro$radGlw[,,131]), col = mypal, main = "Longwave")
```




```
plot(rast(micro$Rbdown[,,131]+micro$Rddown[,,131]), col = mypal, main = "Downward shortwave")
plot(rast(micro$Rdup[,,131]), col = mypal, main = "Upward shortwave")
```

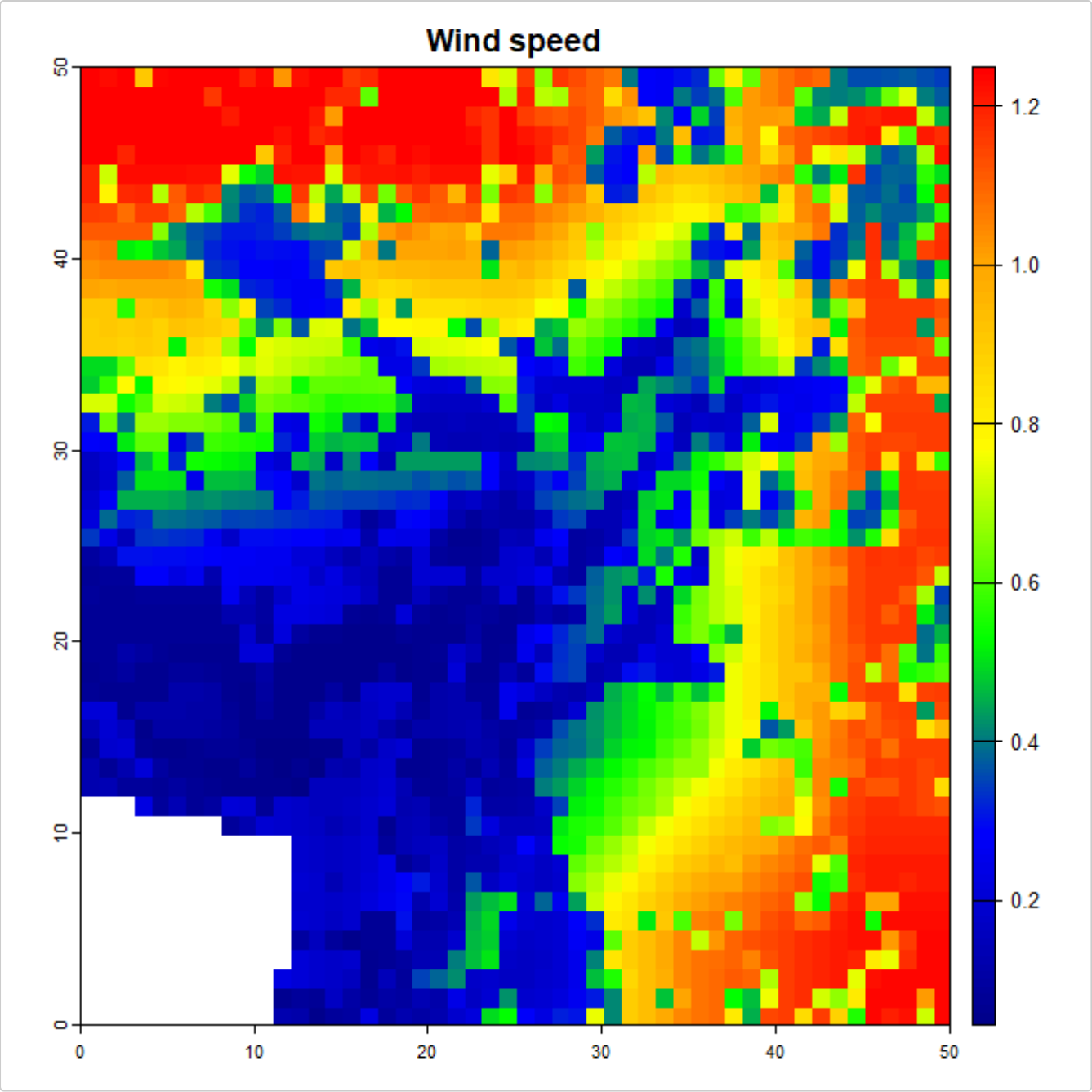


Sensible heat flux and wind

A surface heated by solar radiation will loose some of this heat to the surrounding air, and by virtue of the laws of energy conservation, the air gains this heat. The exchange of heat between a surface and the surrounding air is termed sensible heat exchange and is influenced strongly wind speed. The next stage of modelling is therefore to calculate wind speed.

The function `wind` models two processes. Firstly, direction-dependent terrain shelter coefficients are applied. Secondly, the effects of vegetation on wind speeds are determined. If `reqhgt` is above the vegetation, the shaped of the wind speed above vegetation is determined by the degree of surface drag, in turn contingent upon vegetation height and the plant area index of vegetation. If `reqhgt` is below canopy, the effects of vegetation is to attenuate wind speeds, but the shape of the wind-height profile below and above canopy differs. For a given `reqhgt` some pixels may lie below canopy and some above. This is all handled automatically by function `wind` as in the example below.

```
micro <- wind(micro, reqhgt = 0.05)
par(mfrow=c(1,1))
plot(rast(micro$uz[,,100]), col = mypal, main = "Wind speed")
```



The lower wind speeds in the valley caused by terrain sheltering are evident. The speckle in the figure is caused by vegetation.

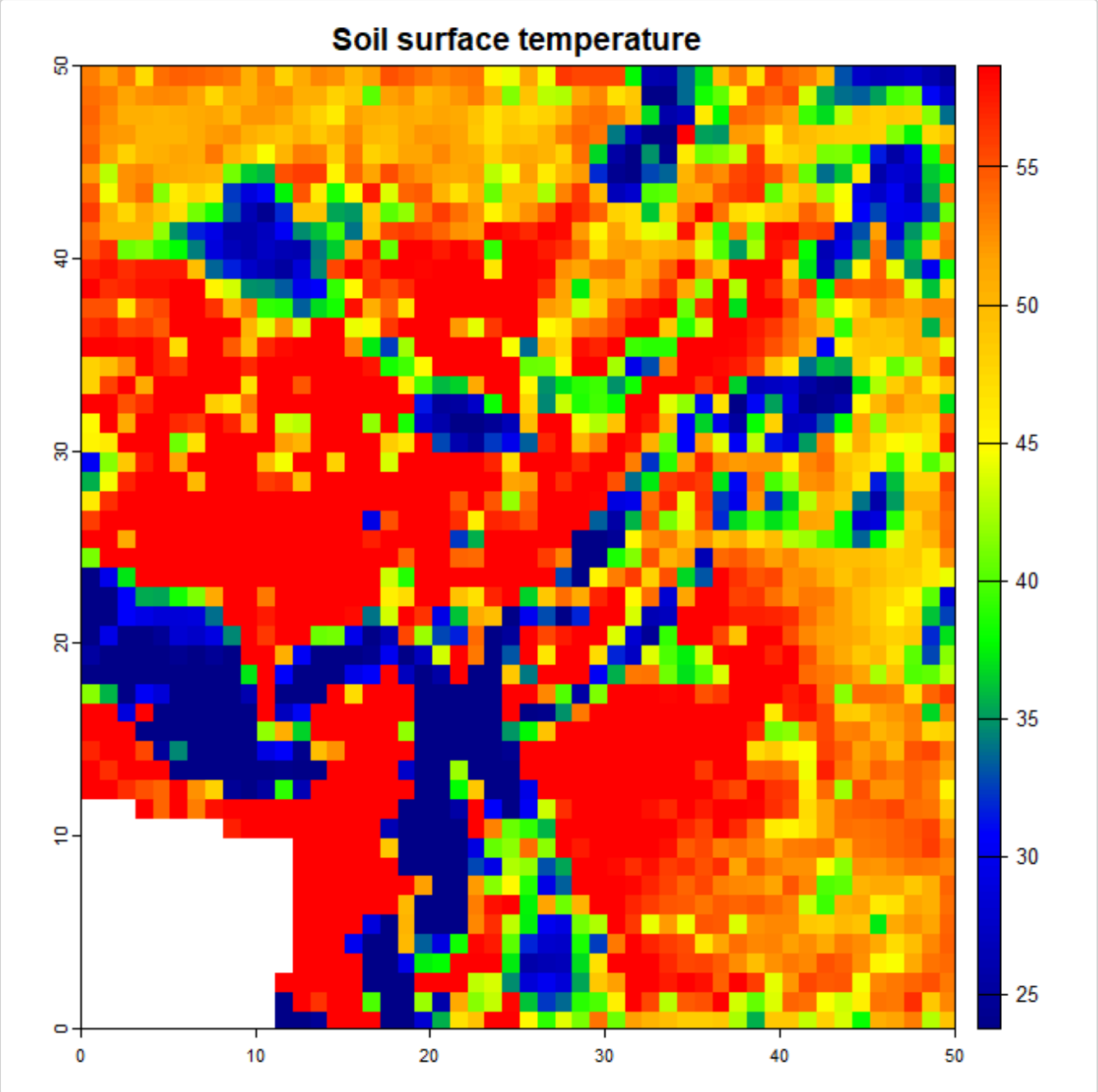
One minor additional point to note is that when calling any of the component functions after creating the model input using e.g. `modelin` there is no need to run the components prior to that. The function automatically checks whether these have been run, and if necessary does so.

Ground surface temperature

Once radiation absorbed by the ground and wind speed have been calculated, ground surface temperature can be computed. This is done using either function `soiltemp_hr` or `soiltemp_dy` depending on whether the model input is daily or hourly. Whereas `twostream` and `wind` handle both hourly or daily data, because ground heat fluxes are calculated in different ways depending on whether inputs are hourly or daily, the functions used for each differ. In the example below, the ground surface temperature on the hottest hour of the year is calculated using `soiltemp_hr` and then plotted.

