https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

palm_csd 25.10: A processing tool for static input data in the PALM model system

Sebastian Schubert[1], Julian Anders[2], Tobias Gronemeier[4], Björn Maronga[2], and Mohamed Salim[1,3]

[1]Technische Universität Berlin, Institute for Ecology, Straße des 17. Juni 135, 10623 Berlin, Germany
[2]Leibniz Universität Hannover, Institute of Meteorology and Climatology, Herrenhäuser Str. 2, 30419 Hannover, Germany
[3]Aswan University, Sahary City, Airport Road, 81528 Aswan, Egypt
[4]pecanode GmbH, Peterstr. 30, 38640 Goslar, Germany

Correspondence: Sebastian Schubert (schubert.2@tu-berlin.de)

Abstract. We present palm_csd version 25.10, the current default preprocessing tool for generating the static driver for the building-resolving large-eddy simulation model PALM. The static driver defines the spatial surface characteristics of the simulation domain. This paper provides a technical description of the updated palm_csd workflow, focusing on the processing of buildings, vegetation, pavement, water bodies, terrain height and land cover in compliance with the PALM Input Data Standard (PIDS). Major extensions introduced since the previous description include the processing of georeferenced raster and vector data with automated reprojection, user-defined domain rotation and nesting, enhanced handling of building parameters, optimized generation of resolved vegetation, estimation of leaf area index from vegetation height, and the derivation of static input for non-building-resolving simulations based on Local Climate Zone classifications. We demonstrate the application of palm_csd using publicly available geodata for the city of Berlin (Germany), covering both building-resolving and LCZ-based simulation setups. Common data inconsistencies and sources of uncertainty in urban geodata are discussed. palm_csd 25.10 provides a reproducible, flexible and continuously maintained framework for transforming heterogeneous geospatial datasets into PALM-compatible static drivers to support both detailed urban morphology and coarser-scale urban climate applications.

1 Introduction

With the advent of modern high-performance computing, detailed simulation of the atmospheric flow field in urban areas has become feasible. Large-eddy simulation (LES)-based microscale atmospheric models, once limited to idealized flows, are now routinely applied to complex urban environments and form the basis of microscale urban climate modeling. This progress has been accompanied by advances in numerical methods, turbulence representation and physical parameterizations of energy and mass exchange at the urban surface. State-of-the-art microscale building-resolving models now resolve the coupled processes of momentum, heat and radiation transfer within the urban canopy layer and integrate comprehensive schemes for surface energy balance, radiative exchange and vegetation–atmosphere interactions (Maronga et al., 2020; Salim et al., 2018; Eichhorn and Kniffka, 2010; Bruse and Fleer, 1998). These developments created the methodological foundation for investigating the physical mechanisms governing urban climate and for addressing challenges such as heat stress, ventilation and outdoor comfort in the context of microclimate research for sustainable city development.

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Building on these advances, the PALM model system (Maronga et al., 2020) has become one of the most widely used modeling frameworks for microscale urban climate studies. PALM employs a LES approach based on the filtered, incompressible Navier–Stokes equations, with buoyancy treated through the Boussinesq approximation. Its modular structure incorporates specialized components such as the urban surface model (Resler et al., 2017), land surface model (Gehrke et al., 2021), plant canopy model (Maronga et al., 2015), radiative transfer model (Krč et al., 2021; Salim et al., 2022), building energy model (Pfafferott et al., 2021) and atmospheric chemistry module (Khan et al., 2021). These components allow PALM to explicitly resolve the three-dimensional interactions between the built environment and the atmosphere, enabling realistic simulations of urban boundary layers, surface energy fluxes and thermal comfort indicators (Anders et al., 2023, 2025).

Realistic PALM simulations require high-quality input data that represent both the spatial configuration of the urban surface and the time-dependent atmospheric forcing. Two categories of input are required: the static driver, which defines all spatially invariant characteristics of the simulation domain, and the dynamic driver, which provides transient meteorological boundary and initial conditions derived from mesoscale models or observations. The static driver includes elements such as topography, building geometry, land cover and vegetation, and is thus of particular importance because it encodes the physical structure and material properties that govern radiative exchange, energy storage and aerodynamic roughness within the urban canopy. Preparing these data, however, remains a demanding and time-consuming task that involves gathering and harmonizing heterogeneous geospatial sources, translating them into the PALM Input Data Standard (PIDS, Heldens et al., 2020) and ensuring internal consistency across a large number of variables. Manual preparation is feasible for small test cases but becomes impractical for city-scale applications or when high spatial resolution is required.

To address this challenge, the Python-based command line palm_csd tool (PALM tool to Create Static Driver) was developed as the standard and officially distributed preprocessor of the PALM model system. palm_csd provides a structured and reproducible workflow for transforming heterogeneous geospatial datasets into PIDS-compliant static drivers. It supports a wide range of input formats, performs automated data consistency checks and ensures that all surface and geometric parameters required by PALM are correctly formatted and physically coherent. The tool was initially established within the MOSAIK project (Maronga et al., 2019; Heldens et al., 2020) to process high-resolution datasets for several German cities and has since become the reference framework upon which more specialized preprocessors, such as the PALM-4U GUI (Winkler et al., 2023), PALM-GEM (Bureš and Resler, 2024), GEO4PALM (Lin et al., 2024) and SanDyPALM (Vogel et al., 2025), have been developed.

Although several new preprocessing tools have emerged recently (Fluck, 2023; Bureš and Resler, 2024; Vogel et al., 2025), palm_csd is designed to be the reference implementation for creating static input data in the PALM model system. Its design follows the PIDS and is continuously updated to reflect ongoing developments and new features in the PALM model itself. This integration ensures that the structure and content of static drivers remain fully compatible with the current PALM release version.

The objective of this paper is to provide a technical description of the palm_csd preprocessor as of version 25.10, including its input data model, internal structure and workflow for generating PALM-compatible static drivers. We describe how the tool processes and validates geospatial input data, assigns surface and material properties, constructs three-dimensional fields such

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

as building masks and leaf area density, and ensures compliance with the PIDS. Furthermore, we highlight recent extensions and improvements introduced in the latest PALM releases, including enhanced data validation, additional input variables and more flexible file handling.

After explaining in detail the features of palm_csd, we present an approach to generate a static driver for Berlin (Germany) based on freely available data, both for detailed building-resolving simulations and for coarse non-building-resolving simulations based on Local Climate Zone data. This is followed by a discussion of palm_csd's features compared to other tools as well as input data availability and the resulting uncertainties.

2 Description of palm_csd

palm_csd supports all maintained Python versions (at the time of writing, 3.10 to 3.14) with extensive automatic tests, including static type checking with mypy, as well as linting and formatting with ruff. All user input is checked for consistency. The configuration file for palm_csd is written in the YAML file format (Schubert, 2026).

Compared to the palm_csd version described in Heldens et al. (2020), in addition to many minor enhancements, we mainly

- added processing of georeferenced raster and vector data, including the calculation of coordinates with support for arbitrary rotation angles,
- added support for building parameters,
- added an optional removal of buildings in a buffer zone at the borders of a domain,
- optimized the generation of resolved vegetation,
- added estimation of the leaf area index from vegetation height,
- added Local Climate Zone-based input and the generation of parameters for an urban parametrization scheme,
- added input data checks and extensive output messages,
- added a statistics and visualization tool,
- added extensive documentation,
- changed the configuration to the YAML format, and
- added automatic tests and improved the code quality.

In the following, the complete features of palm_csd are explained in detail: the general set-up of the domains and input file handling as well as the processing of buildings, vegetation, pavement, water bodies and terrain height. After that, an alternative approach for non-building-resolving simulations is presented based on Local Climate Zones data input as well as a short section about other tools using palm_csd. Throughout the paper, we use the names of pavement, vegetation and water types as used in the set-up of palm_csd, e.g. `bare_soil` instead of "bare soil", to allow for direct usage of these names by the user.

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

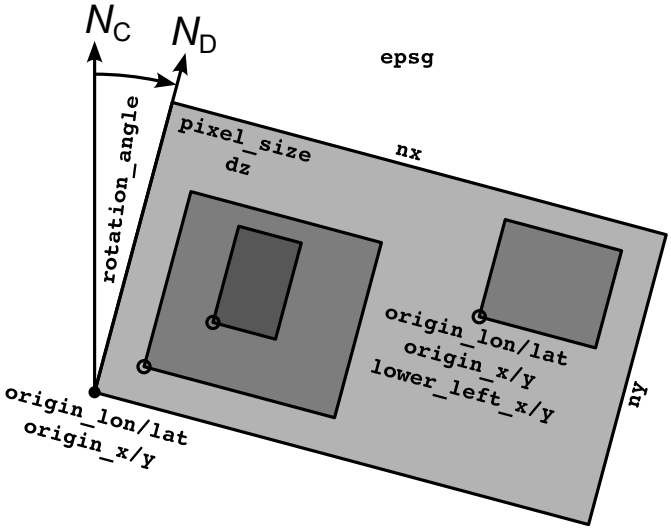


Figure 1. Illustration of nested domains. The `rotation_angle` describes the angle between the projected coordinate system's north axis N_C and the north axis of the domain N_D. Together with the `epsg` code that describes the coordinate reference system, it is applied to all domains. The other parameters are set on a per-domain basis: grid points in the horizontal and vertical directions, `nx`+1 and `ny`+1, horizontal `pixel_size` (`dx` and `dy` in PALM) and the vertical grid spacing `dz`.

