SUEWS Documentation

Release v2020a

SUEWS dev team led by Prof Sue Grimmond

2023-03-05

FOR USERS

CHAPTER ONE

WHAT IS SUEWS?

Surface Urban Energy and Water Balance Scheme (**SUEWS**) [Järvi *et al.*, 2011, Ward *et al.*, 2016] is a neighbourhood/local-scale urban land surface model to simulate the urban radiation, energy and water balances using only commonly measured meteorological variables and information about the surface cover. SUEWS utilises an evaporation-interception approach [Grimmond and Oke, 1991], similar to that used in forests, to model evaporation from urban surfaces.

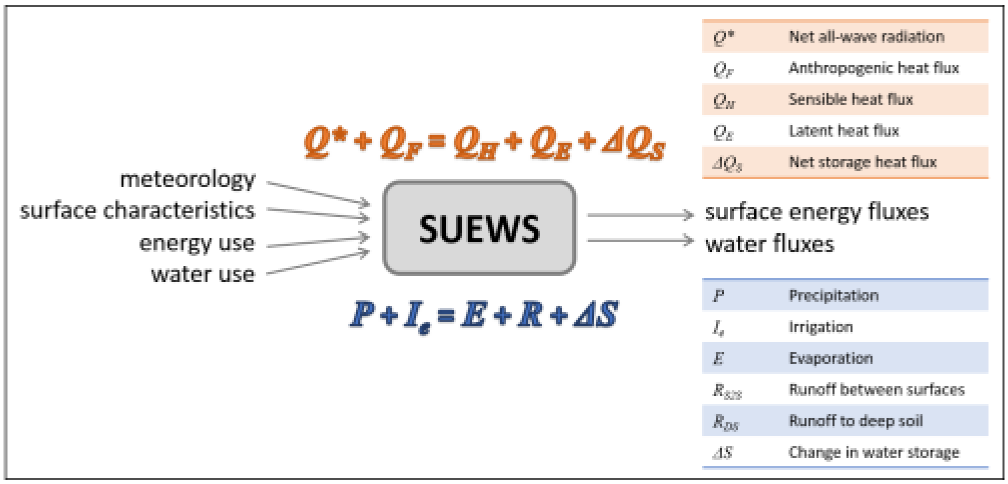


Fig. 1.1: Overview of SUEWS

The model uses seven surface types: paved, buildings, evergreen trees/shrubs, deciduous trees/shrubs, grass, bare soil and water. The surface state for each surface type at each time step is calculated from the running water balance of the canopy where the evaporation is calculated from the Penman-Monteith equation. The soil moisture below each surface type (excluding water) is taken into account. Horizontal movement of water above and below ground level is allowed.

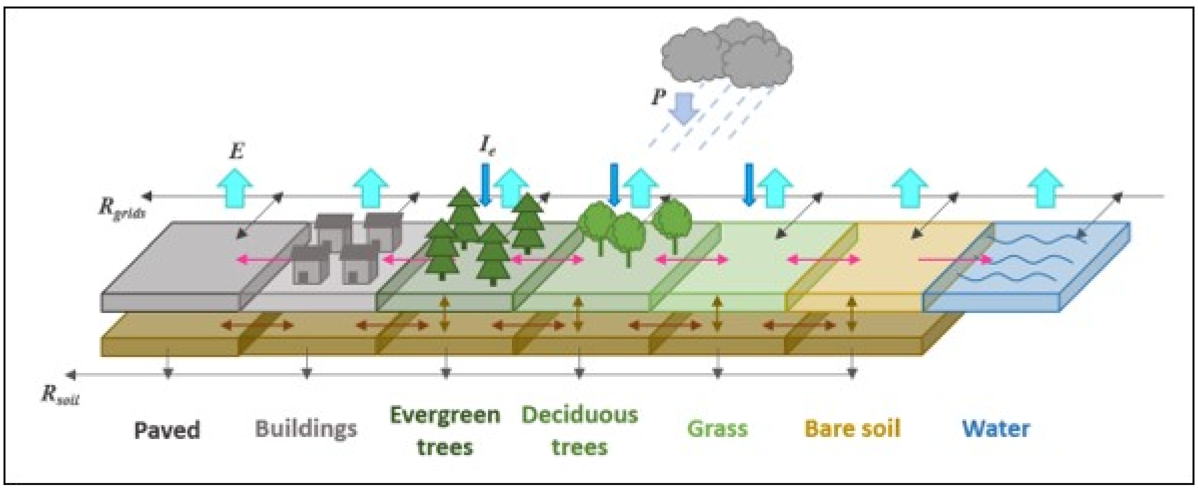


Fig. 1.2: The seven surface types considered in SUEWS

CHAPTER

TWO

HOW TO GET SUEWS?

Please follow the guidance in *Installation* to get SUEWS.

CHAPTER

THREE

HOW TO USE SUEWS?

- **For existing users:**

  Overview of changes in this version, see *Version 2020a (released on 14 May 2020)*. If these changes impact your existing simulations, please see appropriate parts of the manual. It may be necessary to adapt some of your input files for for the current version.

  Tip: A helper python script, *SUEWS table converter*, is provided to help facilitate the conversion of input files between different SUEWS versions.

  Additionally, the manuals for previous versions can be accessed in respective sections under *Version History*.

- **For new users:**

  Before performing SUEWS simulations, new users should read the overview *Introduction*, then follow the steps in *Workflow of using SUEWS* to prepare *input files* for SUEWS.

  Note there are tutorials learning about running SUEWS available *the tutorial*.

CHAPTER

FOUR

HOW TO GET HELP IN USING SUEWS?

Please let us know in the UMEP Community. The developers and other users are willing to help you.

CHAPTER

FIVE

HOW HAS SUEWS BEEN USED?

The scientific details and application examples of SUEWS can be found in *SUEWS-related Publications*.

CHAPTER SIX

HOW TO CITE SUEWS?

Please go to our Zenodo repository for a proper citation of SUEWS.

Tip: Visit the repositories below for different citation styles.

CHAPTER

SEVEN

HOW TO SUPPORT SUEWS?

1. *Cite SUEWS* appropriately in your work.
2. Contribute to the *development*.
3. Report issues via the GitHub page.
4. Provide *suggestions and feedback*.

7.1 Installation

7.1.1 Formal releases

Since 2023, SUEWS is available as a command line tool via its Python wrapper package SuPy (SUEWS in Python) on PyPI and conda-forge.

Note: The Fortran-based binaries build prior to 2023 are still available at the SUEWS download page. However, they are not maintained anymore so users are encouraged to use the Python-based packages instead.

Installing Python

These instructions will set you up with mamba, which makes it easy to install and manage Python packages.

To install the `mamba` Python distribution follow the mamba installation instructions.

This makes installing `supy` and many other packages in the scientific Python ecosystem much easier and quicker. It also provides many pre-compiled binaries that are not available on PyPI.

Tip: `mamba` is a drop-in replacement for `conda` (another widely used Python package manager): `mamba` is faster and solves some common problems with `conda`. More details about `mamba` can be found at mamba.

Installing SuPy

One can install `supy` using `pip`:

```
python3 -m pip install supy --upgrade
```

7.1.2 Development build

The development build can be highly unstable and is not recommended for production use. However, it is automatically constructed every week for testing purposes and we are happy to receive feedback on the development build.

To install the development build of SUEWS, you need to install `supy` in the development mode:

1. git clone the repository:

   ```
   git clone https://github.com/UMEP-dev/SUEWS.git
   ```

2. navigate to the directory of the cloned repository:

   ```
   cd SUEWS
   ```

3. install the package in the development mode:

   ```
   make dev
   ```

7.2 Workflow of using SUEWS

The following is to help with the model setup. Note that there are also starting tutorials for the version of SUEWS in UMEP. The version there is the same (i.e. the executable) as the standalone version so you can swap to that later once you have some familiarity.

7.2.1 Preparatory reading

Read the manual and relevant papers (and references therein):

- Järvi, L., Grimmond, C. S. B., Taka, M., Nordbo, A., Setälä, H., and Strachan, I. B. Development of the surface urban energy and water balance scheme (SUEWS) for cold climate cities. *Geosci. Model Dev.*, 7(4):1691–1711, August 2014. doi:10.5194/gmd-7-1691-2014.
- Järvi, L., Grimmond, C.S.B., and Christen, A. The surface urban energy and water balance scheme (SUEWS): Evaluation in Los Angeles and Vancouver. *J. Hydrol.*, 411(3-4):219–237, December 2011. doi:10.1016/j.jhydrol.2011.10.001.
- Ward, H.C., Kotthaus, S., Järvi, L., and Grimmond, C.S.B. Surface urban energy and water balance scheme (SUEWS): Development and evaluation at two UK sites. *Urban Clim.*, 18:1–32, December 2016. doi:10.1016/j.uclim.2016.05.001.

See other publications with example applications

7.2.2 Decide what type of model run you are interested in

	Available in this release
SUEWS at a point or for an individual area	Yes
SUEWS for multiple grids or areas	Yes
SUEWS with Boundary Layer (BL)	Yes
SUEWS with snow	Yes

7.2.3 Download the program and example data files

Visit the SUEWS download page to receive a link to download the program and example data files. Select the appropriate compiled version for your platform to download. There is also a python-based version in UMEP under the QGIS environment. For python users, `SuPy` - a python wrapper for SUEWS - is also available.

Note, as the definition of long double precision varies between computers (e.g. Mac vs Windows) slightly different results may occur in the output files.

Test/example files are shipped in the archive with the SUEWS executable, which are based on measurements of the London KCL site, 2011 data (denoted `Kc11`)

In the following, `SS` is the site code (e.g. `Kc`), `ss` the grid ID, `YYYY` the year and `tt` the time interval.

Filename	Description	Input/output
SSss_data.txt	Meteorological input	Input file (60-min)
SSss_YYYY_data_5.txt	Meteorological input	Input file (5-min)
InitialConditionsSSss	Initial conditions	Input - _YYYY.nml(+) file
*SUEWS_***.txt*	Property look-up tables	Input text files containing all other input information
RunControl.nml	Sets model run	Input (located in options main directory)
SS_Filechoices.txt	Summary of model run	Output options
SSss_YYYY_5.txt	(Optional) 5-min	Output resolution output file
SSss_YYYY_60.txt	60-min resolution	Output output file
SSss_DailyState.txt	Daily state variables	Output (all years in one file)

(+) There is a second file InitialConditionsSSss_YYYY_EndOfRun.nml or InitialConditionsSSss_YYYY+1.nml in the input directory. At the end of the run, and at the end of each year of the run, these files are written out so that this information could be used to initialize further model runs.

7.2.4 Run the model for example data

Before running the model with your own data, check that you get the same results as the test run example files provided. Copy the example output files elsewhere so you can compare the results. When you run the program it will write over the supplied files.

To run the model you can use **Command Prompt** (in the directory where the programme is located type the model name) or just double click the executable file.

Please see *Troubleshooting* if you have problems running the model.

7.2.5 Preparation of data

Tip: If you need help preparing the data you can use some of the UMEP tools.

The information required to run SUEWS for your site consists of:

- Continuous *meteorological forcing data* for the entire period to be modelled without gaps.
- Knowledge of the *surface and soil conditions immediately prior to the first model timestep*.

> **Note:** If these initial conditions are unknown, model spin-up can help; i.e. run the model and use the output at the end of the run to infer the conditions at the start of the main run). Spin-up is important for getting appropriate initial conditions for the model. An example of a spin-up can be found in Kokkonen *et al.* [2018].

- The *location of the site* (latitude, longitude, altitude).
- Information about the *characteristics of the surface*, including land cover, heights of buildings and trees, radiative characteristics (e.g. albedo, emissivity), drainage characteristics, soil characteristics, snow characteristics, phenological characteristics (e.g. seasonal cycle of LAI).

> **Note:** For guidance on how to derive parameters related to LAI, albedo, surface conductance and surface roughness, the reader is referred to this link.

- Information about *human behaviour*, including energy use and water use (e.g. for irrigation or street cleaning) and snow clearing (if applicable).

> **Note:** The anthropogenic energy use and water use may be provided as a time series in the meteorological forcing file (by setting `EmissionsMethod` = 0) if these data are available or modelled based on parameters provided to the model, including population density, hourly and weekly profiles of energy and water use, information about the proportion of properties using irrigation and the type of irrigation (automatic or manual).

It is particularly important to ensure the following input information is appropriate and representative of the site:

- Fractions of different land cover types and (less so) heights of buildings [Ward *et al.*, 2016]
- Accurate meteorological forcing data, particularly precipitation and incoming shortwave radiation [Kokkonen *et al.*, 2018]
- Initial soil moisture conditions [Best and Grimmond, 2014]
- Anthropogenic heat flux parameters, particularly if there are considerable energy emissions from transport, buildings, metabolism, etc [Ward *et al.*, 2016].
- External water use (if irrigation or street cleaning occurs)
- Snow clearing (if running the snow option)
- Surface conductance parameterisation [Järvi *et al.*, 2011, Ward *et al.*, 2016]

SUEWS can be run either for an individual area or for multiple areas. There is no requirement for the areas to be of any particular shape but here we refer to them as model 'grids'.

Preparation of site characteristics and model parameters

The area to be modelled is described by a set of characteristics that are specified in the *SUEWS_SiteSelect.txt* file. Each row corresponds to one model grid for one year (i.e. running a single grid over three years would require three rows; running two grids over two years would require four rows). Characteristics are often selected by a code for a particular set of conditions. For example, a specific soil type (links to *SUEWS_Soil.txt*) or characteristics of deciduous trees in a particular region (links to *SUEWS_Veg.txt*). The intent is to build a library of characteristics for different types of urban areas. The codes are specified by the user, must be integer values and must be unique within the first column of each input file, otherwise the model will return an error.

Note: The first column of *SUEWS_SiteSelect.txt* the is labelled 'Grid' and can contain repeat values for different years. See *Input files* for details. Note UMEP maybe helpful for components of this.

Land cover

For each grid, the land cover must be classified using the following surface types:

Classification	Surface type	File where characteristics are specified
Non-vegetated	Paved surfaces	*SUEWS_NonVeg.txt*
	Building	*SUEWS_NonVeg.txt*
	Bare soil	*SUEWS_NonVeg.txt*
Vegetation	Evergreen trees	*SUEWS_Veg.txt*
	Deciduous trees	*SUEWS_Veg.txt*
	Grass	*SUEWS_Veg.txt*
Water	Water	*SUEWS_Water.txt*
Snow	Snow	*SUEWS_Snow.txt*

The surface cover fractions (i.e. proportion of the grid taken up by each surface) must be specified in *SUEWS_SiteSelect.txt*. The surface cover fractions are **critical**, so make certain that the different surface cover fractions are appropriate for your site.

For some locations, land cover information may be already available (e.g. from various remote sensing resources). If not, websites like Bing Maps and Google Maps allow you to see aerial images of your site and can be used to estimate the relative proportion of each land cover type. If detailed spatial datasets are available, UMEP allows for a direct link to a GIS environment using QGIS.

Anthropogenic heat flux (Q_F)

You can either model Q_F within SUEWS or provide it as an input.

- To model it population density is needed as an input for LUMPS and SUEWS to calculate Q_F.
- If you have no information about the population of the site we recommend that you use the LUCY model [Allen *et al.*, 2010, Lindberg *et al.*, 2013] to estimate the anthropogenic heat flux which can then be provided as input SUEWS along with the meteorological forcing data.

Alternatively, you can use the updated version of LUCY called LQF, which is included in UMEP.

Other information

The surface cover fractions and population density can have a major impact on the model output. However, it is important to consider the suitability of all parameters for your site. Using inappropriate parameters may result in the model returning an error or, worse, generating output that is simply not representative of your site. Please read the section on *Input files*. Recommended or reasonable ranges of values are suggested for some parameters, along with important considerations for how to select appropriate values for your site.

Data Entry

To create the series of input text files describing the characteristics of your site, there are three options:

1. Data can be entered directly into the input text files. The example (.txt) files provide a template to create your own files which can be edited with *A text editor* directly.
2. Use UMEP.

Note that in all txt files:

- The first two rows are headers: the first row is the column number; the second row is the column name.
- The names and order of the columns should not be altered from the templates, as these are checked by the model and errors will be returned if particular columns cannot be found.
- Since v2017a it is no longer necessary for the meteorological forcing data to have two rows with -9 in column 1 as their last two rows.
- "!" indicates a comment, so any text following "!" on the same line will not be read by the model.
- If data are unavailable or not required, enter the value -999 in the correct place in the input file.
- Ensure the units are correct for all input information. See *Input files* for a description of parameters.

In addition to these text files, the following files are also needed to run the model.

Preparation of the RunControl file

In the *RunControl.nml* file the site name (SS) and directories for the model input and output are given. This means **before running** the model (even the with the example datasets) you must either

1. open the *RunControl.nml* file and edit the input and output file paths and the site name (with a *A text editor*) so that they are correct for your setup, or
2. create the directories specified in the RunControl.nml file

From the given site identification the model identifies the input files and generates the output files. For example if you specify:

```
FileOutputPath = “C:\FolderName\SUEWSOutput\”
```

and use site code SS the model creates an output file:

```
C:\FolderName\SUEWSOutput\SSss_YYYY_TT.txt
```

Note: The path separator differs between Windows (backslash: `\`) and Linux/Mac (slash, or forward slash: `/`).

If the file paths are not correct the program will return an error when run and write the error to the *Error messages: problems.txt* file.

Preparation of the Meteorological forcing data

The model time-step is specified in *RunControl.nml* (5 min is highly recommended). If meteorological forcing data are not available at this resolution, SUEWS has the option to downscale (e.g. hourly) data to the time-step required. See details about the *SSss_YYYY_data_tt.txt* to learn more about choices of data input. Each grid can have its own meteorological forcing file, or a single file can be used for all grids. The forcing data should be representative of the local-scale, i.e. collected (or derived) above the height of the roughness elements (buildings and trees).

Preparation of the InitialConditions file

Information about the surface state and meteorological conditions just before the start of the run are provided in the Initial Conditions file. At the very start of the run, each grid can have its own Initial Conditions file, or a single file can be used for all grids. For details see *Initial Conditions file*.

7.2.6 Run the model for your site

To run the model you can use **Command Prompt** (in the directory where the programme is located type the model name) or just double click the executable file.

Please see *Troubleshooting* if you have problems running the model.

7.2.7 Analyse the output

It is a good idea to perform initial checks that the model output looks reasonable.

Characteristic	Things to check
Leaf area index	**Does the phenology look appropriate?** • what does the seasonal cycle of leaf area index (LAI) look like? • Are the leaves on the trees at approximately the right time of the year?
Kdown	**Is the timing of diurnal cycles correct for the incoming solar radiation?** • Although Kdown is a required input, it is also included in the output file. It is a good idea to check that the timing of Kdown in the output file is appropriate, as problems can indicate errors with the timestamp, incorrect time settings or problems with the disaggregation. In particular, make sure the sign of the longitude is specified correctly in *SUEWS_SiteSelect.txt*. • Checking solar angles (zenith and azimuth) can also be a useful check that the timing is correct.
Albedo	**Is the bulk albedo correct?** • This is critical because a small error has an impact on all the fluxes (energy and hydrology). • If you have measurements of outgoing shortwave radiation compare these with the modelled values. • How do the values compare to literature values for your area?

7.2.8 Summary of files

The table below lists the files required to run SUEWS and the output files produced. SS is the two-letter code (specified in *RunControl.nml*) representing the site name, ss is the grid identification (integer values between 0 and 2,147,483,647 (largest 4-byte integer)) and YYYY is the year. TT is the resolution of the input/output file and tt is the model time-step.

The last column indicates whether the files are needed/produced once per run (1/run), or once per day (1/day), for each year (1/year) or for each grid (1/grid):

```
[B] indicates files used with the CBL part of SUEWS (BLUEWS) and therefore are only needed/
↪produced if this option is selected
[E] indicates files associated with ESTM storage heat flux models and therefore are only needed/
↪produced if this option is selected
```

7.2.9 Get in contact

For issues met in using SUEWS, we recommend the following ways to get in contact with the developers and the SUEWS community:

1. Report issues on our GitHub page.
2. Ask for help by joining the Email-list for SUEWS.

7.3 Input files

SUEWS allows you to input a large number of parameters to describe the characteristics of your site. You should not assume that the example values provided in files or in the tables below are appropriate. Values marked with 'MD' are examples of recommended values (see the suggested references to help decide how appropriate these are for your site/model domain); values marked with 'MU' need to be set (i.e. changed from the example) for your site/model domain.

7.3.1 RunControl.nml

The file **RunControl.nml** is a namelist that specifies the options for the model run. It must be located in the same directory as the executable file.

A sample file of **RunControl.nml** looks like

```
&RunControl
CBLUse=0
SnowUse=0
SOLWEIGUse=0
NetRadiationMethod=3
EmissionsMethod=2
StorageHeatMethod=3
OHMIncQF=0
StabilityMethod=2
RoughLenHeatMethod=2
RoughLenMomMethod=2
SMDMethod=0
WaterUseMethod=0
FileCode='Saeve'
```

(continues on next page)

(continued from previous page)

```
FileInputPath="./Input/"
FileOutputPath="./Output/"
MultipleMetFiles=0
MultipleInitFiles=0
MultipleESTMFiles=1
KeepTstepFilesIn=1
KeepTstepFilesOut=1
WriteOutOption=2
ResolutionFilesOut=3600
Tstep=300
ResolutionFilesIn=3600
ResolutionFilesInESTM=3600
DisaggMethod=1
RainDisaggMethod=100
DisaggMethodESTM=1
SuppressWarnings=1
KdownZen=0
diagnose=0
/
```

Note:

- In *Linux* and *Mac*, please add an empty line after the end slash.
- The file is not case-sensitive.
- The parameters and variables can appear in any order.

The parameters and their setting instructions are provided through the links below:

- *Scheme options*
  - *CBLUse*
  - *SnowUse*
  - *NetRadiationMethod*
  - *BaseTMethod*
  - *EmissionsMethod*
  - *StorageHeatMethod*
  - *OHMIncQF*
  - *StabilityMethod*
  - *RoughLenHeatMethod*
  - *RoughLenMomMethod*
  - *SMDMethod*
  - *WaterUseMethod*
  - *DiagMethod*
- *File related options*
  - *FileCode*
  - *FileInputPath*
  - *FileOutputPath*
  - *MultipleMetFiles*
  - *MultipleInitFiles*
  - *MultipleESTMFiles*
  - *KeepTstepFilesIn*
  - *KeepTstepFilesOut*
  - *WriteOutOption*
  - *SuppressWarnings*
- *Time related options*

  - *Tstep*
  - *ResolutionFilesIn*
  - *ResolutionFilesInESTM*
  - *ResolutionFilesOut*

- *Options related to disaggregation of input data*
  - *DisaggMethod*
  - *KdownZen*
  - *RainDisaggMethod*
  - *RainAmongN*
  - *MultRainAmongN*
  - *MultRainAmongNUpperI*
  - *DisaggMethodESTM*

Scheme options

`CBLUse`

Requirement Required

Description Determines whether a CBL slab model is used to calculate temperature and humidity.

Configuration

`SnowUse`

Requirement Required

Description Determines whether the snow part of the model runs.

Configuration

Value	Comments
0	Snow calculations are not performed.
1	Snow calculations are performed.

`NetRadiationMethod`

Requirement Required

Description Determines method for calculation of radiation fluxes.

Configuration

Value	Comments
0	Uses observed values of Q* supplied in meteorological forcing file.
1	Q* modelled with L↓ observations supplied in meteorological forcing file. Zenith angle not accounted for in albedo calculation.
2	Q* modelled with L↓ modelled using cloud cover fraction supplied in meteorological forcing file [Loridan *et al.*, 2011]. Zenith angle not accounted for in albedo calculation.
3	Q* modelled with L↓ modelled using air temperature and relative humidity supplied in meteorological forcing file [Loridan *et al.*, 2011]. Zenith angle not accounted for in albedo calculation.
11	Same as 1 but with L↑ modelled using surface temperature **Not recommended in this version.**
12	Same as 2 but with L↑ modelled using surface temperature **Not recommended in this version.**
13	Same as 3 but with L↑ modelled using surface temperature **Not recommended in this version.**
100	Q* modelled with L↓ observations supplied in meteorological forcing file. Zenith angle accounted for in albedo calculation. SSss_YYYY_NARPOut.txt file produced. **Not recommended in this version.**
200	Q* modelled with L↓ modelled using cloud cover fraction supplied in meteorological forcing file [Loridan *et al.*, 2011]. Zenith angle accounted for in albedo calculation. SSss_YYYY_NARPOut.txt file produced. **Not recommended in this version.**
300	Q* modelled with L↓ modelled using air temperature and relative humidity supplied in meteorological forcing file [Loridan *et al.*, 2011]. Zenith angle accounted for in albedo calculation. SSss_YYYY_NARPOut.txt file produced. **Not recommended in this version.**
1001	Q* modelled with *SPARTACUS-Surface (SS)* but with L↓ modelled as in 1. **Experimental in this version.**
1002	Q* modelled with *SPARTACUS-Surface (SS)* but with L↓ modelled as in 2. **Experimental in this version.**
1003	Q* modelled with *SPARTACUS-Surface (SS)* but with L↓ modelled as in 3. **Experimental in this version.**

`BaseTMethod`

Requirement Required

Description Determines method for base temperature used in HDD/CDD calculations.

Configuration

Value	Comments
1	V-shape approach: a single `BaseT_HC` is used
2	U-shape approach: `TCritic_Heating_WD` (`TCritic_Heating_WE`) and `TCritic_Cooling_WD` (`TCritic_Cooling_WE`) are used for HDD and CDD calculations in weekdays (weekends), respectively.

`EmissionsMethod`

Requirement Required

Description Determines method for QF calculation.

Configuration

Value	Comments
0	Uses values provided in the meteorological forcing file (SSss_YYYY_data_tt.txt). If you do not want to include QF to the calculation of surface energy balance, you should set values in the meteorological forcing file to zero to prevent calculation of QF. UMEP provides two methods to calculate QF LQF which is simpler GQF which is more complete but requires more data inputs
1	**Not recommended in this version.** Calculated according to Loridan *et al.* [2011] using coefficients specified in *SUEWS_AnthropogenicEmission.txt*. Modelled values will be used even if QF is provided in the meteorological forcing file.
2	**Recommended in this version.** Calculated according to Järvi *et al.* [2011] using coefficients specified in *SUEWS_AnthropogenicEmission.txt* and diurnal patterns specified in *SUEWS_Profiles.txt*. Modelled values will be used even if QF is provided in the meteorological forcing file.
3	Updated Loridan *et al.* [2011] method using daily (not instantaneous) air temperature (HDD(id-1,3)) using coefficients specified in *SUEWS_AnthropogenicEmission.txt*.
4	Järvi *et al.* [2019] method, in addition to anthropogenic heat due to building energy use calculated by Järvi *et al.* [2011], that due to metabolism and traffic is also calculated using coefficients specified in *SUEWS_AnthropogenicEmission.txt* and diurnal patterns specified in *SUEWS_Profiles.txt*. Modelled values will be used even if QF is provided in the meteorological forcing file.

`StorageHeatMethod`

Requirement Required

Description Determines method for calculating storage heat flux QS.

Configuration

Value	Comments
0	Uses observed values of QS supplied in meteorological forcing file.
1	QS modelled using the objective hysteresis model (OHM) [Grimmond *et al.*, 1991] using parameters specified for each surface type.
3	QS modelled using AnOHM [Sun *et al.*, 2017]. **Not recommended in this version.**
4	QS modelled using the Element Surface Temperature Method (ESTM) [Offerle *et al.*, 2005]. **Not recommended in this version.**

`OHMIncQF`

Requirement Required

Description Determines whether the storage heat flux calculation uses Q^* or (Q^* +QF).

Configuration

Value	Comments
0	QS modelled Q* only.
1	QS modelled using Q*+QF.

`StabilityMethod`

Requirement Required

Description Defines which atmospheric stability functions are used.

Configuration

Value	Comments
0	Not used.
1	Not used.
2	• Momentum: – unstable: Dyer [1974] modified by Högström [1988] – stable: Van Ulden and Holtslag [1985] • Heat: Dyer [1974] modified by Högström [1988] **Not recommended in this version.**
3	• Momentum: Campbell and Norman [1998] (Eq 7.27, Pg97) • Heat – unstable: Campbell and Norman [1998] – stable: Campbell and Norman [1998] **Recommended in this version.**
4	• Momentum: Businger *et al.* [1971] modified by Högström [1988] • Heat: Businger *et al.* [1971] modified by Högström [1988] **Not recommended in this version.**

`RoughLenHeatMethod`

Requirement Required

Description Determines method for calculating roughness length for heat.

Configuration

Value	Comments
1	Uses value of 0.1*z0m.
2	Calculated according to Kawai *et al.* [2009].
3	Calculated according to Voogt and Grimmond [2000].
4	Calculated according to Kanda *et al.* [2007].
5	Adaptively using z0m based on pervious coverage: if fully pervious, use method 1; otherwise, use method 2. **Recommended in this version.**

`RoughLenMomMethod`

Requirement Required

Description Determines how aerodynamic roughness length (z0m) and zero displacement height (zdm) are calculated.

Configuration

Value	Comments
1	Values specified in *SUEWS_SiteSelect.txt* are used. **Tip:** Note that UMEP provides tools to calculate these. See Kent *et al.* [2017] for recommendations on methods. Kent *et al.* [2017] have developed a method to include vegetation which is also avaialble within UMEP.
2	z0m and zd are calculated using 'rule of thumb' [Grimmond and Oke, 1999] using mean building and tree height specified in *SUEWS_SiteSelect.txt*. z0m and zd are adjusted with time to account for seasonal variation in porosity of deciduous trees.
3	z0m and zd are calculated based on the Macdonald *et al.* [1998] method using mean building and tree heights, plan area fraction and frontal areal index specified in *SUEWS_SiteSelect.txt*. z0m and zd are adjusted with time to account for seasonal variation in porosity of deciduous trees.

`SMDMethod`

Requirement Required

Description Determines method for calculating soil moisture deficit (SMD).

Configuration

Value	Comments
0	SMD modelled using parameters specified in *SUEWS_Soil.txt*. **Recommended in this version.**
1	Observed SM provided in the meteorological forcing file is used. Data are provided as volumetric soil moisture content. Metadata must be provided in *SUEWS_Soil.txt*.
2	Observed SM provided in the meteorological forcing file is used. Data are provided as gravimetric soil moisture content. Metadata must be provided in *SUEWS_Soil.txt*.

`SOLWEIGUse`

Deprecated since version v2020a.

Requirement Required

Description Determines whether SOLWEIG is used to calculate detailed radiation balance of all facets.

Configuration

Value	Comments
0	SOLWEIG calculations are not performed.
1	SOLWEIG calculations are performed. A grid of mean radiant temperature (Tmrt) is calculated based on high resolution digital surface models.

`WaterUseMethod`

Requirement Required

Description Defines how external water use is calculated.

Configuration

Value	Comments
0	External water use modelled using parameters specified in *SUEWS_Irrigation.txt*.
1	Observations of external water use provided in the meteorological forcing file are used.

`DiagMethod`

Requirement Required

Description Defines how near surface diagnostics are calculated.

Configuration

Value	Comments
0	Use MOST to calculate near surface diagnostics.
1	Use RST to calculate near surface diagnostics.
1	Use a set of criteria based on plan area index, frontal area index and heights of roughness elements to determine if RSL or MOST should be used.

Time related options

`Tstep`

Requirement Required

Description Specifies the model time step [s].

Configuration A value of 300 s (5 min) is strongly recommended. The time step cannot be less than 1 min or greater than 10 min, and must be a whole number of minutes that divide into an hour (i.e. options are 1, 2, 3, 4, 5, 6, 10 min or 60, 120, 180, 240, 300, 360, 600 s).

`ResolutionFilesIn`

Requirement Required

Description Specifies the resolution of the input files [s] which SUEWS will disaggregate to the model time step.

Configuration 1800 s for 30 min or 3600 s for 60 min are recommended.

Note: If `ResolutionFilesIn` is not provided, SUEWS assumes `ResolutionFilesIn` = Tstep.

`ResolutionFilesInESTM`

Requirement Optional

Description Specifies the resolution of the ESTM input files [s] which SUEWS will disaggregate to the model time step.

Configuration The same as for *ResolutionFilesIn*.

`ResolutionFilesOut`

Requirement Required

Description Specifies the resolution of the output files [s].

Configuration 1800 s for 30 min or 3600 s for 60 min are recommended.

File related options

`FileCode`

Requirement Required

Description Alphabetical site identification code (e.g. He, Sc, Kc).

Configuration This must be consistent with names of *meterological input file* and *initial condition files*

`FileInputPath`

Requirement Required

Description Input directory.

Configuration This can be set either as an absolute path or a relative path where the program is initiated.

`FileOutputPath`

Requirement Required

Description Output directory.

Configuration This can be set either as an absolute path or a relative path where the program is initiated.

`MultipleMetFiles`

Requirement Required

Description Specifies whether one single meteorological forcing file is used for all grids or a separate met file is provided for each grid.

Configuration

Value	Comments
0	Single meteorological forcing file used for all grids. No grid number should appear in the file name.
1	Separate meteorological forcing files used for each grid. The grid number should appear in the file name.

`MultipleInitFiles`

Requirement Required

Description Specifies whether one single initial conditions file is used for all grids at the start of the run or a separate initial conditions file is provided for each grid.

Configuration

Value	Comments
0	Single initial conditions file used for all grids. No grid number should appear in the file name.
1	Separate initial conditions files used for each grid. The grid number should appear in the file name.

`MultipleESTMFiles`

Requirement Optional

Description Specifies whether one single ESTM forcing file is used for all grids or a separate file is provided for each grid.

Configuration

Value	Comments
0	Single ESTM forcing file used for all grids. No grid number should appear in the file name.
1	Separate ESTM forcing files used for each grid. The grid number should appear in the file name.

`KeepTstepFilesIn`

Requirement Optional

Description Specifies whether input meteorological forcing files at the resolution of the model time step should be saved.

Configuration

Value	Comments
0	Meteorological forcing files at model time step are not written out. This is the default option Recommended to reduce processing time and save disk space as (e.g. 5-min) files can be large.
1	Meteorological forcing files at model time step are written out.

`KeepTstepFilesOut`

Requirement Optional

Description Specifies whether output meteorological forcing files at the resolution of the model time step should be saved.

Configuration

Value	Comments
0	Output files at model time are not saved. This is the default option. Recommended to save disk space as (e.g. 5-min) files can be large.
1	Output files at model time step are written out.

`WriteOutOption`

Requirement Optional

Description Specifies which variables are written in the output files.

Configuration

Value	Comments
0	All (except snow-related) output variables written. This is the default option.
1	All (including snow-related) output variables written.
2	Writes out a minimal set of output variables (use this to save space or if information about the different surfaces is not required).

`SuppressWarnings`

Requirement Optional

Description Controls whether the warnings.txt file is written or not.

Configuration

Value	Comments
0	The warnings.txt file is written. This is the default option.
1	No warnings.txt file is written. May be useful for large model runs as this file can grow large.

Options related to disaggregation of input data

`DisaggMethod`

Requirement Optional

Description Specifies how meteorological variables in the input file (except rain and snow) are disaggregated to the model time step. Wind direction is not currently downscaled so non -999 values will cause an error.

Configuration

Value	Comments
1	Linear downscaling of averages for all variables, additional zenith check is used for Kdown. This is the default option.
2	Linear downscaling of instantaneous values for all variables, additional zenith check is used for Kdown.
3	WFDEI setting: average Kdown (with additional zenith check); instantaneous for Tair, RH, pres and U. (N.B. WFDEI actually provides Q not RH)

`KdownZen`

Requirement Optional

Description Can be used to switch off zenith checking in Kdown disaggregation. Note that the zenith calculation requires location information obtained from *SUEWS_SiteSelect.txt*. If a single met file is used for all grids, the zenith is calculated for the first grid and the disaggregated data is then applied for all grids.

Configuration

Value	Comments
0	No zenith angle check is applied.
1	Disaggregated Kdown is set to zero when zenith angle exceeds 90 degrees (i.e. sun below horizon) and redistributed over the day. This is the default option.

`RainDisaggMethod`

Requirement Optional

Description Specifies how rain in the meteorological forcing file are disaggregated to the model time step. If present in the original met forcing file, snow is currently disaggregated in the same way as rainfall.

Configuration

Value	Comments
100	Rainfall is evenly distributed among all subintervals in a rainy interval. This is the default option.
101	Rainfall is evenly distributed among among RainAmongN subintervals in a rainy interval – also requires RainAmongN to be set.
102	Rainfall is evenly distributed among among RainAmongN subintervals in a rainy interval for different intensity bins – also requires MultRainAmongN and MultRainAmongNUpperI to be set.

`RainAmongN`

Requirement Optional

Description Specifies the number of subintervals (of length tt) over which to distribute rainfall in each interval (of length TT).

Configuration Must be an integer value. Use with RainDisaggMethod = 101.

`MultRainAmongN`

Requirement Optional

Description Specifies the number of subintervals (of length tt) over which to distribute rainfall in each interval (of length TT) for up to 5 intensity bins. Must take integer values.

Configuration Use with RainDisaggMethod = 102. e.g. MultRainAmongN(1) = 5, MultRainAmongN(2) = 8, MultRainAmongN(3) = 12

`MultRainAmongNUpperI`

Requirement Optional

Description Specifies upper limit for each intensity bin to apply MultRainAmongN.

Configuration Any intensities above the highest specified intensity will use the last MultRainAmongN value and write a warning to *Warning messages: warnings.txt*. Use with RainDisaggMethod = 102. e.g. MultRainAmongNUpperI(1) = 0.5, MultRainAmongNUpperI(2) = 2.0, MultRainAmongNUpperI(3) = 50.0

`DisaggMethodESTM`

Requirement Optional

Description Specifies how ESTM-related temperatures in the input file are disaggregated to the model time step.

Configuration

Value	Comments
1	Linear downscaling of averages.
2	Linear downscaling of instantaneous values.

7.3.2 SUEWS Site Information

Note:

1. We use the following codes for denoting the requirement level of various input variables/parameters for SUEWS throughout this section:

   `MU`

   Parameters which must be supplied and must be specific for the site/grid being run.

   `MD`

   Parameters which must be supplied and must be specific for the site/grid being run (but default values may be ok if these values are not known specifically for the site).

   `O`

   Parameters that are optional, depending on the model settings in *RunControl.nml*. Set any parameters that are not used/not known to '-999'.

   `L`

   Codes that are used to link between the input files, which must

   - be specified in the correct way to link the *main* and *sub-reference* files (similar to key-value pairs);
   - be integers and unique in column 1 of corresponding input files; and
   - match up with column 1 of the corresponding input file, even if those parameters are not used (in which case set all columns except column 1 to '-999' in the corresponding input file), otherwise the model run will fail.

2. We use the following codes for denoting the typical land cover/entity types of SUEWS throughout this section:

   `Paved`

   Paved surface

   `Bldgs`

   Building surface

   `EveTr`

   Evergreen trees and shrubs

   `DecTr`

   Deciduous trees and shrubs

   `Grass`

   Grass surface

   `BSoil`

   Unmanaged land and/or bare soil

   `Water`

   Water surface

Runoff
The water that drains freely off the impervious surface

SoilStore
The water stored in the underlying soil that infiltrates from the pervious surface

The following text files provide SUEWS with information about the study area.

SUEWS_AnthropogenicEmission.txt

Note: Changed in version v2019a: this file is renamed from `SUEWS_AnthropogenicHeat.txt` (prior to v2019a) to include more emission related settings.

SUEWS_AnthropogenicEmission.txt provides the parameters needed to model the anthropogenic heat flux using either the method of Loridan *et al.* [2011] based on air temperature (`EmissionsMethod` = 1 in *RunControl.nml*) or the method of Järvi *et al.* [2011] based on heating and cooling degree days (`EmissionsMethod` = 2 in *RunControl.nml*).

For the method of Järvi *et al.* [2011] (`EmissionsMethod` = 2 in *RunControl.nml*), one can further configure the scheme for calculting *HDD*/ *CDD* via `BaseTMethod` in *RunControl.nml*:

- `BaseTMethod` = 1 ("V-shape" approach): a single `BaseT_HC` is used by omitting the comfort range where neither heating nor cooling is activated.
- `BaseTMethod` = 2 ("U-shape" approach): `TCritic_Heating_WD` (`TCritic_Heating_WE`) and `TCritic_Cooling_WD` (`TCritic_Cooling_WE`) are used for HDD and CDD calculations in weekdays (weekends), respectively, which allows a comfort range between `TCritic_Heating_WD` (`TCritic_Heating_WE`) and `TCritic_Cooling_WD` (`TCritic_Cooling_WE`).

The sub-daily variation in anthropogenic heat flux is modelled according to the daily cycles specified in *SUEWS_Profiles.txt*.

Alternatively, if available, the anthropogenic heat flux can be provided in the met forcing file (and set `EmissionsMethod` = 0 in *RunControl.nml*) by filling the `qf` column with valid values.