```
micropoint<-runpointmodel(climdata,0.05,dtmcaerth,vegp,soilc)
micropoint<-subsetpointmodel(micropoint, tstep = "month", what = "tmax")
```

```
micro <- modelin(micropoint, vegp, soilc, dtmcaerth)
micro <- soiltemp(micro, reqhgt = 0.05)
# Plot ground temperature of hottest hour
par(mfrow=c(1,1))
plot(rast(micro$Tg[,,134]), col = mypal, main = "Soil surface temperature")
```

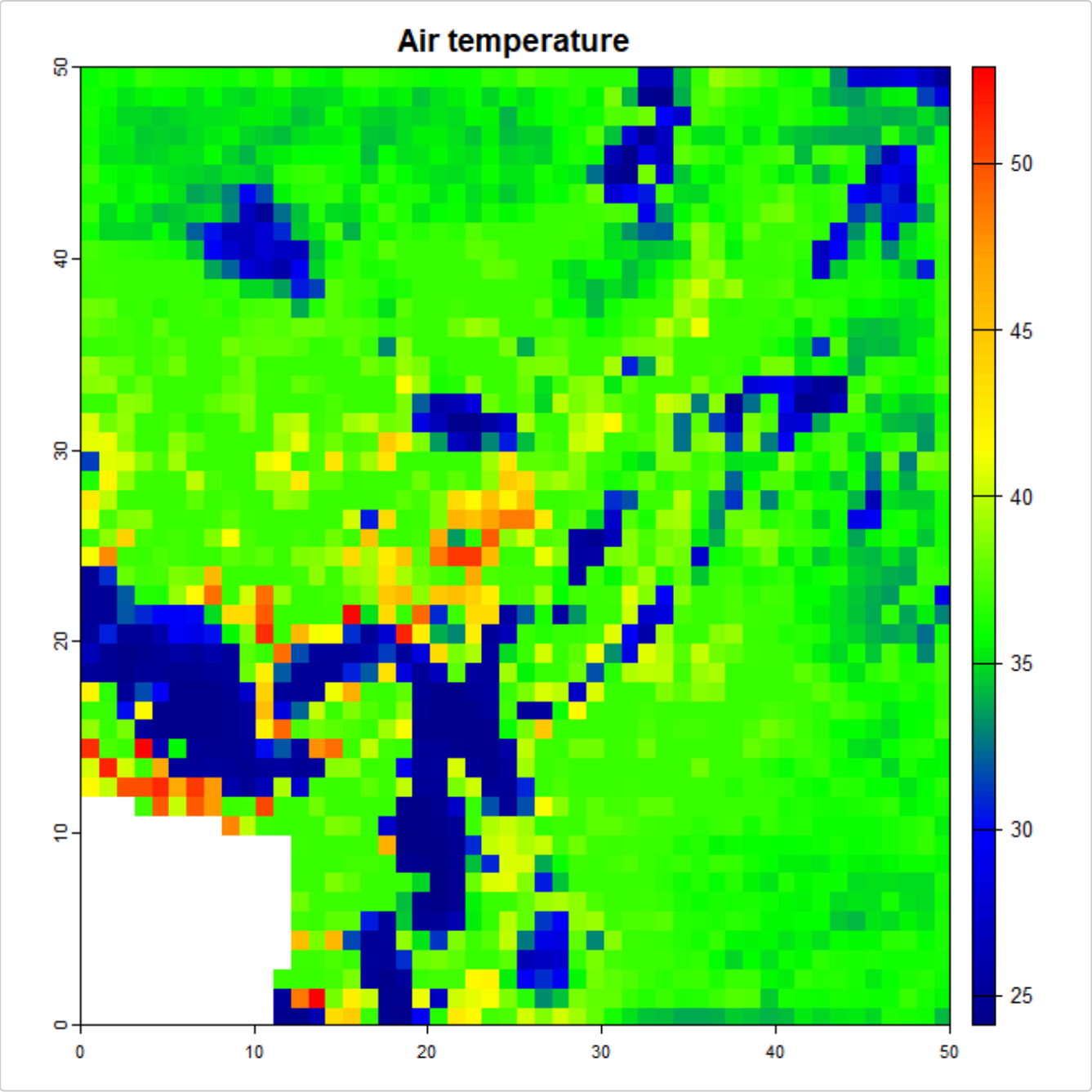


As can be seen, the soil surface temperatures on bare, south-facing slopes get pretty hot

Above ground

After calculating ground surface temperature, there are two pathways, depending on whether microclimatic conditions below or above ground are required. If `reqhgt > 0` then function `aboveground` is called as in the example below. This essentially runs the full microclimate model. Details of the model outputs are specified below.