2.1 Domain set-up, input files and their geographic processing

palm_csd supports the generation of a root domain and an arbitrary number of nested domains. It checks for sufficient space at the borders of the domains to the respective parent domain and for any overlap between the domains. If domains overlap, only one-way nesting is allowed. All domains share one target coordinate system and one rotation angle (Fig. 1). The latter is the angle between the projected coordinate system's north axis N_{C} and the domain's north axis N_{D}. The input data, when in a georeferenced format as described below, will be automatically reprojected and the rotation angle will be applied. This allows datasets with varying input projections to be used without manual conversion, thereby reducing preprocessing and ensuring a consistent final projection.

The position of the domain is set by the coordinates of the lower-left corner of the domain, either in the longitude/latitude system WGS84 or in the target coordinate system. For nested domains, palm_csd will ensure that the nest aligns with its parent by slightly adjusting the lower-left corner coordinates. It will also calculate the distance to the lower-left corner of the root parent domain, which is needed for the set-up of nesting in PALM. Alternatively, this distance can be set directly, without specifying the lower-left corner coordinates of the nest.

palm_csd supports a variety of input file formats, including 2D raster data and vector data. The input files can be in georeferenced formats such as GeoTIFF for rasters or ESRI Shapefiles for vector data. The resampling algorithm can be chosen for the downscaling and upscaling of the input data when reprojecting or changing the grid. In order to preserve the values of categorical data, only nearest-neighbour and mode-resampling are allowed. For all other data types, all algorithms supported by

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

the employed Python GIS package rasterio[1] can be selected. Note that the different methods handle missing values differently. While nearest-neighbour resampling produces a missing value when the centre of the target pixel is closest to a missing value in the source data, the other methods calculate values as soon as a part of the target pixel is covered by a non-missing pixel in the source data. In order to ensure consistency between the different data types, the missing values of nearest-neighbour resampling are applied to all data types.

2.2 Buildings

For palm_csd, each building pixel is defined by a building height, a building type and a unique building identifier (building ID), which is used to associate all pixels belonging to the same building.

If the building input is given as a vector polygon file, at least the building height column needs to be specified. All polygons with a missing building height are ignored. Optionally, the building type and building ID can be specified. If the building type is not given, it is set to `residential_1951_2000`. If the building ID is not given, it is automatically generated with a different value for each polygon.

If the building input is given as raster files, both the building height and the building ID need to be specified, as palm_csd cannot identify separate buildings from the raster data. The building type can be specified as well, but if it is not given, it is set to `residential_1951_2000`.

In addition to 2D buildings, palm_csd supports bridge-like structures. These are defined by their upper height and structural depth. The latter is set for the entire domain via the configuration file. Similar to the standard buildings, the bridge height can be given as a column in a vector polygon or as a raster file. In the latter case, the IDs need to be supplied, whereas in the former case, they are optional. The building type of a bridge is set to `bridge`. When bridges are present in the domain, the generated static driver will include a 3D representation of the buildings, which is required to represent the airspace below the bridges. In all other cases, this field can be enabled manually to, for example, enable the 3D visualization of the domain.

Since the PALM release 24.04, PALM's building parameters, such as building surface albedo, building vegetation cover and building thermal properties, are supplied in the static driver in separate variables (Table 1) with distinct dimensions (Table 2) instead of the single variable `building_pars`. In palm_csd, these parameters can be supplied as attributes to building polygons and as raster files for individual buildings. For example, setting the `building_heat_conductivity_wall_roof_2` attribute or supplying the respective raster data will set the heat conductivity of the second roof layer with the first layer being the outermost layer. More generally, setting `building_heat_conductivity_wall` will define the heat conductivity of all layers of all wall materials (ground floor, above ground floor and roof). In addition, default values can be set for the entire domain via the configuration file with a similar nomenclature.

Optionally, palm_csd can ensure that a buffer zone of a chosen width at the borders of the domain is building-free to enhance the stability of the PALM run.

[1] https://rasterio.readthedocs.io/en/latest/api/rasterio.enums.html#rasterio.enums.Resampling

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Table 1. Building and building surface parameters of PALM and their dimensions (cf. Table 2). In palm_csd, they can be set individually for each building polygon, read from a raster map and defined globally for all building.

Variable	Dimensions	Description
building_albedo_type	(building_surface_type,y,x)	albedo type
building_emissivity	(building_surface_type,y,x)	emissivity
building_fraction	(building_surface_type,y,x)	wall, window and green fractions
building_general_pars	(building_general_par,y,x)	general parameters
building_heat_capacity	(building_surface_type,building_surface_layer,y,x)	heat capacity
building_heat_conductivity	(building_surface_type,building_surface_layer,y,x)	thermal conductivity
building_indoor_pars	(building_indoor_par,y,x)	indoor parameters
building_lai	(building_surface_level,y,x)	leaf-area index at green fraction
building_roughness_length	(building_surface_level,y,x)	roughness length momentum
building_roughness_length_qh	(building_surface_level,y,x)	roughness length heat and moisture
building_thickness	(building_surface_type,building_surface_layer,y,x)	wall-layer thicknesses
building_transmissivity	(building_surface_level,y,x)	transmissivity of windows

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Table 2. Dimensions of PALM's building parameters. In addition, `building_surface_layer` includes surface layers 1 to 4.

(a) `building_surface_level`

Attribute	Indices	Description
`gfl`	0	ground floor level (gfl)
`agfl`	1	above ground floor level (agfl)
`roof`	2	roof

(b) `building_general_pars`

Attribute	Indices	Description
`height_gfl`	0	ground floor level height
`green_type_roof`	1	type of green roof

(c) `building_surface_type`

Attribute	Indices	Description
`wall_gfl`	0	wall gfl
`wall_agfl`	1	wall agfl
`wall_roof`	2	wall roof
`window_gfl`	3	window gfl
`window_agfl`	4	window agfl
`window_roof`	5	window roof
`green_gfl`	6	green gfl
`green_agfl`	7	green agfl
`green_roof`	8	green roof

(d) `building_indoor_pars`

Attribute	Indices	Description
`indoor_temperature_summer`	0	indoor target summer temperature
`indoor_temperature_winter`	1	indoor target winter temperature
`shading_window`	2	shading factor
`g_window`	3	g-value windows
`u_window`	4	u-value windows
`airflow_unoccupied`	5	basic airflow without occupancy of the room
`airflow_occupied`	6	additional airflow dependent on occupancy of the room
`heat_recovery_efficiency`	7	heat recovery efficiency
`effective_surface`	8	dynamic parameter specific effective surface
`inner_heat_storage`	9	dynamic parameter inner heat storage
`ratio_surface_floor`	10	ratio internal surface/floor area
`heating_capacity_max`	11	maximal heating capacity
`cooling_capacity_max`	12	maximal cooling capacity
`heat_gain_high`	13	additional internal heat gains dependent on occupancy
`heat_gain_low`	14	basic internal heat gains without occupancy
`height_storey`	15	storey height
`height_ceiling_construction`	16	ceiling construction height
`heating_factor`	17	anthropogenic heat output factor for heating
`cooling_factor`	18	anthropogenic heat output factor for cooling

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

2.3 Vegetation

PALM represents vegetation in two ways: as flat, vertically unresolved vegetation types or as resolved vegetation described by the 3D distribution of the density of the vegetation. Unresolved vegetation is represented by vegetation types such as `short_grass` or `evergreen_shrubs` together with the leaf area index (LAI). The LAI (unit $\mathrm{m}^2/\mathrm{m}^2$) is defined as the one-sided area of leaves per ground area. The unresolved vegetation is used for vegetation that does not cover the full height of a grid cell. Resolved vegetation is defined by the leaf area density (LAD) and the basal area density (BAD). The LAD (unit $\mathrm{m}^2/\mathrm{m}^3$) is defined as the one-sided area of leaves per volume, while the BAD is defined as the area of branches per volume. This approach is preferred for vegetation that covers at least the height of a grid cell.

The LAI can be reproduced from the LAD by vertically summing over the LAD multiplied by the vertical grid spacing Δz:

$$\mathrm{LAI} = \sum_k \mathrm{LAD}_k \, \Delta \mathrm{z} \, . \tag{1}$$

2.3.1 Unresolved vegetation

palm_csd supports the direct input of the different vegetation types that PALM supports. Additionally, the LAI can be supplied, which is then used instead of the default values in PALM. Note that vegetation types that represent high (grown) vegetation feature large roughness lengths z_0 and $z_{0,\mathrm{h}}$ for momentum and heat, respectively, with $z_0 > 1\,\mathrm{m}$ and $z_{0,h} > 1\,\mathrm{m}$. These vegetation types are therefore not suitable for the vertical grid spacing typically used in building-resolving simulations. If the vertical grid spacing is close to or smaller than the roughness lengths, PALM will crash or will not provide meaningful results. Thus, by default, high vegetation types are replaced by resolved vegetation depending on what is described below.