No.	Column Name	Use	Description
1	*Code*	*L*	Code linking to a corresponding look-up table.
2	*BaseT_HC*	*MU*	Base temperature for heating degree days [°C]
3	*QF_A_WD*	*MU O*	Base value for QF on weekdays [W m^{-2} (Cap ha^{-1} $)^{-1}$]
4	*QF_B_WD*	*MU O*	Parameter related to cooling degree days on weekdays [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]
5	*QF_C_WD*	*MU O*	Parameter related to heating degree days on weekdays [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]
6	*QF_A_WE*	*MU O*	Base value for QF on weekends [W m^{-2} (Cap ha^{-1} $)^{-1}$]
7	*QF_B_WE*	*MU O*	Parameter related to cooling degree days on weekends [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]
8	*QF_C_WE*	*MU O*	Parameter related to heating degree days on weekends [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]
9	*AHMin_WD*	*MU O*	Minimum QF on weekdays [W m^{-2}]

continues on next page

Table 7.1 – continued from previous page

No.	Column Name	Use	Description
10	*AHMin_WE*	*MU O*	Minimum QF on weekends [W m^{-2}]
11	*AHSlope_Heating_WD*	*MU O*	Heating slope of QF on weekdays [W m^{-2} K^{-1}]
12	*AHSlope_Heating_WE*	*MU O*	Heating slope of QF on weekends [W m^{-2} K^{-1}]
13	*AHSlope_Cooling_WD*	*MU O*	Cooling slope of QF on weekdays [W m^{-2} K^{-1}]
14	*AHSlope_Cooling_WE*	*MU O*	Cooling slope of QF on weekends [W m^{-2} K^{-1}]
15	*TCritic_Heating_WD*	*MU O*	Critical heating temperature on weekdays [°C]
16	*TCritic_Heating_WE*	*MU O*	Critical heating temperature on weekends [°C]
17	*TCritic_Cooling_WD*	*MU O*	Critical cooling temperature on weekdays [°C]
18	*TCritic_Cooling_WE*	*MU O*	Critical cooling temperature on weekends [°C]
19	*EnergyUseProfWD*	*MU O*	Code linking to *EnergyUseProfWD* in *SUEWS_Profiles.txt*.
20	*EnergyUseProfWE*	*MU O*	Code linking to *EnergyUseProfWE* in *SUEWS_Profiles.txt*.
21	*ActivityProfWD*	*MU O*	Code linking to *ActivityProfWD* in *SUEWS_Profiles.txt*.
22	*ActivityProfWE*	*MU O*	Code linking to *ActivityProfWE* in *SUEWS_Profiles.txt*.
23	*TraffProfWD*	*MU O*	Code for traffic activity profile (weekdays) linking to *Code* of *SUEWS_Profiles.txt*. Not used in v2018a.
24	*TraffProfWE*	*MU O*	Code for traffic activity profile (weekends) linking to *Code* of *SUEWS_Profiles.txt*. Not used in v2018a.
25	*PopProfWD*	*MU O*	Code for population density profile (weekdays) linking to *Code* of *SUEWS_Profiles.txt*.
26	*PopProfWE*	*MU O*	Code for population density profile (weekends) linking to *Code* of *SUEWS_Profiles.txt*.
27	*MinQFMetab*	*MU O*	Minimum value for human heat emission. [W m^{-2}]
28	*MaxQFMetab*	*MU O*	Maximum value for human heat emission. [W m^{-2}]
29	*MinFCMetab*	*MU O*	Minimum (night) CO2 from human metabolism. [W m^{-2}]
30	*MaxFCMetab*	*MU O*	Maximum (day) CO2 from human metabolism. [W m^{-2}]
31	*FrPDDwe*	*MU O*	Fraction of weekend population to weekday population. [-]
32	*FrFossilFuel_Heat*	*MU O*	Fraction of fossil fuels used for building heating [-]
33	*FrFossilFuel_NonHeat*	*MU O*	Fraction of fossil fuels used for building energy use [-]
34	*EF_umolCO2perJ*	*MU O*	Emission factor for fuels used for building heating.

continues on next page

Table 7.1 – continued from previous page

No.	Column Name	Use	Description
35	*EnEF_v_Jkm*	*MU O*	Emission factor for heat [J k m^{-1}].
36	*FcEF_v_kgkmWD*	*MU O*	CO2 emission factor for weekdays [kg km^{-1}]
37	*FcEF_v_kgkmWE*	*MU O*	CO2 emission factor for weekends [kg km^{-1}]
38	*CO2PointSource*	*MU O*	CO2 emission factor [kg km^{-1}]
39	*TrafficUnits*	*MU O*	Units for the traffic rate for the study area. Not used in v2018a.

An example *SUEWS_AnthropogenicEmission.txt* can be found in the online version.

SUEWS_BiogenCO2.txt

Caution: The BiogenCO2 part is under development and not ready for use.

SUEWS_BiogenCO2.txt provides the parameters needed to model the Biogenic CO2 characteristics of vegetation surfaces.

No.	Column Name	Use	Description
1	*Code*	*L*	Code linking to a corresponding look-up table.
2	*alpha*	*MU O*	The mean apparent ecosystem quantum. Represents the initial slope of the light-response curve.
3	*beta*	*MU O*	The light-saturated gross photosynthesis of the canopy. [umol m^{-2} s^{-1}]
4	*theta*	*MU O*	The convexity of the curve at light saturation.
5	*alpha_enh*	*MU O*	Part of the `alpha` coefficient related to the fraction of vegetation.
6	*beta_enh*	*MU O*	Part of the `beta` coefficient related to the fraction of vegetation.
7	*resp_a*	*MU O*	Respiration coefficient a.
8	*resp_b*	*MU O*	Respiration coefficient b - related to air temperature dependency.
9	*min_respi*	*MU O*	Minimum soil respiration rate (for cold-temperature limit) [umol m^{-2} s^{-1}].

An example *SUEWS_BiogenCO2.txt* can be found online

SUEWS_Conductance.txt

SUEWS_Conductance.txt contains the parameters needed for the Jarvis (1976) [Jarvis, 1976] surface conductance model used in the modelling of evaporation in SUEWS. These values should **not** be changed independently of each other. The suggested values below have been derived using datasets for Los Angeles and Vancouver (see Järvi *et al.* [2011]) and should be used with `gsModel` = 1. An alternative formulation (`gsModel` =2) uses slightly different functional forms and different coefficients (with different units).

No.	Column Name	Use	Description
1	*Code*	*L*	Code linking to a corresponding look-up table.
2	*G1*	*MD*	Related to maximum surface conductance [mm s^{-1}]
3	*G2*	*MD*	Related to Kdown dependence [W m^{-2}]
4	*G3*	*MD*	Related to VPD dependence [units depend on `gsModel`]
5	*G4*	*MD*	Related to VPD dependence [units depend on `gsModel`]
6	*G5*	*MD*	Related to temperature dependence [°C]
7	*G6*	*MD*	Related to soil moisture dependence [mm^{-1}]
8	*TH*	*MD*	Upper air temperature limit [°C]
9	*TL*	*MD*	Lower air temperature limit [°C]
10	*S1*	*MD*	A parameter related to soil moisture dependence [-]
11	*S2*	*MD*	A parameter related to soil moisture dependence [mm]
12	*Kmax*	*MD*	Maximum incoming shortwave radiation [W m^{-2}]
13	`gsModel`	*MD*	Formulation choice for conductance calculation.

An example *SUEWS_Conductance.txt* can be found online

SUEWS_Irrigation.txt

External water use may be used for a wide range of reasons (e.g. cleaning roads, irrigating plants, fountains, washing cars).

SUEWS has two options for External Water use (if non-zero):

1) provide observed data in meteorological forcing file in the `Wuh` column with valid values by setting `WaterUseMethod` = 1 in *RunControl.nml*
2) a simple model that calculates daily water use from the mean daily air temperature, number of days since rain and fraction of irrigated area using automatic/manual irrigation. The user needs to supply coefficients (XXX) for these relations.
   a) sub-daily pattern of water use is detemined from the daily cycles specified in *SUEWS_Profiles.txt*.
   b) surface that the water can be applied to is specified by XX.
   c) water can pond.

No.	Column Name	Use	Description
1	Code	L	Code linking to a corresponding look-up table.
2	Ie_start	MU	Day when irrigation starts [DOY]
3	Ie_end	MU	Day when irrigation ends [DOY]
4	InternalWaterUse	MU	Internal water use [mm h^{-1}]
5	Faut	MU	Fraction of irrigated area that is irrigated using automated systems
6	H_maintain	MU	water depth to maintain used in automatic irrigation (e.g., ponding water due to flooding irrigation in rice crop-field) [mm].
7	Ie_a1	MD	Coefficient for automatic irrigation model [mm d^{-1}]
8	Ie_a2	MD	Coefficient for automatic irrigation model [mm d^{-1} K^{-1}]
9	Ie_a3	MD	Coefficient for automatic irrigation model [mm d^{-2}]
10	Ie_m1	MD	Coefficient for manual irrigation model [mm d^{-1}]
11	Ie_m2	MD	Coefficient for manual irrigation model [mm d^{-1} K^{-1}]
12	Ie_m3	MD	Coefficient for manual irrigation model [mm d^{-2}]
13	DayWat(1)	MU	Irrigation allowed on Sundays [1], if not [0]
14	DayWat(2)	MU	Irrigation allowed on Mondays [1], if not [0]
15	DayWat(3)	MU	Irrigation allowed on Tuesdays [1], if not [0]
16	DayWat(4)	MU	Irrigation allowed on Wednesdays [1], if not [0]
17	DayWat(5)	MU	Irrigation allowed on Thursdays [1], if not [0]
18	DayWat(6)	MU	Irrigation allowed on Fridays [1], if not [0]
19	DayWat(7)	MU	Irrigation allowed on Saturdays [1], if not [0]
20	DayWatPer(1)	MU	Fraction of properties using irrigation on Sundays [0-1]
21	DayWatPer(2)	MU	Fraction of properties using irrigation on Mondays [0-1]
22	DayWatPer(3)	MU	Fraction of properties using irrigation on Tuesdays [0-1]
23	DayWatPer(4)	MU	Fraction of properties using irrigation on Wednesdays [0-1]
24	DayWatPer(5)	MU	Fraction of properties using irrigation on Thursdays [0-1]
25	DayWatPer(6)	MU	Fraction of properties using irrigation on Fridays [0-1]
26	DayWatPer(7)	MU	Fraction of properties using irrigation on Saturdays [0-1]

An example *SUEWS_Irrigation.txt* can be found in the online version.

SUEWS_NonVeg.txt

SUEWS_NonVeg.txt specifies the characteristics for the non-vegetated surface cover types (Paved, Bldgs, BSoil) by linking codes in column 1 of *SUEWS_NonVeg.txt* to the codes specified in *SUEWS_SiteSelect.txt* (Code_Paved, Code_Bldgs, Code_BSoil). Each row should correspond to a particular surface type. For suggestions on how to complete this table, see: *Typical Values*.

No.	Column Name	Use	Description
1	Code	L	Code linking to a corresponding look-up table.
2	AlbedoMin	MU	Effective surface albedo (middle of the day value) for wintertime (not including snow).
3	AlbedoMax	MU	Effective surface albedo (middle of the day value) for summertime.
4	Emissivity	MU	Effective surface emissivity.
5	StorageMin	MD	Minimum water storage capacity for upper surfaces (i.e. canopy).
6	StorageMax	MD	Maximum water storage capacity for upper surfaces (i.e. canopy)
7	WetThreshold	MD	Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface [mm].
8	StateLimit	MD	Upper limit to the surface state. [mm]
9	DrainageEq	MD	Calculation choice for Drainage equation

continues on next page

Table 7.2 – continued from previous page

No.	Column Name	Use	Description
10	*DrainageCoef1*	*MD*	Coefficient D0 [mm h^{-1}] used in *DrainageEq*
11	*DrainageCoef2*	*MD*	Coefficient b [-] used in *DrainageEq*
12	*SoilTypeCode*	*L*	Code for soil characteristics below this surface linking to *Code* of *SUEWS_Soil.txt*
13	*SnowLimPatch*	*O*	Limit for the snow water equivalent when snow cover starts to be patchy [mm]
14	*SnowLimRemove*	*O*	Limit of the snow water equivalent for snow removal from roads and roofs [mm]
15	*OHMCode_SummerWet*	*L*	Code for OHM coefficients to use for this surface during wet conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
16	*OHMCode_SummerDry*	*L*	Code for OHM coefficients to use for this surface during dry conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
17	*OHMCode_WinterWet*	*L*	Code for OHM coefficients to use for this surface during wet conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
18	*OHMCode_WinterDry*	*L*	Code for OHM coefficients to use for this surface during dry conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
19	*OHMThresh_SW*	*MD*	Temperature threshold determining whether summer/winter OHM coefficients are applied [°C]
20	*OHMThresh_WD*	*MD*	Soil moisture threshold determining whether wet/dry OHM coefficients are applied [-]
21	*ESTMCode*	*L*	Code for ESTM coefficients linking to *SUEWS_ESTMCoefficients.txt*
22	*AnOHM_Cp*	*MU*	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]
23	*AnOHM_Kk*	*MU*	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]
24	*AnOHM_Ch*	*MU*	Bulk transfer coefficient for this surface to use in AnOHM [-]

An example *SUEWS_NonVeg.txt* can be found in the online version.

SUEWS_OHMCoefficients.txt

OHM, the Objective Hysteresis Model [Grimmond *et al.*, 1991] calculates the storage heat flux as a function of net all-wave radiation and surface characteristics.

- For each surface, OHM requires three model coefficients (a1, a2, a3). The three should be selected as a set.
- The **SUEWS_OHMCoefficients.txt** file provides these coefficients for each surface type.
- A variety of values has been derived for different materials and can be found in the literature (see: *Typical Values*).
- **Coefficients can be changed depending on:**
    1. surface wetness state (wet/dry) based on the calculated surface wetness state and soil moisture.
    2. season (summer/winter) based on a 5-day running mean air temperature.
- To use the same coefficients irrespective of wet/dry and summer/winter conditions, use the same code for all four OHM columns (*OHMCode_SummerWet*, *OHMCode_SummerDry*, *OHMCode_WinterWet* and *OHMCode_WinterDry*).

Note:

1. AnOHM (set in *RunControl.nml* by *StorageHeatMethod* = 3) does not use the coefficients specified in *SUEWS_OHMCoefficients.txt* but instead requires three parameters to be specified for each surface type

(including snow): heat capacity (*AnOHM_Cp*), thermal conductivity (*AnOHM_Kk*) and bulk transfer coefficient (*AnOHM_Ch*). These are specified in *SUEWS_NonVeg.txt*, *SUEWS_Veg.txt*, *SUEWS_Water.txt* and *SUEWS_Snow.txt*. No additional files are required for AnOHM.

2. AnOHM is under development in v2018b and should NOT be used!

No.	Column Name	Use	Description
1	*Code*	*L*	Code linking to a corresponding look-up table.
2	*a1*	*MU*	Coefficient for Q* term [-]
3	*a2*	*MU*	Coefficient for dQ*/dt term [h]
4	*a3*	*MU*	Constant term [W m^{-2}]

An example *SUEWS_OHMCoefficients.txt* can be found in the online version.

SUEWS_Profiles.txt

SUEWS_Profiles.txt specifies the daily cycle of variables related to human behaviour (energy use, water use and snow clearing). Different profiles can be specified for weekdays and weekends. The profiles are provided at hourly resolution here; the model will then linearly interpolate the profiles to the resolution of the model time step; some profiles may be normalized either by `sum` or by `mean` depending on the activity type while others not(see `Normalisation method` column of *table below*). Thus it does not matter whether columns 2-25 add up to, say 1, 24, or another number, because the model will eventually use the normalised values to rescale the results.

Note:

1. Currently, the snow clearing profiles are not interpolated as these are effectively a switch (0 for off and 1 for on).
2. If the anthropogenic heat flux and water use are specified in the met forcing file, the energy and water use profiles are ignored.

Activity	Description	Normalisation method	Weekday option	Weekend option
Energy use	This profile, in junction with population density (*PopDensDay* and *PopDensNight*), determines the overall anthropogenic heat.	`mean`	*EnergyUseProfWD*	*EnergyUseProfWE*
Population density	This profile, in junction with human activity (*ActivityProfWD* and *ActivityProfWE*), determines the anthropogenic heat due to metabolism.	None	*PopProfWD*	*PopProfWE*
Human activity	This profile, in junction with population density (*PopProfWD* and *PopProfWE*), determines the anthropogenic heat due to metabolism.	None	*ActivityProfWD*	*ActivityProfWE*
Traffic	This profile determines the anthropogenic heat due to traffic.	`mean`	*TraffProfWD*	*TraffProfWE*
Water use (manual)	This profile determines the irrigation under manual operation.	`sum`	*WaterUseProfManuWD*	*WaterUseProfManuWE*
Water use (automatic)	This profile determines the irrigation under automatic operation.	`sum`	*WaterUseProfAutoWD*	*WaterUseProfAutoWE*
Snow removal	This profile determines if snow removal is conducted at the end of each hour.	None	*SnowClearingProfWD*	*SnowClearingProfWE*

- Anthropogenic heat flux (weekday and weekend)
- Water use (weekday and weekend; manual and automatic irrigation)
- Snow removal (weekday and weekend)
- Human activity (weekday and weekend).

No.	Column Name	Use	Description
1	`Code`	*L*	Code linking to a corresponding look-up table.
2	2-25	*MU*	Multiplier for each hour of the day [-] for energy and water use. For SnowClearing, set those hours to 1 when snow removal from paved and roof surface is allowed (0 otherwise) if the snow removal limits set in the *SUEWS_NonVeg.txt* (SnowLimR emove column) are exceeded.

An example *SUEWS_Profiles.txt* can be found in the online version.

SUEWS_SiteSelect.txt

For each year and each grid, site specific surface cover information and other input parameters are provided to SUEWS by *SUEWS_SiteSelect.txt*. The model currently requires a new row for each year of the model run. All rows in this file will be read by the model and run.

No.	Column Name	Use	Description
1	`Grid`	*MU*	a unique number to represent grid
2	`Year`	*MU*	Year [YYYY]
3	`StartDLS`	*MU*	Start of the day light savings [DOY]
4	`EndDLS`	*MU*	End of the day light savings [DOY]
5	`lat`	*MU*	Latitude [deg].
6	`lng`	*MU*	longitude [deg]
7	`Timezone`	*MU*	Time zone [h] for site relative to UTC (east is positive). This should be set according to the times given in the meteorological forcing file(s).
8	`SurfaceArea`	*MU*	Area of the grid [ha].
9	`Alt`	*MU*	Altitude of grids [m].
10	`z`	*MU*	Measurement height [m] for all atmospheric forcing variables set in *SSss_YYYY_data_tt.txt*.
11	`id`	*MD*	Day of year [DOY]
12	`ih`	*MD*	Hour [H]
13	`imin`	*MD*	Minute [M]
14	`Fr_Paved`	*MU*	Surface cover fraction of `Paved` surfaces [-]
15	`Fr_Bldgs`	*MU*	Surface cover fraction of buildings [-]
16	`Fr_EveTr`	*MU*	Surface cover fraction of `EveTr`: evergreen trees and shrubs [-]
17	`Fr_DecTr`	*MU*	Surface cover fraction of deciduous trees and shrubs [-]
18	`Fr_Grass`	*MU*	Surface cover fraction of `Grass` [-]
19	`Fr_Bsoil`	*MU*	Surface cover fraction of bare soil or unmanaged land [-]
20	`Fr_Water`	*MU*	Surface cover fraction of open water [-]
21	`IrrFr_Paved`	*MU*	Fraction of `Paved` that is irrigated [-]
22	`IrrFr_Bldgs`	*MU*	Fraction of `Bldgs` that is irrigated [-]
23	`IrrFr_EveTr`	*MU*	Fraction of `EveTr` that is irrigated [-]
24	`IrrFr_DecTr`	*MU*	Fraction of `DecTr` that is irrigated [-]
25	`IrrFr_Grass`	*MU*	Fraction of `Grass` that is irrigated [-]
26	`IrrFr_BSoil`	*MU*	Fraction of `BSoil` that is irrigated [-]

continues on next page

Table 7.3 – continued from previous page

No.	Column Name	Use	Description
27	IrrFr_Water	MU	Fraction of Water that is irrigated [-]
28	H_Bldgs	MU	Mean building height [m]
29	H_EveTr	MU	Mean height of evergreen trees [m]
30	H_DecTr	MU	Mean height of deciduous trees [m]
31	z0	O	Roughness length for momentum [m]
32	zd	O	Zero-plane displacement [m]
33	FAI_Bldgs	O	Frontal area index for buildings [-]
34	FAI_EveTr	O	Frontal area index for evergreen trees [-]
35	FAI_DecTr	O	Frontal area index for deciduous trees [-]
36	PopDensDay	O	Daytime population density (i.e. workers, tourists) [people ha^{-1}]
37	PopDensNight	O	Night-time population density (i.e. residents) [people ha^{-1}]
38	TrafficRate_WD	O	Weekday traffic rate [veh km m^{-2} s-1] Can be used for CO2 flux calculation - not used in v2018a.
39	TrafficRate_WE	O	Weekend traffic rate [veh km m^{-2} s-1] Can be used for CO2 flux calculation - not used in v2018a.
40	QF0_BEU_WD	O	Building energy use [W m^{-2}]
41	QF0_BEU_WE	O	Building energy use [W m^{-2}]
42	Code_Paved	L	Code for Paved surface characteristics linking to Code of SUEWS_NonVeg.txt
43	Code_Bldgs	L	Code for Bldgs surface characteristics linking to Code of SUEWS_NonVeg.txt
44	Code_EveTr	L	Code for EveTr surface characteristics linking to Code of SUEWS_Veg.txt
45	Code_DecTr	L	Code for DecTr surface characteristics linking to Code of SUEWS_Veg.txt
46	Code_Grass	L	Code for Grass surface characteristics linking to Code of SUEWS_Veg.txt
47	Code_BSoil	L	Code for BSoil surface characteristics linking to Code of SUEWS_NonVeg.txt
48	Code_Water	L	Code for Water surface characteristics linking to Code of SUEWS_Water.txt
49	LUMPS_DrRate	MD	Drainage rate of bucket for LUMPS [mm h^{-1}]
50	LUMPS_Cover	MD	Limit when surface totally covered with water for LUMPS [mm]
51	LUMPS_MaxRes	MD	Maximum water bucket reservoir [mm] Used for LUMPS surface wetness control.
52	NARP_Trans	MD	Atmospheric transmissivity for NARP [-]
53	CondCode	L	Code for surface conductance parameters linking to Code of SUEWS_Conductance.txt
54	SnowCode	L	Code for snow surface characteristics linking to Code of SUEWS_Snow.txt
55	SnowClearingProfWD	L	Code for snow clearing profile (weekdays) linking to Code of SUEWS_Profiles.txt.
56	SnowClearingProfWE	L	Code for snow clearing profile (weekends) linking to Code of SUEWS_Profiles.txt.
57	AnthropogenicCode	L	Code for modelling anthropogenic heat flux linking to Code of SUEWS_AnthropogenicEmission.txt, which contains the model co-efficients for estimation of the anthropogenic heat flux (used if EmissionsMethod = 1, 2 in RunControl.nml).
58	IrrigationCode	L	Code for modelling irrigation linking to Code of SUEWS_Irrigation.txt
59	WaterUseProfManuWD	L	Code for water use profile (manual irrigation, weekdays) linking to Code of SUEWS_Profiles.txt.
60	WaterUseProfManuWE	L	Code for water use profile (manual irrigation, weekends) linking to Code of SUEWS_Profiles.txt.

continues on next page

Table 7.3 – continued from previous page

No.	Column Name	Use	Description
61	*WaterUseProfAutoWD*	*L*	Code for water use profile (automatic irrigation, weekdays) linking to *Code* of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in *Code* of *SUEWS_Profiles.txt*.
62	*WaterUseProfAutoWE*	*L*	Code for water use profile (automatic irrigation, weekends) linking to *Code* of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in *Code* of *SUEWS_Profiles.txt*.
63	*FlowChange*	*MD*	Difference in input and output flows for water surface [mm h^{-1}]
64	*RunoffToWater*	*MD* *MU*	Fraction of above-ground runoff flowing to water surface during flooding [-]
65	*PipeCapacity*	*MD* *MU*	Storage capacity of pipes [mm]
66	*GridConnection1of8*	*MD* *MU*	Number of the 1st grid where water can flow to
67	*Fraction1of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection1of8* [-]
68	*GridConnection2of8*	*MD* *MU*	Number of the 2nd grid where water can flow to
69	*Fraction2of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection2of8* [-]
70	*GridConnection3of8*	*MD* *MU*	Number of the 3rd grid where water can flow to
71	*Fraction3of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection3of8* [-]
72	*GridConnection4of8*	*MD* *MU*	Number of the 4th grid where water can flow to
73	*Fraction4of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection4of8* [-]
74	*GridConnection5of8*	*MD* *MU*	Number of the 5th grid where water can flow to
75	*Fraction5of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection5of8* [-]
76	*GridConnection6of8*	*MD* *MU*	Number of the 6th grid where water can flow to
77	*Fraction6of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection6of8* [-]
78	*GridConnection7of8*	*MD* *MU*	Number of the 7th grid where water can flow to
79	*Fraction7of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection7of8* [-]
80	*GridConnection8of8*	*MD* *MU*	Number of the 8th grid where water can flow to
81	*Fraction8of8*	*MD* *MU*	Fraction of water that can flow to *GridConnection8of8* [-]
82	*WithinGridPavedCode*	*L*	Code that links to the fraction of water that flows from *Paved* surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.
83	*WithinGridBldgsCode*	*L*	Code that links to the fraction of water that flows from *Bldgs* surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*
84	*WithinGridEveTrCode*	*L*	Code that links to the fraction of water that flows from *EveTr* surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.

continues on next page

Table 7.3 – continued from previous page

No.	Column Name	Use	Description
85	WithinGridDecTrCode	L	Code that links to the fraction of water that flows from DecTr surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.
86	WithinGridGrassCode	L	Code that links to the fraction of water that flows from Grass surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.
87	WithinGridBSoilCode	L	Code that links to the fraction of water that flows from BSoil surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.
88	WithinGridWaterCode	L	Code that links to the fraction of water that flows from Water surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.
89	AreaWall	MU	Area of wall within grid (needed for ESTM calculation) [m^2].
90	Fr_ESTMClass_Paved1	MU	Surface cover fraction of Paved surface class 1 used in ESTM calculations
91	Fr_ESTMClass_Paved2	MU	Surface cover fraction of Paved surface class 2 used in ESTM calculations
92	Fr_ESTMClass_Paved3	MU	Surface cover fraction of Paved surface class 3 used in ESTM calculations
93	Code_ESTMClass_Paved1	L	Code linking to *SUEWS_ESTMCoefficients.txt*
94	Code_ESTMClass_Paved2	L	Code linking to *SUEWS_ESTMCoefficients.txt*
95	Code_ESTMClass_Paved3	L	Code linking to *SUEWS_ESTMCoefficients.txt*
96	Fr_ESTMClass_Bldgs1	MU	Surface cover fraction of building class 1 used in ESTM calculations
97	Fr_ESTMClass_Bldgs2	MU	Surface cover fraction of building class 2 used in ESTM calculations
98	Fr_ESTMClass_Bldgs3	MU	Surface cover fraction of building class 3 used in ESTM calculations
99	Fr_ESTMClass_Bldgs4	MU	Surface cover fraction of building class 4 used in ESTM calculations
100	Fr_ESTMClass_Bldgs5	MU	Surface cover fraction of building class 5 used in ESTM calculations
101	Code_ESTMClass_Bldgs1	L	Code linking to *SUEWS_ESTMCoefficients.txt*
102	Code_ESTMClass_Bldgs2	L	Code linking to *SUEWS_ESTMCoefficients.txt*
103	Code_ESTMClass_Bldgs3	L	Code linking to *SUEWS_ESTMCoefficients.txt*
104	Code_ESTMClass_Bldgs4	L	Code linking to *SUEWS_ESTMCoefficients.txt*
105	Code_ESTMClass_Bldgs5	L	Code linking to *SUEWS_ESTMCoefficients.txt*

Attention:

- Two rows of -9 should be placed at end of this file.
- In this file the **column order is important**.
- Surface cover fractions specified from Fr_Paved to Fr_Water should sum up to 1.
- Surface cover fractions specified from Fr_ESTMClass_Paved1 to Fr_ESTMClass_Paved3 should sum up to 1.
- Surface cover fractions specified from Fr_ESTMClass_Bldgs1 to Fr_ESTMClass_Bldgs5 should sum up to 1.
- In this file the **row order is important** for simulations of **multiple grids and multiple years**. Ensure the rows in are arranged so that all grids for a particular year appear on consecutive lines (rather than grouping all years together for a particular grid). See below for a valid example:

```
Grid  Year ...
1     2001 ...
2     2001 ...
1     2002 ...
2     2002 ...
```

Tip: ! can be used to indicate comments in the file. Comments are not read by the programme so they can be used by the user to provide notes for their interpretation of the contents. This is strongly recommended.

Day Light Savings (DLS)

The dates for DLS normally vary for each year and country as they are often associated with a specific set of Sunday mornings at the beginning of summer and autumn. Note it is important to remember leap years. You can check http://www.timeanddate.com/time/dst/ for your city.

Tip: If DLS does not occur give a start and end day immediately after it. Make certain the dummy dates are correct for the hemisphere

- For northern hemisphere, use: 180 181
- For southern hemisphere, use: 365 1

Example when running multiple years (in this case 2008 and 2009 in Canada):

Year	start of daylight savings	end of daylight savings
2008	170	240
2009	172	242

Grid Connections (water flow between grids)

Caution:

- **Not available in this version.**
- columns between `GridConnection1of8` and `GridConnection8of8` in *SUEWS_SiteSelect.txt* can be set to zero.

This section gives an example of water flow between grids, calculated based on the relative elevation of the grids and length of the connecting surface between adjacent grids. For the square grids in the figure, water flow is assumed to be zero between diagonally adjacent grids, as the length of connecting surface linking the grids is very small. Model grids need not be square or the same size.

The table gives example values for the grid connections part of *SUEWS_SiteSelect.txt* for the grids shown in the figure. For each row, only water flowing out of the current grid is entered (e.g. water flows from 234 to 236 and 237, with a larger proportion of water flowing to 237 because of the greater length of connecting surface between 234 and 237 than between 234 and 236. No water is assumed to flow between 234 and 233 or 235 because there is no elevation difference between these grids. Grids 234 and 238 are at the same elevation and only connect at a point, so no water flows between them. Water enters grid 234 from grids 230, 231 and 232 as these are more elevated.

Note: Arrows indicate the water flow in to and out of grid 234, but note that only only water flowing out of each grid is entered in *SUEWS_SiteSelect.txt*

An example *SUEWS_SiteSelect.txt* can be found in the online version.

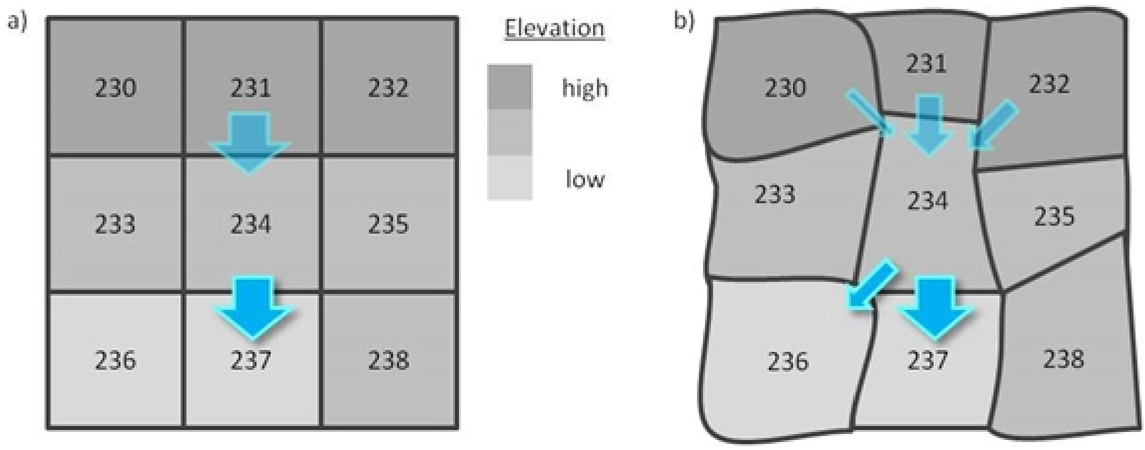


Fig. 7.1: Example grid connections showing water flow between grids.

Grid	GridConnection 1of8	Fraction1of8	GridConnection 2of8	Fraction2of8	GridConnection 3of8	Fraction3of8	GridConnection 4of8	Fraction4of8	GridConnection 5of8	Fraction5of8	GridConnection 6of8	Fraction6of8	GridConnection 7of8	Fraction7of8	GridConnection 8of8	Fraction8of8
230	233	0.90	234	0.10	0	0	0	0	0	0	0	0	0	0	0	0
231	234	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0
232	234	0.20	235	0.80	0	0	0	0	0	0	0	0	0	0	0	0
233	236	1.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0
234	236	0.10	237	0.90	0	0	0	0	0	0	0	0	0	0	0	0
235	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
236	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
237	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
238	237	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 7.2: Example values for the grid connections part of *SUEWS_SiteSelect.txt* for the grids.

SUEWS_Snow.txt

SUEWS_Snow.txt specifies the characteristics for snow surfaces when *SnowUse=1* in *RunControl.nml*. If the snow part of the model is not run, fill this table with '-999' except for the first (Code) column and set *SnowUse=0* in *RunControl.nml*. For a detailed description of the variables, see Järvi et al. (2014) [Järvi *et al.*, 2014].

No.	Column Name	Use	Description
1	*Code*	*L*	Code linking to a corresponding look-up table.
2	*RadMeltFactor*	*MU*	Hourly radiation melt factor of snow [mm W^{-1} h^{-1}]
3	*TempMeltFactor*	*MU*	Hourly temperature melt factor of snow [mm K^{-1} h^{-1}]
4	*AlbedoMin*	*MU*	Effective surface albedo (middle of the day value) for wintertime (not including snow).
5	*AlbedoMax*	*MU*	Effective surface albedo (middle of the day value) for summertime.
6	*Emissivity*	*MU*	Effective surface emissivity.
7	*tau_a*	*MD*	Time constant for snow albedo aging in cold snow [-]
8	*tau_f*	*MD*	Time constant for snow albedo aging in melting snow [-]
9	*PrecipLimAlb*	*MD*	Limit for hourly precipitation when the ground is fully covered with snow [mm]
10	*SnowDensMin*	*MD*	Fresh snow density [kg m^{-3}]
11	*SnowDensMax*	*MD*	Maximum snow density [kg m^{-3}]
12	*tau_r*	*MD*	Time constant for snow density ageing [-]
13	*CRWMin*	*MD*	Minimum water holding capacity of snow [mm]
14	*CRWMax*	*MD*	Maximum water holding capacity of snow [mm]
15	*PrecipLimSnow*	*MD*	Temperature limit when precipitation falls as snow [°C]
16	*OHMCode_SummerWet*	*L*	Code for OHM coefficients to use for this surface during wet conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
17	*OHMCode_SummerDry*	*L*	Code for OHM coefficients to use for this surface during dry conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
18	*OHMCode_WinterWet*	*L*	Code for OHM coefficients to use for this surface during wet conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
19	*OHMCode_WinterDry*	*L*	Code for OHM coefficients to use for this surface during dry conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
20	*OHMThresh_SW*	*MD*	Temperature threshold determining whether summer/winter OHM coefficients are applied [°C]
21	*OHMThresh_WD*	*MD*	Soil moisture threshold determining whether wet/dry OHM coefficients are applied [-]
22	*ESTMCode*	*L*	Code for ESTM coefficients linking to *SUEWS_ESTMCoefficients.txt*
23	*AnOHM_Cp*	*MU*	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]
24	*AnOHM_Kk*	*MU*	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]
25	*AnOHM_Ch*	*MU*	Bulk transfer coefficient for this surface to use in AnOHM [-]

An example *SUEWS_Snow.txt* can be found in the online version.

SUEWS_Soil.txt

SUEWS_Soil.txt specifies the characteristics of the sub-surface soil below each of the non-water surface types (Paved, Bldgs, EveTr, DecTr, Grass, BSoil). The model does not have a soil store below the water surfaces. Note that these sub-surface soil stores are different to the bare soil/unmamnaged surface cover type. Each of the non-water surface types need to link to soil characteristics specified here. If the soil characteristics are assumed to be the same for all surface types, use a single code value to link the characteristics here with the SoilTypeCode columns in *SUEWS_NonVeg.txt* and *SUEWS_Veg.txt*.

Soil moisture can either be provided using observational data in the met forcing file (the `xsmd` column when `SMDMethod` = 1 or 2 in *RunControl.nml*) and providing some soil properties here, or modelled by SUEWS (`SMDMethod` = 0 in *RunControl.nml*).

No.	Column Name	Use	Description
1	`Code`	*L*	Code linking to a corresponding look-up table.
2	`SoilDepth`	*MD*	Depth of soil beneath the surface [mm]
3	`SoilStoreCap`	*MD*	Limit value for `SoilDepth` [mm]
4	`SatHydraulicCond`	*MD*	Hydraulic conductivity for saturated soil [mm s^{-1}]
5	`SoilDensity`	*MD*	Soil density [kg m^{-3}]
6	`InfiltrationRate`	*O*	Infiltration rate.
7	`OBS_SMDepth`	*O*	The depth of soil moisture measurements. [mm]
8	`OBS_SMCap`	*O*	The maximum observed soil moisture. [m^3 m^{-3} or kg kg^{-1}]
9	`OBS_SoilNotRocks`	*O*	Fraction of soil without rocks. [-]

An example *SUEWS_Soil.txt* can be found in the online version.

SUEWS_Veg.txt

SUEWS_Veg.txt specifies the characteristics for the vegetated surface cover types (EveTr, DecTr, Grass) by linking codes in column 1 of SUEWS_Veg.txt to the codes specified in *SUEWS_SiteSelect.txt* (Code_EveTr, Code_DecTr, Code_Grass). Each row should correspond to a particular surface type. For suggestions on how to complete this table, see: *Typical Values*.

No.	Column Name	Use	Description
1	`Code`	*L*	Code linking to a corresponding look-up table.
2	`AlbedoMin`	*MU*	Effective surface albedo (middle of the day value) for wintertime (not including snow).
3	`AlbedoMax`	*MU*	Effective surface albedo (middle of the day value) for summertime.
4	`Emissivity`	*MU*	Effective surface emissivity.
5	`StorageMin`	*MD*	Minimum water storage capacity for upper surfaces (i.e. canopy).
6	`StorageMax`	*MD*	Maximum water storage capacity for upper surfaces (i.e. canopy)
7	`WetThreshold`	*MD*	Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface [mm].
8	`StateLimit`	*MD*	Upper limit to the surface state. [mm]
9	`DrainageEq`	*MD*	Calculation choice for Drainage equation
10	`DrainageCoef1`	*MD*	Coefficient D0 [mm h^{-1}] used in `DrainageEq`
11	`DrainageCoef2`	*MD*	Coefficient b [-] used in `DrainageEq`
12	`SoilTypeCode`	*L*	Code for soil characteristics below this surface linking to `Code` of *SUEWS_Soil.txt*
13	`SnowLimPatch`	*O*	Limit for the snow water equivalent when snow cover starts to be patchy [mm]

continues on next page

Table 7.5 – continued from previous page

No.	Column Name	Use	Description
14	*BaseT*	*MU*	Base Temperature for initiating growing degree days (GDD) for leaf growth. [°C]
15	*BaseTe*	*MU*	Base temperature for initiating sensesance degree days (SDD) for leaf off. [°C]
16	*GDDFull*	*MU*	The growing degree days (GDD) needed for full capacity of the leaf area index (LAI) [°C].
17	*SDDFull*	*MU*	The sensesence degree days (SDD) needed to initiate leaf off. [°C]
18	*LAIMin*	*MD*	leaf-off wintertime value
19	*LAIMax*	*MD*	full leaf-on summertime value
20	*PorosityMin*	*MD*	leaf-off wintertime value Used only for *DecTr* (can affect roughness calculation)
21	*PorosityMax*	*MD*	full leaf-on summertime value Used only for *DecTr* (can affect roughness calculation)
22	*MaxConductance*	*MD*	The maximum conductance of each vegetation or surface type. [mm s^{-1}]
23	*LAIEq*	*MD*	LAI calculation choice.
24	*LeafGrowthPower1*	*MD*	a parameter required by LAI calculation in *LAIEq*
25	*LeafGrowthPower2*	*MD*	a parameter required by LAI calculation [K^{-1}] in *LAIEq*
26	*LeafOffPower1*	*MD*	a parameter required by LAI calculation [K^{-1}] in *LAIEq*
27	*LeafOffPower2*	*MD*	a parameter required by LAI calculation [K^{-1}] in *LAIEq*
28	*OHMCode_SummerWet*	*L*	Code for OHM coefficients to use for this surface during wet conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
29	*OHMCode_SummerDry*	*L*	Code for OHM coefficients to use for this surface during dry conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
30	*OHMCode_WinterWet*	*L*	Code for OHM coefficients to use for this surface during wet conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
31	*OHMCode_WinterDry*	*L*	Code for OHM coefficients to use for this surface during dry conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
32	*OHMThresh_SW*	*MD*	Temperature threshold determining whether summer/winter OHM coefficients are applied [°C]
33	*OHMThresh_WD*	*MD*	Soil moisture threshold determining whether wet/dry OHM coefficients are applied [-]
34	*ESTMCode*	*L*	Code for ESTM coefficients linking to *SUEWS_ESTMCoefficients.txt*
35	*AnOHM_Cp*	*MU*	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]
36	*AnOHM_Kk*	*MU*	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]
37	*AnOHM_Ch*	*MU*	Bulk transfer coefficient for this surface to use in AnOHM [-]
38	*BiogenCO2Code*	*MU*	Code linking to the *Code* column in *SUEWS_BiogenCO2.txt*.

An example *SUEWS_Veg.txt* can be found in the online version.

SUEWS_Water.txt

SUEWS_Water.txt specifies the characteristics for the water surface cover type by linking codes in column 1 of *SUEWS_Water.txt* to the codes specified in *SUEWS_SiteSelect.txt* (*Code_Water*).