```
mout <- aboveground(micro, reqhgt = 0.05)
plot(rast(mout$Tz[,,134]), col = mypal, main = "Air temperature")
```



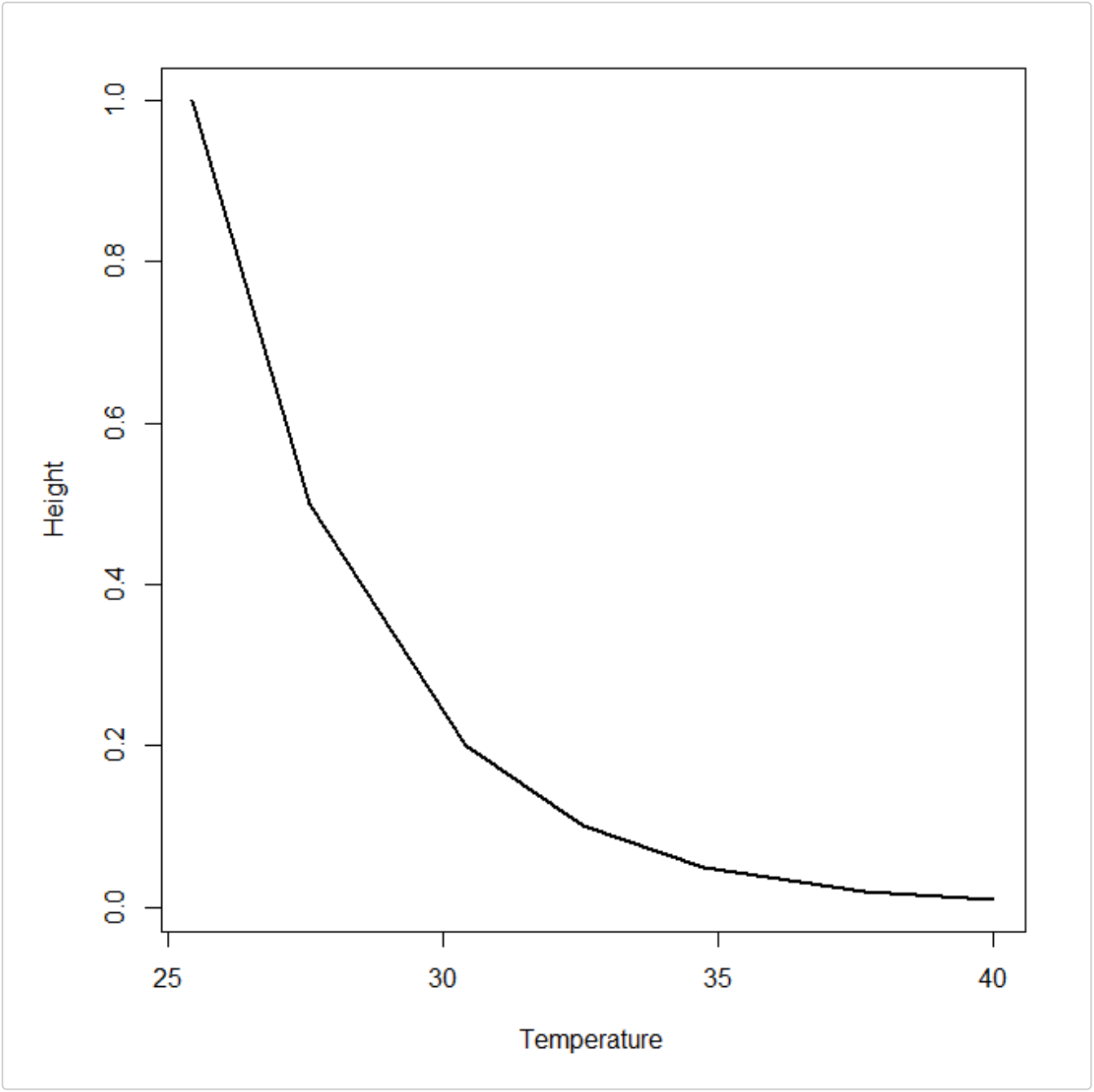
In the example above air temperature in the hottest hour is plotted. Users are free to experiment with plotting other model outputs. The air temperature at 5 cm is somewhat lower than the ground surface temperature, but much higher than ambient temperature on south-facing slopes, as one would expect

Above canopy

The function `aboveground` automatically works out which pixels are below canopy, and which above, but the microclimate is modelled differs. To work out temperatures above canopy, the canopy (and soil surface) are treated as a single layer of homogeneous phytomass and the energy balance solved to derive the mean temperature of the canopy. The canopy is then assumed to exchange heat with the air above it such that close to the heat exchange surface of the canopy, air temperatures similar to canopy temperatures, but are increasingly close to air temperature at reference height (i.e. the at height of the input weather station data). The result is a logarithmic temperature-height (and humidity) profile as re-reproduced in the example below in which the model is run at multiple heights over a small area with spatially uniform, terrain, soil and vegetation properties.

```
dem <- aggregate(rast(dtmcaerth), 10) * 0
# Create spatially uniform vegetation and soil parameters  dataset
```

```r
vegp2 <- list(pai = array(0.05, dim = c(5, 5, 12)),
              hgt = aggregate(rast(vegp$hgt), 10) * 0 + 0.005,
              x = aggregate(rast(vegp$x), 10) * 0 + 1,
              gsmax = aggregate(rast(vegp$x), 10) * 0 + 1,
              leafr = aggregate(rast(vegp$x), 10) * 0 + 0.3,
              clump = array(0, dim = c(5, 5, 12)),
              leafd = aggregate(rast(vegp$x), 10) * 0 + 0.05,
              leaft = aggregate(rast(vegp$x), 10) * 0 + 0.15)
soilc2 <- list(soiltype = aggregate(rast(soilc$soiltype), 10),
               groundr = aggregate(rast(soilc$groundr), 10) * 0 + 0.15)
# Run and subset point model
micropoint <- runpointmodel(climdata, reqhgt = 0.05, dem, vegp2, soilc2)
micropoint <- subsetpointmodel(micropoint, tstep = "month", what = "tmax")
# Create model input
micro <- modelin(micropoint, vegp2, soilc2, dem)
# Run model for multiple heights
reqhgts<-c(0.01,0.02,0.05,0.1,0.2,0.5,1)
temps <- 0
for (i in 1:length(reqhgts)) {
  mout <- aboveground(micro, reqhgt = reqhgts[i])
  temps[i] <- mout$Tz[2, 2, 132] # Extract temperature for hottest hour
}
par(mar = c (5, 5, 2, 2))
plot(reqhgts ~ temps, type = "l", lwd = 2, xlab = "Temperature", ylab = "Height")
```



One minor point worth noting is that when running the point model above ground, the input `reqhgt` is just used to determine whether to return the required variables for modelling microclimate above ground and the point model itself does not need to be run for separate heights as it just returns temperature and other microclimate variables for the heat exchange surface of the canopy. This is not the case below ground, as here the point model returns microclimate parameters specifically associated with the specified depth.

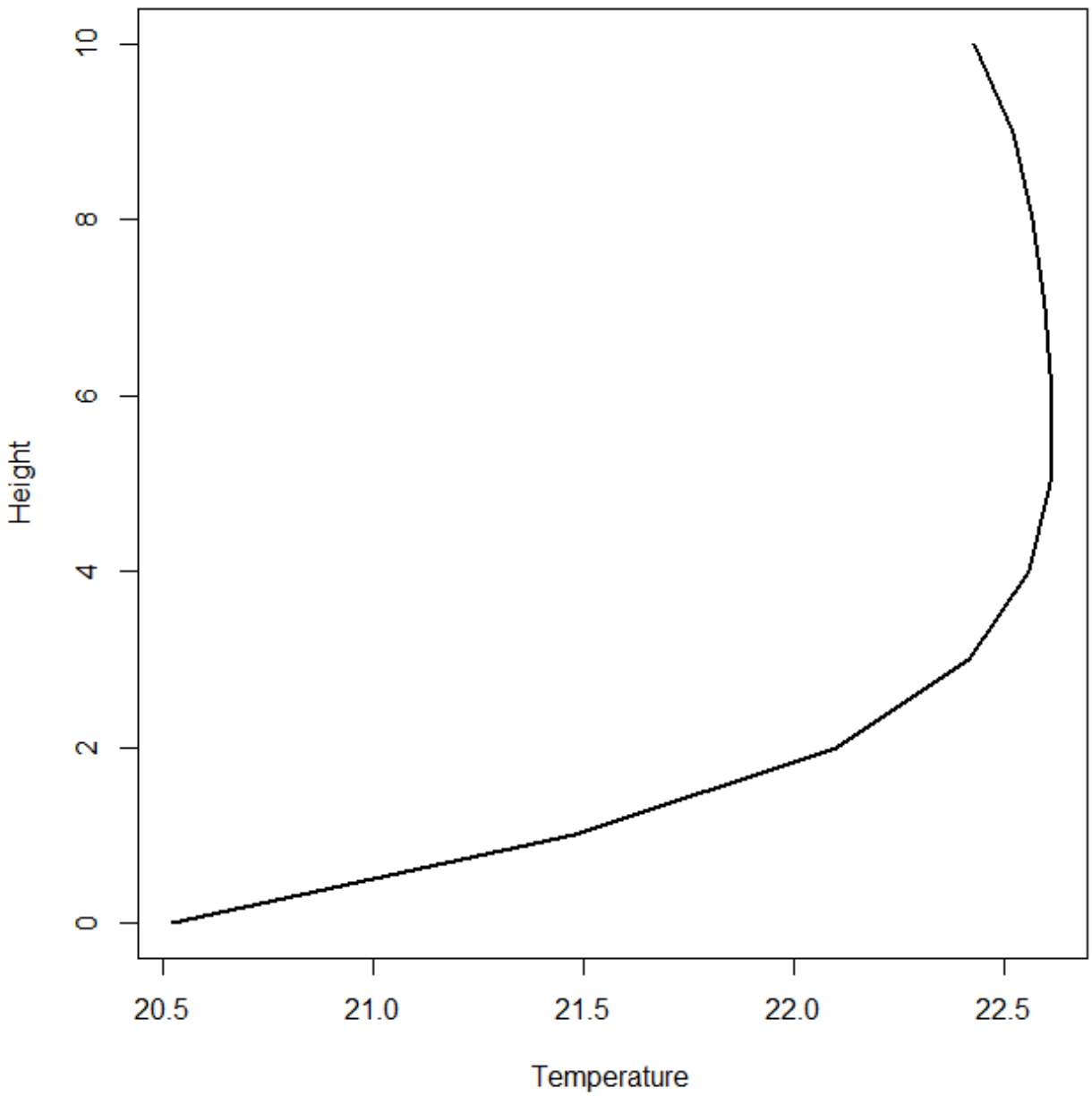
Below canopy

Below canopy, variation in the energy budget within the canopy must be more explicitly handled. The `microclimf` package contains a theoretically-grounded model emulator of Raupach's localised near-field model. In this model, the temperature (or humidity) is assumed to comprise both a `near-field and 'far field' contribution. In essence, the canopy is assumed to comprise multiple layers and far-field contribution is result from heat (or vapour) emanating from the entire canopy downwind of the point of interest. When the net energy balance of the canopy is positive, the result is an approximately linear increase in temperature (or vapour) as one descends through the canopy because at lower heights. However, for far-field-height profile, the effects of ground surface temperature must be accounted for, and the air temperature close to the ground is thus close to ground surface temperature. An additional 'near-field' contribution is then calculated, in the effect determined by the energy budget and the foliage density close to the height of

interest. To calculate the foliage density, plausible assumptions about the vertical distribution of foliage are made, such that foliage density is determined from the plant area index. The model is not unduly sensitive to assumptions about the vertical distribution of foliage.

The net result is a more complex temperature-height (and humidity) profile as re-reproduced in the example below in which the model is run at multiple heights over a small area with spatially uniform, terrain, soil and vegetation properties.