Besides supplying the vegetation type directly, it can be derived from a column that includes strings or values that need to be mapped to PALM's vegetation types. This column could possibly also include pavement or water types. Vegetation cover on buildings can be set as a building parameter, as explained above.

2.3.2 Resolved vegetation

palm_csd can generate LAD and BAD fields from two input variants: single tree input and vegetation patch input. In the single tree case, LAD and BAD are generated from detailed information of individual trees, while in the vegetation patch case, when detailed information is missing, LAD is generated from the vegetation height, LAI and patch type information. If not directly available as input, the LAI can be estimated from the vegetation height h with

$$\mathrm{LAI} = \lambda_{\mathrm{LAI}} \cdot h \, , \tag{2}$$

with a default value of 0.2 for the factor λ_{LAI} following Vogel et al. (2025). Alternatively, separate constant LAIs for low and high vegetation can be defined.

The LAD and BAD fields of single trees are generated from single tree information. A single tree is defined by its height, crown diameter, trunk diameter, crown height (all in m) and shape (see Heldens et al., 2020, for the different available shapes).

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

The input quantities can be specified as a single file or as several vector point files, with the columns representing the respective input data or as separate corresponding raster files. In the case of vector point input, the tree type can also be derived from a column that includes the tree species name as text. The content of this column is compared with the species name used in the tree default table (see Heldens et al., 2020). If one of the tree attributes is missing, default values from this table are used.

In order to capture tree-like vegetation in areas where detailed tree data are not available, palm_csd can generate LAD distributions inside these "vegetation patches"; BAD values are currently not generated. A vegetation patch is identified if one of the following conditions is met: a) the vegetation height is larger than a threshold, b) a vegetation type defined as "high vegetation" is present, or c) a defined patch type is present.

For each identified vegetation patch pixel, one of the following continuous vertical LAD profiles is assumed: The profile of Markkanen et al. (2003) is given (up to a normalization constant) by

$$\mathrm{LAD}_{\mathrm{M03}}(z) \propto \left(\frac{z}{h}\right)^{\alpha-1}\left(1-\frac{z}{h}\right)^{\beta-1} \tag{3}$$

where α and β are the shape parameters of the profile. The profile of Lalic and Mihailovic (2004) is given by

$$\mathrm{LAD}_{\mathrm{LM2004}}(z) = L_{\mathrm{m}}\left(\frac{h-z_{\mathrm{m}}}{h-z}\right)^{n}\exp\left[n\left(1-\frac{h-z_{\mathrm{m}}}{h-z}\right)\right] \quad \text{with} \quad n=\begin{cases}6 & \text{for } 0\le z<z_{\mathrm{m}}\\ 0.5 & \text{for } z_{\mathrm{m}}\le z\le h\end{cases}. \tag{4}$$

Here, z_{m} is the relative height of the highest LAD value L_m and a parameter of this profile. L_m is given by the normalization of the profile to the LAI. In order to support the choice of a profile, palm_csd comes with an interactive tool to visualize the different profiles depending on α, β and z_{m}, respectively.

In order to discretize the LAD profile for the PALM grid, the LAD is integrated over the vertical grid spacing Δz and averaged over the grid cell height z_k to z_{k+1}, where z_k is the lower height of the grid cell and z_{k+1} is the upper height of the grid cell. The resulting LAD value for each grid cell is given by

$$\mathrm{LAD}_k = \frac{1}{\Delta z}\int_{z_k}^{z_{k+1}} \mathrm{LAD}(z)\,\mathrm{d}z\,. \tag{5}$$

When evaluated with the profiles (3) and (4), the expressions include the regularized incomplete beta function and the non-regularized upper incomplete gamma function, respectively. Both functions are taken from the SciPy package (Virtanen et al., 2020).

For both single trees and vegetation patches, it is ensured that there are no LAD and BAD fields within buildings. High vegetation types for which resolved vegetation is generated are replaced by a chosen vegetation type.

2.4 Pavement, street surfaces and street crossings

Pavement is characterized by its material type, which is used to determine the energetic interaction with the surface and the atmosphere, and by its street type, which can be used to prescribe traffic-based pollutant emissions. The street types closely

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

follow the OpenStreetMap classification. Street crossings indicate the locations where pedestrians can cross streets and are used by PALM's multi agent system.

Both pavement and street types can be supplied directly. Alternatively, the pavement type can be derived from a column that includes strings or values that need to be mapped to PALM's pavement types and that possibly also include other types. The only valid value for street crossing is `1` to indicate that pedestrians can cross the street. It can be supplied as vector polygons or as a raster file.

2.5 Water surfaces

Water surfaces are characterized by their type and a prescribed initial water body temperature. In addition to directly supplying the water type numerically, it can be derived from a column that includes strings or values that need to be mapped to PALM's water types and that possibly also include other types. The water temperature can be set for input polygons, as raster files or in the domain configuration on a per-type basis.

2.6 Terrain height

The terrain height can be supplied as a raster file or as a column in a vector polygon. If no terrain height is given for the entire domain, a default value of $0\,\mathrm{m}$ is used. Partially missing values in a domain are not allowed.

palm_csd calculates the minimum terrain height of all domains, subtracts this value from all terrain heights and stores the result in the global `origin_z` attribute of the resulting static driver. The terrain height of a nested domain is shifted such that its mean is equal to the mean of the parent's terrain height in the nest's area to avoid artificial holes in the ground due to discretization. Furthermore, a gradual overlay of the terrain height is applied to avoid sharp gradients at the nest's boundaries.

2.7 Local Climate Zone based input for non-building-resolving simulation

So far, we focused on a detailed representation of the surface properties in the static driver. For coarser, non-building-resolving resolutions, palm_csd can also be used with less detailed Local Climate Zone (LCZ) input data. The LCZ classification (Stewart and Oke, 2012) consists of 17 classes, with classifications available for many urban areas in the World Urban Database and Access Portal Tool (Ching et al., 2018). In addition to standard PALM fields, palm_csd can also generate fields for the urban parametrization scheme DCEP (Schubert et al., 2012), which calculates the urban radiation and energy fluxes. Buildings are represented by infinitely long street canyons characterized by their building width B, street width W and their building height distribution with the average H. Urban impervious surfaces and vegetation are treated as separate tiles with grid cell fractions f_{urb} and $1 - f_{\mathrm{urb}}$, respectively.

Instead of the full data set required otherwise, only the input of an LCZ map and an orography map is needed here. Note that DCEP is currently under revision. The LCZ can be provided with either one layer with values from 1 to 17 for the 17 LCZ classes or a three-layer file with the red/green/blue (RGB) values of each LCZ class.

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Table 3. Assigned default parameters of each Local Climate Zone.

class	vegetation_type	water_type	lai_summer	lai_winter
compact_highrise	interrupted_forest	None	1.0	0.1
compact_midrise	interrupted_forest	None	1.0	0.1
compact_lowrise	interrupted_forest	None	1.0	0.1
open_highrise	interrupted_forest	None	2.0	0.5
open_midrise	interrupted_forest	None	2.0	0.5
open_lowrise	interrupted_forest	None	2.0	0.5
lightweight_lowrise	interrupted_forest	None	1.0	0.1
large_lowrise	interrupted_forest	None	0.5	0.1
sparsely_built	interrupted_forest	None	2.0	0.5
heavy_industry	interrupted_forest	None	0.5	0.0
dense_trees	deciduous_broadleaf_trees	None	4.0	0.8
scattered_trees	interrupted_forest	None	2.0	0.5
bush_scrub	deciduous_shrubs	None	1.0	0.1
low_plants	deciduous_shrubs	None	1.0	0.1
bare_rock_or_paved	bare_soil	None	0.0	0.0
bare_soil_or_sand	bare_soil	None	0.0	0.0
water	None	lake	None	None

2.7.1 PALM parameters

For each LCZ class, several PALM properties are assigned (Table 3): a vegetation type, a water type and a leaf area index for winter and summer. These values can also be adjusted by the user. The vegetation type `interrupted_forest` is assigned to urban LCZ classes under the assumption of low and high vegetation in these areas. The only additionally required data input is the orography.

2.7.2 Derivation of DCEP parameters

According to the definition of the LCZ classification, valid minimum and maximum values of the following parameters are assigned to each LCZ class (Table 4): mean building-height-to-street-width ratio (aspect ratio) λ_S, building surface fraction λ_B, impervious (without buildings) and pervious fraction λ_I and λ_V, and average roughness element height H. For LCZ 1 and LCZ 4, a maximum building height was not defined. We follow Demuzere et al. (2022a) and set these values to 75 m. For the derivation of the required PALM input values, one value within the defined valid range of each parameter is used. This value can be set by the user. The default values are taken from the W2W default values[2].