No.	Column Name	Use	Description
1	*Code*	*L*	Code linking to a corresponding look-up table.
2	*AlbedoMin*	*MU*	Effective surface albedo (middle of the day value) for wintertime (not including snow).
3	*AlbedoMax*	*MU*	Effective surface albedo (middle of the day value) for summertime.

continues on next page

Table 7.6 – continued from previous page

No.	Column Name	Use	Description
4	Emissivity	MU	Effective surface emissivity.
5	StorageMin	MD	Minimum water storage capacity for upper surfaces (i.e. canopy).
6	StorageMax	MD	Maximum water storage capacity for upper surfaces (i.e. canopy)
7	WetThreshold	MD	Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface [mm].
8	StateLimit	MU	Upper limit to the surface state. [mm]
9	WaterDepth	MU	Water depth [mm].
10	DrainageEq	MD	Calculation choice for Drainage equation
11	DrainageCoef1	MD	Coefficient D0 [mm h^{-1}] used in DrainageEq
12	DrainageCoef2	MD	Coefficient b [-] used in DrainageEq
13	OHMCode_SummerWet	L	Code for OHM coefficients to use for this surface during wet conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
14	OHMCode_SummerDry	L	Code for OHM coefficients to use for this surface during dry conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.
15	OHMCode_WinterWet	L	Code for OHM coefficients to use for this surface during wet conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
16	OHMCode_WinterDry	L	Code for OHM coefficients to use for this surface during dry conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.
17	OHMThresh_SW	MD	Temperature threshold determining whether summer/winter OHM coefficients are applied [°C]
18	OHMThresh_WD	MD	Soil moisture threshold determining whether wet/dry OHM coefficients are applied [-]
19	ESTMCode	L	Code for ESTM coefficients linking to *SUEWS_ESTMCoefficients.txt*
20	AnOHM_Cp	MU	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]
21	AnOHM_Kk	MU	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]
22	AnOHM_Ch	MU	Bulk transfer coefficient for this surface to use in AnOHM [-]

An example *SUEWS_Water.txt* can be found in the online version.

SUEWS_WithinGridWaterDist.txt

SUEWS_WithinGridWaterDist.txt specifies the movement of water between surfaces within a grid/area. It allows impervious connectivity to be taken into account.

Each row corresponds to a surface type (linked by the Code in column 1 to the *SUEWS_SiteSelect.txt* columns: WithinGridPavedCode, WithinGridBldgsCode, . . . , WithinGridWaterCode). Each column contains the fraction of water flowing from the surface type to each of the other surface types or to runoff or the sub-surface soil store.

Note:

- The sum of each row (excluding the Code) must equal 1.
- Water **CANNOT** flow from one surface to that same surface, so the diagonal elements should be zero.
- The row corresponding to the water surface should be zero, as there is currently no flow permitted from the water surface to other surfaces by the model.
- Currently water **CANNOT** go to both runoff and soil store (i.e. it must go to one or the other – `Runoff` for impervious surfaces; `SoilStore` for pervious surfaces).

In the table below, for example,

- All flow from paved surfaces goes to runoff;
- 90% of flow from buildings goes to runoff, with small amounts going to other surfaces (mostly paved surfaces as buildings are often surrounded by paved areas);
- All flow from vegetated and bare soil areas goes into the sub-surface soil store;
- The row corresponding to water contains zeros (as it is currently not used).

No.	Column Name	Use	Description
1	`ToPaved`	`MU`	Fraction of water going to `Paved`
2	`ToBldgs`	`MU`	Fraction of water going to `Bldgs`
3	`ToEveTr`	`MU`	Fraction of water going to `EveTr`
4	`ToDecTr`	`MU`	Fraction of water going to `DecTr`
5	`ToGrass`	`MU`	Fraction of water going to `Grass`
6	`ToBSoil`	`MU`	Fraction of water going to `BSoil`
7	`ToWater`	`MU`	Fraction of water going to `Water`
8	`ToRunoff`	`MU`	Fraction of water going to `Runoff`
9	`ToSoilStore`	`MU`	Fraction of water going to `SoilStore`

An example *SUEWS_WithinGridWaterDist.txt* can be found in the online version.

Input Options

`a1`

Description Coefficient for Q* term [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_OHMCoefficients.txt	`MU`	Coefficient for Q* term [-]

`a2`

Description Coefficient for `dQ*/dt` term [h]

Configuration

Referencing Table	Requirement	Comment
SUEWS_OHMCoefficients.txt	`MU`	Coefficient for dQ*/dt term [h]

`a3`

Description Constant term [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_OHMCoefficients.txt	`MU`	Constant term [W m^{-2}]

ActivityProfWD

Description Code linking to *ActivityProfWD* in *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*L*	Code for human activity profile (weekdays) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*. Used for CO2 flux calculation.

ActivityProfWE

Description Code linking to *ActivityProfWE* in *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*L*	Code for human activity profile (weekends) Provides the link to column 1 of *SUEWS_Profiles.txt*. Look the codes Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*. Used for CO2 flux calculation.

AHMin_WD

Description Minimum QF on weekdays [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with *EmissionsMethod* = 1

AHMin_WE

Description Minimum QF on weekends [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with *EmissionsMethod* = 1

AHSlope_Heating_WD

Description Heating slope of QF on weekdays [W m^{-2} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

AHSlope_Heating_WE

Description Heating slope of QF on weekends [W m^{-2} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

AHSlope_Cooling_WD

Description Cooling slope of QF on weekdays [W m^{-2} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

AHSlope_Cooling_WE

Description Cooling slope of QF on weekends [W m^{-2} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

AlbedoMax

Description Effective surface albedo (middle of the day value) for summertime.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MU*	Effective surface albedo (middle of the day value) for summertime. View factors should be taken into account.

continues on next page

Table 7.18 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MU*	Example values [-] • 0.1 EveTr [Oke, 2002] • 0.18 DecTr [Oke, 2002] • 0.21 Grass [Oke, 2002]
SUEWS_Water.txt	*MU*	Example values [-] • 0.1 Water [Oke, 2002]
SUEWS_Snow.txt	*MU*	Example values [-] • 0.85 [Järvi *et al.*, 2014]

AlbedoMin

Description Effective surface albedo (middle of the day value) for wintertime (not including snow).

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MU*	Not currently used for non-vegetated surfaces – set the same as AlbedoMax.
SUEWS_Veg.txt	*MU*	Example values [-] • 0.1 EveTr [Oke, 2002] • 0.18 DecTr [Oke, 2002] • 0.21 Grass [Oke, 2002]
SUEWS_Water.txt	*MU*	Not currently used for water surface - set same as AlbedoMax.
SUEWS_Snow.txt	*MU*	Example values [-] • 0.18 [Järvi *et al.*, 2014]

alpha

Description The mean apparent ecosystem quantum. Represents the initial slope of the light-response curve. [umol CO2 umol photons^-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU O*	Example values: `EmissionsMethod` = 11, 12, 13, 14, 15 or 16: • 0.044 [Ruimy *et al.*, 1995] • 0.0593 [Schmid, 2000] • 0.0205 [Flanagan *et al.*, 2002] `EmissionsMethod` = 21, 22, 23, 24, 25, or 26: 0.031 [Bellucco *et al.*, 2017] `EmissionsMethod` = 31, 32, 33, 34, 35, 36: 0.005 [Bellucco *et al.*, 2017]

Alt

Description Altitude of grids [m].

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Used for both the radiation and water flow between grids. **Not available in this version.**

AnOHM_Ch

Description Bulk transfer coefficient for this surface to use in AnOHM [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MU*	Bulk transfer coefficient for this surface to use in AnOHM [-]
SUEWS_Veg.txt	*MU*	Bulk transfer coefficient for this surface to use in AnOHM [-]
SUEWS_Water.txt	*MU*	Bulk transfer coefficient for this surface to use in AnOHM [-]
SUEWS_Snow.txt	*MU*	Bulk transfer coefficient for this surface to use in AnOHM [-]

AnOHM_Cp

Description Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MU*	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]
SUEWS_Veg.txt	*MU*	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]
SUEWS_Water.txt	*MU*	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]
SUEWS_Snow.txt	*MU*	Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]

`AnOHM_Kk`

Description Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MU*	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]
SUEWS_Veg.txt	*MU*	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]
SUEWS_Water.txt	*MU*	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]
SUEWS_Snow.txt	*MU*	Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]

`AnthropogenicCode`

Description Code for modelling anthropogenic heat flux linking to `Code` of *SUEWS_AnthropogenicEmission.txt*, which contains the model coefficients for estimation of the anthropogenic heat flux (used if `EmissionsMethod` = 1, 2 in *RunControl.nml*).

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_AnthropogenicEmission.txt*.

`AreaWall`

Description Area of wall within grid (needed for ESTM calculation) [m^2].

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Area of wall within grid (needed for ESTM calculation). [m^2]

`BaseT`

Description Base Temperature for initiating growing degree days (GDD) for leaf growth. [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MU*	See section 2.2 Järvi et al. (2011); Appendix A of Järvi *et al.* [2014]. Example values: 5 for EveTr [Järvi *et al.*, 2011]

`BaseTe`

Description Base temperature for initiating sensesance degree days (SDD) for leaf off. [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MU*	See section 2.2 Järvi *et al.* [2011] ; Appendix A Järvi *et al.* [2014] . Example values: 10 EveTr Järvi *et al.* [2011]

`BaseT_HC`

Description Base temperature for heating degree days [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU*	Base temperature for heating degree days [°C] e.g. Sailor and Vasireddy [2006]

`beta`

Description The light-saturated gross photosynthesis of the canopy. [umol m^{-2} s^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU 0*	Example values: `EmissionsMethod` = 11, 12, 13, 14, 15, 16: • 43.35 [Ruimy *et al.*, 1995] • 35 [Schmid, 2000] • 16.3 [Flanagan *et al.*, 2002] `EmissionsMethod` = 21, 22, 23, 24, 25, 26: 17.793 [Bellucco *et al.*, 2017] `EmissionsMethod` = 31, 32, 33, 34, 35, 36: 8.474 [Bellucco *et al.*, 2017]

theta

Description The convexity of the curve at light saturation.

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU 0*	Example value: `EmissionsMethod` = 21, 22, 23, 24, 25, 26: 0.723 [Bellucco *et al.*, 2017] `EmissionsMethod` = 31, 32, 33, 34, 35, 36: 0.96 [Bellucco *et al.*, 2017]

alpha_enh

Description Part of the `alpha` coefficient related to the fraction of vegetation.

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU 0*	Example value: 0.016 [Bellucco *et al.*, 2017]

beta_enh

Description Part of the `beta` coefficient related to the fraction of vegetation.

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU 0*	Example values: 33.454 [Bellucco *et al.*, 2017]

resp_a

Description Respiration coefficient a.

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU O*	Example values: • 1.08 [Schmid, 2000] • 3.229 [Järvi *et al.*, 2012]

`resp_b`

Description Respiration coefficient b - related to air temperature dependency.

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU O*	Example values: • 0.0064 [Schmid, 2000] • 0.0329 [Järvi *et al.*, 2012]

`min_respi`

Description Minimum soil respiration rate (for cold-temperature limit) [umol m^{-2} s^{-1}].

Configuration

Referencing Table	Requirement	Comment
SUEWS_BiogenCO2.txt	*MU O*	Example values: 0.6 estimate from Hyytiälä forest site.

`BiogenCO2Code`

Description Code linking to the `Code` column in *SUEWS_BiogenCO2.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*L*	Code linking to the `Code` column in *SUEWS_BiogenCO2.txt*.

`QF0_BEU_WD`

Description Building energy use [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*0*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

QF0_BEU_WE

Description Building energy use [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*0*	Can be used for CO2 flux calculation.

CO2PointSource

Description CO2 emission factor [kg km^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*0*	CO2 emission factor [kg km^{-1}]

Code

Description Code linking to a corresponding look-up table.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*L*	Code linking to *SUEWS_SiteSelect.txt* for paved surfaces (Code_Paved), buildings (Code_Bldgs) and bare soil surfaces (Code_BSoil). Value of integer is arbitrary but must match codes specified in *SUEWS_SiteSelect.txt*.
SUEWS_Veg.txt	*L*	Code linking to *SUEWS_SiteSelect.txt* for evergreen trees and shrubs (Code_EveTr), deciduous trees and shrubs (Code_DecTr) and grass surfaces (Code_Grass). Value of integer is arbitrary but must match codes specified in *SUEWS_SiteSelect.txt*.

continues on next page

Table 7.41 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Water.txt	*L*	Code linking to *SUEWS_SiteSelect.txt* for water surfaces (Code_Water). Value of integer is arbitrary but must match code specified in *SUEWS_SiteSelect.txt*.
SUEWS_Snow.txt	*L*	Code linking to *SUEWS_SiteSelect.txt* for snow surfaces (SnowCode). Value of integer is arbitrary but must match code specified in *SUEWS_SiteSelect.txt*.
SUEWS_Soil.txt	*L*	Code linking to the SoilTypeCode column in SUEWS_NonVeg.txt (for Paved, Bldgs and BSoil surfaces) and SUEWS_Veg.txt (for EveTr, DecTr and Grass surfaces). Value of integer is arbitrary but must match code specified in *SUEWS_SiteSelect.txt*.
SUEWS_Conductance.txt	*L*	Code linking to the CondCode column in *SUEWS_SiteSelect.txt* . Value of integer is arbitrary but must match code specified in *SUEWS_SiteSelect.txt*.
SUEWS_AnthropogenicEmission.txt	*L*	Code linking to the AnthropogenicCode column in *SUEWS_SiteSelect.txt* . Value of integer is arbitrary but must match code specified in *SUEWS_SiteSelect.txt*.
SUEWS_Irrigation.txt	*L*	Code linking to *SUEWS_SiteSelect.txt* for irrigation modelling (IrrigationCode). Value of integer is arbitrary but must match codes specified in *SUEWS_SiteSelect.txt*.
SUEWS_OHMCoefficients.txt	*L*	Code linking to the `OHMCode_SummerWet`, `OHMCode_SummerDry`, `OHMCode_WinterWet` and `OHMCode_WinterDry` columns in *SUEWS_NonVeg.txt*, *SUEWS_Veg.txt*, *SUEWS_Water.txt* and *SUEWS_Snow.txt* files. Value of integer is arbitrary but must match code specified in *SUEWS_SiteSelect.txt*.

continues on next page

Table 7.41 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*L*	For buildings and paved surfaces, set to zero if there is more than one ESTM class per grid and the codes and surface fractions specified in *SUEWS_SiteSelect.txt* will be used instead.
SUEWS_BiogenCO2.txt	*L*	Code linking to the `BiogenCO2Code` column in *SUEWS_Veg.txt*.

`Code_Bldgs`

Description Code for `Bldgs` surface characteristics linking to `Code` of *SUEWS_NonVeg.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for Bldgs surface characteristics Provides the link to column 1 of *SUEWS_NonVeg.txt*, which contains the attributes describing buildings in this grid for this year. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_NonVeg.txt*.

`Code_BSoil`

Description Code for `BSoil` surface characteristics linking to `Code` of *SUEWS_NonVeg.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_NonVeg.txt*.

`Code_DecTr`

Description Code for `DecTr` surface characteristics linking to `Code` of *SUEWS_Veg.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for DecTr surface characteristics Provides the link to column 1 of *SUEWS_Veg.txt*, which contains the attributes describing deciduous trees and shrubs in this grid for this year. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Veg.txt*.

Code_ESTMClass_Bldgs1

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

Code_ESTMClass_Bldgs2

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

Code_ESTMClass_Bldgs3

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

Code_ESTMClass_Bldgs4

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

`Code_ESTMClass_Bldgs5`

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

`Code_ESTMClass_Paved1`

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

`Code_ESTMClass_Paved2`

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

`Code_ESTMClass_Paved3`

Description Code linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code linking to *SUEWS_ESTMCoefficients.txt*

`Code_EveTr`

Description Code for `EveTr` surface characteristics linking to `Code` of *SUEWS_Veg.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for EveTr surface characteristics Provides the link to column 1 of *SUEWS_Veg.txt*, which contains the attributes describing evergreen trees and shrubs in this grid for this year. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Veg.txt*.

`Code_Grass`

Description Code for `Grass` surface characteristics linking to `Code` of *SUEWS_Veg.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for Grass surface characteristics Provides the link to column 1 of *SUEWS_Veg.txt*, which contains the attributes describing grass surfaces in this grid for this year. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Veg.txt*.

`Code_Paved`

Description Code for `Paved` surface characteristics linking to `Code` of *SUEWS_NonVeg.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for Paved surface characteristics Provides the link to column 1 of *SUEWS_NonVeg.txt*, which contains the attributes describing paved areas in this grid for this year. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_NonVeg.txt*. e.g. 331 means use the characteristics specified in the row of input file *SUEWS_NonVeg.txt* which has 331 in column 1 (Code).

`Code_Water`

Description Code for `Water` surface characteristics linking to `Code` of *SUEWS_Water.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for Water surface characteristics Provides the link to column 1 of *SUEWS_Water.txt*, which contains the attributes describing open water in this grid for this year. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Water.txt*.

`CondCode`

Description Code for surface conductance parameters linking to `Code` of *SUEWS_Conductance.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for surface conductance parameters Provides the link to column 1 of *SUEWS_Conductance.txt*, which contains the parameters for the Jarvis [1976] parameterisation of surface conductance. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Conductance.txt*. e.g. 33 means use the characteristics specified in the row of input file *SUEWS_Conductance.txt* which has 33 in column 1 (Code).

`CRWMax`

Description Maximum water holding capacity of snow [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MD*	Maximum water holding capacity of snow [mm]

`CRWMin`

Description Minimum water holding capacity of snow [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MD*	Minimum water holding capacity of snow [mm]

`DayWat(1)`

Description Irrigation allowed on Sundays [1], if not [0]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Irrigation allowed on Sundays [1], if not [0]

`DayWat(2)`

Description Irrigation allowed on Mondays [1], if not [0]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Irrigation allowed on Mondays [1], if not [0]

`DayWat(3)`

Description Irrigation allowed on Tuesdays [1], if not [0]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Irrigation allowed on Tuesdays [1], if not [0]

`DayWat(4)`

Description Irrigation allowed on Wednesdays [1], if not [0]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Irrigation allowed on Wednesdays [1], if not [0]

`DayWat(5)`

Description Irrigation allowed on Thursdays [1], if not [0]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Irrigation allowed on Thursdays [1], if not [0]

`DayWat(6)`

Description Irrigation allowed on Fridays [1], if not [0]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Irrigation allowed on Fridays [1], if not [0]

`DayWat(7)`

Description Irrigation allowed on Saturdays [1], if not [0]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Irrigation allowed on Saturdays [1], if not [0]

`DayWatPer(1)`

Description Fraction of properties using irrigation on Sundays [0-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of properties using irrigation on Sundays [0-1]

`DayWatPer(2)`

Description Fraction of properties using irrigation on Mondays [0-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of properties using irrigation on Mondays [0-1]

`DayWatPer(3)`

Description Fraction of properties using irrigation on Tuesdays [0-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of properties using irrigation on Tuesdays [0-1]

DayWatPer(4)

Description Fraction of properties using irrigation on Wednesdays [0-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of properties using irrigation on Wednesdays [0-1]

DayWatPer(5)

Description Fraction of properties using irrigation on Thursdays [0-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of properties using irrigation on Thursdays [0-1]

DayWatPer(6)

Description Fraction of properties using irrigation on Fridays [0-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of properties using irrigation on Fridays [0-1]

DayWatPer(7)

Description Fraction of properties using irrigation on Saturdays [0-1]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of properties using irrigation on Saturdays [0-1]

DrainageCoef1

Description Coefficient D0 [mm h^{-1}] used in *DrainageEq*

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MD*	• Example values: – `DrainageEq` = 3, 10 for `Paved` and `Bldgs`; – `DrainageEq` = 2, 0.013 for `BSoil`
SUEWS_Veg.txt	*MD*	• Example values: – `DrainageEq` = 3, 10 for `Grass` (irrigated); – `DrainageEq` = 2, 0.013 for `EveTr`, `DecTr`, `Grass` (unirrigated)
SUEWS_Water.txt	*MD*	Not currently used for water surface

DrainageCoef2

Description Coefficient b [-] used in `DrainageEq`

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MD*	• Example values: – `DrainageEq` = 3, 3 for `Paved` and `Bldgs` – `DrainageEq` = 2, 1.71 for `BSoil`
SUEWS_Veg.txt	*MD*	• Example values: – `DrainageEq` = 3, 3 for `Grass` (irrigated) – `DrainageEq` = 2, 1.71 for `EveTr`, `DecTr`, `Grass` (unirrigated)
SUEWS_Water.txt	*MD*	Not currently used for water surface

DrainageEq

Description Calculation choice for Drainage equation

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MD*	• Options: – 1: Falk and Niemczynowicz [1978] – 2: Halldin *et al.* [1979] (Rutter eqn corrected for c=0, see Calder and Wright [1986]) – 3: for `BSoil` [Falk and Niemczynowicz, 1978] ; for `Paved` and `Bldgs` Coefficients are specified in the following two columns. **Recommended in this version.**
SUEWS_Veg.txt	*MD*	• Options: – 1: Falk and Niemczynowicz [1978] – 2: Halldin *et al.* [1979] (Rutter eqn corrected for c=0, see Calder & Wright (1986) [Calder and Wright, 1986]) – 3: for `EveTr`, `DecTr`, `Grass` (unirrigated) see Falk and Niemczynowicz [1978]. Coefficients are specified in the following two columns. **Recommended in this version.**
SUEWS_Water.txt	*MD*	Not currently used for water surface.

`EF_umolCO2perJ`

Description Emission factor for fuels used for building heating.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

Emissivity

Description Effective surface emissivity.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MU*	Effective surface emissivity. View factors should be taken into account.
SUEWS_Veg.txt	*MU*	Example values [-] • 0.98 EveTr [Oke, 2002] • 0.98 DecTr [Oke, 2002] • 0.93 Grass [Oke, 2002]
SUEWS_Water.txt	*MU*	Example values [-] • 0.95 Water [Oke, 2002]
SUEWS_Snow.txt	*MU*	Example values [-] • 0.99 [Järvi *et al.*, 2014]

EndDLS

Description End of the day light savings [DOY]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	End of the day light savings [DOY] See *Day Light Savings (DLS)*.

EnEF_v_Jkm

Description Emission factor for heat [J k m^{-1}].

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	Emission factor for heat [J k m^{-1}]. Example values: 3.97e6 Sailor and Lu (2004) [Sailor and Lu, 2004]

EnergyUseProfWD

Description Code linking to EnergyUseProfWD in *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*L*	Code for energy use profile (weekdays) Provides the link to column 1 of *SUEWS_Profiles.txt*. Look the codes Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*.

EnergyUseProfWE

Description Code linking to EnergyUseProfWE in *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*L*	Code for energy use profile (weekends) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*.

ESTMCode

Description Code for ESTM coefficients linking to *SUEWS_ESTMCoefficients.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*L*	For paved and building surfaces, it is possible to specify multiple codes per grid (3 for paved, 5 for buildings) using *SUEWS_SiteSelect.txt* . In this case, set ESTMCode here to zero.

continues on next page

Table 7.83 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*L*	Code for ESTM coefficients to use for this surface. Links to SUEWS_ESTMCoefficients.txt . Value of integer is arbitrary but must match code specified in column 1 of SUEWS_ESTMCoefficients.txt.
SUEWS_Water.txt	*L*	Code for ESTM coefficients to use for this surface. Links to SUEWS_ESTMCoefficients.txt . Value of integer is arbitrary but must match code specified in column 1 of SUEWS_ESTMCoefficients.txt.
SUEWS_Snow.txt	*L*	For paved and building surfaces, it is possible to specify multiple codes per grid (3 for paved, 5 for buildings) using *SUEWS_SiteSelect.txt* . In this case, set ESTM code here to zero.

FAI_Bldgs

Description Frontal area index for buildings [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Frontal area index for buildings [-] Required if `RoughLenMomMethod` = 3 in *RunControl.nml* .

FAI_DecTr

Description Frontal area index for deciduous trees [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Frontal area index for deciduous trees [-] Required if `RoughLenMomMethod` = 3 in *RunControl.nml* .

FAI_EveTr

Description Frontal area index for evergreen trees [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Frontal area index for evergreen trees [-] Required if `RoughLenMomMethod` = 3 in *RunControl.nml* .

`Faut`

Description Fraction of irrigated area that is irrigated using automated systems

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Fraction of irrigated area that is irrigated using automated systems (e.g. sprinklers).

`FcEF_v_kgkmWD`

Description CO2 emission factor for weekdays [kg km^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	CO2 emission factor for weekdays [kg km^{-1}] Can be used for CO2 flux calculation.

`FcEF_v_kgkmWE`

Description CO2 emission factor for weekends [kg km^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	CO2 emission factor for weekdays [kg km^{-1}] Can be used for CO2 flux calculation.

`FcEF_v_Jkm`

Description Traffic emission factor for CO2.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*0*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

fcld

Description Cloud fraction [tenths]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*0*	Cloud fraction [tenths]

FlowChange

Description Difference in input and output flows for water surface [mm h^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Difference in input and output flows for water surface [mm h^{-1}] Used to indicate river or stream flow through the grid. Currently not fully tested!

Fraction1of8

Description Fraction of water that can flow to `GridConnection1of8` [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

Fraction2of8

Description Fraction of water that can flow to `GridConnection2of8` [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

`Fraction3of8`

Description Fraction of water that can flow to *GridConnection3of8* [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

`Fraction4of8`

Description Fraction of water that can flow to *GridConnection4of8* [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

`Fraction5of8`

Description Fraction of water that can flow to *GridConnection5of8* [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

`Fraction6of8`

Description Fraction of water that can flow to *GridConnection6of8* [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

`Fraction7of8`

Description Fraction of water that can flow to *GridConnection7of8* [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

Fraction8of8

Description Fraction of water that can flow to `GridConnection8of8` [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of water that can flow to the grid specified in previous column [-]

Fr_Bldgs

Description Surface cover fraction of buildings [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Surface cover fraction of buildings [-]

Fr_Bsoil

Description Surface cover fraction of bare soil or unmanaged land [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Surface cover fraction of bare soil or unmanaged land [-]

Fr_DecTr

Description Surface cover fraction of deciduous trees and shrubs [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Surface cover fraction of deciduous trees and shrubs [-]

Fr_ESTMClass_Bldgs1

Description Surface cover fraction of building class 1 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 94-98 must add up to 1

`Fr_ESTMClass_Bldgs2`

Description Surface cover fraction of building class 2 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 94-98 must add up to 1

`Fr_ESTMClass_Bldgs3`

Description Surface cover fraction of building class 3 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 94-98 must add up to 1

`Fr_ESTMClass_Bldgs4`

Description Surface cover fraction of building class 4 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 94-98 must add up to 1

`Fr_ESTMClass_Bldgs5`

Description Surface cover fraction of building class 5 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 94-98 must add up to 1

`Fr_ESTMClass_Paved1`

Description Surface cover fraction of *Paved* surface class 1 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 88-90 must add up to 1

Fr_ESTMClass_Paved2

Description Surface cover fraction of Paved surface class 2 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 88-90 must add up to 1

Fr_ESTMClass_Paved3

Description Surface cover fraction of Paved surface class 3 used in ESTM calculations

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 88-90 must add up to 1

Fr_EveTr

Description Surface cover fraction of EveTr: evergreen trees and shrubs [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Surface cover fraction of evergreen trees and shrubs [-]

Fr_Grass

Description Surface cover fraction of Grass [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Surface cover fraction of grass [-]

Fr_Paved

Description Surface cover fraction of Paved surfaces [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Columns 14 to 20 must sum to 1 .

Fr_Water

Description Surface cover fraction of open water [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Surface cover fraction of open water [-] (e.g. river, lakes, ponds, swimming pools)

FrFossilFuel_Heat

Description Fraction of fossil fuels used for building heating [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

FrFossilFuel_NonHeat

Description Fraction of fossil fuels used for building energy use [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

FrPDDwe

Description Fraction of weekend population to weekday population. [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Fraction of weekend population to weekday population. [-]

G1

Description Related to maximum surface conductance [mm s^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to maximum surface conductance [mm s^{-1}]

G2

Description Related to Kdown dependence [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to Kdown dependence [W m^{-2}]

G3

Description Related to VPD dependence [units depend on `gsModel`]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to VPD dependence [units depend on gsChoice in *RunControl.nml*]

G4

Description Related to VPD dependence [units depend on `gsModel`]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to VPD dependence [units depend on gsChoice in *RunControl.nml*]

G5

Description Related to temperature dependence [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to temperature dependence [°C]

G6

Description Related to soil moisture dependence [mm^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to soil moisture dependence [mm^{-1}]

gamq_gkgm

Description vertical gradient of specific humidity [g kg^{-1} m^{-1}]

Configuration

Referencing Table	Requirement	Comment
CBL_initial_data.txt	*MU*	vertical gradient of specific humidity (g kg^{-1} m^{-1})

gamt_Km

Description vertical gradient of potential temperature [K m^{-1}]

Configuration

Referencing Table	Requirement	Comment
CBL_initial_data.txt	*MU*	vertical gradient of potential temperature (K m^{-1}) strength of the inversion

GDDFull

Description The growing degree days (GDD) needed for full capacity of the leaf area index (LAI) [°C].

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MU*	This should be checked carefully for your study area using modelled LAI from the DailyState output file compared to known behaviour in the study area. See section 2.2 Järvi *et al.* [2011] ; Appendix A Järvi *et al.* [2014] for more details. Example values: 300 for `EveTr` Järvi *et al.* [2011]

Grid

Description a unique number to represent grid

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Grid numbers do not need to be consecutive and do not need to start at a particular value. Each grid must have a unique grid number. All grids must be present for all years. These grid numbers are referred to in GridConnections (columns 64-79) (N.B. **Not available in this version.**)

GridConnection1of8

Description Number of the 1st grid where water can flow to The next 8 pairs of columns specify the water flow between grids. The first column of each pair specifies the grid that the water flows to (from the current grid, column 1); the second column of each pair specifies the fraction of water that flow to that grid. The fraction (i.e. amount) of water transferred may be estimated based on elevation, the length of connecting surface between grids, presence of walls, etc. Water cannot flow from the current grid to the same grid, so the grid number here must be different to the grid number in column 1. Water can flow to a maximum of 8 other grids. If there is no water flow between grids, or a single grid is run, set to 0. See section on Grid Connections

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	The next 8 pairs of columns specify the water flow between grids. The first column of each pair specifies the grid that the water flows to (from the current grid, column 1); the second column of each pair specifies the fraction of water that flow to that grid. The fraction (i.e. amount) of water transferred may be estimated based on elevation, the length of connecting surface between grids, presence of walls, etc. Water cannot flow from the current grid to the same grid, so the grid number here must be different to the grid number in column 1. Water can flow to a maximum of 8 other grids. If there is no water flow between grids, or a single grid is run, set to 0. See section on Grid Connections

`GridConnection2of8`

Description Number of the 2nd grid where water can flow to

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Number of the grid where water can flow to

`GridConnection3of8`

Description Number of the 3rd grid where water can flow to

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Number of the grid where water can flow to

`GridConnection4of8`

Description Number of the 4th grid where water can flow to

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Number of the grid where water can flow to

GridConnection5of8

Description Number of the 5th grid where water can flow to

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Number of the grid where water can flow to

GridConnection6of8

Description Number of the 6th grid where water can flow to

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Number of the grid where water can flow to

GridConnection7of8

Description Number of the 7th grid where water can flow to

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Number of the grid where water can flow to

GridConnection8of8

Description Number of the 8th grid where water can flow to

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Number of the grid where water can flow to

gsModel

Description Formulation choice for conductance calculation.

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	• 1 [Järvi *et al.*, 2011] • 2 [Ward *et al.*, 2016] **Recommended in this version.**

`H_Bldgs`

Description Mean building height [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Mean building height [m]

`H_DecTr`

Description Mean height of deciduous trees [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Mean height of deciduous trees [m]

`H_EveTr`

Description Mean height of evergreen trees [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Mean height of evergreen trees [m]

`H_maintain`

Description water depth to maintain used in automatic irrigation (e.g., ponding water due to flooding irrigation in rice crop-field) [mm].

Note:

1. *H_maintain* can be positive (e.g., ponding water due to flooding irrigation in rice crop-field) or negative (e.g., soil water store level to maintain: *SoilStoreCap* + *H_maintain*) or zero (e.g., to maintain a maximum soil store level, i.e., *SoilStoreCap*).

2. Disable this feature by setting this parameter to `-999`: then no restrictions will be applied to maintain available water level.

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	water depth to maintain used in automatic irrigation.

`id`

Description Day of year [DOY]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Not used: set to 1 in this version.
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Day of year [DOY]
SSss_YYYY_data_tt.txt	*MU*	Day of year [DOY]
CBL_initial_data.txt	*MU*	Day of year [DOY]

`Ie_a1`

Description Coefficient for automatic irrigation model [mm d^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MD*	Coefficient for automatic irrigation model [mm d -1]

`Ie_a2`

Description Coefficient for automatic irrigation model [mm d^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MD*	Coefficient for automatic irrigation model [mm d -1 K^{-1}]

`Ie_a3`

Description Coefficient for automatic irrigation model [mm d^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MD*	Coefficient for automatic irrigation model [mm d -2]

Ie_end

Description Day when irrigation ends [DOY]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Day when irrigation ends [DOY]

Ie_m1

Description Coefficient for manual irrigation model [mm d^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MD*	Coefficient for manual irrigation model [mm d -1]

Ie_m2

Description Coefficient for manual irrigation model [mm d^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MD*	Coefficient for manual irrigation model [mm d -1 K^{-1}]

Ie_m3

Description Coefficient for manual irrigation model [mm d^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MD*	Coefficient for manual irrigation model [mm d -2]

Ie_start

Description Day when irrigation starts [DOY]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Day when irrigation starts [DOY]

ih

Description Hour [H]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Hour [H] Not used: set to 0 in this version.

imin

Description Minute [M]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Minute [M] Not used: set to 0 in this version.
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Minute [M]
SSss_YYYY_data_tt.txt	*MU*	Minute [M]

InfiltrationRate

Description Infiltration rate.

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*O*	Not currently used

Internal_albedo

Description Albedo of all internal elements for building surfaces only

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Albedo of all internal elements for building surfaces only

Internal_CHbld

Description Bulk transfer coefficient of internal building elements [W m^{-2} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*0*	Bulk transfer coefficient of internal building elements [W m^{-2} K^{-1}] (for building surfaces only and if `IbldCHmod` == 0 in *ESTMinput.nml*

`Internal_CHroof`

Description Bulk transfer coefficient of internal roof [W m^{-2} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*0*	Bulk transfer coefficient of internal roof [W m^{-2} K^{-1}] (for building surfaces only and if `IbldCHmod` == 0 in *ESTMinput.nml*

`Internal_CHwall`

Description Bulk transfer coefficient of internal wall [W m^{-2} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*0*	Bulk transfer coefficient of internal wall [W m^{-2} K^{-1}] (for building surfaces only and if `IbldCHmod` == 0 in *ESTMinput.nml*

`Internal_emissivity`

Description Emissivity of all internal elements for building surfaces only

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Emissivity of all internal elements for building surfaces only

`Internal_k1`

Description Thermal conductivity of the first layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Thermal conductivity of the first layer [W m^{-1} K^{-1}]

Internal_k2

Description Thermal conductivity of the second layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the second layer [W m^{-1} K^{-1}]

Internal_k3

Description Thermal conductivity of the third layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the third layer [W m^{-1} K^{-1}]

Internal_k4

Description Thermal conductivity of the fourth layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the fourth layer [W m^{-1} K^{-1}]

Internal_k5

Description Thermal conductivity of the fifth layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the fifth layer [W m^{-1} K^{-1}]

Internal_rhoCp1

Description Volumetric heat capacity of the first layer[J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Volumetric heat capacity of the first layer[J m^{-3} K^{-1}]

`Internal_rhoCp2`

Description Volumetric heat capacity of the second layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the second layer [J m^{-3} K^{-1}]

`Internal_rhoCp3`

Description Volumetric heat capacity of the third layer[J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the third layer[J m^{-3} K^{-1}]

`Internal_rhoCp4`

Description Volumetric heat capacity of the fourth layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the fourth layer [J m^{-3} K^{-1}]

`Internal_rhoCp5`

Description Volumetric heat capacity of the fifth layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the fifth layer [J m^{-3} K^{-1}]

`Internal_thick1`

Description Thickness of the first layer [m] for building surfaces only

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Thickness of the first layer [m] for building surfaces only; set to -999 for all other surfaces

Internal_thick2

Description Thickness of the second layer [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the second layer [m] (if no second layer, set to -999.)

Internal_thick3

Description Thickness of the third layer [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the third layer [m] (if no third layer, set to -999.)

Internal_thick4

Description Thickness of the fourth layer [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the fourth layer [m] (if no fourth layer, set to -999.)

Internal_thick5

Description Thickness of the fifth layer [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the fifth layer [m] (if no fifth layer, set to -999.)

`InternalWaterUse`

Description Internal water use [mm h^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Irrigation.txt	*MU*	Internal water use [mm h^{-1}]

`IrrFr_Paved`

Description Fraction of `Paved` that is irrigated [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Fraction of paved surfaces that are irrigated [-]

`IrrFr_Bldgs`

Description Fraction of `Bldgs` that is irrigated [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Fraction of rooftop of buildings (e.g., green roofs) that are irrigated [-]

`IrrFr_DecTr`

Description Fraction of `DecTr` that is irrigated [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Fraction of deciduous trees that are irrigated [-]

`IrrFr_EveTr`

Description Fraction of `EveTr` that is irrigated [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Fraction of evergreen trees that are irrigated [-] e.g. 50% of the evergreen trees/shrubs are irrigated

`IrrFr_Grass`

Description Fraction of `Grass` that is irrigated [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Fraction of grass that is irrigated [-]

`IrrFr_BSoil`

Description Fraction of `BSoil` that is irrigated [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Fraction of bare soil that are irrigated [-]

`IrrFr_Water`

Description Fraction of `Water` that is irrigated [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Fraction of water that are irrigated [-]

`IrrigationCode`

Description Code for modelling irrigation linking to `Code` of *SUEWS_Irrigation.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for modelling irrigation Provides the link to column 1 of SUEWS_Irrigation.txt, which contains the model coefficients for estimation of the water use (used if WU_Choice = 0 in *RunControl.nml*). Value of integer is arbitrary but must match code specified in column 1 of SUEWS_Irrigation.txt.

it

Description Hour [H]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Hour [H]
SSss_YYYY_data_tt.txt	*MU*	Hour [H]

iy

Description Year [YYYY]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Year [YYYY]
SSss_YYYY_data_tt.txt	*MU*	Year [YYYY]

kdiff

Description Diffuse radiation [W m^{-2}].

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Recommended if SOLWEIGUse = 1

kdir

Description Direct radiation [W m^{-2}].

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Recommended if SOLWEIGUse = 1

kdown

Description Incoming shortwave radiation [W m^{-2}].

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*MU*	Must be > 0 W m^{-2} .

Kmax

Description Maximum incoming shortwave radiation [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Maximum incoming shortwave radiation [W m^{-2}]

lai

Description Observed leaf area index [m^{-2} m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Observed leaf area index [m^{-2} m^{-2}]

LAIEq

Description LAI calculation choice.

Note: North and South hemispheres are treated slightly differently.

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	Coefficients are specified in the following parameters: `LeafGrowthPower1`, `LeafGrowthPower2`, `LeafOffPower1` and `LeafOffPower2`. Options • 0 Järvi *et al.* [2011] • 1 Järvi *et al.* [2014]

LAIMax

Description full leaf-on summertime value

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	full leaf-on summertime value Example values: - 5.1 EveTr Breuer et al. (2003) [Breuer *et al.*, 2003] - 5.5 DecTr Breuer et al. (2003) [Breuer *et al.*, 2003] - 5.9 Grass Breuer et al. (2003) [Breuer *et al.*, 2003]

LAIMin

Description leaf-off wintertime value

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	leaf-off wintertime value Example values: - 4. EveTr [Järvi *et al.*, 2011] - 1. DecTr [Järvi *et al.*, 2011] - 1.6 Grass [Grimmond and Oke, 1991]

lat

Description Latitude [deg].

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Use coordinate system WGS84. Positive values are northern hemisphere (negative southern hemisphere). Used in radiation calculations. Note, if the total modelled area is small the latitude and longitude could be the same for each grid but small differences in radiation will not be determined. If you are defining the latitude and longitude differently between grids make certain that you provide enough decimal places.

`ldown`

Description Incoming longwave radiation [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Incoming longwave radiation [W m^{-2}]

`LeafGrowthPower1`

Description a parameter required by LAI calculation in `LAIEq`

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	Example values • `LAIEq` = 0: 0.03 [Järvi *et al.*, 2011] • `LAIEq` = 1: 0.04 [Järvi *et al.*, 2014]

`LeafGrowthPower2`

Description a parameter required by LAI calculation [K^{-1}] in `LAIEq`

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	Example values • *LAIEq* = 0: 0.0005 [Järvi *et al.*, 2011] • *LAIEq* = 1: 0.001 [Järvi *et al.*, 2014]

LeafOffPower1

Description a parameter required by LAI calculation [K^{-1}] in *LAIEq*

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	Example values • *LAIEq* = 0: 0.03 [Järvi *et al.*, 2011] • *LAIEq* = 1: -1.5 [Järvi *et al.*, 2014]

LeafOffPower2

Description a parameter required by LAI calculation [K^{-1}] in *LAIEq*

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	Example values • *LAIEq* = 0: 0.0005 [Järvi *et al.*, 2011] • *LAIEq* = 1: 0.0015 [Järvi *et al.*, 2014]

lng

Description longitude [deg]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Use coordinate system WGS84. For compatibility with GIS, negative values are to the west, positive values are to the east (e.g. Vancouver = -123.12; Shanghai = 121.47) Note this is a change of sign convention between v2016a and v2017a See latitude for more details.