```
# Create spatially uniform vegetation a\nd soil parameters  dataset
vegp2 <- list(pai = array(3, dim = c(5, 5, 12)),
              hgt = aggregate(rast(vegp$hgt), 10) * 0 + 10,
              x = aggregate(rast(vegp$x), 10) * 0 + 1,
              gsmax = aggregate(rast(vegp$x), 10) * 0 + 1,
              leafr = aggregate(rast(vegp$x), 10) * 0 + 0.3,
              clump = array(0, dim = c(5, 5, 12)),
              leafd = aggregate(rast(vegp$x), 10) * 0 + 0.05,
              leaft = aggregate(rast(vegp$x), 10) * 0 + 0.15)
soilc2 <- list(soiltype = aggregate(rast(soilc$soiltype), 10),
               groundr = aggregate(rast(soilc$groundr), 10) * 0 + 0.15)
# Run and subset point model (reqhgt just used to determine whether above or below
ground)
dem <- aggregate(rast(dtmcaerth), 10) * 0
micropoint <- runpointmodel(climdata, reqhgt = 10, dem, vegp2, soilc2)
micropoint <- subsetpointmodel(micropoint, tstep = "month", what = "tmax")
# Create model input
micro <- modelin(micropoint, vegp2, soilc2, dem)
# Run model for multiple heights
reqhgts<- 10^(c(-10:10) / 10)
temps <- 0
for (i in 1:length(reqhgts)) {
  mout <- aboveground(micro, reqhgt = reqhgts[i])
  temps[i] <- mout$Tz[2, 2, 132] # Extract temperature for hottest hour
}
par(mar = c (5, 5, 2, 2))
plot(reqhgts ~ temps, type = "l", lwd = 2, xlab = "Temperature", ylab = "Height")
```



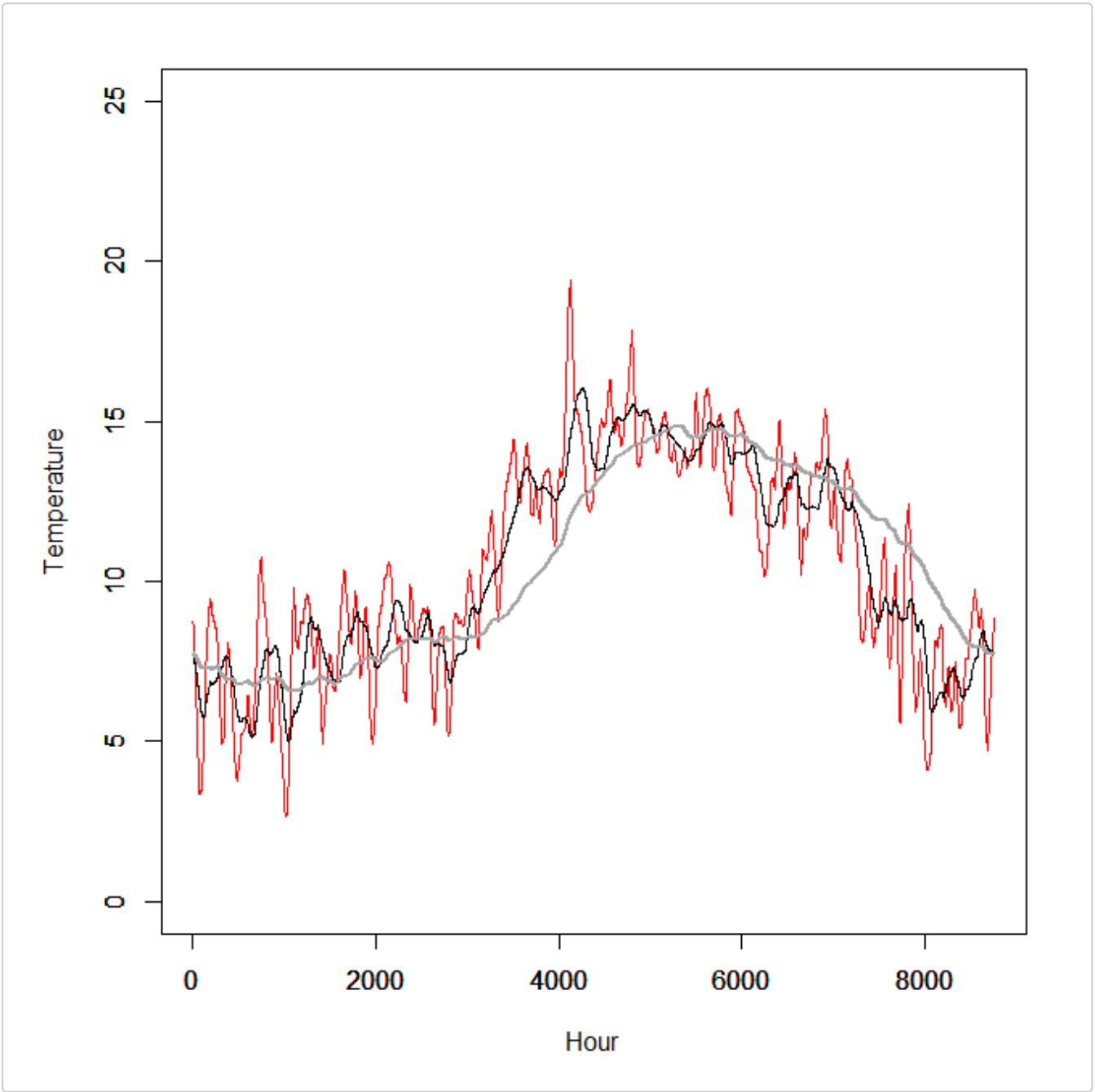
Below ground

If `reqhgt` is negative, functions `below_hr` (hourly) or `below_dy` (daily) are used and it is assumed that microclimatic conditions below ground are needed. The way the model works is to assume, that once ground surface temperatures are calculated, that both the annual and diurnal temperatures cycle is dampened by the specific heat capacity and thermal conductivity of the soil, themselves contingent on soil physical characteristics and water content. There is also a phase shift

This can be seen in the example below in which the model is run in hourly time-increments for the entire year at three different depths and time-series of soil temperature plotted.