[2]https://github.com/matthiasdemuzere/w2w/blob/main/w2w/resources/LCZ_UCP_lookup.csv

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Table 4. Minimum (min), default (def) and maximum (max) value of LCZ parameters: mean building-height-to-street-width ratio (aspect ratio) λ_S, building surface fraction λ_B, impervious (without buildings) and pervious fraction λ_I and λ_V, and average roughness element height H. The latter represents the average building height in the case of urban LCZs. Their default values (avg) is the arithmetic or the geometric mean value of the minimum and maximum value, respectively.

class	λ_S			λ_B			λ_I			λ_V			H		
	min	def	max	min	def	max	min	def	max	min	def	max	min	def	max
compact_highrise	2.00	2.50	None	0.40	0.50	0.60	0.40	0.45	0.60	0.00	0.05	0.10	25.0	avg	75.00
compact_midrise	0.75	1.25	2.00	0.40	0.55	0.70	0.30	0.40	0.50	0.00	0.05	0.20	10.0	avg	25.00
compact_lowrise	0.75	1.25	1.50	0.40	0.55	0.70	0.20	0.35	0.50	0.00	0.10	0.30	3.0	avg	10.00
open_highrise	0.75	1.00	1.25	0.20	0.30	0.40	0.30	0.35	0.40	0.30	0.35	0.40	25.0	avg	75.00
open_midrise	0.30	0.50	0.75	0.20	0.30	0.40	0.30	0.40	0.50	0.20	0.30	0.40	10.0	avg	25.00
open_lowrise	0.30	0.50	0.75	0.20	0.30	0.40	0.20	0.35	0.50	0.30	0.35	0.60	3.0	avg	10.00
lightweight_lowrise	1.00	1.50	2.00	0.60	0.75	0.90	0.00	0.10	0.20	0.00	0.15	0.30	2.0	avg	4.00
large_lowrise	0.10	0.20	0.30	0.30	0.40	0.50	0.40	0.45	0.50	0.00	0.15	0.20	3.0	avg	10.00
sparsely_built	0.10	0.15	0.25	0.10	0.15	0.20	0.00	0.10	0.20	0.60	0.75	0.80	3.0	avg	10.00
heavy_industry	0.20	0.35	0.50	0.20	0.25	0.30	0.20	0.30	0.40	0.40	0.45	0.50	5.0	avg	15.00
dense_trees	1.00	2.00	None	0.00	0.00	0.10	0.00	0.00	0.10	0.90	1.00	1.00	3.0	avg	30.00
scattered_trees	0.50	0.65	0.80	0.00	0.00	0.10	0.00	0.00	0.10	0.90	1.00	1.00	3.0	avg	15.00
bush_scrub	0.70	0.80	0.90	0.00	0.00	0.10	0.00	0.00	0.10	0.90	1.00	1.00	0.0	1.000	2.00
low_plants	0.90	1.00	None	0.00	0.00	0.10	0.00	0.00	0.10	0.90	1.00	1.00	0.0	0.500	1.00
bare_rock_or_paved	0.90	1.00	None	0.00	0.05	0.10	0.90	0.90	1.00	0.00	0.05	0.10	0.0	0.125	0.25
bare_soil_or_sand	0.90	1.00	None	0.00	0.00	0.10	0.00	0.00	0.10	0.90	1.00	1.00	0.0	0.125	0.25
water	0.90	1.00	None	0.00	0.00	0.10	0.00	0.00	0.10	0.90	1.00	1.00	0.0	0.000	0.00

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

The input parameters of DCEP are derived from the LCZ parameters as follows: The urban fraction f_{urb} of a grid cell is considered to be the total impervious fraction of a grid cell $f_{\text{urb}} = \lambda_B + \lambda_I$. The street width W is calculated from the average building height and the aspect ratio with $W = H/\lambda_S$. The building width B is given by $B = \lambda_B/\lambda_I \cdot W$.

We follow the approach of Demuzere et al. (2022a) in the calculation of the distribution of building height: With the probability density function f of a normal distribution with the mean value H and a standard deviation $(H_{\text{max}} - H_{\text{min}})/4$, the fraction p of buildings at a height h is given by

$$p(h) = \int_{h-\Delta H/2}^{h+\Delta H/2} f(x)\,dx\,, \tag{6}$$

with ΔH being the layer thickness. Numerically, the integral is calculated directly using the cumulative distribution function of the given normal distribution from SciPy (Virtanen et al., 2020). In contrast to Demuzere et al. (2022a), the user can choose in (6) between the arithmetic and the geometric mean of building heights. In the original tool, only the arithmetic average is used, while the LCZ definition is based on the geometric mean.

2.8 Coupling and other tools

The palm_csd package comes with the `static_driver_stats` tool that calculates statistics and can visualize any static driver (results, for example, in Fig. 6).

Furthermore, although palm_csd is a comprehensive stand-alone tool, its flexibility also enables seamless coupling with other applications. For example, the LCZ-wizard (Anders and Maronga, 2025) facilitates the customization of artificial urban environments by combining built and natural LCZs. It uses small LCZ-units ($100 \times 100\,\text{m}^2$), which are aggregated into a coherent urban morphology. While the LCZ-wizard can be employed solely to generate building- and vegetation-resolving geodata, its interface with palm_csd extends this functionality by producing consistent static drivers that can be directly applied in PALM microscale simulations.

3 Application example: Berlin, Germany

3.1 Building-resolving set-up

This section shows the basic steps for processing data for the city of Berlin (Germany) with palm_csd. The input data are freely available from the Geoportal[3] and the Umweltatlas[4]. While most of the processing steps are done by palm_csd, some preprocessing is required using GIS tools of the user's choice, for example, with the open-source tool QGIS® (Graser et al., 2025). Its routines are mentioned in the following.

The building height data available for Berlin consist of vector polygons representing the building footprints, along with their attributes, such as building height (Fig. 2a). In this data set, few building heights are missing, which can be extracted

[3]https://www.berlin.de/sen/sbw/stadtdaten/geoportal/
[4]https://www.berlin.de/umweltatlas/

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

(a) Buildings and their attributes from 2023. The colour is based on the building height attribute column `hoehe_mod`.

(b) ALKIS data and their attributes from 2025.

Figure 2. Vector polygons and their attributes for the area of Tiergarten and its surrounding in Berlin (Germany) in UTM Zone 33N projection with north at the top.

from a normalized digital surface model (nDSM) raster. An nDSM represents the height of objects above ground level and can be calculated from a non-normalized DSM raster (object height above sea level, Fig. 3a) subtracted by a digital terrain model (DTM) raster (terrain height above sea level, Fig. 3b). Averages over each building polygon yield the required building heights. In QGIS, both steps are done with the *Raster calculator* and the *Zonal statistics* tool, respectively. The building ID is automatically assigned by palm_csd with a different value for each building polygon.

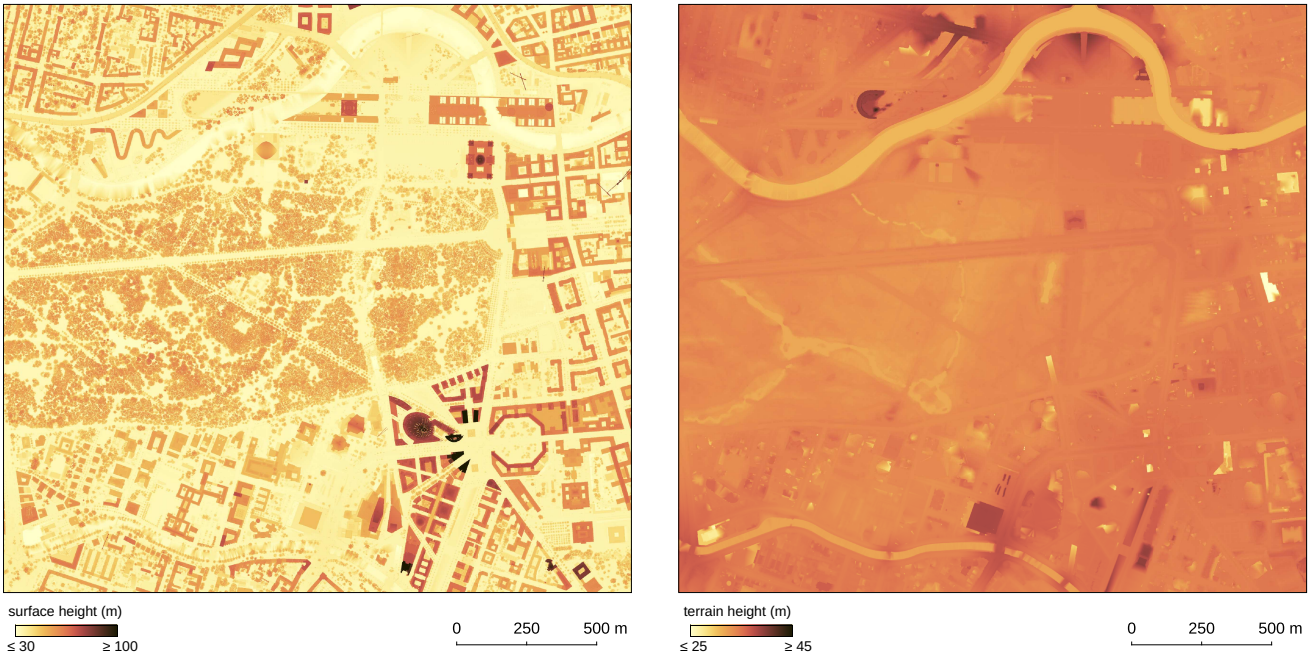
Similarly, other building parameters can be assigned, such as the albedo type, heat conductivity, heat capacity and the surface fractions on a building polygon basis or for the whole domain. For example, in palm_csd, mapping `hcon_wa` to `building_heat_conductivity_wall` assigns the heat conductivity to all layers of all wall elements and mapping `hcap_wag_1` to `building_heat_capacity_wall_gfl_1` assigns the heat capacity of the outermost wall layer of the ground floor. Analogously, by mapping the column `bfrac_gr_r` to `building_fraction_green_roof` and `bfrac_wa_r` to `building_fraction_wall_roof`, the green roofs are defined with their corresponding green and wall fractions. The fraction of windows is automatically set to 0 and, in general, the corresponding wall, green and windows fractions are normalized to 1.