LUMPS_Cover

Description Limit when surface totally covered with water for LUMPS [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Limit when surface totally covered with water [mm] Used for LUMPS surface wetness control. Default recommended value of 1 mm from Loridan *et al.* [2011] .

LUMPS_DrRate

Description Drainage rate of bucket for LUMPS [mm h^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Drainage rate of bucket for LUMPS [mm h^{-1}] Used for LUMPS surface wetness control. Default recommended value of 0.25 mm h^{-1} from Loridan *et al.* [2011] .

LUMPS_MaxRes

Description Maximum water bucket reservoir [mm] Used for LUMPS surface wetness control.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Maximum water bucket reservoir [mm] Used for LUMPS surface wetness control. Default recommended value of 10 mm from Loridan *et al.* [2011] .

MaxQFMetab

Description Maximum value for human heat emission. [W m^{-2}]

Example values: 175 Sailor and Lu (2004) [Sailor and Lu, 2004]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	Maximum value for human heat emission. [W m^{-2}]

MaxFCMetab

Description Maximum (day) CO2 from human metabolism. [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	Maximum (day) CO2 from human metabolism. [W m^{-2}]

MaxConductance

Description The maximum conductance of each vegetation or surface type. [mm s^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	Example values [mm s^{-1}] • 7.4: EveTr [Järvi *et al.*, 2011] • 11.7: DecTr [Järvi *et al.*, 2011] • 33.1: Grass (unirrigated) [Järvi *et al.*, 2011] • 40.: Grass (irrigated) [Järvi *et al.*, 2011]

MinQFMetab

Description Minimum value for human heat emission. [W m^{-2}]

Example values: 75 Sailor and Lu (2004) [Sailor and Lu, 2004]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	Minimum value for human heat emission. [W m^{-2}].

`MinFCMetab`

Description Minimum (night) CO2 from human metabolism. [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	Minimum (night) CO2 from human metabolism. [W m^{-2}]

`NARP_Trans`

Description Atmospheric transmissivity for NARP [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD*	Atmospheric transmissivity for NARP [-] Value must in the range 0-1. Default recommended value of 1.

`nroom`

Description Number of rooms per floor for building surfaces only [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Number of rooms per floor for building surfaces only

`OBS_SMCap`

Description The maximum observed soil moisture. [m^3 m^{-3} or kg kg^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*O*	Use only if soil moisture is observed and provided in the met forcing file and `SMDMethod` = 1 or 2. Use of observed soil moisture not currently tested

`OBS_SMDepth`

Description The depth of soil moisture measurements. [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*O*	Use only if soil moisture is observed and provided in the met forcing file and `SMDMethod` = 1 or 2. Use of observed soil moisture not currently tested

`OBS_SoilNotRocks`

Description Fraction of soil without rocks. [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*O*	Use only if soil moisture is observed and provided in the met forcing file and `SMDMethod` = 1 or 2. Use of observed soil moisture not currently tested

`OHMCode_SummerDry`

Description Code for OHM coefficients to use for this surface during dry conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.

continues on next page

Table 7.213 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Water.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Snow.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.

`OHMCode_SummerWet`

Description Code for OHM coefficients to use for this surface during wet conditions in summer, linking to *SUEWS_OHMCoefficients.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Veg.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.

continues on next page

Table 7.214 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Water.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Snow.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in summer. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.

`OHMCode_WinterDry`

Description Code for OHM coefficients to use for this surface during dry conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Veg.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Water.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.

continues on next page

Table 7.215 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*L*	Code for OHM coefficients to use for this surface during dry conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.

OHMCode_WinterWet

Description Code for OHM coefficients to use for this surface during wet conditions in winter, linking to *SUEWS_OHMCoefficients.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Veg.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Water.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.
SUEWS_Snow.txt	*L*	Code for OHM coefficients to use for this surface during wet conditions in winter. Links to *SUEWS_OHMCoefficients.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_OHMCoefficients.txt*.

OHMThresh_SW

Description Temperature threshold determining whether summer/winter OHM coefficients are applied [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	MD	Temperature threshold determining whether summer/winter OHM coefficients are applied [°C] If 5-day running mean air temperature is greater than or equal to this threshold, OHM coefficients for summertime are applied; otherwise coefficients for wintertime are applied.
SUEWS_Veg.txt	MD	Temperature threshold determining whether summer/winter OHM coefficients are applied [°C] If 5-day running mean air temperature is greater than or equal to this threshold, OHM coefficients for summertime are applied; otherwise coefficients for wintertime are applied.
SUEWS_Water.txt	MD	Temperature threshold determining whether summer/winter OHM coefficients are applied [°C] If 5-day running mean air temperature is greater than or equal to this threshold, OHM coefficients for summertime are applied; otherwise coefficients for wintertime are applied.
SUEWS_Snow.txt	MD	Not actually used for Snow surface as winter wet conditions always assumed.

OHMThresh_WD

Description Soil moisture threshold determining whether wet/dry OHM coefficients are applied [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	MD	Not actually used for building and paved surfaces (as impervious).
SUEWS_Veg.txt	MD	Note that OHM coefficients for wet conditions are applied if the surface is wet.

continues on next page

Table 7.218 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Water.txt	*MD*	Not actually used for water surface (as no soil surface beneath).
SUEWS_Snow.txt	*MD*	Not actually used for Snow surface as winter wet conditions always assumed.

PipeCapacity

Description Storage capacity of pipes [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Storage capacity of pipes [mm] Runoff amounting to less than the value specified here is assumed to be removed by pipes.

PopDensDay

Description Daytime population density (i.e. workers, tourists) [people ha^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Daytime population density (i.e. workers, tourists) [people ha -1] Population density is required if EmissionsMethod = 2 in *RunControl.nml* . The model will use the average of daytime and night-time population densities, unless only one is provided. If daytime population density is unknown, set to -999.

PopDensNight

Description Night-time population density (i.e. residents) [people ha^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Night-time population density (i.e. residents) [people ha -1] Population density is required if EmissionsMethod = 2 in *RunControl.nml* . The model will use the average of daytime and night-time population densities, unless only one is provided. If night-time population density is unknown, set to -999.

`PopProfWD`

Description Code for population density profile (weekdays) linking to `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	Code for population density profile (weekdays).

`PopProfWE`

Description Code for population density profile (weekends) linking to `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*O*	Code for population density profile (weekends)

`PorosityMax`

Description full leaf-on summertime value Used only for `DecTr` (can affect roughness calculation)

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	full leaf-on summertime value Used only for DecTr (can affect roughness calculation)

`PorosityMin`

Description leaf-off wintertime value Used only for `DecTr` (can affect roughness calculation)

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MD*	leaf-off wintertime value Used only for DecTr (can affect roughness calculation)

PrecipLimAlb

Description Limit for hourly precipitation when the ground is fully covered with snow [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MD*	Limit for hourly precipitation when the ground is fully covered with snow. Then snow albedo is reset to AlbedoMax [mm]

PrecipLimSnow

Description Temperature limit when precipitation falls as snow [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MD*	Auer [1974]

pres

Description Barometric pressure [kPa]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*MU*	Barometric pressure [kPa]

qe

Description Latent heat flux [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Latent heat flux [W m^{-2}]

qf

Description Anthropogenic heat flux [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Anthropogenic heat flux [W m^{-2}]

QF_A_WD

Description Base value for QF on weekdays [W m^{-2} (Cap ha^{-1} $)^{-1}$]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 2 Example values: • 0.3081 [Järvi *et al.*, 2011] • 0.1 [Järvi *et al.*, 2014]

QF_A_WE

Description Base value for QF on weekends [W m^{-2} (Cap ha^{-1} $)^{-1}$]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 2 Example values: • 0.3081 [Järvi *et al.*, 2011] • 0.1 [Järvi *et al.*, 2014]

QF_B_WD

Description Parameter related to cooling degree days on weekdays [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 2 Example values: • 0.0099 [Järvi *et al.*, 2011] • 0.0099 [Järvi *et al.*, 2014]

QF_B_WE

Description Parameter related to cooling degree days on weekends [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 2 Example values: • 0.0099 [Järvi *et al.*, 2011] • 0.0099 [Järvi *et al.*, 2014]

QF_C_WD

Description Parameter related to heating degree days on weekdays [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 2 Example values: • 0.0102 [Järvi *et al.*, 2011] • 0.0102 [Järvi *et al.*, 2014]

QF_C_WE

Description Parameter related to heating degree days on weekends [W m^{-2} K^{-1} (Cap ha^{-1} $)^{-1}$]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Example values: • 0.0102 [Järvi *et al.*, 2011] • 0.0102 [Järvi *et al.*, 2014]

q+_gkg

Description specific humidity at the top of CBL [g kg^{-1}]

Configuration

Referencing Table	Requirement	Comment
CBL_initial_data.txt	*MU*	specific humidity at the top of CBL (g kg^{-1})

q_gkg

Description specific humidiy in CBL [g kg^{-1}]

Configuration

Referencing Table	Requirement	Comment
CBL_initial_data.txt	*MU*	specific humidiy in CBL (g kg^{-1})

qh

Description Sensible heat flux [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Sensible heat flux [W m^{-2}]

qn

Description Net all-wave radiation [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Required if `NetRadiationMethod` = 1.

qs

Description Storage heat flux [W m^{-2}]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Storage heat flux [W m^{-2}]

RadMeltFactor

Description Hourly radiation melt factor of snow [mm W^{-1} h^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MU*	Hourly radiation melt factor of snow [mm W^{-1} h^{-1}]

rain

Description Rainfall [mm]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*MU*	Rainfall [mm]

RH

Description Relative Humidity [%]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*MU*	Relative Humidity [%]

RunoffToWater

Description Fraction of above-ground runoff flowing to water surface during flooding [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MD MU*	Fraction of above-ground runoff flowing to water surface during flooding [-] Value must be in the range 0-1. Fraction of above-ground runoff that can flow to the water surface in the case of flooding.

S1

Description A parameter related to soil moisture dependence [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to soil moisture dependence [-] These will change in the future to ensure consistency with soil behaviour

S2

Description A parameter related to soil moisture dependence [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Related to soil moisture dependence [mm] These will change in the future to ensure consistency with soil behaviour

SatHydraulicCond

Description Hydraulic conductivity for saturated soil [mm s^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*MD*	Hydraulic conductivity for saturated soil [mm s^{-1}]

SDDFull

Description The sensesence degree days (SDD) needed to initiate leaf off. [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*MU*	This should be checked carefully for your study area using modelled LAI from the DailyState output file compared to known behaviour in the study area. See section 2.2 of Järvi *et al.* [2011] and Appendix A of Järvi *et al.* [2014] for more details. Example values: • -450: EveTr [Järvi *et al.*, 2011] • -450: DecTr [Järvi *et al.*, 2011] • -450: Grass [Järvi *et al.*, 2011]

snow

Description Snowfall [mm]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Required if SnowUse = 1

`SnowClearingProfWD`

Description Code for snow clearing profile (weekdays) linking to `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for snow clearing profile (weekdays) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*. e.g. 1 means use the characteristics specified in the row of input file SUEWS_Profiles.txt which has 1 in column 1 (Code).

`SnowClearingProfWE`

Description Code for snow clearing profile (weekends) linking to `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for snow clearing profile (weekends) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*. e.g. 1 means use the characteristics specified in the row of input file SUEWS_Profiles.txt which has 1 in column 1 (Code). Providing the same code for `SnowClearingProfWD` and `SnowClearingProfWE` would link to the same row in *SUEWS_Profiles.txt*, i.e. the same profile would be used for weekdays and weekends.

`SnowCode`

Description Code for snow surface characteristics linking to `Code` of SUEWS_Snow.txt

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for snow surface characteristics Provides the link to column 1 of SUEWS_Snow.txt, which contains the attributes describing snow surfaces in this grid for this year. Value of integer is arbitrary but must match code specified in column 1 of SUEWS_Snow.txt.

`SnowDensMax`

Description Maximum snow density [kg m^{-3}]

Configuration

`SnowDensMin`

Description Fresh snow density [kg m^{-3}]

Configuration

`SnowLimPatch`

Description Limit for the snow water equivalent when snow cover starts to be patchy [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*O*	Limit of snow water equivalent when the surface is fully covered with snow. Not needed if `SnowUse` = 0 in *RunControl.nml* . Example values: • 190: Paved [Järvi *et al.*, 2014] • 190: Bldgs [Järvi *et al.*, 2014] • 190: BSoil [Järvi *et al.*, 2014]

continues on next page

Table 7.254 – continued from previous page

Referencing Table	Requirement	Comment
SUEWS_Veg.txt	*0*	Limit of snow water equivalent when the surface is fully covered with snow. Not needed if `SnowUse` = 0 in *RunControl.nml* . Example values: • 190: EveTr [Järvi *et al.*, 2014] • 190: DecTr [Järvi *et al.*, 2014] • 190: Grass [Järvi *et al.*, 2014]

SnowLimRemove

Description Limit of the snow water equivalent for snow removal from roads and roofs [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*0*	Not needed if `SnowUse` = 0 in *RunControl.nml* . **Not available in this version.** Example values [mm] • 40: `Paved` [Järvi *et al.*, 2014] • 100: `Bldgs` [Järvi *et al.*, 2014]

SoilDensity

Description Soil density [kg m^{-3}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*MD*	Soil density [kg m^{-3}]

SoilDepth

Description Depth of soil beneath the surface [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*MD*	Depth of sub-surface soil store [mm] i.e. the depth of soil beneath the surface

`SoilStoreCap`

Description Limit value for `SoilDepth` [mm]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Soil.txt	*MD*	SoilStoreCap must not be greater than SoilDepth.

`SoilTypeCode`

Description Code for soil characteristics below this surface linking to `Code` of *SUEWS_Soil.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*L*	Code for soil characteristics below this surface Provides the link to column 1 of *SUEWS_Soil.txt*, which contains the attributes describing sub-surface soil for this surface type. Value of integer is arbitrary but must match code specified in column 1 of SUEWS_Soil.txt.
SUEWS_Veg.txt	*L*	Code for soil characteristics below this surface Provides the link to column 1 of *SUEWS_Soil.txt*, which contains the attributes describing sub-surface soil for this surface type. Value of integer is arbitrary but must match code specified in column 1 of SUEWS_Soil.txt.

`StartDLS`

Description Start of the day light savings [DOY]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Start of the day light savings [DOY] See *Day Light Savings (DLS)*.

`StateLimit`

Description Upper limit to the surface state. [mm]

Currently only used for the water surface. Set to a large value (e.g. 20000 mm = 20 m) if the water body is substantial (lake, river, etc) or a small value (e.g. 10 mm) if water bodies are very shallow (e.g. fountains). WaterDepth (column 9) must not exceed this value.

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MD*	Currently only used for the water surface
SUEWS_Veg.txt	*MD*	Currently only used for the water surface
SUEWS_Water.txt	*MU*	Surface state cannot exceed this value. Set to a large value (e.g. 20000 mm = 20 m) if the water body is substantial (lake, river, etc) or a small value (e.g. 10 mm) if water bodies are very shallow (e.g. fountains). WaterDepth (column 9) must not exceed this value.

`StorageMax`

Description Maximum water storage capacity for upper surfaces (i.e. canopy)

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MD*	Maximum water storage capacity for upper surfaces (i.e. canopy) Min and max values are to account for seasonal variation (e.g. leaf-on/leaf-off differences for vegetated surfaces). Not currently used for non-vegetated surfaces - set the same as `StorageMin`. Example values: • 0.48 `Paved` • 0.25 `Bldgs` • 0.8 `BSoil`
SUEWS_Veg.txt	*MD*	Maximum water storage capacity for upper surfaces (i.e. canopy) Min/max values are to account for seasonal variation (e.g. leaf-off/leaf-on differences for vegetated surfaces) Only used for `DecTr` surfaces - set `EveTr` and `Grass` values the same as `StorageMin`. Example values: • 1.3: `EveTr` [Breuer *et al.*, 2003] • 0.8: `DecTr` [Breuer *et al.*, 2003] • 1.9: `Grass` [Breuer *et al.*, 2003]
SUEWS_Water.txt	*MD*	Maximum water storage capacity for upper surfaces (i.e. canopy) Min and max values are to account for seasonal variation - not used for water surfaces so set same as `StorageMin`.

`StorageMin`

Description Minimum water storage capacity for upper surfaces (i.e. canopy).

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	`MD`	Minimum water storage capacity for upper surfaces (i.e. canopy). Min/max values are to account for seasonal variation (e.g. leaf-on/leaf-off differences for vegetated surfaces). Not currently used for non-vegetated surfaces - set the same as `StorageMax`. Example values: • 0.48 `Paved` • 0.25 `Bldgs` • 0.8 `BSoil`
SUEWS_Veg.txt	`MD`	Minimum water storage capacity for upper surfaces (i.e. canopy). Min/max values are to account for seasonal variation (e.g. leaf-off/leaf-on differences for vegetated surfaces). Example values: • 1.3 `EveTr` [Breuer *et al.*, 2003] • 0.3 `DecTr` [Breuer *et al.*, 2003] • 1.9 `Grass` [Breuer *et al.*, 2003]
SUEWS_Water.txt	`MD`	Minimum water storage capacity for upper surfaces (i.e. canopy). Min/max values are to account for seasonal variation - not used for water surfaces. Example values: -0.5 `Water`

`SurfaceArea`

Description Area of the grid [ha].

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	`MU`	Area of the grid [ha].

`Surf_k1`

Description Thermal conductivity of the first layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Thermal conductivity of the first layer [W m^{-1} K^{-1}]

Surf_k2

Description Thermal conductivity of the second layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the second layer [W m^{-1} K^{-1}]

Surf_k3

Description Thermal conductivity of the third layer[W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the third layer[W m^{-1} K^{-1}]

Surf_k4

Description Thermal conductivity of the fourth layer[W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the fourth layer[W m^{-1} K^{-1}]

Surf_k5

Description Thermal conductivity of the fifth layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the fifth layer [W m^{-1} K^{-1}]

Surf_rhoCp1

Description Volumetric heat capacity of the first layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Volumetric heat capacity of the first layer [J m^{-3} K^{-1}]

`Surf_rhoCp2`

Description Volumetric heat capacity of the second layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the second layer [J m^{-3} K^{-1}]

`Surf_rhoCp3`

Description Volumetric heat capacity of the third layer[J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the third layer[J m^{-3} K^{-1}]

`Surf_rhoCp4`

Description Volumetric heat capacity of the fourth layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the fourth layer [J m^{-3} K^{-1}]

`Surf_rhoCp5`

Description Volumetric heat capacity of the fifth layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the fifth layer [J m^{-3} K^{-1}]

Surf_thick1

Description Thickness of the first layer [m] for roofs (building surfaces) and ground (all other surfaces)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Thickness of the first layer [m] for roofs (building surfaces) and ground (all other surfaces)

Surf_thick2

Description Thickness of the second layer [m] (if no second layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the second layer [m] (if no second layer, set to -999.)

Surf_thick3

Description Thickness of the third layer [m] (if no third layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the third layer [m] (if no third layer, set to -999.)

Surf_thick4

Description Thickness of the fourth layer [m] (if no fourth layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the fourth layer [m] (if no fourth layer, set to -999.)

Surf_thick5

Description Thickness of the fifth layer [m] (if no fifth layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the fifth layer [m] (if no fifth layer, set to -999.)

`Tair`

Description Air temperature [°C]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*MU*	Air temperature [°C]

`tau_a`

Description Time constant for snow albedo aging in cold snow [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MD*	Time constant for snow albedo aging in cold snow [-]

`tau_f`

Description Time constant for snow albedo aging in melting snow [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MD*	Time constant for snow albedo aging in melting snow [-]

`tau_r`

Description Time constant for snow density ageing [-]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MD*	Time constant for snow density ageing [-]

`TCritic_Heating_WD`

Description Critical heating temperature on weekdays [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

TCritic_Heating_WE

Description Critical heating temperature on weekends [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

TCritic_Cooling_WD

Description Critical cooling temperature on weekdays [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

TCritic_Cooling_WE

Description Critical cooling temperature on weekends [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*MU O*	Use with `EmissionsMethod` = 1

TempMeltFactor

Description Hourly temperature melt factor of snow [mm K^{-1} h^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Snow.txt	*MU*	Hourly temperature melt factor of snow [mm K^{-1} h^{-1}] (In previous model version, this parameter was 0.12)

TH

Description Upper air temperature limit [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	MD	Upper air temperature limit [°C]

Theta+_K

Description potential temperature at the top of CBL [K]

Configuration

Referencing Table	Requirement	Comment
CBL_initial_data.txt	MU	potential temperature at the top of CBL (K)

Theta_K

Description potential temperature in CBL [K]

Configuration

Referencing Table	Requirement	Comment
CBL_initial_data.txt	MU	potential temperature in CBL (K)

Tiair

Description Indoor air temperature [C]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	MU	Indoor air temperature [C]

Timezone

Description Time zone [h] for site relative to UTC (east is positive). This should be set according to the times given in the meteorological forcing file(s).

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	MU	Time zone [h] for site relative to UTC (east is positive). This should be set according to the times given in the meteorological forcing file(s).

TL

Description Lower air temperature limit [°C]

Configuration

Referencing Table	Requirement	Comment
SUEWS_Conductance.txt	*MD*	Lower air temperature limit [°C]

ToBldgs

Description Fraction of water going to Bldgs

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to *Bldgs*

ToBSoil

Description Fraction of water going to BSoil

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to *BSoil*

ToDecTr

Description Fraction of water going to DecTr

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to *DecTr*

ToEveTr

Description Fraction of water going to *EveTr*

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to *EveTr*

ToGrass

Description Fraction of water going to *Grass*

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to `Grass`

`ToPaved`

Description Fraction of water going to `Paved`

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to `Paved`

`ToRunoff`

Description Fraction of water going to `Runoff`

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to `Runoff`

`ToSoilStore`

Description Fraction of water going to `SoilStore`

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to `SoilStore`

`ToWater`

Description Fraction of water going to `Water`

Configuration

Referencing Table	Requirement	Comment
SUEWS_WithinGridWaterDist.txt	*MU*	Fraction of water going to `Water`

`TraffProfWD`

Description Code for traffic activity profile (weekdays) linking to `Code` of *SUEWS_Profiles.txt*. Not used in v2018a.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*0*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

TraffProfWE

Description Code for traffic activity profile (weekends) linking to Code of *SUEWS_Profiles.txt*. Not used in v2018a.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*0*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

TrafficUnits

Description Units for the traffic rate for the study area. Not used in v2018a.

Configuration

Referencing Table	Requirement	Comment
SUEWS_AnthropogenicEmission.txt	*0*	Weekday building energy use [W m-2] Can be used for CO2 flux calculation.

TrafficRate_WD

Description Weekday traffic rate [veh km m^{-2} s-1] Can be used for CO2 flux calculation - not used in v2018a.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*0*	Weekday traffic rate [veh km m-2 s-1] Can be used for CO2 flux calculation.

TrafficRate_WE

Description Weekend traffic rate [veh km m^{-2} s-1] Can be used for CO2 flux calculation - not used in v2018a.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Weekend traffic rate [veh km m-2 s-1] Can be used for CO2 flux calculation.

Troad

Description Ground surface temperature [C] (used when `TsurfChoice` = 1 or 2)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Ground surface temperature [C] (used when `TsurfChoice` = 1 or 2)

Troof

Description Roof surface temperature [C] (used when `TsurfChoice` = 1 or 2)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Roof surface temperature [C] (used when `TsurfChoice` = 1 or 2)

Tsurf

Description Bulk surface temperature [C] (used when `TsurfChoice` = 0)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Bulk surface temperature [C] (used when TsurfCoice = 0)

Twall

Description Wall surface temperature [C] (used when `TsurfChoice` = 1)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	Wall surface temperature [C] (used when `TsurfChoice` = 1)

Twall_e

Description East-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	East-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

Twall_n

Description North-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	North-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

Twall_s

Description South-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	South-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

Twall_w

Description West-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_ESTM_Ts_data_tt.txt	*MU*	West-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

U

Description Wind speed. [m s^{-1}.]Height of the wind speed measurement (*z*) is needed in *SUEWS_SiteSelect.txt* .

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*MU*	Height of the wind speed measurement (z) is needed in *SUEWS_SiteSelect.txt*.

Wall_k1

Description Thermal conductivity of the first layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Thermal conductivity of the first layer [W m^{-1} K^{-1}]

Wall_k2

Description Thermal conductivity of the second layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the second layer [W m^{-1} K^{-1}]

Wall_k3

Description Thermal conductivity of the third layer [W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the third layer [W m^{-1} K^{-1}]

Wall_k4

Description Thermal conductivity of the fourth layer[W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the fourth layer[W m^{-1} K^{-1}]

Wall_k5

Description Thermal conductivity of the fifth layer[W m^{-1} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thermal conductivity of the fifth layer[W m^{-1} K^{-1}]

Wall_rhoCp1

Description Volumetric heat capacity of the first layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Volumetric heat capacity of the first layer [J m^{-3} K^{-1}]

Wall_rhoCp2

Description Volumetric heat capacity of the second layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the second layer [J m^{-3} K^{-1}]

Wall_rhoCp3

Description Volumetric heat capacity of the third layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the third layer [J m^{-3} K^{-1}]

Wall_rhoCp4

Description Volumetric heat capacity of the fourth layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the fourth layer [J m^{-3} K^{-1}]

Wall_rhoCp5

Description Volumetric heat capacity of the fifth layer [J m^{-3} K^{-1}]

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Volumetric heat capacity of the fifth layer [J m^{-3} K^{-1}]

Wall_thick1

Description Thickness of the first layer [m] for building surfaces only; set to -999 for all other surfaces

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*MU*	Thickness of the first layer [m] for building surfaces only; set to -999 for all other surfaces

Wall_thick2

Description Thickness of the second layer [m] (if no second layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the second layer [m] (if no second layer, set to -999.)

Wall_thick3

Description Thickness of the third layer [m] (if no third layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the third layer [m] (if no third layer, set to -999.)

Wall_thick4

Description Thickness of the fourth layer [m] (if no fourth layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the fourth layer [m] (if no fourth layer, set to -999.)

Wall_thick5

Description Thickness of the fifth layer [m] (if no fifth layer, set to -999.)

Configuration

Referencing Table	Requirement	Comment
SUEWS_ESTMCoefficients.txt	*O*	Thickness of the fifth layer [m] (if no fifth layer, set to -999.)

WaterDepth

Description Water depth [mm].

Configuration

Referencing Table	Requirement	Comment
SUEWS_Water.txt	*MU*	Set to a large value (e.g. 20000 mm = 20 m) if the water body is substantial (lake, river, etc) or a small value (e.g. 10 mm) if water bodies are very shallow (e.g. fountains). This value must not exceed StateLimit (column 8).

WaterUseProfAutoWD

Description Code for water use profile (automatic irrigation, weekdays) linking to `Code` of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for water use profile (automatic irrigation, weekdays) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*.

WaterUseProfAutoWE

Description Code for water use profile (automatic irrigation, weekends) linking to `Code` of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for water use profile (automatic irrigation, weekends) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*.

`WaterUseProfManuWD`

Description Code for water use profile (manual irrigation, weekdays) linking to `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for water use profile (manual irrigation, weekdays) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*.

`WaterUseProfManuWE`

Description Code for water use profile (manual irrigation, weekends) linking to `Code` of *SUEWS_Profiles.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code for water use profile (manual irrigation, weekends) Provides the link to column 1 of *SUEWS_Profiles.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_Profiles.txt*.

`wdir`

Description Wind direction [deg].

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	**Not available in this version.**

WetThreshold

Description Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface [mm].

Configuration

Referencing Table	Requirement	Comment
SUEWS_NonVeg.txt	*MD*	Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface. Example values: • 0.6 Paved • 0.6 Bldgs • 1. BSoil
SUEWS_Veg.txt	*MD*	Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface. Example values: • 1.8 EveTr • 1. DecTr • 2. Grass
SUEWS_Water.txt	*MD*	Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface. Example values: • 0.5 Water

WithinGridBldgsCode

Description Code that links to the fraction of water that flows from `Bldgs` surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code that links to the fraction of water that flows from Bldgs surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_WithinGridWaterDist.txt*.

WithinGridBSoilCode

Description Code that links to the fraction of water that flows from `BSoil` surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code that links to the fraction of water that flows from BSoil surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_WithinGridWaterDist.txt*.

WithinGridDecTrCode

Description Code that links to the fraction of water that flows from `DecTr` surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code that links to the fraction of water that flows from DecTr surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_WithinGridWaterDist.txt*.

WithinGridEveTrCode

Description Code that links to the fraction of water that flows from `EveTr` surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code that links to the fraction of water that flows from EveTr surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_WithinGridWaterDist.txt*.

WithinGridGrassCode

Description Code that links to the fraction of water that flows from `Grass` surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code that links to the fraction of water that flows from Grass surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_WithinGridWaterDist.txt*.

WithinGridPavedCode

Description Code that links to the fraction of water that flows from `Paved` surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code that links to the fraction of water that flows from Paved surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt* . Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_WithinGridWaterDist.txt*.

WithinGridWaterCode

Description Code that links to the fraction of water that flows from Water surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*L*	Code that links to the fraction of water that flows from Water surfaces to surfaces in columns 2-10 of *SUEWS_WithinGridWaterDist.txt*. Value of integer is arbitrary but must match code specified in column 1 of *SUEWS_WithinGridWaterDist.txt*.

Wuh

Description External water use [m^3]

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	External water use [m^3]

xsmd

Description Observed soil moisture; can be provided either as volumetric ([m^3 m^{-3}] when `SMDMethod` = 1) or gravimetric quantity ([kg kg^{-1}] when `SMDMethod` = 2). This should be used in conjunction with other soil properties in *SUEWS_Soil.txt*.

Configuration

Referencing Table	Requirement	Comment
SSss_YYYY_data_tt.txt	*O*	Observed soil moisture [m^3 m^{-3} or kg kg^{-1}]

Year

Description Year [YYYY]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	Year [YYYY] Years must be continuous. If running multiple years, ensure the rows in *SUEWS_SiteSelect.txt* are arranged so that all grids for a particular year appear on consecutive lines (rather than grouping all years together for a particular grid).

z

Description Measurement height [m] for all atmospheric forcing variables set in *SSss_YYYY_data_tt.txt*.

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*MU*	z must be greater than the displacement height. Forcing data should be representative of the local-scale, i.e. above the height of the roughness elements.

z0

Description Roughness length for momentum [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Value supplied here is used if `RoughLenMomMethod` = 1 in *RunControl.nml* ; otherwise set to '-999' and a value will be calculated by the model (`RoughLenMomMethod` = 2, 3).

zd

Description Zero-plane displacement [m]

Configuration

Referencing Table	Requirement	Comment
SUEWS_SiteSelect.txt	*O*	Value supplied here is used if `RoughLenMomMethod` = 1 in *RunControl.nml* ; otherwise set to -999 and a value will be calculated by the model (`RoughLenMomMethod` = 2, 3).

zi0

Description initial convective boundary layer height (m)

Configuration

Referencing Table	Requirement	Comment
CBL_initial_data.txt	*MU*	initial convective boundary layer height [m]

Typical Values

Other values to add - please let us know

Generic Properties

Property	General Type	Value	Description	Reference
Albedo	Non Vegetated	0.09	Paved Helsinki	Järvi *et al.* [2014]
	Non Vegetated	0.15	Buildings Helsinki	Järvi *et al.* [2014]
	Non Vegetated	0.19	Bare Soil, Helsinki	Järvi *et al.* [2014]
	Non Vegetated	0.12	Paved	Oke [2002]
	Non Vegetated	0.15	Buildings	Oke [2002]
	Non Vegetated	0.21	Bare Soil	Oke [2002]
Emissivity	Non Vegetated	0.95	Paved	Oke [2002]
	Non Vegetated	0.91	Buildings	Oke [2002]
	Non Vegetated	0.93	Bare Soil	Oke [2002]
Surface Water storage capacity	Non Vegetated	0.48	Paved	Davies and Hollis [1981]
	Non Vegetated	0.25	Buildings	Falk and Niemczynowicz [1978]
Albedo	Vegetation	0.1	EveTr	
	Vegetation	0.12	DecTr	
	Vegetation	0.18	Grass	
	Vegetated	0.1	EveTr Helsinki	Järvi *et al.* [2014]
	Vegetated	0.16	DecTr Helsinki	Järvi *et al.* [2014]
	Vegetated	0.19	Grass Helsinki	Järvi *et al.* [2014]
	Vegetated	0.1	EveTr	Oke [2002]
	Vegetated	0.18	DecTr	Oke [2002]
	Vegetated	0.21	Grass	Oke [2002]
Emissivity	Vegetated	0.98	EveTr	Oke [2002]

continues on next page

Table 7.354 – continued from previous page

Property	General Type	Value	Description	Reference
	Vegetated	0.98	DecTr	Oke [2002]
	Vegetated	0.93	Grass	Oke [2002]
water Storage Minimum capacity (mm)	Vegetated	1.3	EveTr	Breuer *et al.* [2003]
	Vegetated	0.3	DecTr	Breuer *et al.* [2003]
	Vegetated	1.9	Grass	Breuer *et al.* [2003]
Maximum water storage capacity of this surface [mm]	Vegetated	1.3	EveTr	Breuer *et al.* [2003]
	Vegetated	0.8	DecTr	Grimmond and Oke (1991)
	Vegetated	1.9	Grass	Breuer *et al.* [2003]
Albedo Max (leaf on)	Vegetated	0.12	DecTr	
	Vegetated	0.18	Grass	
	Vegetated	0.1	EveTr Helsinki	Järvi *et al.* [2014]
	Vegetated	0.16	DecTr Helsinki	Järvi *et al.* [2014]
	Vegetated	0.19	Grass Helsinki	Järvi *et al.* [2014]
	Vegetated	0.1	EveTr	Oke [2002]
	Vegetated	0.18	DecTr	Oke [2002]
	Vegetated	0.21	Grass	Oke [2002]
Emissivity *View factors should be taken into account	Vegetated	0.98	EveTr	Oke [2002]
Emissivity *View factors should be taken into account	Vegetated	0.98	DecTr	Oke [2002]
Emissivity *View factors should be taken into account	Vegetated	0.93	Grass	Oke [2002]
Minimum water storage capacity of this surface [mm] • Min & max values are to account for seasonal variation (e.g. leaf-on/leaf-off differences for vegetated surfaces).	Vegetated	1.3	EveTr	Breuer *et al.* [2003]
	Vegetated	0.3	DecTr	Breuer *et al.* [2003]
	Vegetated	1.9	Grass	Breuer *et al.* [2003]

continues on next page

Table 7.354 – continued from previous page

Property	General Type	Value	Description	Reference
	Vegetated	1.3	EveTr	Breuer *et al.* [2003]
	Vegetated	0.8	DecTr	Grimmond and Oke (1991)
	Vegetated	1.9	Grass	Breuer *et al.* [2003]
AlbedoMin	Water	0.1	Water	Oke [2002]
AlbedoMax	Water	0.1	Water	Oke [2002]
Emissivity	Water	0.95	Water	Oke [2002]
Minimum water storage capacity of this surface [mm]	Water	0.5	Water	
Maximum water storage capacity for upper surfaces (i.e. canopy)	Water	0.5	Water	
WetThreshold	water	0.5	Water	
StateLimit • Upper limit to the surface state [mm] • State cannot exceed this value. • Set to a large value (e.g. 20000 mm = 20 m) if the water body is substantial (lake| river| etc) or a small value (e.g. 10 mm) if water bodies are very shallow (e.g. fountains).	Water	20000	Water	
RadMeltFactor	Snow	0.0016	Hourly radiation melt factor of snow [mm W-1 h-1]	
TempMeltFactor	Snow	0.12	Hourly temperature melt factor of snow [mm °C -1 h-1]	
AlbedoMin	Snow	0-1	Minimum snow albedo [-] - 0.18	Järvi *et al.* [2014]

continues on next page

Table 7.354 – continued from previous page

Property	General Type	Value	Description	Reference
AlbedoMax * Maximum snow albedo (fresh snow) [-]	Snow	0.85		Järvi *et al.* [2014]
Emissivity * Effective surface emissivity. * View factors should be taken into account	Snow	0.99	Snow	Järvi *et al.* [2014]
tau_a * Time constant for snow albedo aging in cold snow [-]	Snow	0.018		Järvi *et al.* [2014]
tau_f *Time constant for snow albedo aging in melting snow [-]	Snow	0.11		Järvi *et al.* [2014]
PrecipiLimAlb	Snow	2	Limit for hourly precipitation when the ground is fully covered with snow. Then snow albedo is reset to AlbedoMax [mm]	
snowDensMin	Snow	100	Fresh snow density [kg m-3]	
snowDensMax	Snow	400	Maximum snow density [kg m-3]	
tau_r *Time constant for snow density ageing [-]	Snow	0.043		Järvi *et al.* [2014]
CRWMin *Minimum water holding capacity of snow [mm]	Snow	0.05		Järvi *et al.* [2014]
CRWMax *Maximum water holding capacity of snow [mm]	Snow	0.2		Järvi *et al.* [2014]
PrecipLimSnow	Snow	2.2	Temperature limit when precipitation falls as snow [°C]	Auer [1974]
SoilDepth	Snow	350	Depth of sub-surface soil store [mm] *depth of soil beneath the surface	
SoilStoreCap	Soil	150	Capacity of sub-surface soil store [mm]	

continues on next page

Table 7.354 – continued from previous page

Property	General Type	Value	Description	Reference
			how much water can be stored in the sub-surface soil when at maximum capacity.	
			(SoilStoreCap must not be greater than SoilDepth.)	
SatHydraulicCond	Soil	0.0005	Hydraulic conductivity for saturated soil [mm s-1]	
SoilDensity	Soil	1.16	Soil density [kg m-3]	
InfiltrationRate	Soil		Infiltration rate [mm h-1]	
OBS_SMDepth	Soil		Depth of soil moisture measurements [mm]	
OBS_SMCap	Soil		Maxiumum observed soil moisture [m3 m-3 or kg kg-1]	
OBS_SoilNotRocks	Soil		Fraction of soil without rocks [-]	

Storage Heat Flux Related

- Values determined from the literature
- If you have recommendations for others to be included please let us know.
- In the model run, canyons are excluded

Surface type	Description	a1	a2	a3	Reference
Canyon	E-W canyon	0.71	0.04	-39.7	Yoshida *et al.* [1990]
	N-S canyon	0.32	0.01	-27.7	Nunez and Oke [1977]
Vegetation	Mixed forest	0.11	0.11	-12.3	McCaughey [1985]
	Short grass	0.32	0.54	-27.4	Doll *et al.* [1985]
	Bare soil	0.38	0.56	-27.3	Novak [1981]
	Bare soil (wet)	0.33	0.07	-34.9	Fuchs and Hadas [1972]
	Bare soil (dry)	0.65	0.43	-36.5	Fuchs and Hadas [1972]
	Bare soil	0.36	0.27	-42.4	Asaeda and Ca [1993]
	Water Shallow – Turbid	0.5	0.21	-39.1	South *et al.* [1998]
	Unirrigated grass (Crops)	0.21	0.11	-16.1	Grimmond [1992]
	Short irrigated grass	0.35	-0.01	-26.3	Grimmond [1992]
Roof	Tar and gravel, Vancouver	0.17	0.1	-17	Yap [1973]
	Uppsala	0.44	0.57	-28.9	Taesler [1980]
	Membrane and concrete, Kyoto	0.82	0.34	-55.7	Yoshida *et al.* [1990]
	Average gravel/tar/conc. flat industrial, Vancouver	0.25	0.92	-22	Meyn and Oke [2009]
	Dry –gravel/tar/conc. flat industrial, Vancouver	0.25	0.7	-22	Meyn and Oke [2009]

continues on next page

Table 7.355 – continued from previous page

Surface type	Description	a1	a2	a3	Reference
	Wet – gravel/tar/conc. flat industrial, Vancouver	0.25	0.7	-22	Meyn and Oke [2009]
	Bitumen spread over flat industrial membrane, Vancouver	0.06	0.28	-3	Meyn and Oke [2009]
	Asphalt shingle on plywood residential roof , Vancouver	0.14	0.33	-6	Meyn and Oke [2009]
	Star – high albedo asphalt shingle residential roof	0.09	0.18	-1	Meyn and Oke [2009]
	Star - Ceramic Tile	0.07	0.26	-6	Meyn and Oke [2009]
	Star - Slate Tile	0.08	0.32	0	Meyn and Oke [2009]
	Helsinki – Suburban	0.19	0.54	-15.1	Järvi *et al.* [2014]
	Montreal – Suburban	0.12	0.24	-4.5	Järvi *et al.* [2014]
	Montreal – Urban	0.26	0.85	-21.4	Järvi *et al.* [2014]
Impervious	Concrete	0.81	0.1	-79.9	Doll *et al.* [1985]
	Concrete	0.85	0.32	-28.5	Asaeda and Ca [1993]
	Asphalt	0.36	0.23	-19.3	NARITA *et al.* [1984]
	Asphalt	0.64	0.32	-43.6	Asaeda and Ca [1993]
	Asphalt	0.82	0.68	-20.1	Anandakumar [1999]
	Asphalt (winter)	0.72	0.54	-40.2	Anandakumar [1999]
	Asphalt (summer)	0.83	-0.83	-24.6	Anandakumar [1999]

The above text files (used to be stored as worksheets in **SUEWS_SiteInfo.xlsm** for versions prior to v2018a) can be edited directly (see *Data Entry*). Please note this file is subject to possible changes from version to version due to new features, modifications, etc. Please be aware of using the correct copy of this worksheet that are always shipped with the SUEWS public release.