```
# Run microclimate model for reqhgt = -0.05m
micropoint1 <- runpointmodel(climdata, reqhgt = -0.05, dem, vegp2, soilc2)
micro1 <- modelin(micropoint1, vegp2, soilc2, dem)
mout1 <- belowground(micro1, reqhgt = -0.05)
T1<-mout1$Tz[2,2,]
# Run microclimate model for reqhgt = -0.2m
micropoint2 <- runpointmodel(climdata, reqhgt = -0.2, dem, vegp2, soilc2)
```

```
micro2 <- modelin(micropoint2, vegp2, soilc2, dem)
mout2 <- belowground(micro2, reqhgt = -0.2)
T2<-mout2$Tz[2,2,]
# Run microclimate model for reqhgt = -1m
micropoint3 <- runpointmodel(climdata, reqhgt = -1, dem, vegp2, soilc2)
micro3 <- modelin(micropoint3, vegp2, soilc2, dem)
mout3 <- belowground(micro3, reqhgt = -1)
T3<-mout3$Tz[2,2,]
# PLot soil temperature time-series
plot(T1, type="l", ylim = c(0, 25), col = "red", ylab = "Temperature", xlab = "Hour")
par(new = TRUE)
plot(T2, type="l", ylim = c(0, 25), ylab = "", xlab = "")
par(new = TRUE)
plot(T3, type="l", col = "darkgray", lwd = 2, ylim = c(0, 25), ylab = "", xlab = "")
```



It should be noted that the below ground microclimate model works best if the model is run over full time sequences. A simpler approximation method is used when running in the model with sub-set versions of the model.

Running the whole model

The whole model is run using `runmicro`. With `method = "R"` this function is essentially a wrapper function that run the various component functions and the call either `aboveground` or `belowground`. With `method = "Cpp"` the full model is run using c++ code, which avoids the need to store full arrays of variables returned by model sub-components in internal memory. Code for running the model in entirety is presented in the quick start section above.

Model output and formats

When running `runmicro` the model returns an an object off class `microout`. If `reqhgt > 0` then the following outputs are returned by default, though note that the user have the option to specify which model outputs are returned using parameter `out', which is a vector of logicals indicating which variables to return ordered as for the listed outputs when 'reqhgt > 0' as below.

- Model outputs:
  - `Tz` - an array of air temperatures at `reqhgt` (deg C)
  - `tleaf` - an array of leaf temperatures at `reqhgt` (or average canopy temperatures for pixels where `reqhgt` is above vegetation (deg C)
  - `T0` - an array of ground surface temperatures (deg C)
    - `relhum` - an array of relative humidities at `reqhgt` (Percentage)
  - `soilm` - an array of soil moisture fractions in the top 10 cm of the soil (m^3 / m^3)
  - `windspeed` - an array of wind speeds at `reqhgt` (m/s)
  - `Rdirdown` - an array of downward direct (beam) radiation fluxes at `reqhgt` (W/m^2)
  - `Rdifdown` - an array of the downward diffuse radiation fluxes at `reqhgt` (W/m^2)
  - `Rlwdown` - an array of the downward longwave radiation fluxes at `reqhgt` (W/m^2)
  - `Rswup` - an array of upward shortwave radiation fluxes at `reqhgt` (W/m^2). Assumed to comprise entirely diffuse radiation.
  - `Rlwup` - an array of upward longwave radiation fluxes at `reqhgt` (W/m^2).

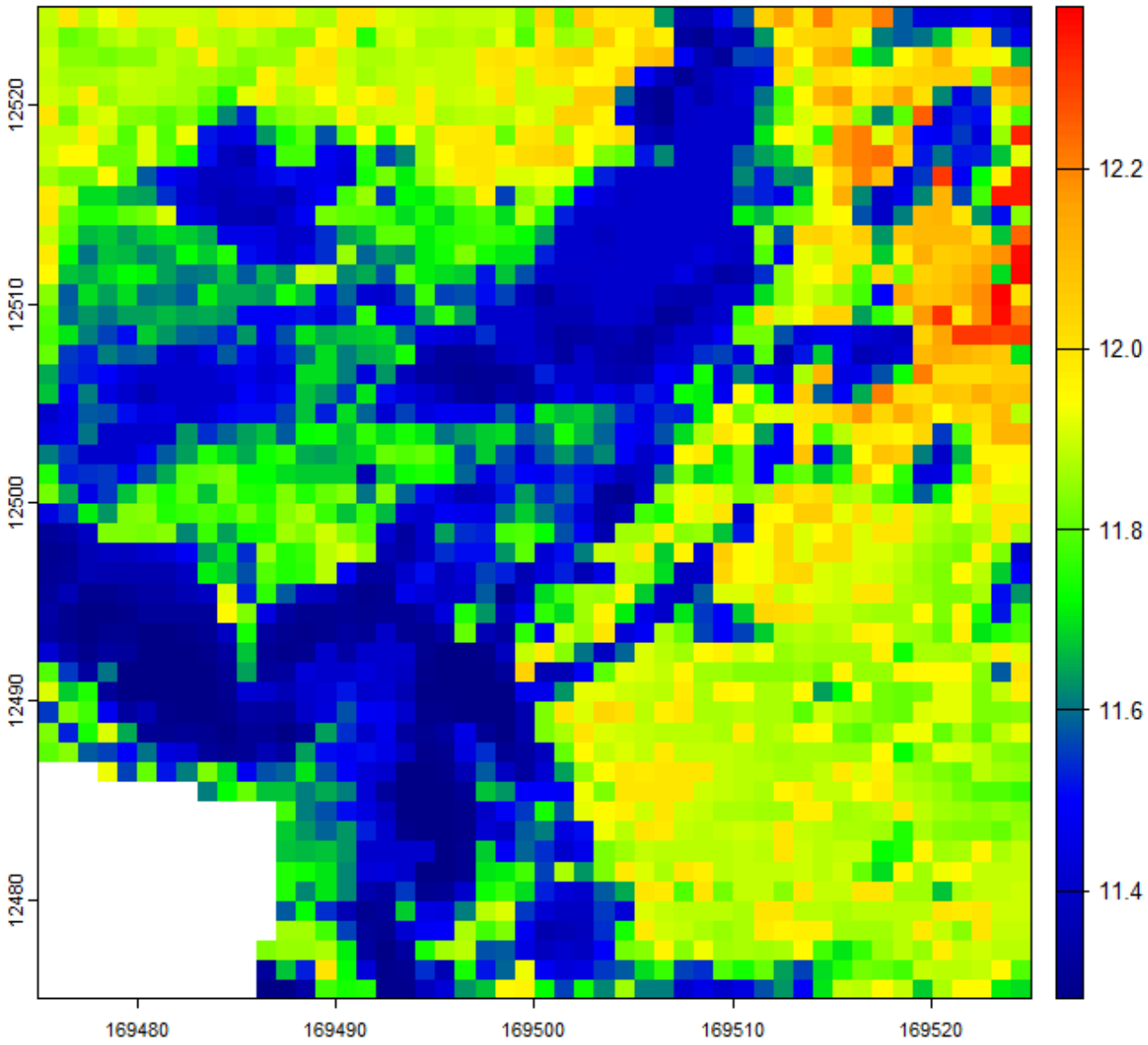
If `reqhgt = 0` the same model outputs are returned, with the exception of wind speed, which is always 0 at the ground surface and relative humidity, where a value for the gorund surface in addition to a value for soil moisture is meaningless.

If `reqhgt < 0` then the same model outputs are return, but `tleaf`, `relhum`, `windspeed`, `Rdirdown`, `Rdifdown`, `Rlwdown`, `Rswup` and `Rlwup` are all set to `NA`.

The returned model object is a list with each of these variables returned as arrays.

It is also possible to write the outputs as netCDF4 files using function `writetonc`. This function takes an object of class `microout` and writes the outputs to the working directory in netCDF4 format. To save disk space, the data are stored as integers and therefore e.g. temperature is mulitplied by 100 prior to writing the data out. To handle the observation times a POSIXlt object of times must be added to the model output prior to writing. In the following example, the model is run,the POSIXlt object added and data written out and then read back in using the terra package:

```
micropoint <- runpointmodel(climdata, reqhgt = 0.05, dtmcaerth, vegp, soilc)
micropoint <- subsetpointmodel(micropoint, tstep = "month", what = "tmax")
mout <- runmicro(micropoint, reqhgt = 0.05, vegp, soilc, dtmcaerth)
mout$tme <- as.POSIXlt(micropoint$weather$obs_time, tz = "UTC")
writetonc(mout, "modelout.nc", dtmcaerth, reqhgt = 0.05)
Tz<-rast("modelout.nc", "Tz")
par(mfrow=c(1,1))
plot(Tz[[12]]/100, col = mypal)
```



Running the model with arrays of climate data

In the examples above, the climate data provided as inputs to the model are provided as a data.frame. However, it may be the case that microclimate surfaces are required over larger areas over which the input climate varies. This situation is at the model input stage and when running the point microclimate model, and by converting the climate variables to multi-layer SpatRasters rather than vectors as in the example below. Here a dummy list of climate arrays is created. Functions `runpointmodela` and `subsetpointmodela` are then called. These functions run and subset the point microclimate model over every grid cell of the array. A coarse resolution dtm is then created that matches the extent, coordinate reference system and resolution of the climate arrays as this is used during the model input handling stage. The grid model can then be run as spreviously: the function checks the format of the data passed to it prior to running the model. HOwever, there are some additional inputs that need to be passed to the function to handle elevation adjustments to the coarse-resolution climate data.

```
# Internal functions
.rast <- function(m,tem) {
  r<-rast(m)
```

```
  ext(r)<-ext(tem)
  crs(r)<-crs(tem)
  r
}
.ta<-function(x,dtm,xdim=5,ydim=5) {
   a<-array(rep(x,each=ydim*xdim),dim=c(ydim,xdim,length(x)))
   .rast(a,dtm)
}
# Create dummy array datasets
dtm <- rast(dtmcaerth) # unpack raster
climarrayr<-list(temp = .ta(climdata$temp, dtm),
  relhum = .ta(climdata$relhum, dtm),
  pres = .ta(climdata$pres, dtm),
  swdown = .ta(climdata$swdown, dtm),
  difrad = .ta(climdata$difrad, dtm),
  lwdown = .ta(climdata$lwdown, dtm),
  windspeed = .ta(climdata$windspeed, dtm),
  winddir = .ta(climdata$winddir, dtm),
  precip = .ta(climdata$precip, dtm))
tme <- as.POSIXlt(climdata$obs_time, tz="UTC")
# Run and subset point model array (using subset defaults)
micropointa <- runpointmodela(climarrayr, tme, reqhgt = 0.05, dtm, vegp, soilc)
micropointa <- subsetpointmodela(micropointa)
# Create coare-resolution dtm matching resolution of climate data
dtmc <- aggregate(dtm, 10, fun = "mean", na.rm = TRUE)
# Run model wiht no altitude correction
mout <- runmicro(micropointa, reqhgt = 0.05, vegp, soilc, dtm, dtmc, altcorrect = 0)
```

The model takes a little longer to run this way, both because of the need to run the point model over multiple grid cells and because the point model outputs are resampled prior to running the grid model, but the code is still fairly efficient.

If `altcorrect == 0', no elevation correction is performed. If`altcorrect > 0' difference between each pixel of dtm and dtmc are calculated and an elevation lapse rate correction is applied to the temperature and pressure data to account for these elevation differences. If `altcorrect`= 1, a fixed lapse rate of 5 degrees per 100m is applied to the temperature data. If `altcorrect`= 2, humidity-dependent lapse rates are calculated and applied.

Running the model over large areas

R stores all data into internal memory and although, by using C++ code, memory requirements are substantially improved, the RAM requirements of the microclimate model can be pretty high if outputs are desired for numerous timesteps at high-resolution over large areas. To circumvent this issue the model can be run as tiles. However, the way that terrain shading effects and topographic wetness are calculated means that model outputs are prone to edge effects. To circumvent this issue, the function `runmicro_big` and `runmicro_biga` can be used, the former handling climate data provided as a data.frame and the latter as a list of arrays. These functions calculated the wind shelter and topographic wetness over the entire study area before running the model in tiles,

In the example below, the application of `runmicro_big` is shown. First some example data are downloaded from Zenodo. These data are 10m resolution model inputs for the entire Lizard Peninsula in Cornwall, an area of approximately 400 square kilometers.

In the first example, using `runmicro_big`, the model is run in tiles with the climate data provided as a data.frame. The code first runs the point microclimate model. The relevant terrain variables are then calculated over the whole study area and the model is then run in tiles. The example takes quite a while to

run, but is included here for illustration purposes. Data are stored in a subfolder called `microut` in the directory specified by `pathout`.

Some warnings are given when running the function to notify the user that direct radiation values in a few instances close to dawn and dusk are higher than expected clear-sky radiation values. This is taken care of by the model - the excess is assigned as diffuse radiation, so is nothing to worry about.

```
# Download example data from Zenodo
url <- "https://zenodo.org/records/15008936/files/runmicrobig.zip"
pathout<-"C:/Temp/tiles/"
dir.create(pathout)
setwd(pathout)
download.file(url, "modeldata.zip")
unzip("modeldata.zip")
# Read in spatial data
big_vegp <- readRDS("vegp_big.RDS")
big_soilc <-readRDS("soilc_big.RDS")
dtm<-rast("dem.tif")
# Read in climate dataframe
climdatadf <- readRDS("climdatapoint.RDS")
# Run and subset point model
micropoint <- runpointmodel(climdatadf, reqhgt = 0.05, dtm, big_vegp, big_soilc)
micropoint <- subsetpointmodel(micropoint, tstep = "month", what = "tmax")
# Run the model in tiles
runmicro_big(micropoint, reqhgt = 0.05, pathout = pathout, big_vegp, big_soilc, dtm)
```

The optimal tile size is automatically calculated. Entirely blank tiles (i.e anything with NA in the digital elevation dataset are skipped). Users have the option to specify the tile size or format of the data written to disk and there are quite a few other options for flexibly handling how the model is run. The help file associated with `runmicro_big` gives details. One useful feature is the ability to specify a tile overlap to avoid possible tiling effects. Model output variables for overlapping tiles can then, once converted to `terra::SpatRasters`, can be merged using function `mosaicblend`. This applies a distance weigting to the overlapping area so that sharp boundary effects are avoided. Another potentially useful option is to write the model outputs as compressed nc files.

In the second example, using `runmicro_biga`, 1km resolution gridded climate data are used to drive the model. It is assumed that data have already been dowenloaded and unzipped. Again, the the code first runs the point microclimate model, but here the point model must be run for each of the ~400 1km grid cells, which itself takes a little while. Again, the relevant terrain variables are then calculated over the whole study area and the model is then run in tiles. As with `runmicro_big`, the size of tiles and the degree of overlap can be specified as user inputs.