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Table 5. Proposed mapping of the ALKIS classes to PALM pavement, vegetation and water types when applied to Berlin (Germany).

ALKIS class	land use	mapped PALM class
AX_Bahnverkehr	rail transport	`bare_soil`
AX_FlaecheBesondererFunktionalerPraegung	special functional area	`asphalt_concrete_mix`
AX_FlaecheGemischterNutzung	mixed-use area	`bare_soil`
AX_Fliessgewaesser	watercourse	`river`
AX_Flugverkehr	air transport	`asphalt_concrete_mix`
AX_Friedhof	cemetery	`short_grass`
AX_Gehoelz	shrubs and bushes	`evergreen_shrubs`
AX_Hafenbecken	harbour basin	`river`
AX_Halde	spoil heap	`bare_soil`
AX_Heide	heathland	`short_grass`
AX_IndustrieUndGewerbeflaeche	industrial and commercial area	`asphalt_concrete_mix`
AX_Landwirtschaft	agricultural land	`crops_mixed_farming`
AX_Moor	bog	`bogs_marsches`
AX_Platz	square	`concrete`
AX_Schiffsverkehr	shipping traffic	`river`
AX_SportFreizeitUndErholungsflaeche	sports, leisure and recreation area	`bare_soil`
AX_StehendesGewaesser	standing water body	`lake`
AX_Strassenverkehr	road transport	`asphalt_concrete_mix`
AX_Sumpf	swamp	`bogs_marsches`
AX_TagebauGrubeSteinbruch	open-pit mine	`bare_soil`
AX_UnlandVegetationsloseFlaeche	barren land / vegetation-free area	`bare_soil`
AX_Wald	forest	`mixed_forest_woodland`
AX_Weg	path	`concrete`
AX_Wohnbauflaeche	residential area	`asphalt_concrete_mix`

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026



(a) Digital Surface Model from 2021.

(b) Digital Terrain Model from 2021.

Figure 3. Height rasters for the area of Tiergarten and its surrounding in Berlin (Germany) in UTM Zone 33N projection with north at the top.

In order to derive a vegetation, pavement and water type, the land-use data set from ALKIS® (Amtliche Liegenschaftskatasterinformationssystem) is employed as vector polygons (Fig. 2b). Each land-use class from ALKIS is directly mapped to a corresponding PALM type as proposed in Table 5 for the area of interest. Water surface temperatures are manually added as a column to the ALKIS data in QGIS. Alternatively, water temperatures can be set on a per-type, domain-wide basis in palm_csd's configuration file.

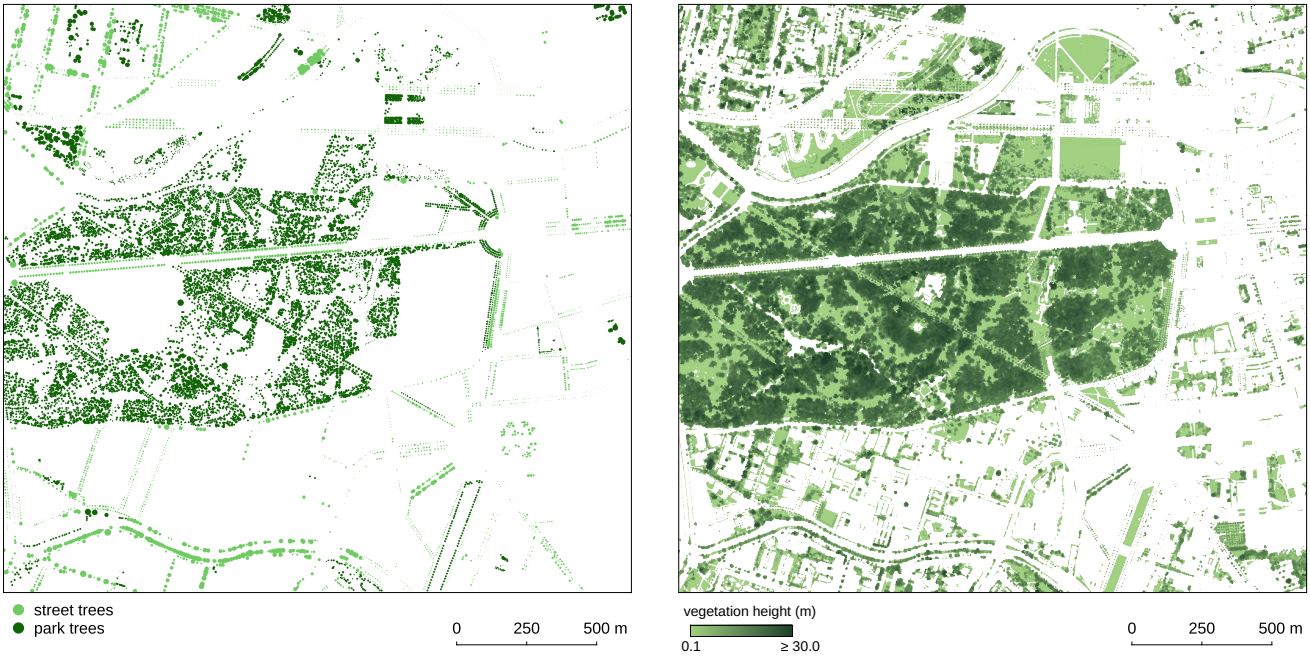
Two single tree data sets, street trees and park trees, are available as vector points for Berlin, which include the crown diameter, tree height, trunk circumference and tree type name (Fig. 4a). These data sets can be directly used by palm_csd, only the trunk diameter needs to be calculated from the trunk circumference, e.g. with QGIS' *Field calculator*. In order to include vegetation not captured by both tree data sets, the vegetation height raster available for Berlin can be employed (Fig. 4b). With this, vegetation patches are identified where this raster is above a certain height threshold.

Alternatively, for domains without precalculated vegetation height, it can be derived from an nDSM. In order to identify vegetation in the nDSM, the normalized difference vegetation index (NDVI) can be used, which is given by:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{Red}}{\mathrm{NIR} + \mathrm{Red}} \tag{7}$$

with Red and NIR standing for the spectral reflectance measurements acquired in the red (visible) and near-infrared regions, respectively (Fig. 5). Pixels with an NDVI value greater than a threshold (e.g. 0.22) are classified as vegetation; everything

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026



(a) Street trees (light green) and park trees (dark green) vector points from 2025.

(b) Vegetation height raster mostly based on data from 2020.

Figure 4. Single trees and vegetation height for the area of Tiergarten and its surrounding in Berlin (Germany) in UTM Zone 33N projection with north at the top.

below is classified as non-vegetation. The required reflectances are often available from orthophotos that include the NIR in addition to the Red, Green and Blue values.

The terrain height is directly derived from the DTM (Fig. 3b).

For the purpose of this example, we generate two static drivers (Fig. 6) by defining two domain sections in the configuration file: a root domain and a higher-resolution nest domain. Since all input data are fully georeferenced, the same input data section can be used for both domains.

3.2 Non-building-resolving set-up based on LCZ input

For non-building-resolving, coarser simulations, the data demand for a basic static driver is much lower. Here, we use the LCZ by Demuzere et al. (2022b, 2023) and a DTM from the Geoportal of the state of Brandenburg[5].

With the input in Fig. 7, the static driver can be generated directly after defining a target domain and geographical projection. Fig. 8 shows the generated urban fraction in each grid cell from this static driver, with higher values in the city centre when using the defaults in Table 4.