Tip:

1. See *SUEWS input converter* for conversion of input file between different versions.
2. Typical values for various properties *can be found here*.

7.3.3 Initial Conditions file

To start the model, information about the conditions at the start of the run is required. This information is provided in initial conditions file. One file can be specified for each grid (`MultipleInitFiles=1` in *RunControl.nml*, filename includes grid number) or, alternatively, a single file can be specified for all grids (MultipleInitFiles=0 in *RunControl.nml*, no grid number in the filename). After that, a new InitialConditionsSSss_YYYY.nml file will be written for each grid for the following years. It is recommended that you look at these files (written to the input directory) to check the status of various surfaces at the end or the run. This may help you get more realistic starting values if you are uncertain what they should be. Note this file will be created for each year for multiyear runs for each grid. If the run finishes before the end of the year the InitialConditions file is still written and the file name is appended with '_EndofRun'.

A sample file of **InitialConditionsSSss_YYYY.nml** looks like

```
&InitialConditions
LeavesOutInitially=0
SoilstorePavedState=150
SoilstoreBldgsState=150
SoilstoreEveTrstate=150
SoilstoreDecTrState=150
SoilstoreGrassState=150
SoilstoreBSoilState=150
```

(continues on next page)

(continued from previous page)

```
BoInit=10
/
```

The two most important pieces of information in the initial conditions file is the soil moisture and state of vegetation at the start of the run. This is the minimal information required; other information can be provided if known, otherwise SUEWS will make an estimate of initial conditions.

The parameters and their setting instructions are provided through the links below:

Note: Variables can be in any order

- *Soil moisture states*
  - *SoilstorePavedState*
  - *SoilstoreBldgsState*
  - *SoilstoreEveTrState*
  - *SoilstoreDecTrState*
  - *SoilstoreGrassState*
  - *SoilstoreBSoilState*
- *Vegetation parameters*
  - *LeavesOutInitially*
  - *GDD_1_0*
  - *GDD_2_0*
  - *LAIinitialEveTr*
  - *LAIinitialDecTr*
  - *LAIinitialGrass*
  - *albEveTr0*
  - *albDecTr0*
  - *albGrass0*
  - *decidCap0*
  - *porosity0*
- *Recent meteorology*
  - *DaysSinceRain*
  - *Temp_C0*
- *Above ground state*
  - *PavedState*
  - *BldgsState*
  - *EveTrState*
  - *DecTrState*
  - *GrassState*
  - *BSoilState*
  - *WaterState*
- *Snow related parameters*
  - *SnowInitially*
  - *SnowWaterPavedState*
  - *SnowWaterBldgsState*
  - *SnowWaterEveTrState*
  - *SnowWaterDecTrState*
  - *SnowWaterGrassState*
  - *SnowWaterBSoilState*
  - *SnowWaterWaterState*
  - *SnowPackPaved*
  - *SnowPackBldgs*
  - *SnowPackEveTr*
  - *SnowPackDecTr*
  - *SnowPackGrass*
  - *SnowPackBSoil*
  - *SnowPackWater*
  - *SnowFracPaved*
  - *SnowFracBldgs*
  - *SnowFracEveTr*
  - *SnowFracDecTr*
  - *SnowFracGrass*
  - *SnowFracBSoil*
  - *SnowFracWater*

- *SnowDensPaved*
- *SnowDensBldgs*
- *SnowDensEveTr*
- *SnowDensDecTr*
- *SnowDensGrass*
- *SnowDensBSoil*
- *SnowDensWater*
- *SnowAlb0*

Soil moisture states

SoilstorePavedState

Requirement Required

Description Initial water stored in soil beneath *Paved* surface [mm]

Configuration For maximum values, see the used soil code in *SUEWS_Soil.txt*

SoilstoreBldgsState

Requirement Required

Description Initial water stored in soil beneath *Bldgs* surface [mm]

Configuration For maximum values, see the used soil code in *SUEWS_Soil.txt*

SoilstoreEveTrState

Requirement Required

Description Initial water stored in soil beneath *EveTr* surface [mm]

Configuration For maximum values, see the used soil code in *SUEWS_Soil.txt*

SoilstoreDecTrState

Requirement Required

Description Initial water stored in soil beneath *DecTr* surface [mm]

Configuration For maximum values, see the used soil code in *SUEWS_Soil.txt*

SoilstoreGrassState

Requirement Required

Description Initial water stored in soil beneath *Grass* surface [mm]

Configuration For maximum values, see the used soil code in *SUEWS_Soil.txt*

SoilstoreBSoilState

Requirement Required

Description Initial water stored in soil beneath *BSoil* surface [mm]

Configuration For maximum values, see the used soil code in *SUEWS_Soil.txt*

Vegetation parameters

LeavesOutInitially

> **Requirement** Optional
>
> **Description** Flag for initial leave status [1 or 0]
>
> **Configuration** If the model run starts in winter when trees are bare, set *LeavesOutInitially* = 0 and the vegetation parameters will be set accordingly based on the values set in SUEWS_SiteInfo.xlsm. If the model run starts in summer when leaves are fully out, set *LeavesOutInitially* = 1 and the vegetation parameters will be set accordingly based on the values set in SUEWS_SiteInfo.xlsm. Not LeavesOutInitially can only be set to 0, 1 or -999 (fractional values cannot be used to indicate partial leaf-out). The value of *LeavesOutInitially* overrides any values provided for the individual vegetation parameters. To prevent *LeavesOutInitially* from setting the initial conditions, either omit it from the namelist or set to -999. If values are provided individually, they should be consistent the information provided in SUEWS_Veg.txt and the time of year. If values are provided individually, values for all required surfaces must be provided (i.e. specifying only *albGrass0* but not *albDecTr0* nor *albEveTr0* is not permitted).

GDD_1_0

> **Requirement** Optional
>
> **Description** GDD related initial value
>
> **Configuration** Cannot be negative. If leaves are already full, then this should be the same as *GDDFull* in *SUEWS_Veg.txt*. If winter, set to 0. It is important that the vegetation characteristics are set correctly (i.e. for the start of the run in summer/winter).

GDD_2_0

> **Requirement** Optional
>
> **Description** GDD related initial value
>
> **Configuration** Cannot be positive If the leaves are full but in early/mid summer then set to 0. If late summer or autumn , this should be a negative value. If leaves are off , then use the values of *SDDFull* in *SUEWS_Veg.txt* to guide your minimum value. It is important that the vegetation characteristics are set correctly (i.e. for the start of the run in summer/winter).

LAIinitialEveTr

> **Requirement** Optional
>
> **Description** Initial LAI for evergreen trees *EveTr*.
>
> **Configuration** The recommended values can be found from *SUEWS_Veg.txt*

LAIinitialDecTr

> **Requirement** Optional
>
> **Description** Initial LAI for deciduous trees *DecTr*.
>
> **Configuration** The recommended values can be found from *SUEWS_Veg.txt*

LAIinitialGrass

> **Requirement** Optional
>
> **Description** Initial LAI for irrigated grass *Grass*.

Configuration The recommended values can be found from *SUEWS_Veg.txt*

`albEveTr0`

Requirement Optional

Description Albedo of evergreen surface `EveTr` on day 0 of run

Configuration The recommended values can be found from *SUEWS_Veg.txt*

`albDecTr0`

Requirement Optional

Description Albedo of deciduous surface `DecTr` on day 0 of run

Configuration The recommended values can be found from *SUEWS_Veg.txt*

`albGrass0`

Requirement Optional

Description Albedo of grass surface `Grass` on day 0 of run

Configuration The recommended values can be found from *SUEWS_Veg.txt*

`decidCap0`

Requirement Optional

Description Storage capacity of deciduous surface `DecTr` on day 0 of run.

Configuration The recommended values can be found from *SUEWS_Veg.txt*

`porosity0`

Requirement Optional

Description Porosity of deciduous vegetation on day 0 of run.

Configuration This varies between 0.2 (leaf-on) and 0.6 (leaf-off). The recommended values can be found from *SUEWS_Veg.txt*

Recent meteorology

`DaysSinceRain`

Requirement Optional

Description Days since rain [d]

Configuration Important to use correct value if starting in summer season If starting when external water use is not occurring it will be reset with the first rain so can just be set to 0. If unknown, SUEWS sets to zero by default. Used to model irrigation.

`Temp_C0`

Requirement Optional

Description Initial air temperature [degC]

Configuration If unknown, SUEWS uses the mean temperature for the first day of the run.

Above ground state

`PavedState`

Requirement Optional

Description Initial wetness condition on *Paved*

Configuration If unknown, model assumes dry surfaces (acceptable as rainfall or irrigation will update these states quickly).

`BldgsState`

Requirement Optional

Description Initial wetness condition on *Bldgs*

Configuration If unknown, model assumes dry surfaces (acceptable as rainfall or irrigation will update these states quickly).

`EveTrState`

Requirement Optional

Description Initial wetness condition on *EveTr*

Configuration If unknown, model assumes dry surfaces (acceptable as rainfall or irrigation will update these states quickly).

`DecTrState`

Requirement Optional

Description Initial wetness condition on *DecTr*

Configuration If unknown, model assumes dry surfaces (acceptable as rainfall or irrigation will update these states quickly).

`GrassState`

Requirement Optional

Description Initial wetness condition on *Grass*

Configuration If unknown, model assumes dry surfaces (acceptable as rainfall or irrigation will update these states quickly).

`BSoilState`

Requirement Optional

Description Initial wetness condition on *BSoil*

Configuration If unknown, model assumes dry surfaces (acceptable as rainfall or irrigation will update these states quickly).

`WaterState`

Requirement Optional

Description Initial wetness condition on *Water*

Configuration For a large water body (e.g. river, sea, lake) set WaterState to a large value, e.g. 20000 mm; for small water bodies (e.g. ponds, fountains) set WaterState to smaller value, e.g. 1000 mm. This value must not exceed StateLimit specified in SUEWS_Water.txt . If unknown, model uses value of WaterDepth specified in SUEWS_Water.txt .

Snow related parameters

`SnowInitially`

Requirement Optional

Description Flag for initial snow status [0 or 1]

Configuration If the model run starts when there is no snow on the ground, set `SnowInitially` = 0 and the snow-related parameters will be set accordingly. If the model run starts when there is snow on the ground, the following snow-related parameters must be set appropriately. The value of `SnowInitially` overrides any values provided for the individual snow-related parameters. To prevent `SnowInitially` from setting the initial conditions, either omit it from the namelist or set to -999. If values are provided individually, they should be consistent the information provided in *SUEWS_Snow.txt* .

`SnowWaterPavedState`

Requirement Optional

Description Initial amount of liquid water in the snow on paved surfaces `Paved`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowWaterBldgsState`

Requirement Optional

Description Initial amount of liquid water in the snow on buildings `Bldgs`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowWaterEveTrState`

Requirement Optional

Description Initial amount of liquid water in the snow on evergreen trees `EveTr`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowWaterDecTrState`

Requirement Optional

Description Initial amount of liquid water in the snow on deciduous trees `DecTr`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowWaterGrassState`

Requirement Optional

Description Initial amount of liquid water in the snow on grass surfaces `Grass`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowWaterBSoilState`

Requirement Optional

Description Initial amount of liquid water in the snow on bare soil surfaces `BSoil`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowWaterWaterState`

Requirement Optional

Description Initial amount of liquid water in the snow in water *Water*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowPackPaved`

Requirement Optional

Description Initial snow water equivalent if the snow on paved surfaces *Paved*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowPackBldgs`

Requirement Optional

Description Initial snow water equivalent if the snow on buildings *Bldgs*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowPackEveTr`

Requirement Optional

Description Initial snow water equivalent if the snow on evergreen trees *EveTr*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowPackDecTr`

Requirement Optional

Description Initial snow water equivalent if the snow on deciduous trees *DecTr*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowPackGrass`

Requirement Optional

Description Initial snow water equivalent if the snow on grass surfaces *Grass*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowPackBSoil`

Requirement Optional

Description Initial snow water equivalent if the snow on bare soil surfaces *BSoil*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowPackWater`

Requirement Optional

Description Initial snow water equivalent if the snow on water *Water*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowFracPaved`

Requirement Optional

Description Initial plan area fraction of snow on paved surfaces `Paved`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowFracBldgs`

Requirement Optional

Description Initial plan area fraction of snow on buildings `Bldgs`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowFracEveTr`

Requirement Optional

Description Initial plan area fraction of snow on evergreen trees `EveTr`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowFracDecTr`

Requirement Optional

Description Initial plan area fraction of snow on deciduous trees `DecTr`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowFracGrass`

Requirement Optional

Description Initial plan area fraction of snow on grass surfaces `Grass`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowFracBSoil`

Requirement Optional

Description Initial plan area fraction of snow on bare soil surfaces `BSoil`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowFracWater`

Requirement Optional

Description Initial plan area fraction of snow on water `Water`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowDensPaved`

Requirement Optional

Description Initial snow density on paved surfaces `Paved`

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowDensBldgs`

Requirement Optional

Description Initial snow density on buildings *`Bldgs`*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowDensEveTr`

Requirement Optional

Description Initial snow density on evergreen trees *`EveTr`*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowDensDecTr`

Requirement Optional

Description Initial snow density on deciduous trees *`DecTr`*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowDensGrass`

Requirement Optional

Description Initial snow density on grass surfaces *`Grass`*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowDensBSoil`

Requirement Optional

Description Initial snow density on bare soil surfaces *`BSoil`*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowDensWater`

Requirement Optional

Description Initial snow density on *`Water`*

Configuration The recommended values can be found from *SUEWS_Snow.txt*

`SnowAlb0`

Requirement Optional

Description Initial snow albedo

Configuration The recommended values can be found from *SUEWS_Snow.txt*

7.3.4 Meteorological Input File

SUEWS is designed to run using commonly measured meteorological variables (e.g. incoming solar radiation, air temperature, relative humidity, pressure, wind speed, etc.).

When preparing this input file, please note the following:

- Required inputs must be continuous – i.e. **gap fill** any missing data.
- Temporal information (i.e., `iy`, `id`, `it` and `imin`) should be in **local time** and indicate the ending timestamp of corresponding periods: e.g. for hourly data, `2021-09-12 13:00` indicates a record for the period between `2021-09-12 12:00` (inclusive) and `2021-09-12 13:00` (exclusive).
- The *table* below gives the must-use (`MU`) and optional (`O`) additional input variables. If an optional input variable (`O`) is not available or will not be used by the model, enter '-999' for this column.
- One single meteorological file can be used for all grids (**MultipleMetFiles=0** in *RunControl.nml*, no grid number in file name) if appropriate for the study area.
- Separate met files can be used for each grid if data are available (**MultipleMetFiles=1** in *RunControl.nml*, file-name includes grid number).
- The meteorological forcing file names should be appended with the temporal resolution in minutes: `tt` in `SS_YYYY_data_tt.txt` (or `SSss_YYYY_data_tt.txt` for multiple grids).
- Separate met forcing files should be provided for each year.
- Files do not need to start/end at the start/end of the year, but they must contain a whole number of days.
- The meteorological input file should match the information given in *SUEWS_SiteSelect.txt*.
- If a *partial year* is used that specific year must be given in SUEWS_SiteSelect.txt.
- If *multiple years* are used, all years should be included in SUEWS_SiteSelect.txt.
- If a *whole year* (e.g. 2011) is intended to be modelled using and hourly resolution dataset, the number of lines in the met data file should be 8760 and begin and end with:

```
iy      id  it  imin
2011    1   1   0 ...
...
2012    1   0   0 ...
```

SSss_YYYY_data_tt.txt

Changed in version v2017a: Since v2017a forcing files no longer need to end with two rows containing '-9' in the first column.

Main meteorological data file.

No.	Use	Column Name	Description
1	*MU*	iy	Year [YYYY]
2	*MU*	id	Day of year [DOY]
3	*MU*	it	Hour [H]
4	*MU*	imin	Minute [M]
5	*O*	qn	Net all-wave radiation [W m^{-2}] (Required if `NetRadiationMethod` = 0.)
6	*O*	qh	Sensible heat flux [W m^{-2}]
7	*O*	qe	Latent heat flux [W m^{-2}]
8	*O*	qs	Storage heat flux [W m^{-2}]
9	*O*	qf	Anthropogenic heat flux [W m^{-2}]
10	*MU*	U	Wind speed [m s-1] (measurement height (`z`) is needed in *SUEWS_SiteSelect.txt*)
11	*MU*	RH	Relative Humidity [%] (measurement height (`z`) is needed in *SUEWS_SiteSelect.txt*)
12	*MU*	Tair	Air temperature [°C] (measurement height (`z`) is needed in *SUEWS_SiteSelect.txt*)
13	*MU*	pres	Barometric pressure [kPa] (measurement height (`z`) is needed in *SUEWS_SiteSelect.txt*)
14	*MU*	rain	Rainfall [mm] (measurement height (`z`) is needed in *SUEWS_SiteSelect.txt*)
15	*MU*	kdown	Incoming shortwave radiation [W m^{-2}] Must be > 0 W m^{-2}.
16	*O*	snow	Snow cover fraction (0 – 1) [-] (Required if `SnowUse` = 1)
17	*O*	ldown	Incoming longwave radiation [W m^{-2}]
18	*O*	fcld	Cloud fraction [tenths]
19	*O*	Wuh	External water use [m^3]
20	*O*	xsmd	Observed soil moisture [m^3 m^{-3}] or [kg kg^{-1}]
21	*O*	lai	Observed leaf area index [m^{-2} m^{-2}]
22	*O*	kdiff	Diffuse radiation [W m^{-2}] **Recommended in this version.** if `SOLWEIGUse` = 1
23	*O*	kdir	Direct radiation [W m^{-2}] **Recommended in this version.** if `SOLWEIGUse` = 1
24	*O*	wdir	Wind direction [°] **Not available in this version.**

7.3.5 CBL input files

Main references for this part of the model: Onomura *et al.* [2015] and Cleugh and Grimmond [2001].

If CBL slab model is used (`CBLUse = 1` in *RunControl.nml*) the following files are needed.

Filename	Purpose
CBL_initial_data.txt	Gives initial data every morning * when CBL slab model starts running. * filename must match the InitialData_FileName in CBLInput.nml * fixed formats.
CBLInput.nml	Specifies run options, parameters and input file names. * Can be in any order

CBL_initial_data.txt

This file should give initial data every morning when CBL slab model starts running. The file name should match the InitialData_FileName in CBLInput.nml.

Definitions and example file of initial values prepared for Sacramento.

No.	Column name	Description
1	id	Day of year [DOY]
2	zi0	Initial convective boundary layer height (m)
3	gamt_Km	Vertical gradient of potential temperature (K m^{-1}) strength of the inversion
4	gamq_gkgm	Vertical gradient of specific humidity (g kg^{-1} m^{-1})
5	Theta+_K	Potential temperature at the top of CBL (K)
6	q+_gkg	Specific humidity at the top of CBL (g kg^{-1})
7	Theta_K	Potential temperature in CBL (K)
8	q_gkg	Specific humidiy in CBL (g kg^{-1})

- gamt_Km and gamq_gkgm written to two significant figures are required for the model performance in appropriate ranges [Onomura *et al.*, 2015].

id	zi0	gamt_Km	gamq_gkgm	Theta+_K	q+_gkg	theta_K	q_gkg
234	188	0.0032	0.00082	290.4	9.6	288.7	8.3
235	197	0.0089	0.089	290.2	8.4	288.3	8.7

CBLInput.nml

sample file of **CBLInput.nml** looks like

```
&CBLInput
EntrainmentType=1       ! 1.Tennekes and Driedonks(1981), 2.McNaughton and␣
↪Springgs(1986), 3.Rayner and Watson(1991),4.Tennekes(1973),
QH_choice=1             ! 1.suews  2.lumps 3.obs
CO2_included=0
cblday(236)=1
cblday(258)=1
cblday(259)=1
cblday(260)=1
cblday(285)=1
cblday(297)=1
wsb=-0.01
InitialData_use=1
InitialDataFileName='CBLinputfiles/CBL_initial_data.txt'
sondeflag=0
FileSonde(234)='CBLinputfiles\Sonde_Sc_1991_0822_0650.txt'
FileSonde(235)='CBLinputfiles\Sonde_Sc_1991_0823_0715.txt'
FileSonde(236)='CBLinputfiles\Sonde_Sc_1991_0824_0647.txt'
FileSonde(238)='CBLinputfiles\Sonde_Sc_1991_0826_0642.txt'
FileSonde(239)='CBLinputfiles\Sonde_Sc_1991_0827_0640.txt'
FileSonde(240)='CBLinputfiles\Sonde_Sc_1991_0828_0640.txt'
/
```

Note: The file contents can be in any order.

The parameters and their setting instructions are provided through *the links below*:

- *EntrainmentType*
- *QH_Choice*
- *InitialData_use*
- *Sondeflag*
- *CBLday(id)*
- *CO2_included*
- *FileSonde(id)*
- *InitialDataFileName*
- *Wsb*

CBLinput

EntrainmentType

Requirement Required

Description Determines entrainment scheme. See Cleugh and Grimmond 2000 [16] for details.

Configuration

Value	Comments
1	Tennekes and Driedonks (1981) - **Recommended in this version.**
2	McNaughton and Springs (1986)
3	Rayner and Watson (1991)
4	Tennekes (1973)

QH_Choice

Requirement Required

Description Determines QH used for CBL model.

Configuration

Value	Comments
1	QH modelled by SUEWS
2	QH modelled by LUMPS
3	Observed QH values are used from the meteorological input file

InitialData_use

Requirement Required

Description Determines initial values (see *CBL_initial_data.txt*)

Configuration

Value	Comments
0	All initial values are calculated. **Not available in this version.**
1	Take zi0, gamt_Km and gamq_gkgm from input data file. Theta+_K, q+_gkg, Theta_K and q_gkg are calculated using Temp_C, avrh and Pres_kPa in meteorological input file.
2	Take all initial values from input data file (see CBL_Initial_data.txt).

`Sondeflag`

Requirement Required

Description to fill

Configuration

Value	Comments
0	Does not read radiosonde vertical profile data - **Recommended in this version.**
1	Reads radiosonde vertical profile data

`CBLday(id)`

Requirement Required

Description Set CBLday(id) = 1 If CBL model is set to run for DOY 175–177, CBLday(175) = 1, CBLday(176) = 1, CBLday(177) = 1

Configuration to fill

`CO2_included`

Requirement Required

Description Set to zero in current version

Configuration to fill

`FileSonde(id)`

Requirement Required

Description If Sondeflag=1, write the file name including the path from site directory e.g. FileSonde(id)= 'CBLinputfilesXXX.txt', XXX is an arbitrary name.

Configuration to fill

`InitialDataFileName`

Requirement Required

Description If InitialData_use 1, write the file name including the path from site directory e.g. InitialDataFileName='CBLinputfilesCBL_initial_data.txt'

Configuration to fill

`Wsb`

Requirement Required

Description Subsidence velocity ($m\ s^{-1}$) in eq. 1 and 2 of Onomura et al. (2015) [17] . (-0.01 $m\ s^{-1}$ **Recommended in this version.**)

Configuration to fill

7.3.6 ESTM input files

SUEWS_ESTMCoefficients.txt

Note ESTM is under development in this release and should not be used!

The Element Surface Temperature Method (ESTM) [Offerle *et al.*, 2005] calculates the net storage heat flux from surface temperatures. In the method the three-dimensional urban volume is reduced to four 1-d elements (i.e. building roofs, walls, and internal mass and ground (road, vegetation, etc)). The storage heat flux is calculated from the heat conduction through the different elements. For the inside surfaces of the roof and walls, and both surfaces for the internal mass (ceilings/floors, internal walls), the surface temperature of the element is determined by setting the conductive heat transfer out of (in to) the surface equal to the radiative and convective heat losses (gains). Each element (roof, wall, internal element and ground) can have maximum five layers and each layer has three parameters tied to it: thickness (x), thermal conductivity (k), volumetric heat capacity (rhoCp).

If ESTM is used (`StorageHeatMethod` =4), the files *SUEWS_ESTMCoefficients.txt*, *ESTMinput.nml* and *SSss_YYYY_ESTM_Ts_data_tt.txt* should be prepared.

SUEWS_ESTMCoefficients.txt contains the parameters for the layers of each of the elements (roofs, wall, ground, internal mass).

- If less than five layers are used, the parameters for unused layers should be set to -999.
- The ESTM coefficients with the prefix *Surf_* must be specified for each surface type (plus snow) but the *Wall_* and *Internal_* variables apply to the building surfaces only.
- For each grid, one set of ESTM coefficients must be specified for each surface type; for paved and building surfaces it is possible to specify up to three and five sets of coefficients per grid (e.g. to represent different building materials) using the relevant columns in *SUEWS_SiteSelect.txt*. For the model to use these columns in site select, the ESTMCode column in *SUEWS_NonVeg.txt* should be set to zero.

The following input files are required if ESTM is used to calculate the storage heat flux.

ESTMinput.nml

ESTMinput.nml specifies the model settings and default values.

A sample file of **ESTMinput.nml** looks like

```
&ESTMinput
TsurfChoice= 0
evolveTibld= 0      ! !!!!!FO!!!!! 0 originally
ibldCHmod  = 0
LBC_soil   = 13.00             !!FO!! 4, 8 or 17 degC - could be set as the annual mean␣
↪air temperature (12.8 degC for London)
THEAT_ON   = 18.
THEAT_OFF  = 22.
THEAT_FIX  = 19.
/
```

Note: The file contents can be in any order.

The parameters and their setting instructions are provided through *the links below*:

- *TsurfChoice*
- *evolveTibld*
- *IbldCHmod*
- *LBC_soil*
- *Theat_fix*
- *Theat_off*
- *Theat_on*

ESTMinput

TsurfChoice

Requirement Required

Description Source of surface temperature data used.

Configuration

Value	Comments
0	*Tsurf* in *SSss_YYYY_ESTM_Ts_data_tt.txt* used for all surface elements.
1	Input surface temperature are different for ground, roof and wall.
2	Wall surface temperature is different for four directions.

evolveTibld

Requirement Required

Description Source of internal building temperature (Tibld)

Configuration

Value	Comments
0	*Tiair* in *SSss_YYYY_ESTM_Ts_data_tt.txt* used.
1	Tibld calculated considering the effect of anthropogenic heat from HVAC
2	Tibld calculated without considering the influence of HVAC.

IbldCHmod

Requirement Required

Description Method to calculate internal convective heat exchange coefficients (CH) for internal building, wall and roof if evolveTibld is 1 or 2.

Configuration

Value	Comments
0	CHs are read from SUEWS_ESTMcoefficients.txt.
1	CHs are calculated based on ASHRAE (2001)
2	CHs are calculated based on Awbi (1998).

LBC_soil

Requirement Required

Description Soil temperature at lowest boundary condition [C]

Configuration to fill

Theat_fix

Requirement Required

Description Ideal internal building temperature [C]

Configuration to fill

Theat_off

Requirement Required

Description Temperature at which heat control is turned off (used when evolveTibld=1) [C]

Configuration to fill

Theat_on

Requirement Required

Description Temperature at which heat control is turned on (used when evolveTibld =1) [C]

Configuration to fill

SSss_YYYY_ESTM_Ts_data_tt.txt

SSss_YYYY_ESTM_Ts_data_tt.txt contains a time-series of input surface temperature for roof, wall, ground and internal elements.

No.	Column Name	Use	Description
1	*iy*	*MU*	Year [YYYY]
2	*id*	*MU*	Day of year [DOY]
3	*it*	*MU*	Hour [H]
4	*imin*	*MU*	Minute [M]
5	*Tiair*	*MU*	Indoor air temperature [C]
6	*Tsurf*	*MU*	Bulk surface temperature [C] (used when *TsurfChoice* = 0)
7	*Troof*	*MU*	Roof surface temperature [C] (used when *TsurfChoice* = 1 or 2)
8	*Troad*	*MU*	Ground surface temperature [C] (used when *TsurfChoice* = 1 or 2)
9	*Twall*	*MU*	Wall surface temperature [C] (used when *TsurfChoice* = 1)
10	*Twall_n*	*MU*	North-facing wall surface temperature [C] (used when *TsurfChoice* = 2)
11	*Twall_e*	*MU*	East-facing wall surface temperature [C] (used when *TsurfChoice* = 2)
12	*Twall_s*	*MU*	South-facing wall surface temperature [C] (used when *TsurfChoice* = 2)
13	*Twall_w*	*MU*	West-facing wall surface temperature [C] (used when *TsurfChoice* = 2)

7.3.7 SUEWS-SPARTACUS (SS) input files

To run SUEWS-SS the SS specific files that need to be modified are:

- *RunControl.nml*
- *SUEWS_SPARTACUS.nml*

Non-SS specific SUEWS input file parameters also need to have appropriate values. For example, LAI, albedos and emissivities are used by SUEWS-SS as explained in *More background information*.

RunControl.nml

See *NetRadiationMethod* (sensible values are 1001, 1002 or 1003) in *RunControl.nml* parameter.

SUEWS_SPARTACUS.nml

This file is used to specify the SS model options when coupled to SUEWS.

A sample file of **SUEWS_SPARTACUS.nml** is shown below:

```
&Spartacus
nlayers = 1
use_sw_direct_albedo = false
n_vegetation_region_urban = 1
nsw = 1
nlw = 1
nspec = 1
n_stream_sw_urban = 8
n_stream_lw_urban = 8
sw_dn_direct_frac = 0.0
air_ext_sw = 0.0
air_ssa_sw = 0.95
veg_ssa_sw = 0.13
air_ext_lw = 0.0
air_ssa_lw = 0.0
veg_ssa_lw = 0.01
ground_albedo_dir_mult_fact = 1.
/
```

The parameters and their setting instructions are provided through the links below:

Geometry-related options

- nlayers
- *n_vegetation_region_urban*
- height
- building_frac
- building_scale
- veg_frac
- veg_scale
- veg_contact_fraction
- wall_specular_frac

Shortwave-related options

- *use_sw_direct_albedo*
- *sw_dn_direct_frac*
- *n_stream_sw_urban*
- *air_ext_sw*
- *air_ssa_sw*
- *veg_ssa_sw*
- *ground_albedo_dir_mult_fact*
- roof_albedo
- wall_albedo

Longwave-related options

- *n_stream_lw_urban*
- *air_ext_lw*
- *air_ssa_lw*
- *veg_ssa_lw*
- veg_fsd
- roof_emissivity
- wall_emissivity
- roof_albedo_dir_mult_fact

7.3.8 SUEWS input converter

Note: The SUEWS table converter has been integrated into SuPy as a command line tool *suews-convert* since v2020a. Please install SuPy and run *suews-convert* to convert input tables from an older version to a newer one.

Usage

Please refer to the *SuPy API page*.

Example (from 2018a to 2020a)

Assuming your 2018a files are all included in the folder `your_2018a_folder` and your desirable converted files should be placed in a new folder `your_2020a_folder`, please do the following in your command line tool:

```
suews-convert -f 2018a -t 2020a -i your_2018a_folder -o your_2020a_folder
```

Tip: *suews-convert* will use the `RunControl.nml` file in your original folder to determine the location of input tables.

7.4 Output files

7.4.1 Runtime diagnostic information

Error messages: problems.txt

If there are problems running the program serious error messages will be written to problems.txt.

- Serious problems will usually cause the program to stop after writing the error message. If this is the case, the last line of *Error messages: problems.txt* will contain a non-zero number (the error code).
- If the program runs successfully, problems.txt file ends with:

```
Run completed.
0
```

SUEWS has a large number of error messages included to try to capture common errors to help the user determine what the problem is. If you encounter an error that does not provide an error message please capture the details so we can hopefully provide better error messages in future.

See *Troubleshooting* section for help solving problems. If the file paths are not correct the program will return an error when run (see *Workflow of using SUEWS*).

Warning messages: warnings.txt

- If the program encounters a more minor issue it will not stop but a warning may be written to warnings.txt. It is advisable to check the warnings to ensure there is not a more serious problem.
- The warnings.txt file can be large (over several GBs) given warning messages are written out during a large scale simulation, you can use `tail/head` to view the ending/starting part without opening the whole file on Unix-like systems (Linux/mac OS), which may slow down your system.
- To prevent warnings.txt from being written, set *SuppressWarnings* to 1 in *RunControl.nml*.
- Warning messages are usually written with a grid number, timestamp and error count. If the problem occurs in the initial stages (i.e. before grid numbers and timestamps are assigned, these are printed as 00000).

Summary of model parameters: SS_FileChoices.txt

For each run, the model parameters specified in the input files are written out to the file SS_FileChoices.txt.

7.4.2 Model output files

Note: Temporal information in output files (i.e., `iy`, `id`, `it` and `imin` if existing) are in **local time** (i.e. consistent with *Meteorological Input File*) and indicate the ending timestamp of corresponding periods: e.g. for hourly data, `2021-09-12 13:00` indicates a record for the period between `2021-09-12 12:00` (inclusive) and `2021-09-12 13:00` (exclusive).

SSss_YYYY_SUEWS_TT.txt

SUEWS produces the main output file (SSss_YYYY_SUEWS_tt.txt) with time resolution (TT min) set by *ResolutionFilesOut* in *RunControl.nml*.

Before these main data files are written out, SUEWS provides a summary of the column names, units and variables included in the file Ss_YYYY_TT_OutputFormat.txt (one file per run).

The variables included in the main output file are determined according to *WriteOutOption* set in *RunControl.nml*.

Column	Name	WriteOutOption	Description
1	Year	0,1,2	Year [YYYY]
2	DOY	0,1,2	Day of year [DOY]
3	Hour	0,1,2	Hour [H]
4	Min	0,1,2	Minute [M]
5	Dectime	0,1,2	Decimal time [-]
6	Kdown	0,1,2	Incoming shortwave radiation [W m^{-2}]
7	Kup	0,1,2	Outgoing shortwave radiation [W m^{-2}]
8	Ldown	0,1,2	Incoming longwave radiation [W m^{-2}]
9	Lup	0,1,2	Outgoing longwave radiation [W m^{-2}]
10	Tsurf	0,1,2	Bulk surface temperature [°C]
11	QN	0,1,2	Net all-wave radiation [W m^{-2}]
12	QF	0,1,2	Anthropogenic heat flux [W m^{-2}]
13	QS	0,1,2	Storage heat flux [W m^{-2}]
14	QH	0,1,2	Sensible heat flux (calculated using SUEWS) [W m^{-2}]

continues on next page

Table 7.356 – continued from previous page

Column	Name	WriteOutOption	Description
15	QE	0,1,2	Latent heat flux (calculated using SUEWS) [W m^{-2}]
16	QHlumps	0,1	Sensible heat flux (calculated using LUMPS) [W m^{-2}]
17	QElumps	0,1	Latent heat flux (calculated using LUMPS) [W m^{-2}]
18	QHresis	0,1	Sensible heat flux (calculated using resistance method) [W m^{-2}]
19	Rain	0,1,2	Rain [mm]
20	Irr	0,1,2	Irrigation [mm]
21	Evap	0,1,2	Evaporation [mm]
22	RO	0,1,2	Runoff [mm]
23	TotCh	0,1,2	Change in surface and soil moisture stores [mm]
24	SurfCh	0,1,2	Change in surface moisture store [mm]
25	State	0,1,2	Surface wetness state [mm]
26	NWtrState	0,1,2	Surface wetness state (for non-water surfaces) [mm]
27	Drainage	0,1,2	Drainage [mm]
28	SMD	0,1,2	Soil moisture deficit [mm]
29	FlowCh	0,1	Additional flow into water body [mm]
30	AddWater	0,1	Additional water flow received from other grids [mm]
31	ROSoil	0,1	Runoff to soil (sub-surface) [mm]
32	ROPipe	0,1	Runoff to pipes [mm]
33	ROImp	0,1	Above ground runoff over impervious surfaces [mm]
34	ROVeg	0,1	Above ground runoff over vegetated surfaces [mm]
35	ROWater	0,1	Runoff for water body [mm]
36	WUInt	0,1	Internal water use [mm]
37	WUEveTr	0,1	Water use for irrigation of evergreen trees [mm]
38	WUDecTr	0,1	Water use for irrigation of deciduous trees [mm]
39	WUGrass	0,1	Water use for irrigation of grass [mm]
40	SMDPaved	0,1	Soil moisture deficit for paved surface [mm]
41	SMDBldgs	0,1	Soil moisture deficit for building surface [mm]
42	SMDEveTr	0,1	Soil moisture deficit for evergreen surface [mm]
43	SMDDecTr	0,1	Soil moisture deficit for deciduous surface [mm]
44	SMDGrass	0,1	Soil moisture deficit for grass surface [mm]
45	SMDBSoil	0,1	Soil moisture deficit for bare soil surface [mm]
46	StPaved	0,1	Surface wetness state for paved surface [mm]
47	StBldgs	0,1	Surface wetness state for building surface [mm]
48	StEveTr	0,1	Surface wetness state for evergreen tree surface [mm]
49	StDecTr	0,1	Surface wetness state for deciduous tree surface [mm]
50	StGrass	0,1	Surface wetness state for grass surface [mm]
51	StBSoil	0,1	Surface wetness state for bare soil surface [mm]
52	StWater	0,1	Surface wetness state for water surface [mm]
53	Zenith	0,1,2	Solar zenith angle [°]
54	Azimuth	0,1,2	Solar azimuth angle [°]
55	AlbBulk	0,1,2	Bulk albedo [-]
56	Fcld	0,1,2	Cloud fraction [-]
57	LAI	0,1,2	Leaf area index [m 2 m^{-2}]
58	z0m	0,1	Roughness length for momentum [m]
59	zdm	0,1	Zero-plane displacement height [m]
60	ustar	0,1,2	Friction velocity [m s^{-1}]
61	Lob	0,1,2	Obukhov length [m]
62	RA	0,1	Aerodynamic resistance [s m^{-1}]
63	RS	0,1	Surface resistance [s m^{-1}]

continues on next page

Table 7.356 – continued from previous page

Column	Name	WriteOutOption	Description
64	Fc	0,1,2	CO2 flux [umol m^{-2} s^{-1}]
65	FcPhoto	0,1	CO2 flux from photosynthesis [umol m^{-2} s^{-1}]
66	FcRespi	0,1	CO2 flux from respiration [umol m^{-2} s^{-1}]
67	FcMetab	0,1	CO2 flux from metabolism [umol m^{-2} s^{-1}]
68	FcTraff	0,1	CO2 flux from traffic [umol m^{-2} s^{-1}]
69	FcBuild	0,1	CO2 flux from buildings [umol m^{-2} s^{-1}]
70	FcPoint	0,1	CO2 flux from point source [umol m^{-2} s^{-1}]
71	QNSnowFr	1	Net all-wave radiation for snow-free area [W m^{-2}]
72	QNSnow	1	Net all-wave radiation for snow area [W m^{-2}]
73	AlbSnow	1	Snow albedo [-]
74	QM	1	Snow-related heat exchange [W m^{-2}]
75	QMFreeze	1	Internal energy change [W m^{-2}]
76	QMRain	1	Heat released by rain on snow [W m^{-2}]
77	SWE	1	Snow water equivalent [mm]
78	MeltWater	1	Meltwater [mm]
79	MeltWStore	1	Meltwater store [mm]
80	SnowCh	1	Change in snow pack [mm]
81	SnowRPaved	1	Snow removed from paved surface [mm]
82	SnowRBldgs	1	Snow removed from building surface [mm]
83	Ts	0,1,2	Skin temperature [°C]
84	T2	0,1,2	Air temperature at 2 m agl [°C]
85	Q2	0,1,2	Air specific humidity at 2 m agl [g kg^{-1}]
86	U10	0,1,2	Wind speed at 10 m agl [m s^{-1}]
87	RH2	0,1,2	Relative humidity at 2 m agl [%]

SSss_DailyState.txt

Contains information about the state of the surface and soil and vegetation parameters at a time resolution of one day. One file is written for each grid so it may contain multiple years.