```
# Assumes data already downloaded from Zenodo - see above
pathout<-"C:/Temp/tiles/"
setwd(pathout)
big_vegp <- readRDS("vegp_big.RDS")
big_soilc <-readRDS("soilc_big.RDS")
dtm<-rast("dem.tif")
# Read in gridded climate data
climdatag <- readRDS("climdatagrid.RDS")
# Create other input variables
tme <- as.POSIXlt(c(0:8783) * 3600, origin = "2020-01-01 00:00", tz = "UTC")
dtmc <- aggregate(dtm, 100, fun = "mean", na.rm = TRUE)
# Run and subset point model over each 1km grid cell
micropointa <- runpointmodela(climdatag, tme, reqhgt = 0.05, dtm, big_vegp, big_soilc)
```

```
micropointa <- subsetpointmodela(micropointa, tstep = "month", what = "tmax")
# Run microclimate model in tiles (data saved to "C:/Temp/tiles2/")
runmicro_big(micropointa, reqhgt = 0.05, pathout = "C:/Temp/tiles2/", big_vegp,
big_soilc, dtm, dtmc, altcorrect = 0)
```

Note that in contrast to when run using a point model input, altitudinal corrections can be handled using the control parameter `altcorrect`.

Bioclim variables

It is common practice to seek to model the distributions of species using bioclimate variables, and the most commonly used are those available from Worldclim. While it is far from the case that simply modelling species distributions using microclimate data will circumvent all the numerous issues associated with doing so using coarse-resolution macroclimate data, a function for modelling microclimate equivalents of the standard 19 bioclim variables is nonetheless provided.

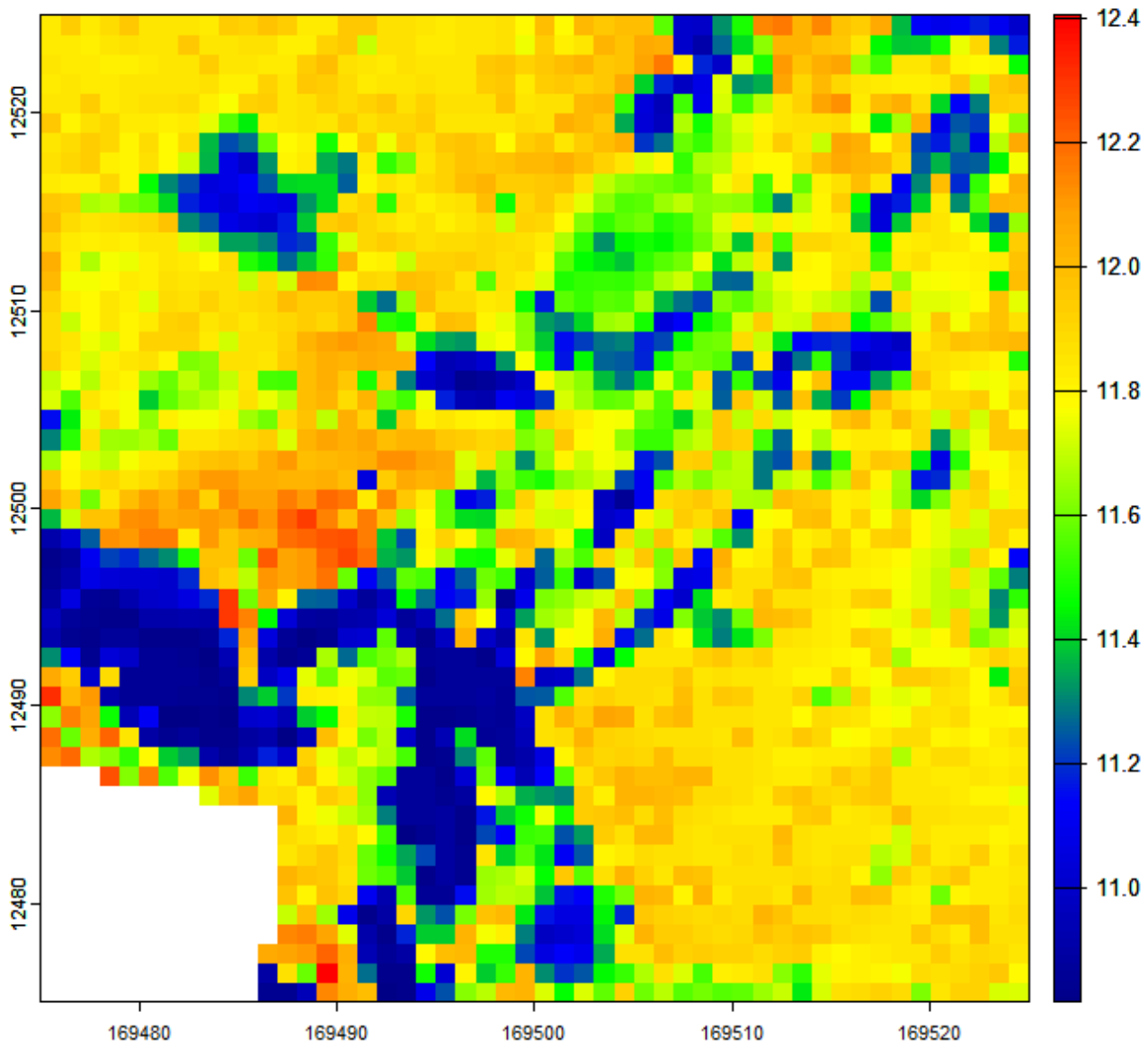
Because rainfall typically does not vary that much at fine- spatial resolution, the function instead calculates soil moisture equivalents for the rainfall-associated bioclim variables (in the top 10 cm of the soil).

To enhance computational efficiency the microclimate model is run for selected days only. Thus, to compute “mean annual temperature”, the mean ambient temperature of each day in the input weather data is calculated, the day with median temperatures in each month selected and the mean across months calculated. This is not, strictly speaking, the same as the mean temperature, but differences are likely to be minor, and for each year of data supplied, there is an approximately 30-fold gain in computational efficiency by calculating e.g. BIO1 in this way. Similarly, to calculate maximum temperature (BIO1), the day of the year with the hottest ambient temperature is selected, and microclimate temperatures calculated on this day only. This ignores the possibility that on a slightly cooler, but sunnier day, microclimate temperatures may be hotter at certain locations. AS with the various runmodel options, If `hourly = TRUE` all hours within a given day are selected and calculations performed on hourly data. If If `hourly = FALSE` only the hours corresponding to times when hourly temperatures are at their daily maximum and minimum and selected. This results in a c. 10-fold increase in computational efficiency, but cannot pick out areas where terrain results in near-ground temperatures reaching a maximum later in the afternoon than the peak in ambient temperature. If weather data for more than one year are supplied, only one set of median, maximum and minimum monthly temperature data are selected representing an average across years. Resultantly, there is little computational penalty if providing data for multiple years in comparison to one year of data.

However, here we illustrate the function for one year of data only using the default inputs provided with the package. Note that this function flexibly handles the weather data being provided as either a data.frame or as a list of arrays. In the example below it is provided as a data.frame. The function takes ~20 seconds to run. The function returns a multilayer SpatRast of each of the bioclim variables

The input parameter `temp` indicates whether to return air or leaf temperature-derived bioclim variables. If `temp = leaf`, for grid cells where `reqhgt` is above vegetation, vertically averaged canopy temperature is used.

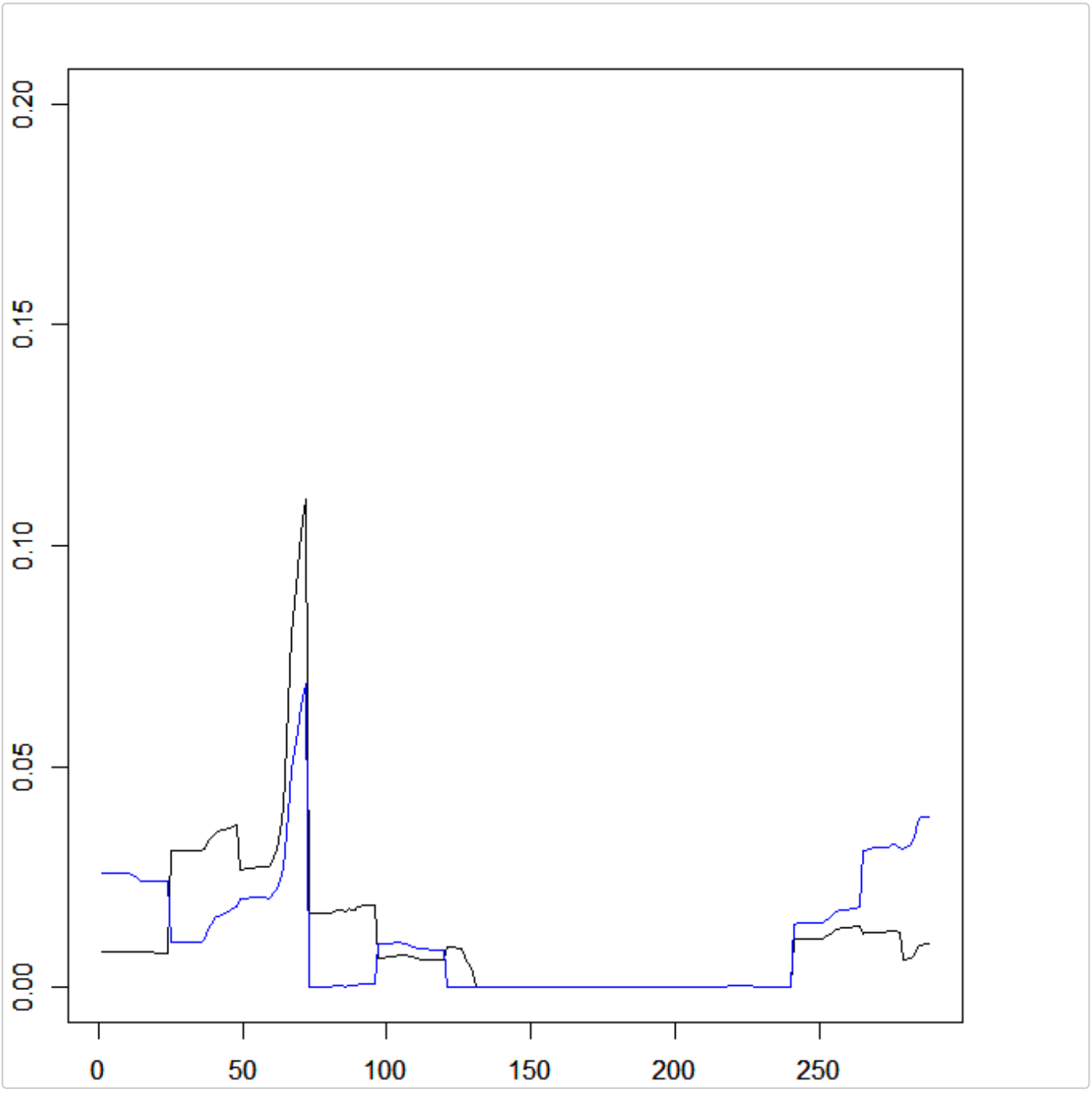
```
mypal <- colorRampPalette(c("darkblue", "blue", "green", "yellow", "orange", "red"))
(255)
# Set back to inbuilt datasets
vegp <- microclimf::vegp
soilc <- microclimf::soilc
bioclim <- runbioclim(climdata, 0.05, vegp, soilc, dtm, temp = "air")
plot(bioclim[[1]], col = mypal) # BIO1
```



Snow

In all the examples above, no account of snow cover is taken, primarily because the location used for the examples, being nearly sub-tropical, is snow free. However, the same may not be true of other places, and the option to account for snow cover is included. This is handled by setting `snow = TRUE` in the input of `runmicro` and then also passing the outputs of the snow depth model. There are two options for running the snow model: a quick and slower method. The distinction between the only matters when microclimate is modelling for a subset of days. Using the slow method a full hourly model is run for each grid cell and snow depth, as one would expect, is partially contingent on snow depth in the previous time-step. The resulting output is then subset if required. Using the quick method, a point snow model is run for every hour, but the full grid model is only run for the subset of days for which snow depths are required. To ensure a sensible snow budget the point model is used to calculate hourly melt, and the nature of the terrain and vegetation used to calculate a melt factor by which to multiply snow melt derived form the point model. The application of both approaches is shown below.