[5]https://geoportal.brandenburg.de/

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

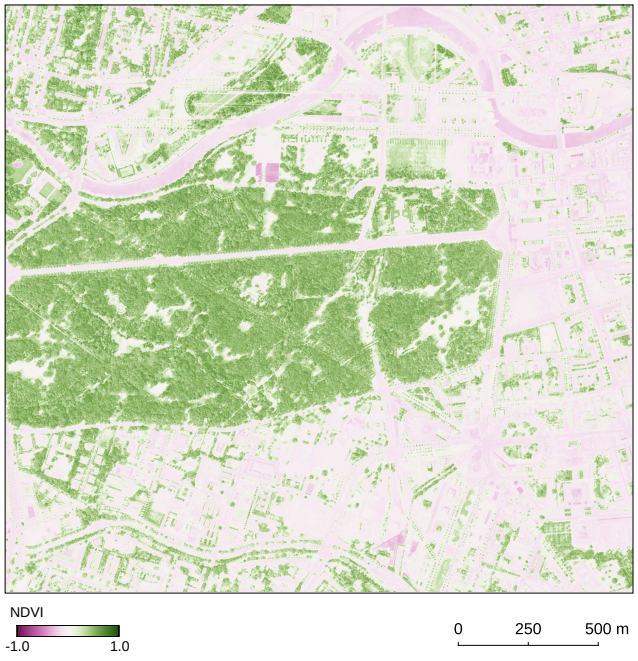
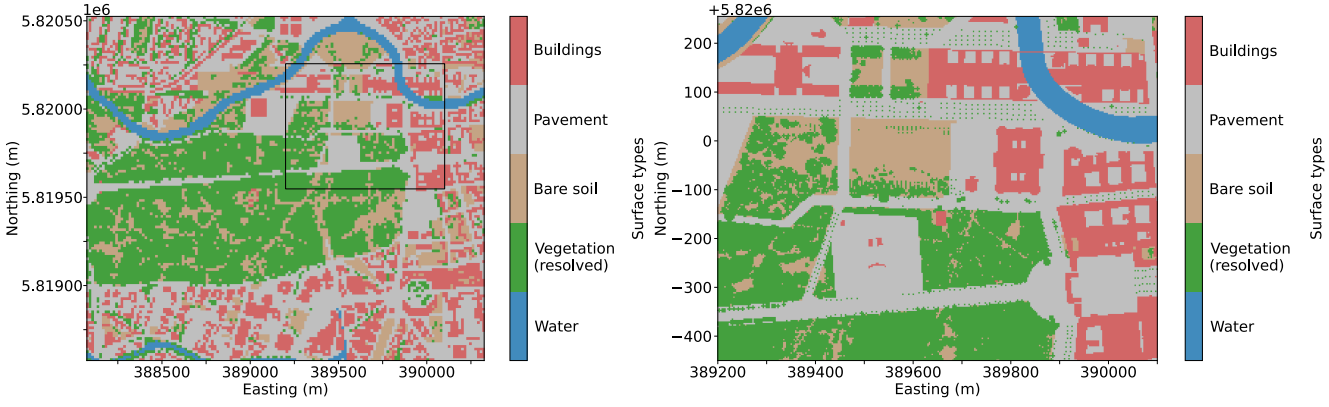


Figure 5. NDVI raster based on orthophotos from August 2020 for the area of Tiergarten and its surrounding in Berlin (Germany) in UTM Zone 33N projection with north at the top.

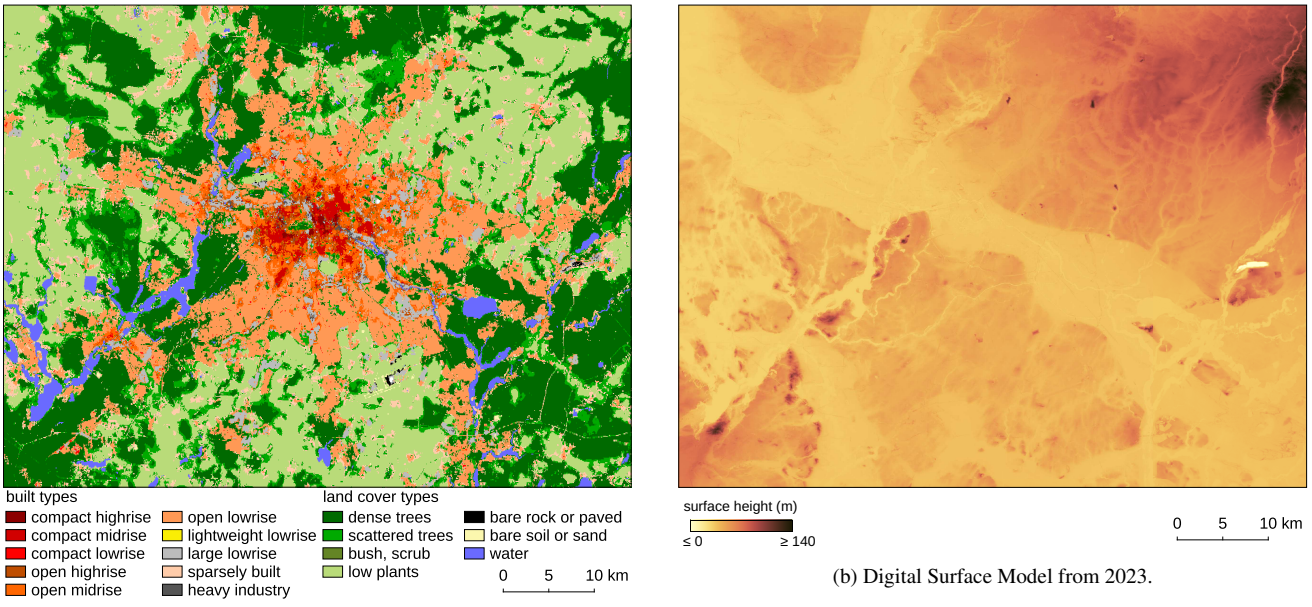


(a) Root domain.

(b) Nest domain.

Figure 6. Visualization of the generated static drivers in UTM Zone 33N projection with north at the top. The black rectangle in (a) indicates the position of the domain in (b).

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026



(a) LCZ data from 2022.

(b) Digital Surface Model from 2023.

Figure 7. Input data for the non-building-resolving static driver generation in UTM Zone 33N projection with north at the top.

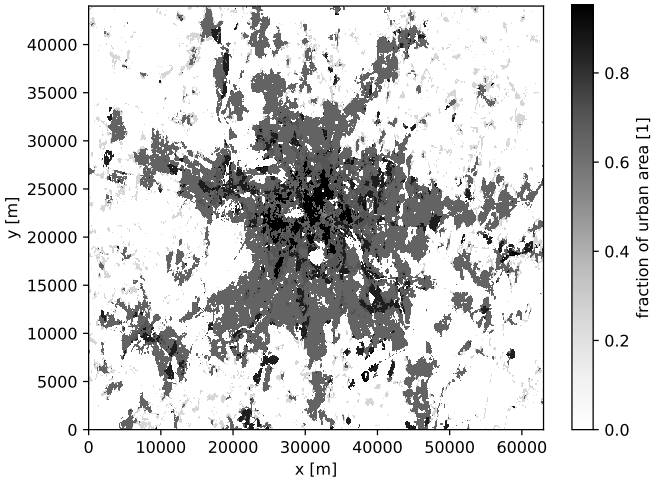


Figure 8. Fraction of the urban surfaces in a grid cell from the generated static driver.

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

4 Discussion

palm_csd is a comprehensive open-source tool for creating the static driver for PALM, with well defined input data. It is a Python-based command line program without a graphical user interface, intended to be used in conjunction with any GIS application such as QGIS (Graser et al., 2025). It offers detailed handling of building parameters and vegetation. While palm_csd's standard settings work for most situations, palm_csd can be adjusted for different circumstances, including idealized data input. It supports arbitrary rotation angles and domain nesting. palm_csd is well documented, actively maintained, closely following the development of PALM and is easy to install using only widely-used Python packages.

Depending on the availability, consistency and accuracy of the available urban data, the preprocessing steps for palm_csd can be labour-intensive. There are no consistent, world-wide available, high-resolution urban morphology and land cover data sets. Thus, static driver processing tools offering general data access (e.g. Lin et al., 2024) can only use data that are too coarse for fine building-resolving simulations. For national states (e.g. Germany) or larger administrative units (e.g. the European Union), suitable and complete data sets are usually also not available, for example, because of incomplete or coarse building height data (Bernard et al., 2022; Szatmári et al., 2022). However, initiatives like the Infrastructure for Spatial Information in the European Community (INSPIRE) of the European Union (European Union, 2024) aim at the consolidation and public availability of geodata. Still, an international standard for geodata is lacking. Thus, currently, data preparation beyond political borders might require intensive data preparation, even within single national states (e.g. across the federal borders of Germany), which is often time-consuming and requires expert knowledge (Masson et al., 2020).

Out of the required input data of palm_csd, vegetation data, in particular the LAI, are usually the most difficult to acquire on a fine scale. For example, satellite derived LAI, as calculated by the SENTINEL2 Toolbox (Weiss et al., 2020), has a resolution that is equal to or coarser than $10\,\mathrm{m}$. When comparing to in situ measurements, it shows overall good performance but with an underestimation over heterogeneous canopies (Brown et al., 2021), which are likely to occur in urban areas. Variations in quality can result in varying quality of the static driver that might even be visible in the PALM simulation results. Detailed local measurements using e.g. airborne laser scanning or terrestrial laser scanning (D'hont et al., 2024) offer the possibility to directly derive the LAD and BAD (Li et al., 2017; Kamoske et al., 2019) but are very resource-intensive, both in terms of scanning and computation.