Column	Name	Description
1	Year	Year [YYYY]
2	DOY	Day of year [DOY]
3	Hour	Hour of the last timestep of a day [HH]
4	Min	Minute of the last timestep of a day [MM]
5	HDD1_h	Heating degree days [°C d]
6	HDD2_c	Cooling degree days [°C d]
7	HDD3_Tmean	Average daily air temperature in forcing data [°C]
8	HDD4_T5d	5-day running-mean air temperature in forcing data [°C]
9	P_day	Daily total precipitation [mm]
10	DaysSR	Days since rain [days]
11	GDD_EveTr	Growing degree days for evergreen tree [°C d]
12	GDD_DecTr	Growing degree days for deciduous tree [°C d]
13	GDD_Grass	Growing degree days for grass [°C d]
14	SDD_EveTr	Senescence degree days for evergreen tree [°C d]
15	SDD_DecTr	Senescence degree days for deciduous tree [°C d]
16	SDD_Grass	Senescence degree days for grass [°C d]
17	Tmin	Daily minimum temperature in forcing data [°C]

continues on next page

Table 7.357 – continued from previous page

Column	Name	Description
18	Tmax	Daily maximum temperature in forcing data [°C]
19	DLHrs	Day length [h]
20	LAI_EveTr	Leaf area index of evergreen trees [m^{-2} m^{-2}]
21	LAI_DecTr	Leaf area index of deciduous trees [m^{-2} m^{-2}]
22	LAI_Grass	Leaf area index of grass [m^{-2} m^{-2}]
23	DecidCap	Moisture storage capacity of deciduous trees [mm]
24	Porosity	Porosity of deciduous trees [-]
25	AlbEveTr	Albedo of evergreen trees [-]
26	AlbDecTr	Albedo of deciduous trees [-]
27	AlbGrass	Albedo of grass [-]
28	WU_EveTr1	Total water use for evergreen trees [mm]
29	WU_EveTr2	Automatic water use for evergreen trees [mm]
30	WU_EveTr3	Manual water use for evergreen trees [mm]
31	WU_DecTr1	Total water use for deciduous trees [mm]
32	WU_DecTr2	Automatic water use for deciduous trees [mm]
33	WU_DecTr3	Manual water use for deciduous trees [mm]
34	WU_Grass1	Total water use for grass [mm]
35	WU_Grass2	Automatic water use for grass [mm]
36	WU_Grass3	Manual water use for grass [mm]
37	deltaLAI	Change in leaf area index (normalised 0-1) [-]
38	LAIlumps	Leaf area index used in LUMPS (normalised 0-1) [-]
39	AlbSnow	Snow albedo [-]
40	DensSnow_Paved	Snow density - paved surface [kg m^{-3}]
41	DensSnow_Bldgs	Snow density - building surface [kg m^{-3}]
42	DensSnow_EveTr	Snow density - evergreen surface [kg m^{-3}]
43	DensSnow_DecTr	Snow density - deciduous surface [kg m^{-3}]
44	DensSnow_Grass	Snow density - grass surface [kg m^{-3}]
45	DensSnow_BSoil	Snow density - bare soil surface [kg m^{-3}]
46	DensSnow_Water	Snow density - water surface [kg m^{-3}]
47	a1	OHM cofficient a1 - [-]
48	a2	OHM cofficient a2 [W m^{-2} h^{-1}]
49	a3	OHM cofficient a3 - [W m^{-2}]

InitialConditionsSSss_YYYY.nml

At the end of the model run (or the end of each year in the model run) a new InitialConditions file is written out (to the input folder) for each grid, see *Initial Conditions file*

SSss_YYYY_snow_TT.txt

SUEWS produces a separate output file for snow (when `SnowUse` = 1 in *RunControl.nml*) with details for each surface type.

File format of SSss_YYYY_snow_TT.txt

Column	Name	Description
1	iy	Year [YYYY]
2	id	Day of year [DOY]

continues on next page

Table 7.358 – continued from previous page

Column	Name	Description
3	it	Hour [H]
4	imin	Minute [M]
5	dectime	Decimal time [-]
6	SWE_Paved	Snow water equivalent – paved surface [mm]
7	SWE_Bldgs	Snow water equivalent – building surface [mm]
8	SWE_EveTr	Snow water equivalent – evergreen surface [mm]
9	SWE_DecTr	Snow water equivalent – deciduous surface [mm]
10	SWE_Grass	Snow water equivalent – grass surface [mm]
11	SWE_BSoil	Snow water equivalent – bare soil surface [mm]
12	SWE_Water	Snow water equivalent – water surface [mm]
13	Mw_Paved	Meltwater – paved surface [mm h^{-1}]
14	Mw_Bldgs	Meltwater – building surface [mm h^{-1}]
15	Mw_EveTr	Meltwater – evergreen surface [mm h^{-1}]
16	Mw_DecTr	Meltwater – deciduous surface [mm h^{-1}]
17	Mw_Grass	Meltwater – grass surface [mm h^{-1} 1]
18	Mw_BSoil	Meltwater – bare soil surface [mm h^{-1}]
19	Mw_Water	Meltwater – water surface [mm h^{-1}]
20	Qm_Paved	Snowmelt-related heat – paved surface [W m^{-2}]
21	Qm_Bldgs	Snowmelt-related heat – building surface [W m^{-2}]
22	Qm_EveTr	Snowmelt-related heat – evergreen surface [W m^{-2}]
23	Qm_DecTr	Snowmelt-related heat – deciduous surface [W m^{-2}]
24	Qm_Grass	Snowmelt-related heat – grass surface [W m^{-2}]
25	Qm_BSoil	Snowmelt-related heat – bare soil surface [W m^{-2}]
26	Qm_Water	Snowmelt-related heat – water surface [W m^{-2}]
27	Qa_Paved	Advective heat – paved surface [W m^{-2}]
28	Qa_Bldgs	Advective heat – building surface [W m^{-2}]
29	Qa_EveTr	Advective heat – evergreen surface [W m^{-2}]
30	Qa_DecTr	Advective heat – deciduous surface [W m^{-2}]
31	Qa_Grass	Advective heat – grass surface [W m^{-2}]
32	Qa_BSoil	Advective heat – bare soil surface [W m^{-2}]
33	Qa_Water	Advective heat – water surface [W m^{-2}]
34	QmFr_Paved	Heat related to freezing of surface store – paved surface [W m^{-2}]
35	QmFr_Bldgs	Heat related to freezing of surface store – building surface [W m^{-2}]
36	QmFr_EveTr	Heat related to freezing of surface store – evergreen surface [W m^{-2}]
37	QmFr_DecTr	Heat related to freezing of surface store – deciduous surface [W m^{-2}]
38	QmFr_Grass	Heat related to freezing of surface store – grass surface [W m^{-2}]
39	QmFr_BSoil	Heat related to freezing of surface store – bare soil surface [W m^{-2}]
40	QmFr_Water	Heat related to freezing of surface store – water [W m^{-2}]
41	fr_Paved	Fraction of snow – paved surface [-]
42	fr_Bldgs	Fraction of snow – building surface [-]
43	fr_EveTr	Fraction of snow – evergreen surface [-]
44	fr_DecTr	Fraction of snow – deciduous surface [-]
45	fr_Grass	Fraction of snow – grass surface [-]
46	Fr_BSoil	Fraction of snow – bare soil surface [-]
47	RainSn_Paved	Rain on snow – paved surface [mm]
48	RainSn_Bldgs	Rain on snow – building surface [mm]
49	RainSn_EveTr	Rain on snow – evergreen surface [mm]
50	RainSn_DecTr	Rain on snow – deciduous surface [mm]
51	RainSn_Grass	Rain on snow – grass surface [mm]

continues on next page

Table 7.358 – continued from previous page

Column	Name	Description
52	RainSn_BSoil	Rain on snow – bare soil surface [mm]
53	RainSn_Water	Rain on snow – water surface [mm]
54	qn_PavedSnow	Net all-wave radiation – paved surface [W m^{-2}]
55	qn_BldgsSnow	Net all-wave radiation – building surface [W m^{-2}]
56	qn_EveTrSnow	Net all-wave radiation – evergreen surface [W m^{-2}]
57	qn_DecTrSnow	Net all-wave radiation – deciduous surface [W m^{-2}]
58	qn_GrassSnow	Net all-wave radiation – grass surface [W m^{-2}]
59	qn_BSoilSnow	Net all-wave radiation – bare soil surface [W m^{-2}]
60	qn_WaterSnow	Net all-wave radiation – water surface [W m^{-2}]
61	kup_PavedSnow	Reflected shortwave radiation – paved surface [W m^{-2}]
62	kup_BldgsSnow	Reflected shortwave radiation – building surface [W m^{-2}]
63	kup_EveTrSnow	Reflected shortwave radiation – evergreen surface [W m^{-2}]
64	kup_DecTrSnow	Reflected shortwave radiation – deciduous surface [W m^{-2}]
65	kup_GrassSnow	Reflected shortwave radiation – grass surface [W m^{-2}]
66	kup_BSoilSnow	Reflected shortwave radiation – bare soil surface [W m^{-2}]
67	kup_WaterSnow	Reflected shortwave radiation – water surface [W m^{-2}]
68	frMelt_Paved	Amount of freezing melt water – paved surface [mm]
69	frMelt_Bldgs	Amount of freezing melt water – building surface [mm]
70	frMelt_EveTr	Amount of freezing melt water – evergreen surface [mm]
71	frMelt_DecTr	Amount of freezing melt water – deciduous surface [mm]
72	frMelt_Grass	Amount of freezing melt water – grass surface [mm]
73	frMelt_BSoil	Amount of freezing melt water – bare soil surface [mm]
74	frMelt_Water	Amount of freezing melt water – water surface [mm]
75	MwStore_Paved	Melt water store – paved surface [mm]
76	MwStore_Bldgs	Melt water store – building surface [mm]
77	MwStore_EveTr	Melt water store – evergreen surface [mm]
78	MwStore_DecTr	Melt water store – deciduous surface [mm]
79	MwStore_Grass	Melt water store – grass surface [mm]
80	MwStore_BSoil	Melt water store – bare soil surface [mm]
81	MwStore_Water	Melt water store – water surface [mm]
82	DensSnow_Paved	Snow density – paved surface [kg m^{-3}]
83	DensSnow_Bldgs	Snow density – building surface [kg m^{-3}]
84	DensSnow_EveTr	Snow density – evergreen surface [kg m^{-3}]
85	DensSnow_DecTr	Snow density – deciduous surface [kg m^{-3}]
86	DensSnow_Grass	Snow density – grass surface [kg m^{-3}]
87	DensSnow_BSoil	Snow density – bare soil surface [kg m^{-3}]
88	DensSnow_Water	Snow density – water surface [kg m^{-3}]
89	Sd_Paved	Snow depth – paved surface [mm]
90	Sd_Bldgs	Snow depth – building surface [mm]
91	Sd_EveTr	Snow depth – evergreen surface [mm]
92	Sd_DecTr	Snow depth – deciduous surface [mm]
93	Sd_Grass	Snow depth – grass surface [mm]
94	Sd_BSoil	Snow depth – bare soil surface [mm]
95	Sd_Water	Snow depth – water surface [mm]
96	Tsnow_Paved	Snow surface temperature – paved surface [°C]
97	Tsnow_Bldgs	Snow surface temperature – building surface [°C]
98	Tsnow_EveTr	Snow surface temperature – evergreen surface [°C]
99	Tsnow_DecTr	Snow surface temperature – deciduous surface [°C]
100	Tsnow_Grass	Snow surface temperature – grass surface [°C]

continues on next page

Table 7.358 – continued from previous page

Column	Name	Description
101	Tsnow_BSoil	Snow surface temperature – bare soil surface [°C]
102	Tsnow_Water	Snow surface temperature – water surface [°C]

SSss_YYYY_RSL_TT.txt

SUEWS produces a separate output file for wind, temperature and humidity profiles in the roughness sublayer at 30 levels (see *Wind, Temperature and Humidity Profiles in the Roughness Sublayer* level details).

File format of SSss_YYYY_RSL_TT.txt:

Column	Name	Description
1	Year	Year [YYYY]
2	DOY	Day of year [DOY]
3	Hour	Hour [H]
4	Min	Minute [M]
5	Dectime	Decimal time [-]
6	z_1	Height at level 1 [m]
7	z_2	Height at level 2 [m]
8	z_3	Height at level 3 [m]
9	z_4	Height at level 4 [m]
10	z_5	Height at level 5 [m]
11	z_6	Height at level 6 [m]
12	z_7	Height at level 7 [m]
13	z_8	Height at level 8 [m]
14	z_9	Height at level 9 [m]
15	z_10	Height at level 10 [m]
16	z_11	Height at level 11 [m]
17	z_12	Height at level 12 [m]
18	z_13	Height at level 13 [m]
19	z_14	Height at level 14 [m]
20	z_15	Height at level 15 [m]
21	z_16	Height at level 16 [m]
22	z_17	Height at level 17 [m]
23	z_18	Height at level 18 [m]
24	z_19	Height at level 19 [m]
25	z_20	Height at level 20 [m]
26	z_21	Height at level 21 [m]
27	z_22	Height at level 22 [m]
28	z_23	Height at level 23 [m]
29	z_24	Height at level 24 [m]
30	z_25	Height at level 25 [m]
31	z_26	Height at level 26 [m]
32	z_27	Height at level 27 [m]
33	z_28	Height at level 28 [m]
34	z_29	Height at level 29 [m]
35	z_30	Height at level 30 [m]
36	U_1	Wind speed at level 1 [m s^{-1}]
37	U_2	Wind speed at level 2 [m s^{-1}]
38	U_3	Wind speed at level 3 [m s^{-1}]

continues on next page

Table 7.359 – continued from previous page

Column	Name	Description
39	U_4	Wind speed at level 4 [m s^{-1}]
40	U_5	Wind speed at level 5 [m s^{-1}]
41	U_6	Wind speed at level 6 [m s^{-1}]
42	U_7	Wind speed at level 7 [m s^{-1}]
43	U_8	Wind speed at level 8 [m s^{-1}]
44	U_9	Wind speed at level 9 [m s^{-1}]
45	U_10	Wind speed at level 10 [m s^{-1}]
46	U_11	Wind speed at level 11 [m s^{-1}]
47	U_12	Wind speed at level 12 [m s^{-1}]
48	U_13	Wind speed at level 13 [m s^{-1}]
49	U_14	Wind speed at level 14 [m s^{-1}]
50	U_15	Wind speed at level 15 [m s^{-1}]
51	U_16	Wind speed at level 16 [m s^{-1}]
52	U_17	Wind speed at level 17 [m s^{-1}]
53	U_18	Wind speed at level 18 [m s^{-1}]
54	U_19	Wind speed at level 19 [m s^{-1}]
55	U_20	Wind speed at level 20 [m s^{-1}]
56	U_21	Wind speed at level 21 [m s^{-1}]
57	U_22	Wind speed at level 22 [m s^{-1}]
58	U_23	Wind speed at level 23 [m s^{-1}]
59	U_24	Wind speed at level 24 [m s^{-1}]
60	U_25	Wind speed at level 25 [m s^{-1}]
61	U_26	Wind speed at level 26 [m s^{-1}]
62	U_27	Wind speed at level 27 [m s^{-1}]
63	U_28	Wind speed at level 28 [m s^{-1}]
64	U_29	Wind speed at level 29 [m s^{-1}]
65	U_30	Wind speed at level 30 [m s^{-1}]
66	T_1	Air temperature at level 1 [°C]
67	T_2	Air temperature at level 2 [°C]
68	T_3	Air temperature at level 3 [°C]
69	T_4	Air temperature at level 4 [°C]
70	T_5	Air temperature at level 5 [°C]
71	T_6	Air temperature at level 6 [°C]
72	T_7	Air temperature at level 7 [°C]
73	T_8	Air temperature at level 8 [°C]
74	T_9	Air temperature at level 9 [°C]
75	T_10	Air temperature at level 10 [°C]
76	T_11	Air temperature at level 11 [°C]
77	T_12	Air temperature at level 12 [°C]
78	T_13	Air temperature at level 13 [°C]
79	T_14	Air temperature at level 14 [°C]
80	T_15	Air temperature at level 15 [°C]
81	T_16	Air temperature at level 16 [°C]
82	T_17	Air temperature at level 17 [°C]
83	T_18	Air temperature at level 18 [°C]
84	T_19	Air temperature at level 19 [°C]
85	T_20	Air temperature at level 20 [°C]
86	T_21	Air temperature at level 21 [°C]
87	T_22	Air temperature at level 22 [°C]

continues on next page

Table 7.359 – continued from previous page

Column	Name	Description
88	T_23	Air temperature at level 23 [°C]
89	T_24	Air temperature at level 24 [°C]
90	T_25	Air temperature at level 25 [°C]
91	T_26	Air temperature at level 26 [°C]
92	T_27	Air temperature at level 27 [°C]
93	T_28	Air temperature at level 28 [°C]
94	T_29	Air temperature at level 29 [°C]
95	T_30	Air temperature at level 30 [°C]
96	q_1	Specific humidity at level 1 [g kg^{-1}]
97	q_2	Specific humidity at level 2 [g kg^{-1}]
98	q_3	Specific humidity at level 3 [g kg^{-1}]
99	q_4	Specific humidity at level 4 [g kg^{-1}]
100	q_5	Specific humidity at level 5 [g kg^{-1}]
101	q_6	Specific humidity at level 6 [g kg^{-1}]
102	q_7	Specific humidity at level 7 [g kg^{-1}]
103	q_8	Specific humidity at level 8 [g kg^{-1}]
104	q_9	Specific humidity at level 9 [g kg^{-1}]
105	q_10	Specific humidity at level 10 [g kg^{-1}]
106	q_11	Specific humidity at level 11 [g kg^{-1}]
107	q_12	Specific humidity at level 12 [g kg^{-1}]
108	q_13	Specific humidity at level 13 [g kg^{-1}]
109	q_14	Specific humidity at level 14 [g kg^{-1}]
110	q_15	Specific humidity at level 15 [g kg^{-1}]
111	q_16	Specific humidity at level 16 [g kg^{-1}]
112	q_17	Specific humidity at level 17 [g kg^{-1}]
113	q_18	Specific humidity at level 18 [g kg^{-1}]
114	q_19	Specific humidity at level 19 [g kg^{-1}]
115	q_20	Specific humidity at level 20 [g kg^{-1}]
116	q_21	Specific humidity at level 21 [g kg^{-1}]
117	q_22	Specific humidity at level 22 [g kg^{-1}]
118	q_23	Specific humidity at level 23 [g kg^{-1}]
119	q_24	Specific humidity at level 24 [g kg^{-1}]
120	q_25	Specific humidity at level 25 [g kg^{-1}]
121	q_26	Specific humidity at level 26 [g kg^{-1}]
122	q_27	Specific humidity at level 27 [g kg^{-1}]
123	q_28	Specific humidity at level 28 [g kg^{-1}]
124	q_29	Specific humidity at level 29 [g kg^{-1}]
125	q_30	Specific humidity at level 30 [g kg^{-1}]

SSss_YYYY_BL_TT.txt

Meteorological variables modelled by CBL portion of the model are output in to this file created for each day with time step (see *CBL input files*).

Column	Name	Description	Units
1	iy	Year [YYYY]	
2	id	Day of year [DoY]	
3	it	Hour [H]	
4	imin	Minute [M]	
5	dectime	Decimal time [-]	
6	zi	Convectibe boundary layer height	m
7	Theta	Potential temperature in the inertial sublayer	K
8	Q	Specific humidity in the inertial sublayer	g kg^{-1}
9	theta+	Potential temperature just above the CBL	K
10	q+	Specific humidity just above the CBL	g kg^{-1}
11	Temp_C	Air temperature	°C
12	RH	Relative humidity	%
13	QH_use	Sensible heat flux used for calculation	W m^{-2}
14	QE_use	Latent heat flux used for calculation	W m^{-2}
15	Press_hPa	Pressure used for calculation	hPa
16	avu1	Wind speed used for calculation	m s^{-1}
17	ustar	Friction velocity used for calculation	m s^{-1}
18	avdens	Air density used for calculation	kg m^{-3}
19	lv_J_kg	Latent heat of vaporization used for calculation	J kg^{-1}
20	avcp	Specific heat capacity used for calculation	J kg^{-1} K^{-1}
21	gamt	Vertical gradient of potential temperature	K m^{-1}
22	gamq	Vertical gradient of specific humidity	kg kg^{-1} m^{-1}

SSss_YYYY_ESTM_TT.txt

If the ESTM model option is run, the following output file is created.

Note: First time steps of storage output could give NaN values during the initial converging phase.

ESTM output file format

Column	Name	Description	Units
1	iy	Year	
2	id	Day of year	
3	it	Hour	
4	imin	Minute	
5	dectime	Decimal time	
6	QSnet	Net storage heat flux (QSwall+QSground+QS)	W m^{-2}
7	QSair	Storage heat flux into air	W m^{-2}
8	QSwall	Storage heat flux into wall	W m^{-2}
9	QSroof	Storage heat flux into roof	W m^{-2}
10	QSground	Storage heat flux into ground	W m^{-2}
11	QSibld	Storage heat flux into internal elements in buildling	W m^{-2}

continues on next page

Table 7.360 – continued from previous page

Column	Name	Description	Units
12	Twall1	Temperature in the first layer of wall (outer-most)	K
13	Twall2	Temperature in the first layer of wall	K
14	Twall3	Temperature in the first layer of wall	K
15	Twall4	Temperature in the first layer of wall	K
16	Twall5	Temperature in the first layer of wall (inner-most)	K
17	Troof1	Temperature in the first layer of roof (outer-most)	K
18	Troof2	Temperature in the first layer of roof	K
19	Troof3	Temperature in the first layer of roof	K
20	Troof4	Temperature in the first layer of roof	K
21	Troof5	Temperature in the first layer of ground (inner-most)	K
22	Tground1	Temperature in the first layer of ground (outer-most)	K
23	Tground2	Temperature in the first layer of ground	K
24	Tground3	Temperature in the first layer of ground	K
25	Tground4	Temperature in the first layer of ground	K
26	Tground5	Temperature in the first layer of ground (inner-most)	K
27	Tibld1	Temperature in the first layer of internal elements	K
28	Tibld2	Temperature in the first layer of internal elements	K
29	Tibld3	Temperature in the first layer of internal elements	K
30	Tibld4	Temperature in the first layer of internal elements	K
31	Tibld5	Temperature in the first layer of internal elements	K
32	Tabld	Air temperature in buildings	K

SSss_YYYY_SPARTACUS_TT.txt

If the SPARTACUS model option is run, the following output file is created.

SPARTACUS output file format

Column	Name	Description
1	Year	Year [YYYY]
2	DOY	Day of year [DOY]
3	Hour	Hour [H]
4	Min	Minute [M]
5	Dectime	Decimal time [-]
6	alb	Albedo at top-of-canopy. Average of diffuse and direct albedos weighted by the amount of diffuse and dire
7	emis	Emissivity at top-of-canopy
8	Lemission	Longwave upward emission at top-of-canopy [Wm-2]
9	Lup	Longwave upward (emission+reflected) at top-of-canopy [Wm-2]
10	Kup	Shortwave upward (reflected) at top-of-canopy [Wm-2]
11	Qn	Net all-wave radiation at top-of-canopy [Wm-2]
12	LCAAbs1	Longwave absorption rate in clear-air part of layer 1 [Wm-2]
13	LCAAbs2	Longwave absorption rate in clear-air part of layer 2 [Wm-2]
14	LCAAbs3	Longwave absorption rate in clear-air part of layer 3 [Wm-2]
15	LCAAbs4	Longwave absorption rate in clear-air part of layer 4 [Wm-2]
16	LCAAbs5	Longwave absorption rate in clear-air part of layer 5 [Wm-2]
17	LCAAbs6	Longwave absorption rate in clear-air part of layer 6 [Wm-2]
18	LCAAbs7	Longwave absorption rate in clear-air part of layer 7 [Wm-2]
19	LCAAbs8	Longwave absorption rate in clear-air part of layer 8 [Wm-2]
20	LCAAbs9	Longwave absorption rate in clear-air part of layer 9 [Wm-2]

Table 7.361 – continued from previous page

Column	Name	Description
21	LCAAbs10	Longwave absorption rate in clear-air part of layer 10 [Wm-2]
22	LCAAbs11	Longwave absorption rate in clear-air part of layer 11 [Wm-2]
23	LCAAbs12	Longwave absorption rate in clear-air part of layer 12 [Wm-2]
24	LCAAbs13	Longwave absorption rate in clear-air part of layer 13 [Wm-2]
25	LCAAbs14	Longwave absorption rate in clear-air part of layer 14 [Wm-2]
26	LCAAbs15	Longwave absorption rate in clear-air part of layer 15 [Wm-2]
27	LWallNet1	Net longwave flux into walls in layer 1 [Wm-2]
28	LWallNet2	Net longwave flux into walls in layer 2 [Wm-2]
29	LWallNet3	Net longwave flux into walls in layer 3 [Wm-2]
30	LWallNet4	Net longwave flux into walls in layer 4 [Wm-2]
31	LWallNet5	Net longwave flux into walls in layer 5 [Wm-2]
32	LWallNet6	Net longwave flux into walls in layer 6 [Wm-2]
33	LWallNet7	Net longwave flux into walls in layer 7 [Wm-2]
34	LWallNet8	Net longwave flux into walls in layer 8 [Wm-2]
35	LWallNet9	Net longwave flux into walls in layer 9 [Wm-2]
36	LWallNet10	Net longwave flux into walls in layer 10 [Wm-2]
37	LWallNet11	Net longwave flux into walls in layer 11 [Wm-2]
38	LWallNet12	Net longwave flux into walls in layer 12 [Wm-2]
39	LWallNet13	Net longwave flux into walls in layer 13 [Wm-2]
40	LWallNet14	Net longwave flux into walls in layer 14 [Wm-2]
41	LWallNet15	Net longwave flux into walls in layer 15 [Wm-2]
42	LRfNet1	Net longwave flux into roofs in layer 1 [Wm-2]
43	LRfNet2	Net longwave flux into roofs in layer 2 [Wm-2]
44	LRfNet3	Net longwave flux into roofs in layer 3 [Wm-2]
45	LRfNet4	Net longwave flux into roofs in layer 4 [Wm-2]
46	LRfNet5	Net longwave flux into roofs in layer 5 [Wm-2]
47	LRfNet6	Net longwave flux into roofs in layer 6 [Wm-2]
48	LRfNet7	Net longwave flux into roofs in layer 7 [Wm-2]
49	LRfNet8	Net longwave flux into roofs in layer 8 [Wm-2]
50	LRfNet9	Net longwave flux into roofs in layer 9 [Wm-2]
51	LRfNet10	Net longwave flux into roofs in layer 10 [Wm-2]
52	LRfNet11	Net longwave flux into roofs in layer 11 [Wm-2]
53	LRfNet12	Net longwave flux into roofs in layer 12 [Wm-2]
54	LRfNet13	Net longwave flux into roofs in layer 13 [Wm-2]
55	LRfNet14	Net longwave flux into roofs in layer 14 [Wm-2]
56	LRfNet15	Net longwave flux into roofs in layer 15 [Wm-2]
57	LRfIn1	Longwave flux into roofs in layer 1 [Wm-2]
58	LRfIn2	Longwave flux into roofs in layer 2 [Wm-2]
59	LRfIn3	Longwave flux into roofs in layer 3 [Wm-2]
60	LRfIn4	Longwave flux into roofs in layer 4 [Wm-2]
61	LRfIn5	Longwave flux into roofs in layer 5 [Wm-2]
62	LRfIn6	Longwave flux into roofs in layer 6 [Wm-2]
63	LRfIn7	Longwave flux into roofs in layer 7 [Wm-2]
64	LRfIn8	Longwave flux into roofs in layer 8 [Wm-2]
65	LRfIn9	Longwave flux into roofs in layer 9 [Wm-2]
66	LRfIn10	Longwave flux into roofs in layer 10 [Wm-2]
67	LRfIn11	Longwave flux into roofs in layer 11 [Wm-2]
68	LRfIn12	Longwave flux into roofs in layer 12 [Wm-2]
69	LRfIn13	Longwave flux into roofs in layer 13 [Wm-2]

Table 7.361 – continued from previous page

Column	Name	Description
70	LRfIn14	Longwave flux into roofs in layer 14 [Wm-2]
71	LRfIn15	Longwave flux into roofs in layer 15 [Wm-2]
72	LTopNet	Top-of-canopy net longwave flux [Wm-2]
73	LGrndNet	Net longwave flux into the ground [Wm-2]
74	LTopDn	Top-of-canopy downwelling longwave flux [Wm-2]
75	KCAAbs1	Shortwave absorption rate in clear-air part of layer 1 [Wm-2]
76	KCAAbs2	Shortwave absorption rate in clear-air part of layer 2 [Wm-2]
77	KCAAbs3	Shortwave absorption rate in clear-air part of layer 3 [Wm-2]
78	KCAAbs4	Shortwave absorption rate in clear-air part of layer 4 [Wm-2]
79	KCAAbs5	Shortwave absorption rate in clear-air part of layer 5 [Wm-2]
80	KCAAbs6	Shortwave absorption rate in clear-air part of layer 6 [Wm-2]
81	KCAAbs7	Shortwave absorption rate in clear-air part of layer 7 [Wm-2]
82	KCAAbs8	Shortwave absorption rate in clear-air part of layer 8 [Wm-2]
83	KCAAbs9	Shortwave absorption rate in clear-air part of layer 9 [Wm-2]
84	KCAAbs10	Shortwave absorption rate in clear-air part of layer 10 [Wm-2]
85	KCAAbs11	Shortwave absorption rate in clear-air part of layer 11 [Wm-2]
86	KCAAbs12	Shortwave absorption rate in clear-air part of layer 12 [Wm-2]
87	KCAAbs13	Shortwave absorption rate in clear-air part of layer 13 [Wm-2]
88	KCAAbs14	Shortwave absorption rate in clear-air part of layer 14 [Wm-2]
89	KCAAbs15	Shortwave absorption rate in clear-air part of layer 15 [Wm-2]
90	KWallNet1	Net shortwave flux into walls in layer 1 [Wm-2]
91	KWallNet2	Net shortwave flux into walls in layer 2 [Wm-2]
92	KWallNet3	Net shortwave flux into walls in layer 3 [Wm-2]
93	KWallNet4	Net shortwave flux into walls in layer 4 [Wm-2]
94	KWallNet5	Net shortwave flux into walls in layer 5 [Wm-2]
95	KWallNet6	Net shortwave flux into walls in layer 6 [Wm-2]
96	KWallNet7	Net shortwave flux into walls in layer 7 [Wm-2]
97	KWallNet8	Net shortwave flux into walls in layer 8 [Wm-2]
98	KWallNet9	Net shortwave flux into walls in layer 9 [Wm-2]
99	KWallNet10	Net shortwave flux into walls in layer 10 [Wm-2]
100	KWallNet11	Net shortwave flux into walls in layer 11 [Wm-2]
101	KWallNet12	Net shortwave flux into walls in layer 12 [Wm-2]
102	KWallNet13	Net shortwave flux into walls in layer 13 [Wm-2]
103	KWallNet14	Net shortwave flux into walls in layer 14 [Wm-2]
104	KWallNet15	Net shortwave flux into walls in layer 15 [Wm-2]
105	KRfNet1	Net shortwave flux into roofs in layer 1 [Wm-2]
106	KRfNet2	Net shortwave flux into roofs in layer 2 [Wm-2]
107	KRfNet3	Net shortwave flux into roofs in layer 3 [Wm-2]
108	KRfNet4	Net shortwave flux into roofs in layer 4 [Wm-2]
109	KRfNet5	Net shortwave flux into roofs in layer 5 [Wm-2]
110	KRfNet6	Net shortwave flux into roofs in layer 6 [Wm-2]
111	KRfNet7	Net shortwave flux into roofs in layer 7 [Wm-2]
112	KRfNet8	Net shortwave flux into roofs in layer 8 [Wm-2]
113	KRfNet9	Net shortwave flux into roofs in layer 9 [Wm-2]
114	KRfNet10	Net shortwave flux into roofs in layer 10 [Wm-2]
115	KRfNet11	Net shortwave flux into roofs in layer 11 [Wm-2]
116	KRfNet12	Net shortwave flux into roofs in layer 12 [Wm-2]
117	KRfNet13	Net shortwave flux into roofs in layer 13 [Wm-2]
118	KRfNet14	Net shortwave flux into roofs in layer 14 [Wm-2]

Table 7.361 – continued from previous page

Column	Name	Description
119	KRfNet15	Net shortwave flux into roofs in layer 15 [Wm-2]
120	KRfIn1	Shortwave flux into roofs in layer 1 [Wm-2]
121	KRfIn2	Shortwave flux into roofs in layer 2 [Wm-2]
122	KRfIn3	Shortwave flux into roofs in layer 3 [Wm-2]
123	KRfIn4	Shortwave flux into roofs in layer 4 [Wm-2]
124	KRfIn5	Shortwave flux into roofs in layer 5 [Wm-2]
125	KRfIn6	Shortwave flux into roofs in layer 6 [Wm-2]
126	KRfIn7	Shortwave flux into roofs in layer 7 [Wm-2]
127	KRfIn8	Shortwave flux into roofs in layer 8 [Wm-2]
128	KRfIn9	Shortwave flux into roofs in layer 9 [Wm-2]
129	KRfIn10	Shortwave flux into roofs in layer 10 [Wm-2]
130	KRfIn11	Shortwave flux into roofs in layer 11 [Wm-2]
131	KRfIn12	Shortwave flux into roofs in layer 12 [Wm-2]
132	KRfIn13	Shortwave flux into roofs in layer 13 [Wm-2]
133	KRfIn14	Shortwave flux into roofs in layer 14 [Wm-2]
134	KRfIn15	Shortwave flux into roofs in layer 15 [Wm-2]
135	KTopDnDir	Direct shortwave flux into roofs [Wm-2]
136	KTopNet	Top-of-canopy net shortwave flux [Wm-2]
137	KGrndDnDir	Direct downwelling shortwave flux into the ground [Wm-2]
138	KGrndNet	Net shortwave flux into the ground [Wm-2]

7.5 Troubleshooting

7.5.1 How to report an issue of this manual?

Please submit your issue via our GitHub page.

7.5.2 How to join your email-list?

Please join our email-list here.

7.5.3 How to create a directory?

Please search the web using this phrase if you do not know how to create a folder or directory

7.5.4 How to unzip a file

Please search the web using this phrase if you do not know how to unzip a file

7.5.5 A text editor

A program to edit plain text files. If you search on the web using the phrase 'text editor' you will find numerous programs. These include for example, NotePad, EditPad, Text Pad etc

7.5.6 Command prompt

From Start select run –type cmd – this will open a window. Change directory to the location of where you stored your files. The following website may be helpful if you do not know what a command prompt is: http://dosprompt.info/

7.5.7 Day of year [DOY]

January 1st is day 1, February 1st is day 32. If you search on the web using the phrase 'day of year calendar' you will find tables that allow rapid conversions. Remember that after February 28th DOY will be different between leap years and non-leap years.

7.5.8 ESTM output

First time steps of storage output could give NaN values during the initial converging phase.

7.5.9 First things to Check if the program seems to have problems

- Check the problems.txt file.
- Check file options – in RunControl.nml.
- Look in the output directory for the SS_FileChoices.txt. This allows you to check all options that were used in the run. You may want to compare it with the original version supplied with the model.
- Note there can not be missing time steps in the data. If you need help with this you may want to checkout **`UMEP`_**

A pop-up saying "file path not found"

This means the program cannot find the file paths defined in RunControl.nml file. Possible solutions:

- Check that you have created the folder that you specified in RunControl.nml.
- Check does the output directory exist?
- Check that you have a single or double quotes around the FileInputPath, FileOutputPath and FileCode

===="%sat_vap_press.f temp=0.0000 pressure dectime"==== Temperature is zero in the calculation of water vapour pressure parameterization.

- You don't need to worry if the temperature should be (is) 0°C.
- If it should not be 0°C this suggests that there is a problem with the data.

%T changed to fit limits

- [TL =0.1]/ [TL =39.9] You may want to change the coefficients for surface resistance. If you have data from these temperatures, we would happily determine them.

%Iteration loop stopped for too stable conditions.

- [zL]/[USTAR] This warning indicates that the atmospheric stability gets above 2. In these conditions MO theory is not necessarily valid. The iteration loop to calculate the Obukhov length and friction velocity is stopped so that stability does not get too high values. This is something you do not need to worry as it does not mean wrong input data.

"Reference to undefined variable, array element or function result"

- Parameter(s) missing from input files.

See also the error messages provided in problems.txt and warnings.txt

7.6 SUEWS-related Software

7.6.1 SuPy: SUEWS that speaks Python

- **What is SuPy?**

SuPy is a Python-enhanced urban climate model with SUEWS as its computation core.

The scientific rigour in SuPy results is thus guaranteed by SUEWS (see *SUEWS publications* and *Parameterisations and sub-models within SUEWS*).

Meanwhile, the data analysis ability of SuPy is greatly enhanced by the Python-based SciPy Stack, notably numpy and pandas.

More details are described in our SuPy paper.

- **How to get SuPy?**

  SuPy is available on all major platforms (macOS, Windows, Linux) for Python 3.7+ (64-bit only) via PyPI:

  ```
  python3 -m pip install supy --upgrade
  ```

- **How to use SuPy?**
  - Please follow *Quickstart of SuPy* and *other tutorials*.
  - Please see *API reference* for details.
  - Please see *FAQ* if any issue.
- **How to contribute to SuPy?**
  - Add your development via Pull Request

- Report issues via the GitHub page.
- Cite our SuPy paper.
- Provide suggestions and feedback.

Tutorials

To familiarise users with SuPy urban climate modelling and to demonstrate the functionality of SuPy, we provide the following tutorials in Jupyter notebooks:

Quickstart of SuPy

This quickstart demonstrates the essential and simplest workflow of `supy` in SUEWS simulation:

1. *load input files*
2. *run simulation*
3. *examine results*

More advanced use of `supy` are available in the *tutorials*

Before we start, we need to load the following necessary packages.

[1]:
```
import matplotlib.pyplot as plt
import supy as sp
import pandas as pd
import numpy as np
from pathlib import Path

%matplotlib inline
```

```
/opt/homebrew/Caskroom/mambaforge/base/envs/supy/lib/python3.9/site-packages/pandas/core/
↪reshape/merge.py:916: FutureWarning: In a future version, the Index constructor will␣
↪not infer numeric dtypes when passed object-dtype sequences (matching Series behavior)
  key_col = Index(lvals).where(~mask_left, rvals)
```

[2]:
```
sp.show_version()
```

```
SuPy versions
-------------
supy: 2022.9.19.dev1-dirty
supy_driver: 2021a15
```

Load input files

For existing SUEWS users:

First, a path to SUEWS `RunControl.nml` should be specified, which will direct `supy` to locate input files.

[3]:
```
path_runcontrol = Path('../sample_run') / 'RunControl.nml'
```

[4]:
```
df_state_init = sp.init_supy(path_runcontrol)
```

```
2022-09-20 23:04:30,116 - SuPy - INFO - All cache cleared.
```

A sample `df_state_init` looks below (note that `.T` is used here to produce a nicer tableform view):

[5]:
```
df_state_init.filter(like='method').T
```

[5]:
```
grid                              1
var                       ind_dim
aerodynamicresistancemethod 0       2
basetmethod               0         1
evapmethod                0         2
diagmethod                0         2
emissionsmethod           0         2
netradiationmethod        0         3
roughlenheatmethod        0         2
roughlenmommethod         0         2
smdmethod                 0         0
stabilitymethod           0         3
storageheatmethod         0         1
waterusemethod            0         0
```

Following the convention of SUEWS, `supy` loads meteorological forcing (met-forcing) files at the grid level.

[6]:
```
grid = df_state_init.index[0]
df_forcing = sp.load_forcing_grid(path_runcontrol, grid)
# by default, two years of forcing data are included;
# to save running time for demonstration, we only use one year in this demo
df_forcing=df_forcing.loc['2012'].iloc[1:]
```

```
2022-09-20 23:04:30,731 - SuPy - INFO - All cache cleared.
```

For new users to SUEWS/SuPy:

To ease the input file preparation, a helper function `load_SampleData` is provided to get the sample input for SuPy simulations

[7]:
```
df_state_init, df_forcing = sp.load_SampleData()
grid = df_state_init.index[0]
# by default, two years of forcing data are included;
# to save running time for demonstration, we only use one year in this demo
df_forcing=df_forcing.loc['2012'].iloc[1:]
```

```
2022-09-20 23:04:34,398 - SuPy - INFO - All cache cleared.
```

Overview of SuPy input

df_state_init

df_state_init includes model Initial state consisting of:

- surface characteristics (e.g., albedo, emissivity, land cover fractions, etc.; full details refer to SUEWS documentation)
- model configurations (e.g., stability; full details refer to SUEWS documentation)

Detailed description of variables in df_state_init refers to *SuPy input*

Surface land cover fraction information in the sample input dataset:

[8]:
```
df_state_init.loc[:,['bldgh','evetreeh','dectreeh']]
```

[8]:
```
var     bldgh evetreeh dectreeh
ind_dim     0        0        0
grid
1        22.0     13.1     13.1
```

[9]:
```
df_state_init.filter(like='sfr_surf')
```

[9]:
```
var     sfr_surf
ind_dim     (0,)  (1,) (2,)  (3,)  (4,) (5,)  (6,)
grid
1           0.43  0.38  0.0  0.02  0.03  0.0  0.14
```

df_forcing

df_forcing includes meteorological and other external forcing information.

Detailed description of variables in df_forcing refers to *SuPy input*.