```
# Run and subset micropoint model using inbuilt datasets
climdata$temp <- climdata$temp - 8 # Make it colder so there is snow
micropoint <- runpointmodel(climdata, reqhgt = 0.05, dtmcaerth, vegp, soilc)
micropoint <- subsetpointmodel(micropoint, tstep = "month", what = "tmin")
# Run the snow model using the quick method
smod1 <- runsnowmodel(climdata, micropoint, vegp, soilc, dtmcaerth, method = "fast")
# Run the snow model using the slow method
smod2 <- runsnowmodel(climdata, micropoint, vegp, soilc, dtmcaerth, method = "slow")
# Compare mean ground snow depths through time
sdepth1 <- apply(smod1$groundsnowdepth, 3, mean, na.rm = TRUE)
sdepth2 <- apply(smod2$groundsnowdepth, 3, mean, na.rm = TRUE)
plot(sdepth1, type = "l", ylim = c(0, 0.2))
par(new = TRUE)
plot(sdepth2, type = "l", ylim = c(0, 0.2), col = "blue")
```



In addition to ground snow depth, total snow water equivalent and average snowpack and ground snow temperature are returned. Though the quick method is a bit crude, the resulting microclimate outputs are

very similar to those obtained using the slow method, only really deviating when the two methjods give different estimates of snow cover. Here we demonstrate this by running and comparing the microclimate model using both snow outputs.

```
        # Run microclimate model with snow using outputs from the quick model
        mout1 <- runmicro(micropoint, reqhgt = 0.05, vegp, soilc, dtmcaerth, snow = TRUE, snowmod
= smod1)
        # Run microclimate model with snow using outputs from the quick model
        mout2 <- runmicro(micropoint, reqhgt = 0.05, vegp, soilc, dtmcaerth, snow = TRUE, snowmod
= smod2)
        Tz1 <- apply(mout1$Tz, 3, mean, na.rm = TRUE)
        Tz2 <- apply(mout2$Tz, 3, mean, na.rm = TRUE)
        plot(Tz1, type = "l", ylim = c(-10, 25), ylab = "Temperature")
        par(new = T)
        plot(Tz2, type = "l", ylim = c(-10, 25), col = "blue", ylab = "")
```