Uncertainties in the input data can have considerable effects on the PALM results (cf. Belda et al., 2021; Resler et al., 2021). In addition, data are seldom up-to-date, especially in cities. Building data and land-cover data often represent different points in time and might not fit together perfectly. For small focus areas, this always requires careful data preparation.

Currently, as of version 25.10, palm_csd creates building data from 2D input data only. For future versions, we plan to add support for the input of 3D building data, e.g. in the CityGML format (Open Geospatial Consortium, 2023). In particular, support for the newly developed cut-cell method will be implemented once it is fully evaluated and documented. Currently, only PALM-GEM can generate the required input parameters but also requires the set-up of a PostgreSQL database on the system. We also plan to extend the tree database with species-dependent allometric functions to more realistically represent

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

trees of different ages. Furthermore, we will add support for generating fields for the single-layer urban canopy parametrization scheme PALM-SLUrb (Karttunen et al., 2025).

5 Conclusions

palm_csd allows the user of PALM working with realistic set-ups to create the complex input data required to represent the study domain. Although data preparation is still mandatory, palm_csd guides the user in collecting data and opens a wide range of options to include realism in the domain set-up. palm_csd answers the question of how to get surface-coverage data into the PALM simulation and simultaneously leaves the user enough room to construct a detailed built-up and vegetated study domain using already existing and well-established GIS tools.

As palm_csd is shipped with PALM, it always generates input data compatible with the latest PALM release. In order to ensure that, palm_csd is continuously developed further and automatically tested for each release.

Code availability. The current version of palm_csd is available from the PALM model system project website at https://gitlab.palm-model.org under the GNU General Public License v3.0. The exact version of the model used to produce the results presented in this paper (version 25.10) is archived on Zenodo under https://doi.org/10.5281/zenodo.18267017. The archived repository contains the source code corresponding to this release (Schubert, 2026).

Author contributions. SS designed and wrote most of palm_csd after the version discussed in Heldens et al. (2020); JA, TG, MS and BM contributed to the development; BM designed and wrote the original version of palm_csd; SS, MS, JA and TG wrote the manuscript draft; all authors reviewed and edited the manuscript.

Competing interests. One author is a member of the editorial board of journal GMD.

Acknowledgements. Sebastian Schubert was partly and Mohamed Salim fully funded by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) as part of the research project "UrbanGreenEye" (Grant No. 50EW2201C). Julian Anders and Björn Maronga were supported by the German Research Foundation (DFG) under grant MA 6383/5-1 (project number: 515096414).

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

References

Anders, J. and Maronga, B.: Urban Microscale Simulations Based on a Local Climate Zone Wizard: Concept and Validation Using the PALM Model System, Urban Climate, 63, 102 576, https://doi.org/10.1016/j.uclim.2025.102576, 2025.

Anders, J., Schubert, S., Sauter, T., Tunn, S., Schneider, C., and Salim, M.: Modelling the impact of an urban development project on microclimate and outdoor thermal comfort in a mid-latitude city, Energy and Buildings, 296, 113 324, https://doi.org/10.1016/j.enbuild.2023.113324, 2023.

Anders, J., Schubert, S., Maronga, B., and Salim, M.: Simplifying heat stress assessment: Evaluating meteorological variables as single indicators of outdoor thermal comfort in urban environments, Building and Environment, 274, 112 658, https://doi.org/10.1016/j.buildenv.2025.112658, 2025.

Belda, M., Resler, J., Geletič, J., Krč, P., Maronga, B., Sühring, M., Kurppa, M., Kanani-Sühring, F., Fuka, V., Eben, K., Benešová, N., and Auvinen, M.: Sensitivity Analysis of the PALM Model System 6.0 in the Urban Environment, Geoscientific Model Development, 14, 4443–4464, https://doi.org/10.5194/gmd-14-4443-2021, 2021.

Bernard, J., Bocher, E., Le Saux Wiederhold, E., Leconte, F., and Masson, V.: Estimation of Missing Building Height in OpenStreetMap Data: A French Case Study Using GeoClimate 0.0.1, Geoscientific Model Development, 15, 7505–7532, https://doi.org/10.5194/gmd-15-7505-2022, 2022.

Brown, L. A., Fernandes, R., Djamai, N., Meier, C., Gobron, N., Morris, H., Canisius, F., Bai, G., Lerebourg, C., Lanconelli, C., Clerici, M., and Dash, J.: Validation of Baseline and Modified Sentinel-2 Level 2 Prototype Processor Leaf Area Index Retrievals over the United States, ISPRS Journal of Photogrammetry and Remote Sensing, 175, 71–87, https://doi.org/10.1016/j.isprsjprs.2021.02.020, 2021.

Bruse, M. and Fleer, H.: Simulating surface–plant–air interactions inside urban environments with a three dimensional numerical model, Environmental Modelling &; Software, 13, 373–384, https://doi.org/10.1016/s1364-8152(98)00042-5, 1998.

Bureš, M. and Resler, J.: PALM-GeM: Geospatial Data Merging and Preprocessing into PALM, https://doi.org/10.5281/ZENODO.11067859, 2024.

Ching, J., Mills, G., Bechtel, B., See, L., Feddema, J., Wang, X., Ren, C., Brousse, O., Martilli, A., Neophytou, M., Mouzourides, P., Stewart, I., Hanna, A., Ng, E., Foley, M., Alexander, P., Aliaga, D., Niyogi, D., Shreevastava, A., Bhalachandran, P., Masson, V., Hidalgo, J., Fung, J., Andrade, M., Baklanov, A., Dai, W., Milcinski, G., Demuzere, M., Brunsell, N., Pesaresi, M., Miao, S., Mu, Q., Chen, F., and Theeuwes, N.: WUDAPT: An Urban Weather, Climate, and Environmental Modeling Infrastructure for the Anthropocene, Bulletin of the American Meteorological Society, 99, 1907–1924, https://doi.org/10.1175/BAMS-D-16-0236.1, 2018.

Demuzere, M., Argüeso, D., Zonato, A., and Kittner, J.: W2W: A Python Package That Injects WUDAPT's Local Climate Zone Information in WRF, Journal of Open Source Software, 7, 4432, https://doi.org/10.21105/joss.04432, 2022a.

Demuzere, M., Kittner, J., Martilli, A., Mills, G., Moede, C., Stewart, I. D., van Vliet, J., and Bechtel, B.: A Global Map of Local Climate Zones to Support Earth System Modelling and Urban-Scale Environmental Science, Earth System Science Data, 14, 3835–3873, https://doi.org/10.5194/essd-14-3835-2022, 2022b.

Demuzere, M., Kittner, J., Martilli, A., Mills, G., Moede, C., Stewart, I. D., van Vliet, J., and Bechtel, B.: Global Map of Local Climate Zones, https://doi.org/10.5281/zenodo.8419340, 2023.

D'hont, B., Calders, K., Bartholomeus, H., Lau, A., Terryn, L., Verhelst, T. E., and Verbeeck, H.: Evaluating Airborne, Mobile and Terrestrial Laser Scanning for Urban Tree Inventories: A Case Study in Ghent, Belgium, Urban Forestry & Urban Greening, 99, 128 428, https://doi.org/10.1016/j.ufug.2024.128428, 2024.

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Eichhorn, J. and Kniffka, A.: The numerical flow model MISKAM: State of development and evaluation of the basic version, Meteorologische Zeitschrift, 19, 81–90, https://doi.org/10.1127/0941-2948/2010/0425, 2010.

European Union: Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 Establishing an Infrastructure for Spatial Information in the European Community (INSPIRE), http://data.europa.eu/eli/dir/2007/2/oj, 2024.

Fluck, S.: palmpy code repository, Github [code], https://github.com/stefanfluck/palmpy, last accessed: 13 October 2025, 2023.

Gehrke, K. F., Sühring, M., and Maronga, B.: Modeling of land–surface interactions in the PALM model system 6.0: land surface model description, first evaluation, and sensitivity to model parameters, Geoscientific Model Development, 14, 5307–5329, https://doi.org/10.5194/gmd-14-5307-2021, 2021.

Graser, A., Sutton, T., and Bernasocchi, M.: The QGIS project: Spatial without compromise, Patterns, 6, 101 265, https://doi.org/10.1016/j.patter.2025.101265, 2025.

Heldens, W., Burmeister, C., Kanani-Sühring, F., Maronga, B., Pavlik, D., Sühring, M., Zeidler, J., and Esch, T.: Geospatial input data for the PALM model system 6.0: Model requirements, data sources, and processing, Geoscientific Model Development, 2020, 1–62, https://doi.org/10.5194/gmd-13-5833-2020, 2020.

Kamoske, A. G., Dahlin, K. M., Stark, S. C., and Serbin, S. P.: Leaf Area Density from Airborne LiDAR: Comparing Sensors and Resolutions in a Temperate Broadleaf Forest Ecosystem, Forest Ecology and Management, 433, 364–375, https://doi.org/10.1016/j.foreco.2018.11.017, 2019.

Karttunen, S., Sühring, M., O'Connor, E., and Järvi, L.: PALM-SLUrb V24.04: A Single-Layer Urban Canopy Model for the PALM Model System – Model Description and First Evaluation, Geoscientific Model Development, 18, 5725–5757, https://doi.org/10.5194/gmd-18-5725-2025, 2025.