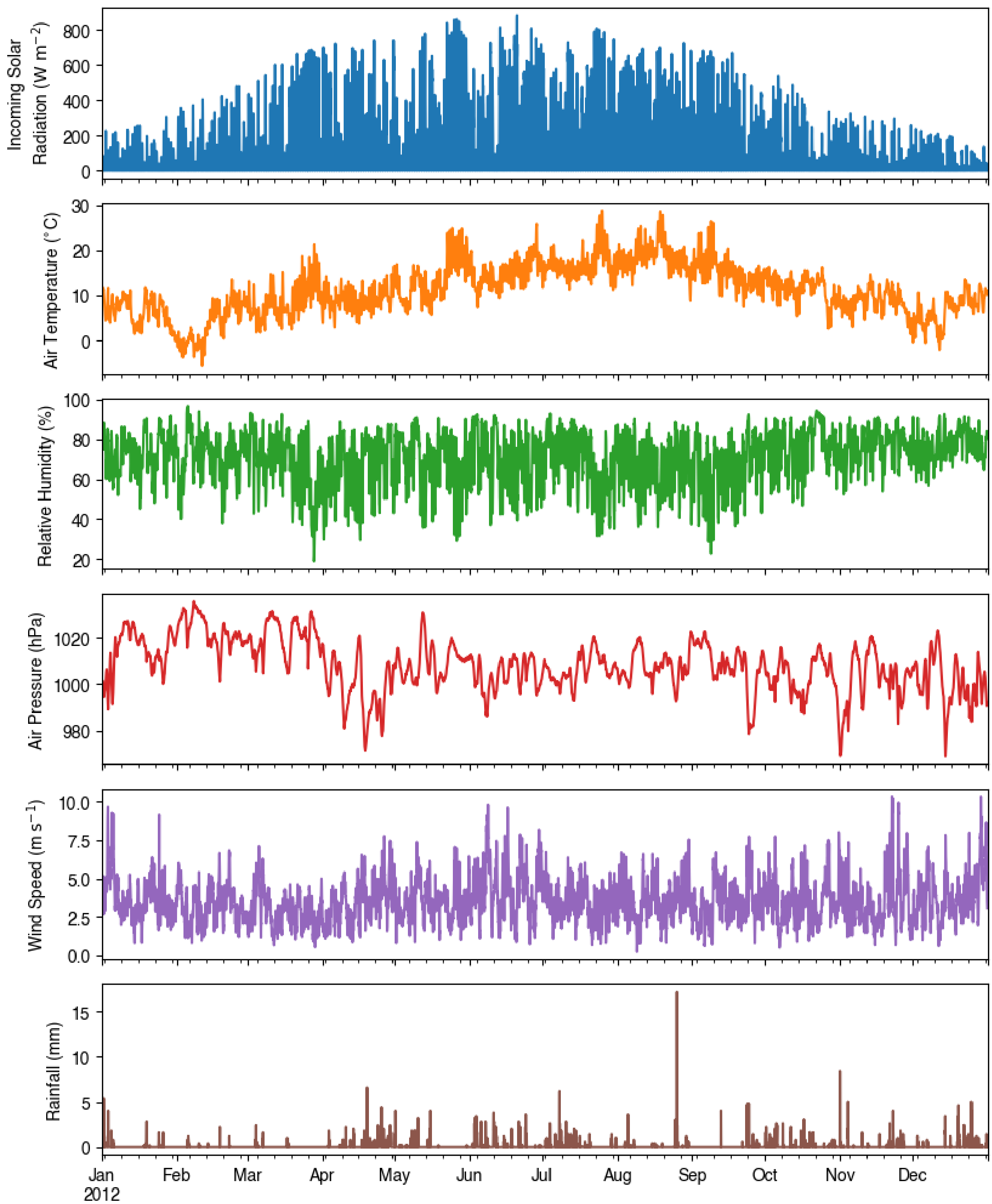
Below is an overview of forcing variables of the sample data set used in the following simulations.

[10]:
```
list_var_forcing = [
    "kdown",
    "Tair",
    "RH",
    "pres",
    "U",
    "rain",
]
dict_var_label = {
    "kdown": "Incoming Solar\n Radiation ($ \mathrm{W \ m^{-2}}$)",
    "Tair": "Air Temperature ($^{\circ}}$C)",
    "RH": r"Relative Humidity (%)",
    "pres": "Air Pressure (hPa)",
    "rain": "Rainfall (mm)",
    "U": "Wind Speed (m $\mathrm{s^{-1}}$)",
}
df_plot_forcing_x = (
    df_forcing.loc[:, list_var_forcing].copy().shift(-1).dropna(how="any")
```

(continues on next page)

(continued from previous page)

```
)
df_plot_forcing = df_plot_forcing_x.resample("1h").mean()
df_plot_forcing["rain"] = df_plot_forcing_x["rain"].resample("1h").sum()

axes = df_plot_forcing.plot(subplots=True, figsize=(8, 12), legend=False,)
fig = axes[0].figure
fig.tight_layout()
fig.autofmt_xdate(bottom=0.2, rotation=0, ha="center")
for ax, var in zip(axes, list_var_forcing):
    _ = ax.set_ylabel(dict_var_label[var])
```


Incoming Solar Radiation (W m^{-2})
800
600
400
200
0
Air Temperature (°C)
30
20
10
0
Relative Humidity (%)
100
80
60
40
20
Air Pressure (hPa)
1020
1000
980
Wind Speed (m s^{-1})
10.0
7.5
5.0
2.5
0.0
Rainfall (mm)
15
10
5
0
Jan
2012
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Modification of SuPy input

Given `pandas.DataFrame` is the core data structure of SuPy, all operations, including modification, output, demonstration, etc., on SuPy inputs (`df_state_init` and `df_forcing`) can be done using `pandas`-based functions/methods.

Specifically, for modification, the following operations are essential:

locating data

Data can be located in two ways, namely: 1. by name via `.loc <http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#selection-by-label>`__; 2. by position via `.iloc <http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#selection-by-position>`__.

[11]:
```
# view the surface fraction variable: `sfr`
df_state_init.loc[:,'sfr_surf']
```

[11]:
```
ind_dim  (0,)  (1,)  (2,)  (3,)  (4,)  (5,)  (6,)
grid
1        0.43  0.38   0.0  0.02  0.03   0.0  0.14
```

[12]:
```
# view the second row of `df_forcing`, which is a pandas Series
df_forcing.iloc[1]
```

[12]:
```
iy       2012.000000
id          1.000000
it          0.000000
imin       10.000000
qn       -999.000000
qh       -999.000000
qe       -999.000000
qs       -999.000000
qf       -999.000000
U           5.176667
RH         86.195000
Tair       11.620000
pres     1001.833333
rain        0.000000
kdown       0.173333
snow     -999.000000
ldown    -999.000000
fcld     -999.000000
Wuh         0.000000
xsmd     -999.000000
lai      -999.000000
kdiff    -999.000000
kdir     -999.000000
wdir     -999.000000
isec        0.000000
Name: 2012-01-01 00:10:00, dtype: float64
```

[13]:
```
# view a particular position of `df_forcing`, which is a value
df_forcing.iloc[8,9]
```

[13]:
```
4.78
```

setting new values

Setting new values is very straightforward: after locating the variables/data to modify, just set the new values accordingly:

[14]:
```
# modify surface fractions
df_state_init.loc[:,'sfr_surf']=[.1,.1,.2,.3,.25,.05,0]
# check the updated values
df_state_init.loc[:,'sfr_surf']
```

[14]:
```
ind_dim  (0,)  (1,)  (2,)  (3,)  (4,)  (5,)  (6,)
grid
1         0.1   0.1   0.2   0.3  0.25  0.05     0
```

Run simulations

Once met-forcing (via `df_forcing`) and initial conditions (via `df_state_init`) are loaded in, we call `sp.run_supy` to conduct a SUEWS simulation, which will return two `pandas` `DataFrames`: `df_output` and `df_state`.

[15]:
```
df_output, df_state_final = sp.run_supy(df_forcing, df_state_init)
```

```
2022-09-20 23:04:39,021 - SuPy - INFO - ====================
2022-09-20 23:04:39,021 - SuPy - INFO - Simulation period:
2022-09-20 23:04:39,022 - SuPy - INFO -   Start: 2012-01-01 00:05:00
2022-09-20 23:04:39,022 - SuPy - INFO -   End: 2012-12-31 23:55:00
2022-09-20 23:04:39,023 - SuPy - INFO -
2022-09-20 23:04:39,023 - SuPy - INFO - No. of grids: 1
2022-09-20 23:04:39,024 - SuPy - INFO - SuPy is running in serial mode
2022-09-20 23:04:49,343 - SuPy - INFO - Execution time: 10.3 s
2022-09-20 23:04:49,344 - SuPy - INFO - ====================
```

`df_output`

`df_output` is an ensemble output collection of major SUEWS output groups, including:

- SUEWS: the essential SUEWS output variables
- DailyState: variables of daily state information
- snow: snow output variables (effective when `snowuse = 1` set in `df_state_init`)

Detailed description of variables in `df_output` refers to SuPy output

[16]:
```
df_output.columns.levels[0]
```

[16]:
```
Index(['SUEWS', 'snow', 'RSL', 'BEERS', 'debug', 'SPARTACUS', 'ESTMExt',
       'DailyState'],
      dtype='object', name='group')
```

df_state_final

df_state_final is a DataFrame for holding:

1. all model states if save_state is set to True when calling sp.run_supy (supy may run significantly slower for a large simulation);
2. or, only the final state if save_state is set to False (the default setting), in which mode supy has a similar performance as the standalone compiled SUEWS executable.

Entries in df_state_final have the same data structure as df_state_init and can thus be used for other SUEWS simulations starting at the timestamp as in df_state_final.

Detailed description of variables in df_state_final refers to SuPy output

[17]:
```
df_state_final.T.head()
```

[17]:
```
datetime                  2012-01-01 00:05:00 2013-01-01 00:00:00
grid                                        1                   1
var              ind_dim
ah_min           (0,)                    15.0                15.0
                 (1,)                    15.0                15.0
ah_slope_cooling (0,)                     2.7                 2.7
                 (1,)                     2.7                 2.7
ah_slope_heating (0,)                     2.7                 2.7
```

Examine results

Thanks to the functionality inherited from pandas and other packages under the PyData stack, compared with the standard SUEWS simulation workflow, supy enables more convenient examination of SUEWS results by statistics calculation, resampling, plotting (and many more).

Ouptut structure

df_output is organised with MultiIndex (grid,timestamp) and (group,varaible) as index and columns, respectively.

[18]:
```
df_output.head()
```

[18]:
```
group                        SUEWS                                      \
var                          Kdown      Kup       Ldown         Lup
grid datetime
1    2012-01-01 00:05:00  0.176667  0.02332  344.179805  371.582645
     2012-01-01 00:10:00  0.173333  0.02288  344.190048  371.657938
     2012-01-01 00:15:00  0.170000  0.02244  344.200308  371.733243
     2012-01-01 00:20:00  0.166667  0.02200  344.210586  371.808562
     2012-01-01 00:25:00  0.163333  0.02156  344.220882  371.883893

group                                                                   \
var                          Tsurf         QN          QF         QS
grid datetime
1    2012-01-01 00:05:00  11.607452 -27.249493  40.574001 -6.382243
     2012-01-01 00:10:00  11.622405 -27.317436  39.724283 -6.228797
     2012-01-01 00:15:00  11.637359 -27.385375  38.874566 -6.082788
```

(continues on next page)

(continued from previous page)

```
     2012-01-01 00:20:00  11.652312 -27.453309  38.024849 -5.943907
     2012-01-01 00:25:00  11.667265 -27.521237  37.175131 -5.811855

group                                         ...     DailyState  \
var                                QH         QE  ... DensSnow_Paved
grid datetime                                 ...
1    2012-01-01 00:05:00  19.664156  0.042594  ...            NaN
     2012-01-01 00:10:00  18.593922  0.041722  ...            NaN
     2012-01-01 00:15:00  17.531131  0.040849  ...            NaN
     2012-01-01 00:20:00  16.475472  0.039975  ...            NaN
     2012-01-01 00:25:00  15.426648  0.039101  ...            NaN

group                                                                 \
var                      DensSnow_Bldgs DensSnow_EveTr DensSnow_DecTr
grid datetime
1    2012-01-01 00:05:00            NaN            NaN            NaN
     2012-01-01 00:10:00            NaN            NaN            NaN
     2012-01-01 00:15:00            NaN            NaN            NaN
     2012-01-01 00:20:00            NaN            NaN            NaN
     2012-01-01 00:25:00            NaN            NaN            NaN

group                                                                         \
var                      DensSnow_Grass DensSnow_BSoil DensSnow_Water  a1  a2
grid datetime
1    2012-01-01 00:05:00            NaN            NaN            NaN NaN NaN
     2012-01-01 00:10:00            NaN            NaN            NaN NaN NaN
     2012-01-01 00:15:00            NaN            NaN            NaN NaN NaN
     2012-01-01 00:20:00            NaN            NaN            NaN NaN NaN
     2012-01-01 00:25:00            NaN            NaN            NaN NaN NaN

group
var                       a3
grid datetime
1    2012-01-01 00:05:00 NaN
     2012-01-01 00:10:00 NaN
     2012-01-01 00:15:00 NaN
     2012-01-01 00:20:00 NaN
     2012-01-01 00:25:00 NaN

[5 rows x 924 columns]
```

Here we demonstrate several typical scenarios for SUEWS results examination.

The essential SUEWS output collection is extracted as a separate variable for easier processing in the following sections. More advanced slicing techniques are available in pandas documentation.

[19]:
```
df_output_suews = df_output['SUEWS']
```

Statistics Calculation

We can use the `.describe()` method for a quick overview of the key surface energy balance budgets.

```
[20]: df_output_suews.loc[:, ['QN', 'QS', 'QH', 'QE', 'QF']].describe()
```

```
[20]: var              QN             QS             QH             QE  \
count  105407.000000  105407.000000  105407.000000  105407.000000
mean       39.883231       5.830107      62.666636      50.411038
std       132.019300      49.161894      77.074237      78.484562
min       -86.331686     -75.287258    -177.705269       0.000000
25%       -42.499510     -27.895414      16.069451       0.676206
50%       -25.749393      -8.183901      43.844985      14.712552
75%        74.815479      19.121287      85.722951      69.135212
max       679.848644     237.932439     480.795771     624.179069

var              QF
count  105407.000000
mean       79.024549
std        31.231867
min        26.327536
25%        50.058031
50%        82.883410
75%       104.812507
max       160.023207
```

Plotting

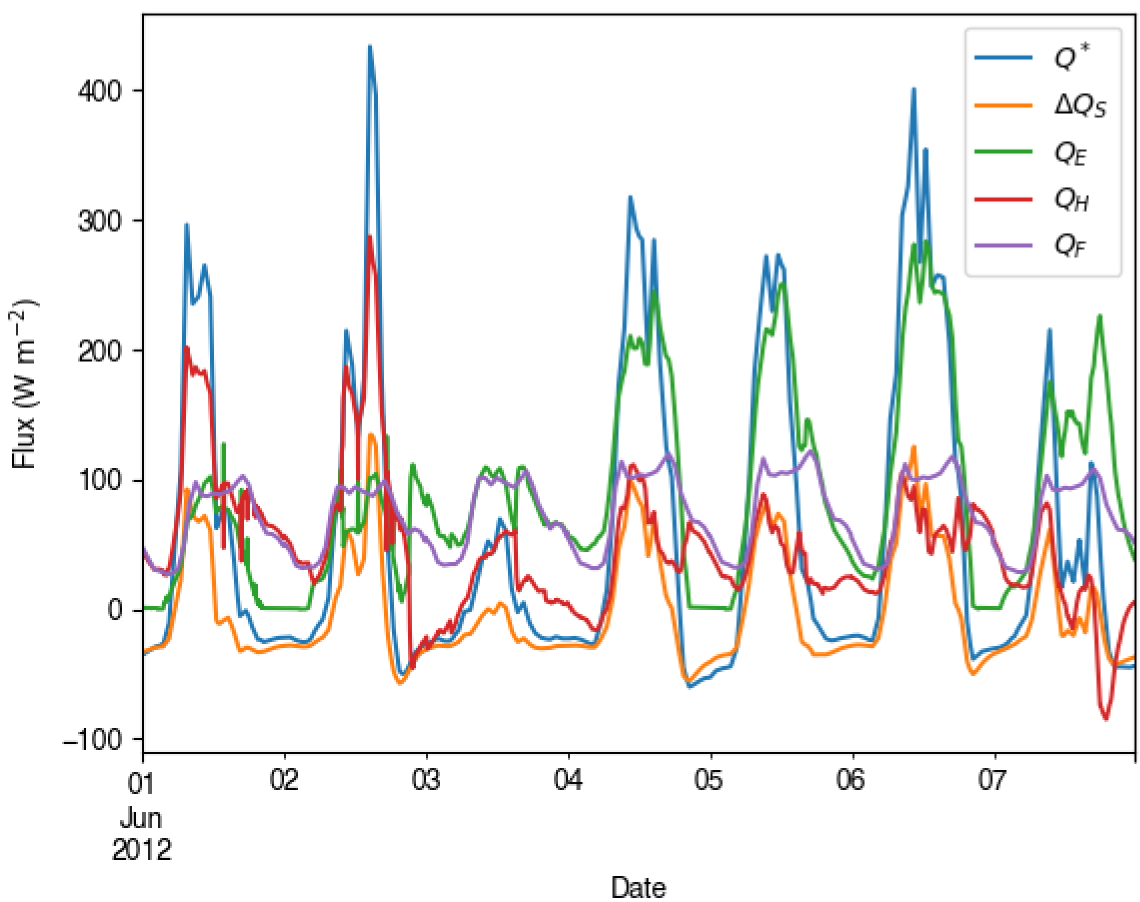
Basic example

Plotting is very straightforward via the `.plot` method bounded with `pandas.DataFrame`. Note the usage of `loc` for two slices of the output `DataFrame`.

```
[21]: # a dict for better display variable names
dict_var_disp = {
    'QN': '$Q^*$',
    'QS': r'$\Delta Q_S$',
    'QE': '$Q_E$',
    'QH': '$Q_H$',
    'QF': '$Q_F$',
    'Kdown': r'$K_{\downarrow}$',
    'Kup': r'$K_{\uparrow}$',
    'Ldown': r'$L_{\downarrow}$',
    'Lup': r'$L_{\uparrow}$',
    'Rain': '$P$',
    'Irr': '$I$',
    'Evap': '$E$',
    'RO': '$R$',
    'TotCh': '$\Delta S$',
}
```

Quick look at the simulation results:

[22]:
```
ax_output = df_output_suews\
    .loc[grid]\
    .loc['2012 6 1':'2012 6 7',
         ['QN', 'QS', 'QE', 'QH', 'QF']]\
    .rename(columns=dict_var_disp)\
    .plot()
_ = ax_output.set_xlabel('Date')
_ = ax_output.set_ylabel('Flux ($ \mathrm{W \ m^{-2}}$)')
_ = ax_output.legend()
```



More examples

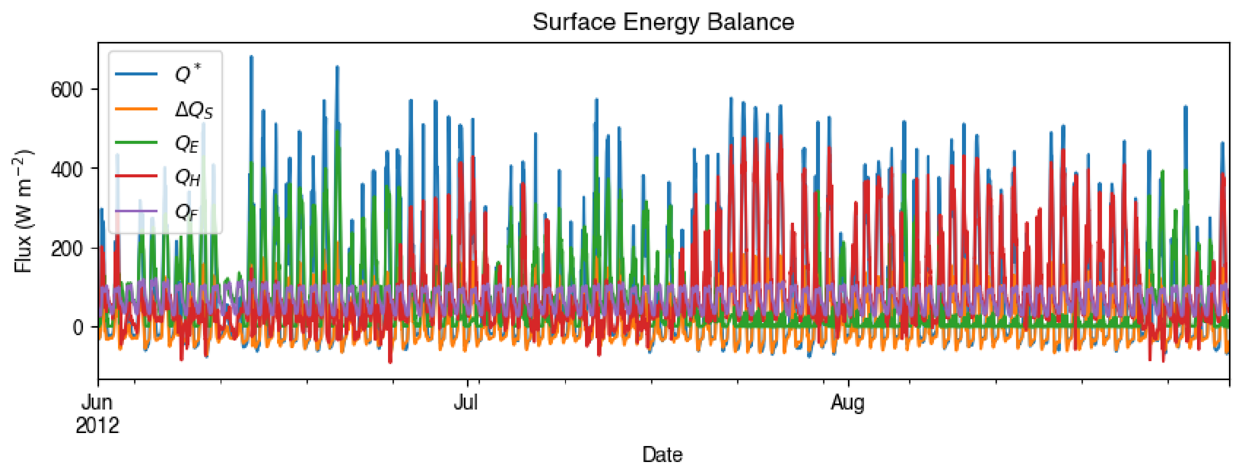
Below is a more complete example for examination of urban energy balance over the whole summer (June to August).

[23]:
```
# energy balance
ax_output = (
    df_output_suews.loc[grid]
    .loc["2012 6":"2012 8", ["QN", "QS", "QE", "QH", "QF"]]
    .rename(columns=dict_var_disp)
    .plot(figsize=(10, 3), title="Surface Energy Balance",)
)
_ = ax_output.set_xlabel("Date")
```

(continues on next page)

(continued from previous page)

```
_ = ax_output.set_ylabel("Flux ($ \mathrm{W \ m^{-2}}$)")
_ = ax_output.legend()
```



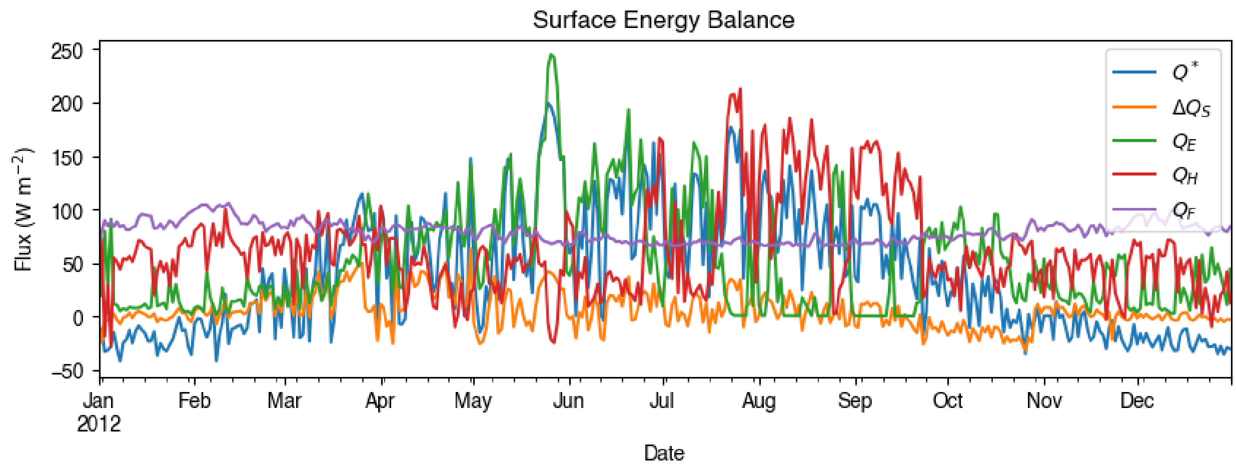
Resampling

The suggested runtime/simulation frequency of SUEWS is `300 s`, which usually results in a large output and may be over-weighted for storage and analysis. Also, you may feel an apparent slowdown in producing the above figure as a large amount of data were used for the plotting. To slim down the result size for analysis and output, we can `resample` the default output very easily.

[24]:
```
rsmp_1d = df_output_suews.loc[grid].resample("1d")
# daily mean values
df_1d_mean = rsmp_1d.mean()
# daily sum values
df_1d_sum = rsmp_1d.sum()
```

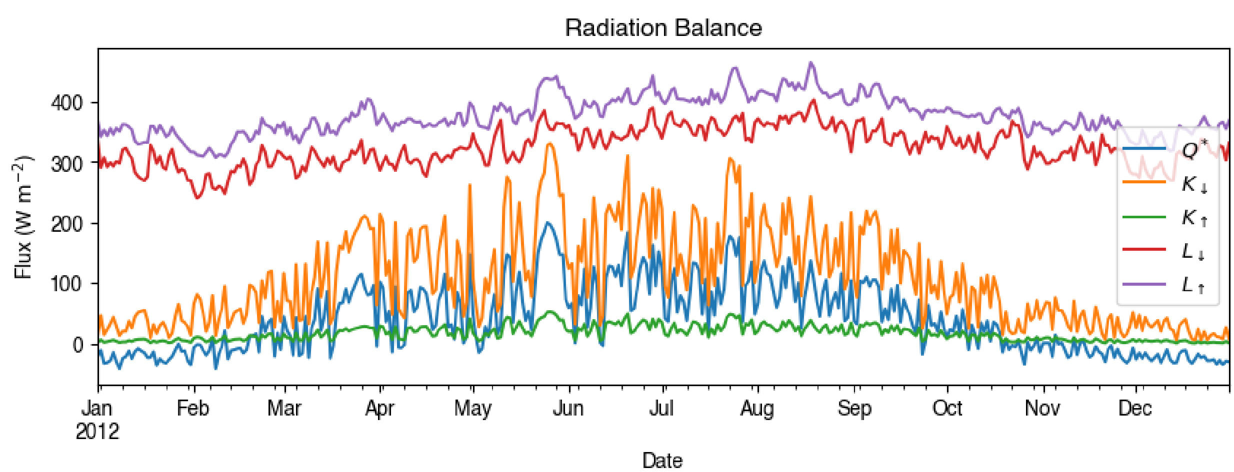
We can then re-examine the above energy balance at hourly scale and plotting will be significantly faster.

[25]:
```
# energy balance
ax_output = (
    df_1d_mean.loc[:, ["QN", "QS", "QE", "QH", "QF"]]
    .rename(columns=dict_var_disp)
    .plot(figsize=(10, 3), title="Surface Energy Balance",)
)
_ = ax_output.set_xlabel("Date")
_ = ax_output.set_ylabel("Flux ($ \mathrm{W \ m^{-2}}$)")
_ = ax_output.legend()
```

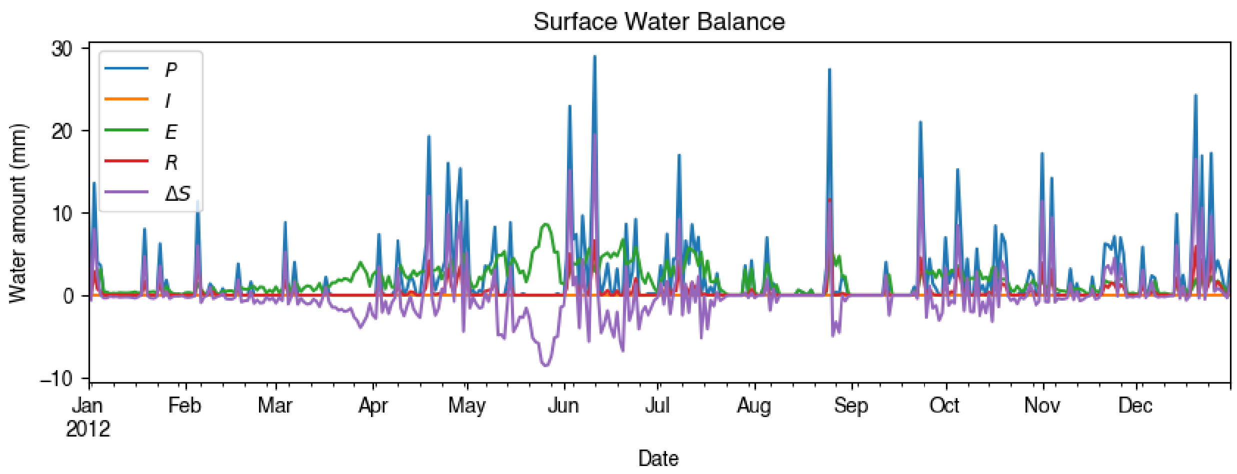


Then we use the hourly results for other analyses.

[26]:
```
# radiation balance
ax_output = (
    df_1d_mean.loc[:, ["QN", "Kdown", "Kup", "Ldown", "Lup"]]
    .rename(columns=dict_var_disp)
    .plot(figsize=(10, 3), title="Radiation Balance",)
)
_ = ax_output.set_xlabel("Date")
_ = ax_output.set_ylabel("Flux ($ \mathrm{W \ m^{-2}}$)")
_ = ax_output.legend()
```



[27]:
```
# water balance
ax_output = (
    df_1d_sum.loc[:, ["Rain", "Irr", "Evap", "RO", "TotCh"]]
    .rename(columns=dict_var_disp)
    .plot(figsize=(10, 3), title="Surface Water Balance",)
)
_ = ax_output.set_xlabel("Date")
_ = ax_output.set_ylabel("Water amount (mm)")
_ = ax_output.legend()
```



Get an overview of partitioning in energy and water balance at monthly scales:

[28]:
```
# get a monthly Resampler
df_plot = df_output_suews.loc[grid].copy()
df_plot.index = df_plot.index.set_names("Month")
rsmp_1M = df_plot.shift(-1).dropna(how="all").resample("1M", kind="period")
# mean values
df_1M_mean = rsmp_1M.mean()
# sum values
df_1M_sum = rsmp_1M.sum()
```

[29]:
```
# month names
name_mon = [x.strftime("%b") for x in rsmp_1M.groups]
# create subplots showing two panels together
fig, axes = plt.subplots(2, 1, sharex=True)
# surface energy balance
df_1M_mean.loc[:, ["QN", "QS", "QE", "QH", "QF"]].rename(columns=dict_var_disp).plot(
    ax=axes[0],  # specify the axis for plotting
    figsize=(10, 6),  # specify figure size
    title="Surface Energy Balance",
    kind="bar",
)
# surface water balance
df_1M_sum.loc[:, ["Rain", "Irr", "Evap", "RO", "TotCh"]].rename(
    columns=dict_var_disp
).plot(
    ax=axes[1],  # specify the axis for plotting
    title="Surface Water Balance",
    kind="bar",
)

# annotations
_ = axes[0].set_ylabel("Mean Flux ($ \mathrm{W \ m^{-2}}$)")
_ = axes[0].legend()
_ = axes[1].set_xlabel("Month")
_ = axes[1].set_ylabel("Total Water Amount (mm)")
```

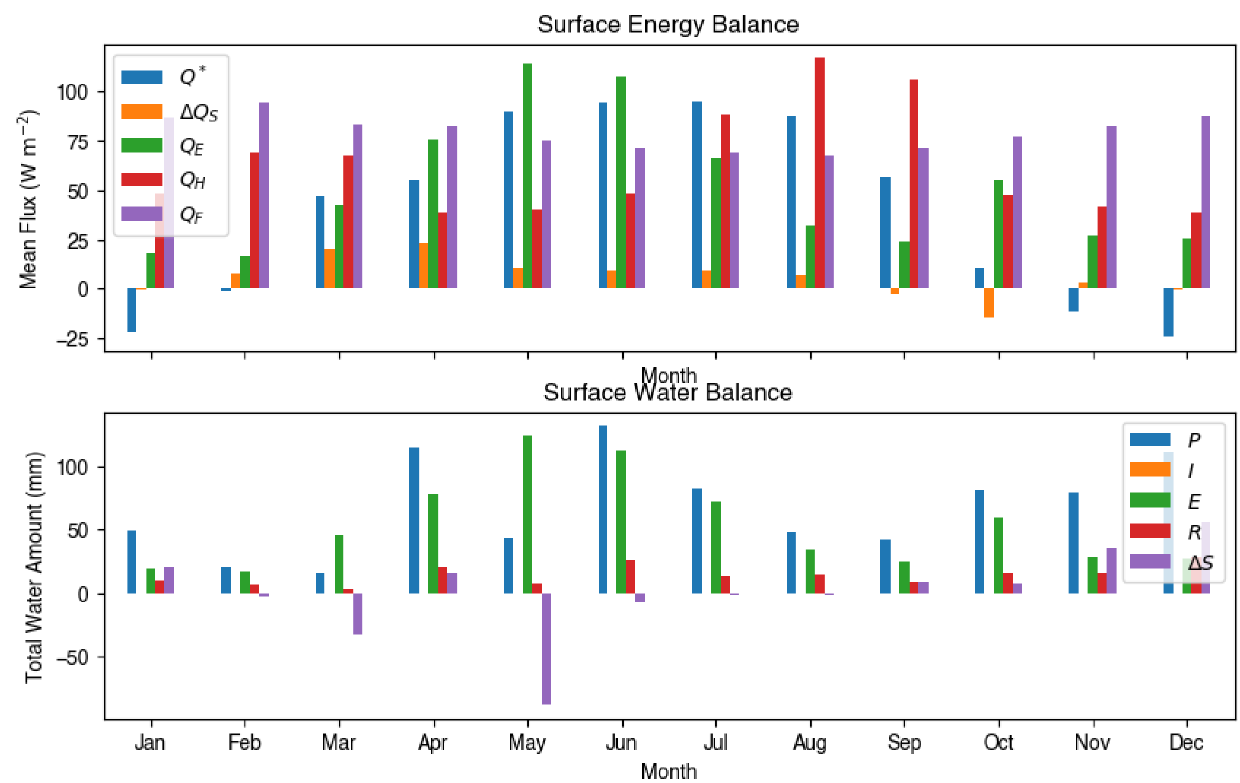
(continues on next page)

(continued from previous page)

```
_ = axes[1].xaxis.set_ticklabels(name_mon, rotation=0)
_ = axes[1].legend()
```

[29]: <AxesSubplot:title={'center':'Surface Energy Balance'}, xlabel='Month'>

[29]: <AxesSubplot:title={'center':'Surface Water Balance'}, xlabel='Month'>



Output

The supy output can be saved as `txt` files for further analysis using supy function `save_supy`.

[30]:
```
df_output
```

[30]:
```
group                         SUEWS                                      \
var                           Kdown      Kup        Ldown         Lup
grid datetime
1    2012-01-01 00:05:00    0.176667  0.02332  344.179805  371.582645
     2012-01-01 00:10:00    0.173333  0.02288  344.190048  371.657938
     2012-01-01 00:15:00    0.170000  0.02244  344.200308  371.733243
     2012-01-01 00:20:00    0.166667  0.02200  344.210586  371.808562
     2012-01-01 00:25:00    0.163333  0.02156  344.220882  371.883893
...                              ...      ...         ...         ...
     2012-12-31 23:35:00    0.000000  0.00000  330.263407  363.676342
     2012-12-31 23:40:00    0.000000  0.00000  330.263407  363.676342
     2012-12-31 23:45:00    0.000000  0.00000  330.263407  363.676342
     2012-12-31 23:50:00    0.000000  0.00000  330.263407  363.676342
     2012-12-31 23:55:00    0.000000  0.00000  330.263407  363.676342
```

(continues on next page)

(continued from previous page)

```
group                                                                   \
var                          Tsurf         QN         QF        QS
grid datetime
1    2012-01-01 00:05:00  11.607452 -27.249493  40.574001 -6.382243
     2012-01-01 00:10:00  11.622405 -27.317436  39.724283 -6.228797
     2012-01-01 00:15:00  11.637359 -27.385375  38.874566 -6.082788
     2012-01-01 00:20:00  11.652312 -27.453309  38.024849 -5.943907
     2012-01-01 00:25:00  11.667265 -27.521237  37.175131 -5.811855
...                             ...        ...        ...       ...
     2012-12-31 23:35:00  10.140000 -33.412935  53.348682 -4.399144
     2012-12-31 23:40:00  10.140000 -33.412935  52.422737 -4.397669
     2012-12-31 23:45:00  10.140000 -33.412935  51.496792 -4.395831
     2012-12-31 23:50:00  10.140000 -33.412935  50.570847 -4.393681
     2012-12-31 23:55:00  10.140000 -33.412935  46.174492 -4.391264

group                                           ...     DailyState  \
var                             QH         QE  ... DensSnow_Paved
grid datetime                                   ...
1    2012-01-01 00:05:00  19.664156   0.042594  ...            NaN
     2012-01-01 00:10:00  18.593922   0.041722  ...            NaN
     2012-01-01 00:15:00  17.531131   0.040849  ...            NaN
     2012-01-01 00:20:00  16.475472   0.039975  ...            NaN
     2012-01-01 00:25:00  15.426648   0.039101  ...            NaN
...                             ...        ...  ...            ...
     2012-12-31 23:35:00   0.904146  23.430745  ...            NaN
     2012-12-31 23:40:00   0.394992  23.012479  ...            NaN
     2012-12-31 23:45:00  -0.121686  22.601374  ...            NaN
     2012-12-31 23:50:00  -0.645680  22.197273  ...            NaN
     2012-12-31 23:55:00  -2.949124  20.101945  ...            0.0

group                                                                     \
var                      DensSnow_Bldgs DensSnow_EveTr DensSnow_DecTr
grid datetime
1    2012-01-01 00:05:00            NaN            NaN            NaN
     2012-01-01 00:10:00            NaN            NaN            NaN
     2012-01-01 00:15:00            NaN            NaN            NaN
     2012-01-01 00:20:00            NaN            NaN            NaN
     2012-01-01 00:25:00            NaN            NaN            NaN
...                                 ...            ...            ...
     2012-12-31 23:35:00            NaN            NaN            NaN
     2012-12-31 23:40:00            NaN            NaN            NaN
     2012-12-31 23:45:00            NaN            NaN            NaN
     2012-12-31 23:50:00            NaN            NaN            NaN
     2012-12-31 23:55:00            0.0            0.0            0.0

group                                                                     \
var                      DensSnow_Grass DensSnow_BSoil DensSnow_Water
grid datetime
1    2012-01-01 00:05:00            NaN            NaN            NaN
     2012-01-01 00:10:00            NaN            NaN            NaN
     2012-01-01 00:15:00            NaN            NaN            NaN
```

(continues on next page)

(continued from previous page)

```
     2012-01-01 00:20:00            NaN            NaN            NaN
     2012-01-01 00:25:00            NaN            NaN            NaN
...                                 ...            ...            ...
     2012-12-31 23:35:00            NaN            NaN            NaN
     2012-12-31 23:40:00            NaN            NaN            NaN
     2012-12-31 23:45:00            NaN            NaN            NaN
     2012-12-31 23:50:00            NaN            NaN            NaN
     2012-12-31 23:55:00            0.0            0.0            0.0

group
var                             a1      a2      a3
grid datetime
1    2012-01-01 00:05:00       NaN     NaN     NaN
     2012-01-01 00:10:00       NaN     NaN     NaN
     2012-01-01 00:15:00       NaN     NaN     NaN
     2012-01-01 00:20:00       NaN     NaN     NaN
     2012-01-01 00:25:00       NaN     NaN     NaN
...                            ...     ...     ...
     2012-12-31 23:35:00       NaN     NaN     NaN
     2012-12-31 23:40:00       NaN     NaN     NaN
     2012-12-31 23:45:00       NaN     NaN     NaN
     2012-12-31 23:50:00       NaN     NaN     NaN
     2012-12-31 23:55:00   0.36935  0.3242  8.0995

[105407 rows x 924 columns]
```

[31]:
```
list_path_save = sp.save_supy(df_output, df_state_final)
```

[32]:
```
for file_out in list_path_save:
    print(file_out.name)
```

```
1_2012_DailyState.txt
1_2012_SUEWS_60.txt
1_2012_RSL_60.txt
1_2012_BEERS_60.txt
1_2012_debug_60.txt
1_2012_ESTMExt_60.txt
df_state.csv
```

Impact Studies Using SuPy

Aim

In this tutorial, we aim to perform sensitivity analysis using supy in a parallel mode to investigate the impacts on urban climate of

1. surface properties: the physical attributes of land covers (e.g., albedo, water holding capacity, etc.)
2. background climate: longterm meteorological conditions (e.g., air temperature, precipitation, etc.)

load supy and sample dataset

[1]:
```
from dask import dataframe as dd
import supy as sp

import pandas as pd
import numpy as np

from time import time
```

```
/opt/homebrew/Caskroom/mambaforge/base/envs/supy/lib/python3.9/site-packages/pandas/core/
↪reshape/merge.py:916: FutureWarning: In a future version, the Index constructor will␣
↪not infer numeric dtypes when passed object-dtype sequences (matching Series behavior)
  key_col = Index(lvals).where(~mask_left, rvals)
```

[2]:
```
# load sample datasets
df_state_init, df_forcing = sp.load_SampleData()

# by default, two years of forcing data are included;
# to save running time for demonstration, we only use one year in this demo
df_forcing=df_forcing.loc['2012'].iloc[1:]

# perform an example run to get output samples for later use
df_output, df_state_final = sp.run_supy(df_forcing, df_state_init)
```

```
2022-06-15 21:25:48,325 - SuPy - INFO - All cache cleared.
2022-06-15 21:25:49,399 - SuPy - INFO - ====================
2022-06-15 21:25:49,399 - SuPy - INFO - Simulation period:
2022-06-15 21:25:49,400 - SuPy - INFO -   Start: 2012-01-01 00:05:00
2022-06-15 21:25:49,400 - SuPy - INFO -   End: 2012-12-31 23:55:00
2022-06-15 21:25:49,400 - SuPy - INFO -
2022-06-15 21:25:49,401 - SuPy - INFO - No. of grids: 1
2022-06-15 21:25:49,401 - SuPy - INFO - SuPy is running in serial mode
2022-06-15 21:25:54,675 - SuPy - INFO - Execution time: 5.3 s
2022-06-15 21:25:54,676 - SuPy - INFO - ====================
```

Surface properties: surface albedo

Examine the default albedo values loaded from the sample dataset

[3]:
```
df_state_init.alb
```

[3]:

ind_dim	(0,)	(1,)	(2,)	(3,)	(4,)	(5,)	(6,)
grid							
1	0.1	0.12	0.1	0.18	0.21	0.18	0.1

Copy the initial condition `DataFrame` to have a *clean slate* for our study

Note: `DataFrame.copy()` defaults to `deepcopy`

[16]:
```
df_state_init_test = df_state_init.copy()
```

Set the `Bldg` land cover to 100% for this study

[17]:
```
df_state_init_test.sfr_surf = 0
df_state_init_test.loc[:, ('sfr_surf', '(1,)')] = 1
df_state_init_test.sfr_surf
```

[17]:
```
ind_dim  (0,)  (1,)  (2,)  (3,)  (4,)  (5,)  (6,)
grid
1           0     1     0     0     0     0     0
```

Construct a `df_state_init_x` dataframe to perform `supy` simulations with specified albedo

[18]:
```
# create a `df_state_init_x` with different surface properties
n_test = 48
list_alb_test = np.linspace(0.1, 0.8, n_test).round(2)
df_state_init_x = df_state_init_test.append(
    [df_state_init_test]*(n_test-1), ignore_index=True)

# here we modify surface albedo
df_state_init_x.loc[:, ('alb', '(1,)')] = list_alb_test
df_state_init_x.index=df_state_init_x.index.rename('grid')
```

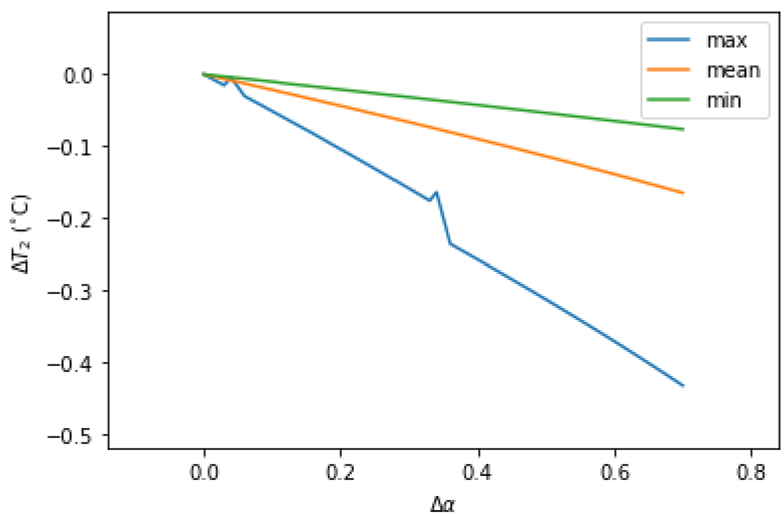
Conduct simulations with `supy`

[20]:
```
df_forcing_part = df_forcing.loc["2012 01":"2012 07"]
df_res_alb_test, df_state_final_x = sp.run_supy(
    df_forcing_part,
    df_state_init_x,
    logging_level=90,
)
```

Examine the simulation results

[21]:
```
# choose results of July 2012 for analysis
df_res_alb_test_july = df_res_alb_test.SUEWS.unstack(0).loc["2012 7"]
df_res_alb_T2_stat = df_res_alb_test_july.T2.describe()
df_res_alb_T2_diff = df_res_alb_T2_stat.transform(
    lambda x: x - df_res_alb_T2_stat.iloc[:, 0]
)
df_res_alb_T2_diff.columns = list_alb_test - list_alb_test[0]
```

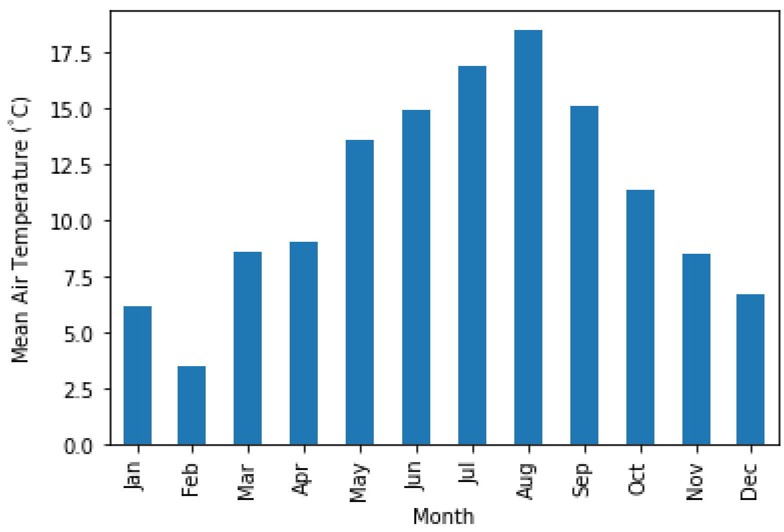
[22]:
```
ax_temp_diff = df_res_alb_T2_diff.loc[["max", "mean", "min"]].T.plot()
_ = ax_temp_diff.set_ylabel("$\Delta T_2$ ($^{\circ}$C)")
_ = ax_temp_diff.set_xlabel(r"$\Delta\alpha$")
ax_temp_diff.margins(x=0.2, y=0.2)
```



Background climate: air temperature

Examine the monthly climatology of air temperature loaded from the sample dataset

[23]:
```
df_plot = df_forcing.Tair.loc["2012"].resample("1m").mean()
ax_temp = df_plot.plot.bar(color="tab:blue")
_ = ax_temp.set_xticklabels(df_plot.index.strftime("%b"))
_ = ax_temp.set_ylabel("Mean Air Temperature ($^\degree$C)")
_ = ax_temp.set_xlabel("Month")
```



Construct a function to perform parallel `supy` simulations with specified `diff_airtemp_test`: the difference in air temperature between the one used in simulation and loaded from sample dataset.