Khan, B., Banzhaf, S., Chan, E. C., Forkel, R., Kanani-Sühring, F., Ketelsen, K., Kurppa, M., Maronga, B., Mauder, M., Raasch, S., Russo, E., Schaap, M., and Sühring, M.: Development of an atmospheric chemistry model coupled to the PALM model system 6.0: implementation and first applications, Geoscientific Model Development, 14, 1171–1193, https://doi.org/10.5194/gmd-14-1171-2021, 2021.

Krč, P., Resler, J., Sühring, M., Schubert, S., Salim, M. H., and Fuka, V.: Radiative Transfer Model 3.0 integrated into the PALM model system 6.0, Geoscientific Model Development, 14, 3095–3120, https://doi.org/10.5194/gmd-14-3095-2021, 2021.

Lalic, B. and Mihailovic, D. T.: An Empirical Relation Describing Leaf-Area Density inside the Forest for Environmental Modeling, Journal of Applied Meteorology and Climatology, 43, 641–645, https://doi.org/10.1175/1520-0450(2004)043<0641:AERDLD>2.0.CO;2, 2004.

Li, S., Dai, L., Wang, H., Wang, Y., He, Z., and Lin, S.: Estimating Leaf Area Density of Individual Trees Using the Point Cloud Segmentation of Terrestrial LiDAR Data and a Voxel-Based Model, Remote Sensing, 9, 1202, https://doi.org/10.3390/rs9111202, 2017.

Lin, D., Zhang, J., Khan, B., Katurji, M., and Revell, L. E.: GEO4PALM v1.1: an open-source geospatial data processing toolkit for the PALM model system, Geoscientific Model Development, 17, 815–845, https://doi.org/10.5194/gmd-17-815-2024, 2024.

Markkanen, T., Rannik, Ü., Marcolla, B., Cescatti, A., and Vesala, T.: Footprints and Fetches for Fluxes over Forest Canopies with Varying Structure and Density, Boundary-Layer Meteorology, 106, 437–459, https://doi.org/10.1023/A:1021261606719, 2003.

Maronga, B., Gryschka, M., Heinze, R., Hoffmann, F., Kanani-Sühring, F., Keck, M., Ketelsen, K., Letzel, M. O., Sühring, M., and Raasch, S.: The Parallelized Large-Eddy Simulation Model (PALM) Version 4.0 for Atmospheric and Oceanic Flows: Model Formulation, Recent Developments, and Future Perspectives, Geoscientific Model Development, 8, 2515–2551, https://doi.org/10.5194/gmd-8-2515-2015, 2015.

Maronga, B., Gross, G., Raasch, S., Banzhaf, S., Forkel, R., Heldens, W., Kanani-Sühring, F., Matzarakis, A., Mauder, M., Pavlik, D., Pfafferott, J., Schubert, S., Seckmeyer, G., Sieker, H., and Winderlich, K.: Development of a new urban climate model

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

based on the model PALM – Project overview, planned work, and first achievements, Meteorologische Zeitschrift, 28, 105–119, https://doi.org/10.1127/metz/2019/0909, 2019.

Maronga, B., Banzhaf, S., Burmeister, C., Esch, T., Forkel, R., Fröhlich, D., Fuka, V., Gehrke, K. F., Geletič, J., Giersch, S., Gronemeier, T., Groß, G., Heldens, W., Hellsten, A., Hoffmann, F., Inagaki, A., Kadasch, E., Kanani-Sühring, F., Ketelsen, K., Khan, B. A., Knigge, C., Knoop, H., Krč, P., Kurppa, M., Maamari, H., Matzarakis, A., Mauder, M., Pallasch, M., Pavlik, D., Pfafferott, J., Resler, J., Rissmann, S., Russo, E., Salim, M., Schrempf, M., Schwenkel, J., Seckmeyer, G., Schubert, S., Sühring, M., von Tils, R., Vollmer, L., Ward, S., Witha, B., Wurps, H., Zeidler, J., and Raasch, S.: Overview of the PALM model system 6.0, Geoscientific Model Development, 13, 1335–1372, https://doi.org/10.5194/gmd-13-1335-2020, 2020.

Masson, V., Heldens, W., Bocher, E., Bonhomme, M., Bucher, B., Burmeister, C., de Munck, C., Esch, T., Hidalgo, J., Kanani-Sühring, F., Kwok, Y.-T., Lemonsu, A., Lévy, J.-P., Maronga, B., Pavlik, D., Petit, G., See, L., Schoetter, R., Tornay, N., Votsis, A., and Zeidler, J.: City-Descriptive Input Data for Urban Climate Models: Model Requirements, Data Sources and Challenges, Urban Climate, 31, 100 536, https://doi.org/10.1016/j.uclim.2019.100536, 2020.

Open Geospatial Consortium: OGC City Geography Markup Language (CityGML) Part 2: GML Encoding Standard, Version 3.0, http://www.opengis.net/doc/IS/CityGML-2/3.0, 2023.

Pfafferott, J., Rißmann, S., Sühring, M., Kanani-Sühring, F., and Maronga, B.: Building indoor model in PALM-4U: indoor climate, energy demand, and the interaction between buildings and the urban microclimate, Geoscientific Model Development, 14, 3511–3519, https://doi.org/10.5194/gmd-14-3511-2021, 2021.

Resler, J., Krč, P., Belda, M., Juruš, P., Benešová, N., Lopata, J., Vlček, O., Damašková, D., Eben, K., Derbek, P., Maronga, B., and Kanani-Sühring, F.: PALM-USM v1.0: A new urban surface model integrated into the PALM large-eddy simulation model, Geoscientific Model Development, 10, 3635–3659, https://doi.org/10.5194/gmd-10-3635-2017, 2017.

Resler, J., Eben, K., Geletič, J., Krč, P., Rosecký, M., Sühring, M., Belda, M., Fuka, V., Halenka, T., Huszár, P., Karlický, J., Benešová, N., Ďoubalová, J., Honzáková, K., Keder, J., Nápravníková, Š., and Vlček, O.: Validation of the PALM Model System 6.0 in a Real Urban Environment: A Case Study in Dejvice, Prague, the Czech Republic, Geoscientific Model Development, 14, 4797–4842, https://doi.org/10.5194/gmd-14-4797-2021, 2021.

Salim, M. H., Schlünzen, K. H., Grawe, D., Boettcher, M., Gierisch, A. M. U., and Fock, B. H.: The microscale obstacle-resolving meteorological model MITRAS v2.0: model theory, Geoscientific Model Development, 11, 3427–3445, https://doi.org/10.5194/gmd-11-3427-2018, 2018.

Salim, M. H., Schubert, S., Resler, J., Krč, P., Maronga, B., Kanani-Sühring, F., Sühring, M., and Schneider, C.: Importance of radiative transfer processes in urban climate models: a study based on the PALM 6.0 model system, Geoscientific Model Development, 15, 145–171, https://doi.org/10.5194/gmd-15-145-2022, 2022.

Schubert, S.: palm_csd 25.10, https://doi.org/10.5281/ZENODO.18267017, 2026.

Schubert, S., Grossman-Clarke, S., and Martilli, A.: A Double-Canyon Radiation Scheme for Multi-Layer Urban Canopy Models, Boundary-Layer Meteorology, 145, 439–468, https://doi.org/10.1007/s10546-012-9728-3, 2012.

Stewart, I. D. and Oke, T. R.: Local Climate Zones for Urban Temperature Studies, Bulletin of the American Meteorological Society, 93, 1879–1900, https://doi.org/10.1175/BAMS-D-11-00019.1, 2012.

Szatmári, D., Kopecká, M., and Feranec, J.: Accuracy Assessment of the Building Height Copernicus Data Layer: A Case Study of Bratislava, Slovakia, Land, 11, 590, https://doi.org/10.3390/land11040590, 2022.

https://doi.org/10.5194/egusphere-2026-355
Preprint. Discussion started: 4 March 2026

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods, 17, 261–272, https://doi.org/10.1038/s41592-019-0686-2, 2020.

Vogel, J., Stadler, S., Chockalingam, G., Afshari, A., Henning, J., and Winkler, M.: SanDyPALM v1. 0: static and dynamic drivers for the PALM model to facilitate urban microclimate simulations, Geoscientific Model Development, 18, 6063–6094, https://doi.org/10.5194/gmd-18-6063-2025, 2025.

Weiss, M., Baret, F., and Jay, S.: S2ToolBox Level 2 Products: LAI, FAPAR, FCOVER Version 2.0, Tech. rep., Institut national de recherche pour l'agriculture, l'alimentation et l'environnement, https://hal.inrae.fr/hal-03584016v1, 2020.

Winkler, M., Stadler, S., Radon, J., and Henning, J.: PALM-4U GUI: A Cloud Based User-Friendly Graphical User Interface for the Urban Climate Model PALM-4U, in: Building Simulation 2023, vol. 18 of *Building Simulation*, pp. 1232–1239, IBPSA, https://doi.org/10.26868/25222708.2023.1670, 2023.