Note

forcing data `df_forcing` has different data structure from `df_state_init`; so we need to modify `run_supy_mgrids` to implement a `run_supy_mclims` for different climate scenarios*

Let's start the implementation of `run_supy_mclims` with a small problem of four forcing groups (i.e., climate scenarios), where the air temperatures differ from the baseline scenario with a constant bias.

[24]:
```
# save loaded sample datasets
df_forcing_part_test = df_forcing.loc['2012 1':'2012 7'].copy()
df_state_init_test = df_state_init.copy()
```

[25]:
```
from dask import delayed
# create a dict with four forcing conditions as a test
n_test = 4
list_TairDiff_test = np.linspace(0., 2, n_test).round(2)
dict_df_forcing_x = {
    tairdiff: df_forcing_part_test.copy()
    for tairdiff in list_TairDiff_test}
for tairdiff in dict_df_forcing_x:
    dict_df_forcing_x[tairdiff].loc[:, 'Tair'] += tairdiff

dd_forcing_x = {
    k: delayed(sp.run_supy)(df, df_state_init_test,logging_level=90)[0]
    for k, df in dict_df_forcing_x.items()}


df_res_tairdiff_test0 = delayed(pd.concat)(
    dd_forcing_x,
```

(continues on next page)

(continued from previous page)

```
    keys=list_TairDiff_test,
    names=['tairdiff'],
)
```

[26]:
```
# test the performance of a parallel run
t0 = time()
df_res_tairdiff_test = df_res_tairdiff_test0\
    .compute(scheduler='threads')\
    .reset_index('grid', drop=True)
t1 = time()
t_par = t1 - t0
print(f'Execution time: {t_par:.2f} s')
```

```
Execution time: 12.16 s
```

[27]:
```
# function for multi-climate `run_supy`
# wrapping the above code into one
def run_supy_mclims(df_state_init, dict_df_forcing_mclims):
    dd_forcing_x = {
        k: delayed(sp.run_supy)(df, df_state_init_test,logging_level=90)[0]
        for k, df in dict_df_forcing_x.items()}
    df_output_mclims0 = delayed(pd.concat)(
        dd_forcing_x,
        keys=list(dict_df_forcing_x.keys()),
        names=['clm'],
    ).compute(scheduler='threads')
    df_output_mclims = df_output_mclims0.reset_index('grid', drop=True)

    return df_output_mclims
```

Construct `dict_df_forcing_x` with multiple forcing `DataFrames`

[28]:
```
# save loaded sample datasets
df_forcing_part_test = df_forcing.loc['2012 1':'2012 7'].copy()
df_state_init_test = df_state_init.copy()

# create a dict with a number of forcing conditions
n_test = 12 # can be set with a smaller value to save simulation time
list_TairDiff_test = np.linspace(0., 2, n_test).round(2)
dict_df_forcing_x = {
    tairdiff: df_forcing_part_test.copy()
    for tairdiff in list_TairDiff_test}
for tairdiff in dict_df_forcing_x:
    dict_df_forcing_x[tairdiff].loc[:, 'Tair'] += tairdiff
```

Perform simulations

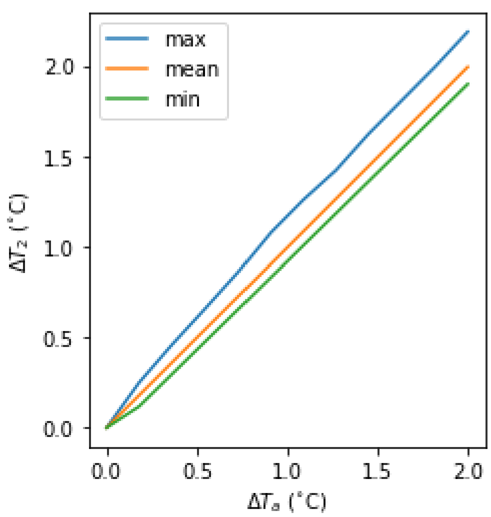
[29]:
```
# run parallel simulations using `run_supy_mclims`
t0 = time()
df_airtemp_test_x = run_supy_mclims(df_state_init_test, dict_df_forcing_x)
t1 = time()
t_par = t1-t0
print(f'Execution time: {t_par:.2f} s')
```

```
Execution time: 35.35 s
```

Examine the results

[30]:
```
df_airtemp_test = df_airtemp_test_x.SUEWS.unstack(0)
df_temp_diff = df_airtemp_test.T2.transform(lambda x: x - df_airtemp_test.T2[0.0])
df_temp_diff_ana = df_temp_diff.loc["2012 7"]
df_temp_diff_stat = df_temp_diff_ana.describe().loc[["max", "mean", "min"]].T
```

[31]:
```
ax_temp_diff_stat=df_temp_diff_stat.plot()
_=ax_temp_diff_stat.set_ylabel('$\\Delta T_2$ ($^{\\circ}$C)')
_=ax_temp_diff_stat.set_xlabel('$\\Delta T_{a}$ ($^{\\circ}$C)')
ax_temp_diff_stat.set_aspect('equal')
```



The T_2 results indicate the increased T_a has different impacts on the T_2 metrics (minimum, mean and maximum) but all increase linearly with T_a. The maximum T_2 has the stronger response compared to the other metrics.

Interaction between SuPy and external models

Introduction

SUEWS can be coupled to other models that provide or require forcing data using the SuPy single timestep running mode. We demonstrate this feature with a simple online anthropogenic heat flux model.

Anthropogenic heat flux (Q_F) is an additional term to the surface energy balance in urban areas associated with human activities (Gabey et al., 2018; Grimmond, 1992; Nie et al., 2014; 2016; Sailor, 2011). In most cities, the largest emission source is from buildings (Hamilton et al., 2009; Iamarino et al., 2011; Sailor, 2011) and is highly dependent on outdoor ambient air temperature.

load necessary packages

[1]:
```
import supy as sp
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import seaborn as sns

%matplotlib inline
# sp.show_version()
```

run `SUEWS` with default settings

[2]:
```
# load sample run dataset
df_state_init, df_forcing = sp.load_SampleData()
# turn off the snow module as unnecessary at the sample site
df_state_init.loc[:, "snowuse"] = 0

# copy `df_state_init` as the basis for later simulations
df_state_init_def = df_state_init.copy()

# by default, two years of forcing data are included;
# to save running time for demonstration, we only use one year in this demo
df_forcing = df_forcing.loc["2012"].iloc[1:]

# set QF as zero for later comparison
df_forcing_def = df_forcing.copy()
grid = df_state_init_def.index[0]
df_state_init_def.loc[:, "emissionsmethod"] = 0
df_forcing_def["qf"] = 0



# run supy
df_output, df_state = sp.run_supy(df_forcing_def, df_state_init_def)
df_output_def = df_output.loc[grid, "SUEWS"]
```

```
2020-07-06 10:55:01,875 - SuPy - INFO - All cache cleared.
2020-07-06 10:55:05,017 - SuPy - INFO - ====================
2020-07-06 10:55:05,018 - SuPy - INFO - Simulation period:
2020-07-06 10:55:05,019 - SuPy - INFO -   Start: 2012-01-01 00:05:00
2020-07-06 10:55:05,019 - SuPy - INFO -   End: 2012-12-31 23:55:00
2020-07-06 10:55:05,020 - SuPy - INFO -
2020-07-06 10:55:05,021 - SuPy - INFO - No. of grids: 1
2020-07-06 10:55:05,021 - SuPy - INFO - SuPy is running in serial mode
2020-07-06 10:55:20,390 - SuPy - INFO - Execution time: 15.4 s
2020-07-06 10:55:20,391 - SuPy - INFO - ====================
```

a simple QF model: `QF_simple`

model description

For demonstration purposes we have created a very simple model instead of using the SUEWS Q_F (Järvi et al. 2011) with feedback from outdoor air temperature. The simple Q_F model considers only building heating and cooling:

$$Q_F = \begin{cases} (T_2 - T_C) \times C_B, \; T_2 > T_C \\ (T_H - T_2) \times H_B, \; T_2 < T_H \\ Q_{F0} \end{cases}$$

where T_C (T_H) is the cooling (heating) threshold temperature of buildings, $_B$ ($_B$) is the building cooling (heating) rate, and $_{F0}$ is the baseline anthropogenic heat. The parameters used are: $_C$ ($_H$) set as 20 °C (10 °C), $_B$ ($_B$) set as 1.5 $\mathrm{W\ m^{-2}\ K^{-1}}$ (3 $\mathrm{W\ m^{-2}\ K^{-1}}$) and Q_{F0} is set as 0 $\mathrm{W\ m^{-2}}$, implying other building activities (e.g. lighting, water heating, computers) are zero and therefore do not change the temperature or change with temperature.

implementation

[3]:
```python
def QF_simple(T2):
    qf_cooling = (T2-20)*5 if T2 > 20 else 0
    qf_heating = (10-T2)*10 if T2 < 10 else 0
    qf_res = np.max([qf_heating, qf_cooling])*0.3
    return qf_res
```

Visualise the `QF_simple` model:

[4]:
```python
ser_temp = pd.Series(np.arange(-5, 45, 0.5),
                     index=np.arange(-5, 45, 0.5)).rename('temp_C')
ser_qf_heating = ser_temp.loc[-5:10].map(QF_simple).rename(
    r'heating:$(T_H-T_a) \times H_B$')
ser_qf_cooling = ser_temp.loc[20:45].map(QF_simple).rename(
    r'cooling: $(T_a-T_C) \times C_B$')
ser_qf_zero = ser_temp.loc[10:20].map(QF_simple).rename('baseline: $Q_{F0}$')
df_temp_qf = pd.concat([ser_temp, ser_qf_cooling, ser_qf_heating, ser_qf_zero],
                       axis=1).set_index('temp_C')
ax_qf_func = df_temp_qf.plot()
_=ax_qf_func.set_xlabel('$T_2$ ($^\circ$C)')
_=ax_qf_func.set_ylabel('$Q_F$ ($ \mathrm{W \ m^{-2}}$)')
```

(continues on next page)

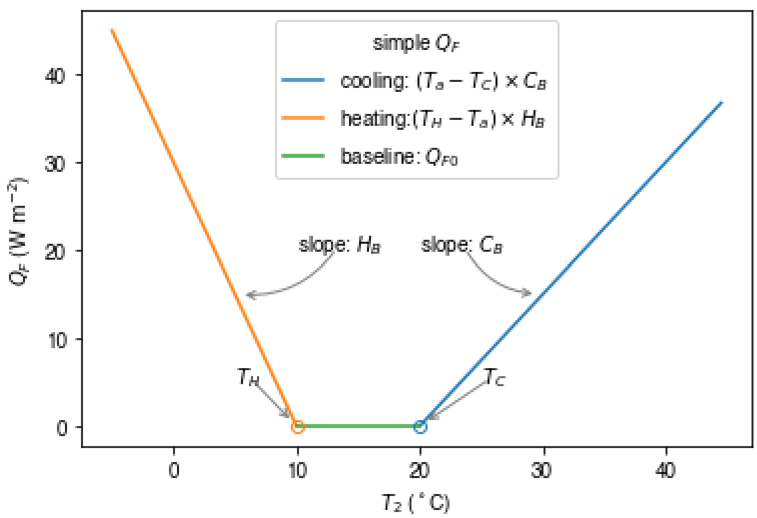
(continued from previous page)

```
_=ax_qf_func.legend(title='simple $Q_F$')
_=ax_qf_func.annotate(
    "$T_C$",
    xy=(20, 0),
    xycoords='data',
    xytext=(25, 5),
    textcoords='data',
    arrowprops=dict(
        arrowstyle="->",
        color="0.5",
        shrinkA=5,
        shrinkB=5,
        patchA=None,
        patchB=None,
        connectionstyle='arc3',
    ),
)

_=ax_qf_func.annotate(
    "$T_H$",
    xy=(10, 0),
    xycoords='data',
    xytext=(5, 5),
    textcoords='data',
    arrowprops=dict(
        arrowstyle="->",
        color="0.5",
        shrinkA=5,
        shrinkB=5,
        patchA=None,
        patchB=None,
        connectionstyle='arc3',
    ),
)
_=ax_qf_func.annotate(
    "slope: $C_B$",
    xy=(30, QF_simple(30)),
    xycoords='data',
    xytext=(20, 20),
    textcoords='data',
    arrowprops=dict(
        arrowstyle="->",
        color="0.5",
        shrinkA=5,
        shrinkB=5,
        patchA=None,
        patchB=None,
        connectionstyle='arc3, rad=0.3',
    ),
)
_=ax_qf_func.annotate(
    "slope: $H_B$",
```

(continues on next page)

(continued from previous page)

```
    xy=(5, QF_simple(5)),
    xycoords='data',
    xytext=(10, 20),
    textcoords='data',
    arrowprops=dict(
        arrowstyle="->",
        color="0.5",
        shrinkA=5,
        shrinkB=5,
        patchA=None,
        patchB=None,
        connectionstyle='arc3, rad=-0.3',
    ),
)
_=ax_qf_func.plot(10, 0, 'o', color='C1', fillstyle='none')
_ = ax_qf_func.plot(20, 0, 'o', color='C0', fillstyle='none')
```



communication between supy and QF_simple

construct a new coupled function

The coupling between the simple Q_F model and SuPy is done via the low-level function `suews_cal_tstep`, which is an interface function in charge of communications between SuPy frontend and the calculation kernel. By setting SuPy to receive external Q_F as forcing, at each timestep, the simple Q_F model is driven by the SuPy output T_2 and provides SuPy with Q_F, which thus forms a two-way coupled loop.

[5]:
```
# load extra low-level functions from supy to construct interactive functions
from supy._post import pack_df_output, pack_df_state
from supy._run import suews_cal_tstep, pack_grid_dict


def run_supy_qf(df_forcing_test, df_state_init_test):
    grid = df_state_init_test.index[0]
```

(continues on next page)

(continued from previous page)

```
    df_state_init_test.loc[grid, 'emissionsmethod'] = 0

    df_forcing_test = df_forcing_test\
        .assign(
            metforcingdata_grid=0,
            ts5mindata_ir=0,
        )\
        .rename(
            # remanae is a workaround to resolve naming inconsistency between
            # suews fortran code interface and input forcing file headers
            columns={
                '%' + 'iy': 'iy',
                'id': 'id',
                'it': 'it',
                'imin': 'imin',
                'qn': 'qn1_obs',
                'qh': 'qh_obs',
                'qe': 'qe',
                'qs': 'qs_obs',
                'qf': 'qf_obs',
                'U': 'avu1',
                'RH': 'avrh',
                'Tair': 'temp_c',
                'pres': 'press_hpa',
                'rain': 'precip',
                'kdown': 'avkdn',
                'snow': 'snowfrac_obs',
                'ldown': 'ldown_obs',
                'fcld': 'fcld_obs',
                'Wuh': 'wu_m3',
                'xsmd': 'xsmd',
                'lai': 'lai_obs',
                'kdiff': 'kdiff',
                'kdir': 'kdir',
                'wdir': 'wdir',
            }
        )

    t2_ext = df_forcing_test.iloc[0].temp_c
    qf_ext = QF_simple(t2_ext)

    # initialise dicts for holding results
    dict_state = {}
    dict_output = {}

    # starting tstep
    t_start = df_forcing_test.index[0]
    # convert df to dict with `itertuples` for better performance
    dict_forcing = {
        row.Index: row._asdict()
        for row in df_forcing_test.itertuples()
    }
```

(continues on next page)

(continued from previous page)

```
    # dict_state is used to save model states for later use
    dict_state = {(t_start, grid): pack_grid_dict(series_state_init)
                  for grid, series_state_init in df_state_init_test.iterrows()}

    # just use a single grid run for the test coupling
    for tstep in df_forcing_test.index:
        # load met forcing at `tstep`
        met_forcing_tstep = dict_forcing[tstep]
        # inject `qf_ext` to `met_forcing_tstep`
        met_forcing_tstep['qf_obs'] = qf_ext

        # update model state
        dict_state_start = dict_state[(tstep, grid)]

        dict_state_end, dict_output_tstep = suews_cal_tstep(
            dict_state_start, met_forcing_tstep)
        # the fourth to the last is `T2` stored in the result array
        t2_ext = dict_output_tstep['dataoutlinesuews'][-4]
        qf_ext = QF_simple(t2_ext)

        dict_output.update({(tstep, grid): dict_output_tstep})
        dict_state.update({(tstep + tstep.freq, grid): dict_state_end})

    # pack results as easier DataFrames
    df_output_test = pack_df_output(dict_output).swaplevel(0, 1)
    df_state_test = pack_df_state(dict_state).swaplevel(0, 1)
    return df_output_test.loc[grid, 'SUEWS'], df_state_test
```

simulations for summer and winter months

The simulation using SuPy coupled is performed for London 2012. The data analysed are a summer (July) and a winter (December) month. Initially Q_F is 0 $\mathrm{W\ m^{-2}}$ the T_2 is determined and used to determine $Q_{F[1]}$ which in turn modifies $T_{2[1]}$ and therefore modifies $Q_{F[2]}$ and the diagnosed $T_{2[2]}$.

spin-up run (January to June) for summer simulation

[6]:
```
df_output_june, df_state_jul = sp.run_supy(
    df_forcing.loc[:'2012 6'], df_state_init)
```

```
2020-07-06 10:55:20,909 - SuPy - INFO - ====================
2020-07-06 10:55:20,909 - SuPy - INFO - Simulation period:
2020-07-06 10:55:20,910 - SuPy - INFO -   Start: 2012-01-01 00:05:00
2020-07-06 10:55:20,911 - SuPy - INFO -   End: 2012-06-30 23:55:00
2020-07-06 10:55:20,911 - SuPy - INFO -
2020-07-06 10:55:20,912 - SuPy - INFO - No. of grids: 1
2020-07-06 10:55:20,913 - SuPy - INFO - SuPy is running in serial mode
2020-07-06 10:55:27,168 - SuPy - INFO - Execution time: 6.3 s
2020-07-06 10:55:27,169 - SuPy - INFO - ====================
```

spin-up run (July to October) for winter simulation

[7]:
```
df_output_oct, df_state_dec = sp.run_supy(
    df_forcing.loc['2012 7':'2012 11'], df_state_jul)
```

```
2020-07-06 10:55:27,176 - SuPy - INFO - ====================
2020-07-06 10:55:27,177 - SuPy - INFO - Simulation period:
2020-07-06 10:55:27,178 - SuPy - INFO -   Start: 2012-07-01 00:00:00
2020-07-06 10:55:27,179 - SuPy - INFO -   End: 2012-11-30 23:55:00
2020-07-06 10:55:27,179 - SuPy - INFO -
2020-07-06 10:55:27,180 - SuPy - INFO - No. of grids: 1
2020-07-06 10:55:27,181 - SuPy - INFO - SuPy is running in serial mode
2020-07-06 10:55:33,139 - SuPy - INFO - Execution time: 6.0 s
2020-07-06 10:55:33,140 - SuPy - INFO - ====================
```

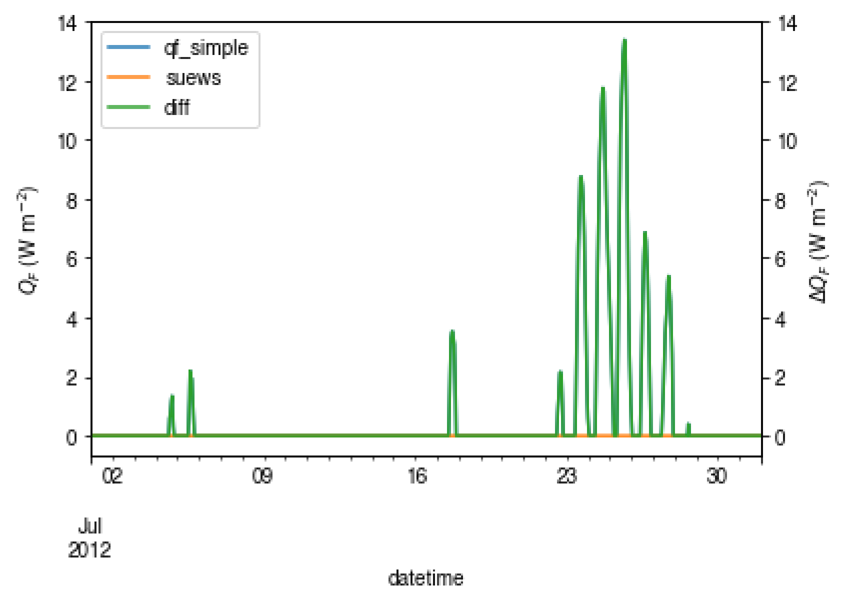
coupled simulation

[8]:
```
df_output_test_summer, df_state_summer_test = run_supy_qf(
    df_forcing.loc["2012-07"], df_state_jul.copy()
)
df_output_test_winter, df_state_winter_test = run_supy_qf(
    df_forcing.loc["2012-12"], df_state_dec.copy()
)
```

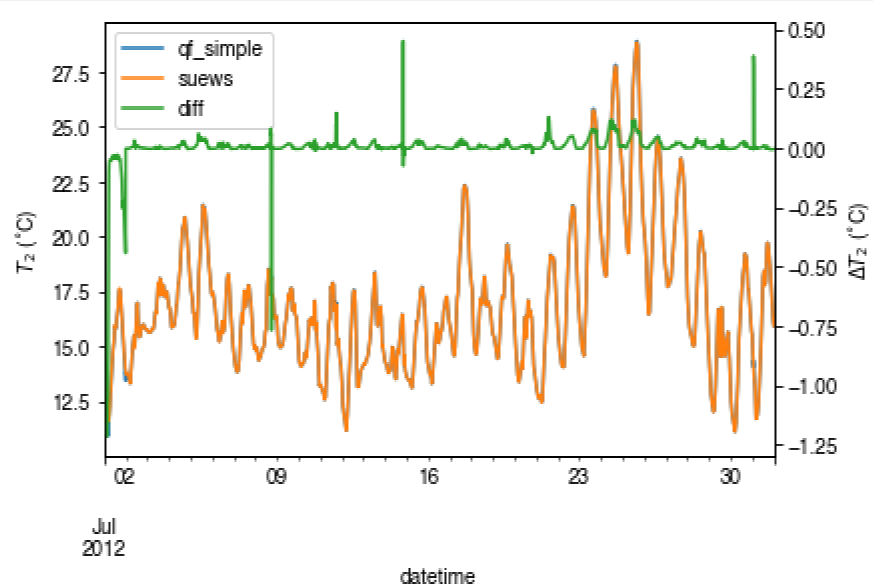
examine the results

sumer

[9]:
```
var = "QF"
var_label = "$Q_F$ ($ \mathrm{W \ m^{-2}}$)"
var_label_right = "$\Delta Q_F$ ($ \mathrm{W \ m^{-2}}$)"
period = "2012-07"
df_test = df_output_test_summer
y1 = df_test.loc[period, var].rename("qf_simple")
y2 = df_output_def.loc[period, var].rename("suews")
y3 = (y1 - y2).rename("diff")
df_plot = pd.concat([y1, y2, y3], axis=1)
ax = df_plot.plot(secondary_y="diff")
_ = ax.set_ylabel(var_label)
_ = ax.right_ax.set_ylabel(var_label_right)
lines = ax.get_lines() + ax.right_ax.get_lines()
_ = ax.legend(lines, [l.get_label() for l in lines], loc="best")
```

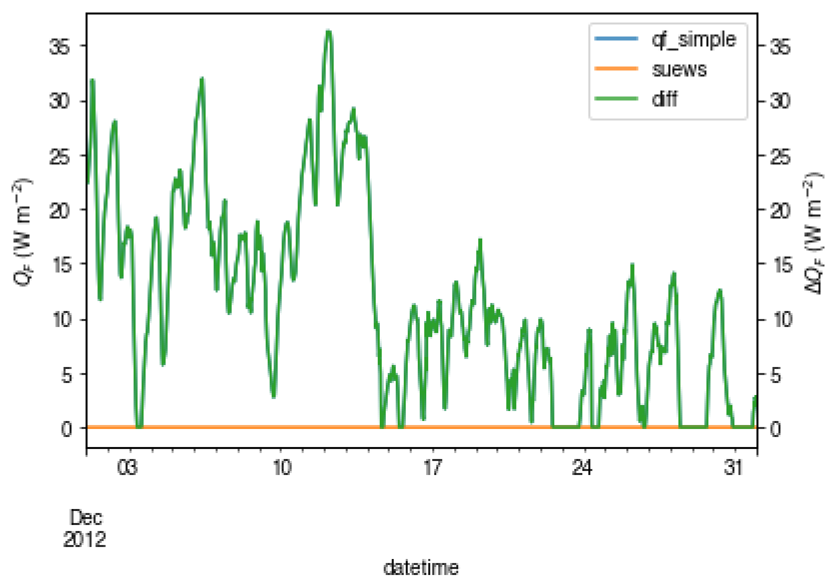


[10]:
```
var = "T2"
var_label = "$T_2$ ($^{\circ}$C)"
var_label_right = "$\Delta T_2$ ($^{\circ}$C)"
period = "2012-07"
df_test = df_output_test_summer
y1 = df_test.loc[period, var].rename("qf_simple")
y2 = df_output_def.loc[period, var].rename("suews")
y3 = (y1 - y2).rename("diff")
df_plot = pd.concat([y1, y2, y3], axis=1)
ax = df_plot.plot(secondary_y="diff")
_ = ax.set_ylabel(var_label)
_ = ax.right_ax.set_ylabel(var_label_right)
lines = ax.get_lines() + ax.right_ax.get_lines()
_ = ax.legend(lines, [l.get_label() for l in lines], loc="best")
```

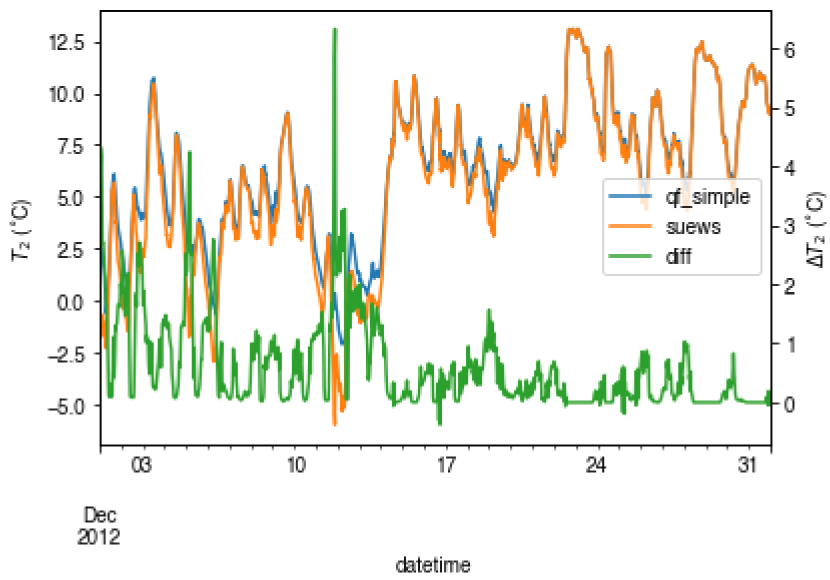


winter

[11]:
```
var = "QF"
var_label = "$Q_F$ ($ \mathrm{W \ m^{-2}}$)"
var_label_right = "$\Delta Q_F$ ($ \mathrm{W \ m^{-2}}$)"
period = "2012 12"
df_test = df_output_test_winter
y1 = df_test.loc[period, var].rename("qf_simple")
y2 = df_output_def.loc[period, var].rename("suews")
y3 = (y1 - y2).rename("diff")
df_plot = pd.concat([y1, y2, y3], axis=1)
ax = df_plot.plot(secondary_y="diff")
_ = ax.set_ylabel(var_label)
_ = ax.right_ax.set_ylabel(var_label_right)
lines = ax.get_lines() + ax.right_ax.get_lines()
_ = ax.legend(lines, [l.get_label() for l in lines], loc="best")
```



[12]:
```
var = "T2"
var_label = "$T_2$ ($^{\circ}$C)"
var_label_right = "$\Delta T_2$ ($^{\circ}$C)"
period = "2012 12"
df_test = df_output_test_winter
y1 = df_test.loc[period, var].rename("qf_simple")
y2 = df_output_def.loc[period, var].rename("suews")
y3 = (y1 - y2).rename("diff")
df_plot = pd.concat([y1, y2, y3], axis=1)
ax = df_plot.plot(secondary_y="diff")
_ = ax.set_ylabel(var_label)
_ = ax.right_ax.set_ylabel(var_label_right)
lines = ax.get_lines() + ax.right_ax.get_lines()
_ = ax.legend(lines, [l.get_label() for l in lines], loc="center right")
```



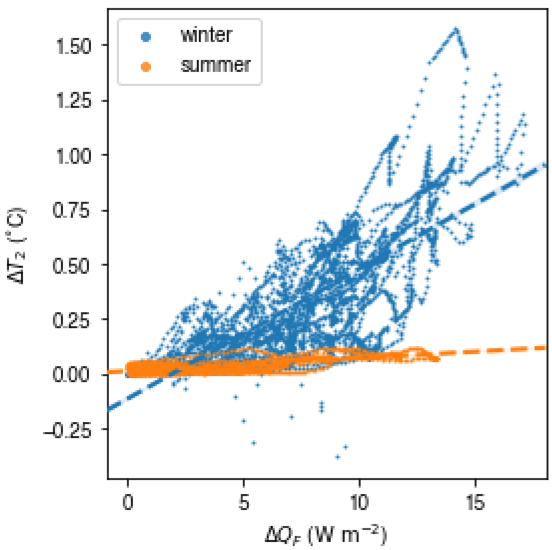
comparison in ΔQ_F-$\Delta T2$ feedback between summer and winter

[13]:
```
# filter results using `where` to choose periods when `QF_simple` is effective
# (i.e. activated by outdoor air temperatures)
df_diff_summer = (
    (df_output_test_summer - df_output_def)
    .where(df_output_def.T2 > 20, np.nan)
    .dropna(how="all", axis=0)
)
df_diff_winter = (
    (df_output_test_winter - df_output_def)
    .where(df_output_test_winter.T2 < 10, np.nan)
    .dropna(how="all", axis=0)
    .loc["20121215":]
)



df_diff_season = pd.concat(
    [df_diff_winter.assign(season="winter"), df_diff_summer.assign(season="summer"),]
).loc[:, ["season", "QF", "T2"]]
g = sns.lmplot(
    data=df_diff_season,
    x="QF",
    y="T2",
    hue="season",
    height=4,
    truncate=False,
    markers="o",
    legend_out=False,
    scatter_kws={"s": 1, "zorder": 0, "alpha": 0.8,},
```

(continues on next page)

(continued from previous page)

```
    line_kws={"zorder": 6, "linestyle": "--"},
)
_ = g.set_axis_labels(
    "$\Delta Q_F$ ($ \mathrm{W \ m^{-2}}$)", "$\Delta T_2$ ($^{\circ}$C)",
)
_ = g.ax.legend(markerscale=4)
_ = g.despine(top=False, right=False)
```



The above figure indicates a positive feedback, as Q_F is increased there is an elevated T_2 but with different magnitudes given the non-linearlity in the SUEWS modelling system. Of particular note is the positive feedback loop under warm air temperatures: the anthropogenic heat emissions increase which in turn elevates the outdoor air temperature causing yet more anthropogenic heat release. Note that London is relatively cool so the enhancement is much less than it would be in warmer cities.

Set up SuPy for Your Own Site

This tutorial aims to demonstrate how to set up SuPy for your own site to model the surface energy balance (SEB).

Please note: SuPy is a Python-enhanced urban climate model with SUEWS, Surface Urban Energy and Water Balance Scheme, as its computation core.

We thus strongly recommend/encourage users to have a good understanding of SUEWS first before diving into the SuPy world.

In this tutorial, We will use an AmeriFlux site US-AR1 as example:

starting by preparation of input data, we show how to specify site characteristics and choose proper scheme options, then conduct simulations, finally provide some demo figures to help understand the simulation results.

A brief structure is as follows:

1. *Preparing the input data*;
2. *Running a simulation*;
3. *Examination of results*; and
4. *Further exploration*

Boilerplate code

[1]:
```
import matplotlib.pyplot as plt
import supy as sp
import pandas as pd
import numpy as np
from pathlib import Path
%matplotlib inline
```

Prepare input data

Site-specific configuration of surface parameters

Given `pandas.DataFrame` as the core data structure of SuPy, all operations, including modification, output, demonstration, etc., on SuPy inputs (`df_state_init` and `df_forcing`) can be done using `pandas`-based functions/methods. Please see SuPy quickstart for methods to do so.

Below we will modify several key properties of the chosen site with appropriate values to run SuPy. First, we copy the `df_state_init` to have a new DataFrame for manipulation.

[2]:
```
df_state_init,df_forcing=sp.load_SampleData()
df_state_amf = df_state_init.copy()
```

```
2020-07-06 11:24:40,102 - SuPy - INFO - All cache cleared.
```

[3]:
```
# site identifier
name_site = 'US-AR1'
```

Details for determining the proper values of selected physical parameters can be found here.

location

[4]:
```
# latitude
df_state_amf.loc[:, 'lat'] = 41.37
# longitude
df_state_amf.loc[:, 'lng'] = -106.24
# altitude
df_state_amf.loc[:, 'alt'] = 611.
```

land cover fraction

Land covers in SUEWS

[5]:
```
# view the surface fraction variable: `sfr`
df_state_amf.loc[:, 'sfr'] = 0
df_state_amf.loc[:, ('sfr', '(4,)')] = 1
df_state_amf.loc[:, 'sfr']
```

[5]:
```
ind_dim  (0,)  (1,)  (2,)  (3,)  (4,)  (5,)  (6,)
grid
1         0.0   0.0   0.0   0.0   1.0   0.0   0.0
```

albedo

[6]:
```
# we only set values for grass as the modelled site has a single land cover type: grass.
df_state_amf.albmax_grass = 0.19
df_state_amf.albmin_grass = 0.14
```

[7]:
```
# initial albedo value
df_state_amf.loc[:, 'albgrass_id'] = 0.14
```

LAI/phenology

[8]:
```
df_state_amf.filter(like='lai')
```

[8]:
```
var     laimax           laimin           laipower                          ...  \
ind_dim   (0,) (1,) (2,)   (0,) (1,) (2,)   (0, 0) (0, 1) (0, 2) (1, 0)  ...
grid                                                                     ...
1          5.1  5.5  5.9    4.0  1.0  1.6     0.04   0.04   0.04  0.001  ...

var                               laitype          laicalcyes lai_id
ind_dim  (3, 0)  (3, 1)  (3, 2)    (0,) (1,) (2,)          0   (0,) (1,) (2,)
grid
1        0.0015  0.0015  0.0015     1.0  1.0  1.0          1    4.0  1.0  1.6

[1 rows x 25 columns]
```

[9]:
```
# properties to control vegetation phenology
# you can skip the details for and just set them as provided below

# LAI paramters
df_state_amf.loc[:, ('laimax', '(2,)')] = 1
df_state_amf.loc[:, ('laimin', '(2,)')] = 0.2
# initial LAI
df_state_amf.loc[:, ('lai_id', '(2,)')] = 0.2

# BaseT
df_state_amf.loc[:, ('baset', '(2,)')] = 5
# BaseTe
df_state_amf.loc[:, ('basete', '(2,)')] = 20

# SDDFull
df_state_amf.loc[:, ('sddfull', '(2,)')] = -1000
# GDDFull
df_state_amf.loc[:, ('gddfull', '(2,)')] = 1000
```

surface resistance

```
[10]: # parameters to model surface resistance
df_state_amf.maxconductance = 18.7
df_state_amf.g1 = 1
df_state_amf.g2 = 104.215
df_state_amf.g3 = 0.424
df_state_amf.g4 = 0.814
df_state_amf.g5 = 36.945
df_state_amf.g6 = 0.025
```

measurement height

```
[11]: # height where forcing variables are measured/collected
df_state_amf.z = 2.84
```

urban feature

```
[12]: # disable anthropogenic heat by setting zero population
df_state_amf.popdensdaytime = 0
df_state_amf.popdensnighttime = 0
```

check `df_state`

```
[13]: # this procedure is to double-check proper values are set in `df_state_amf`
sp.check_state(df_state_amf)
```

```
2020-07-06 11:24:43,372 - SuPy - INFO - SuPy is validating `df_state`...
2020-07-06 11:24:43,574 - SuPy - INFO - All checks for `df_state` passed!
```

prepare forcing conditions

Here we use the SuPy utility function `read_forcing` to read in forcing data from an external file in the format of SUEWS input. Also note, this `read_forcing` utility will also resample the forcing data to a proper temporal resolution to run SuPy/SUEWS, which is usually 5 min (300 s).

load and resample forcing data

UMEP workshop users: please note the AMF file path might be DIFFERENT from yours; please set it to the location where your downloaded file is placed.

---

[15]:
```
# load forcing data from an external file and resample to a resolution of 300 s.
# Note this dataset has been gap-filled.
df_forcing_amf = sp.util.read_forcing("data/US-AR1_2010_data_60.txt", tstep_mod=300)

# this procedure is to double-check proper forcing values are set in `df_forcing_amf`
_ = sp.check_forcing(df_forcing_amf)
```

```
2020-07-06 11:24:44,453 - SuPy - INFO - SuPy is validating `df_forcing`...
2020-07-06 11:24:46,299 - SuPy - ERROR - Issues found in `df_forcing`:
`kdown` should be between [0, 1400] but `-1.298` is found at 2010-01-01 00:05:00
```

The checker detected invalid values in variable kdown: negative incoming solar radiation is found. We then need to fix this as follows:

[16]:
```
# modify invalid values
df_forcing_amf.kdown = df_forcing_amf.kdown.where(df_forcing_amf.kdown > 0, 0)
```

[17]:
```
# check `df_forcing` again
_ = sp.check_forcing(df_forcing_amf)
```

```
2020-07-06 11:24:46,312 - SuPy - INFO - SuPy is validating `df_forcing`...
2020-07-06 11:24:48,523 - SuPy - INFO - All checks for `df_forcing` passed!
```

examine forcing data

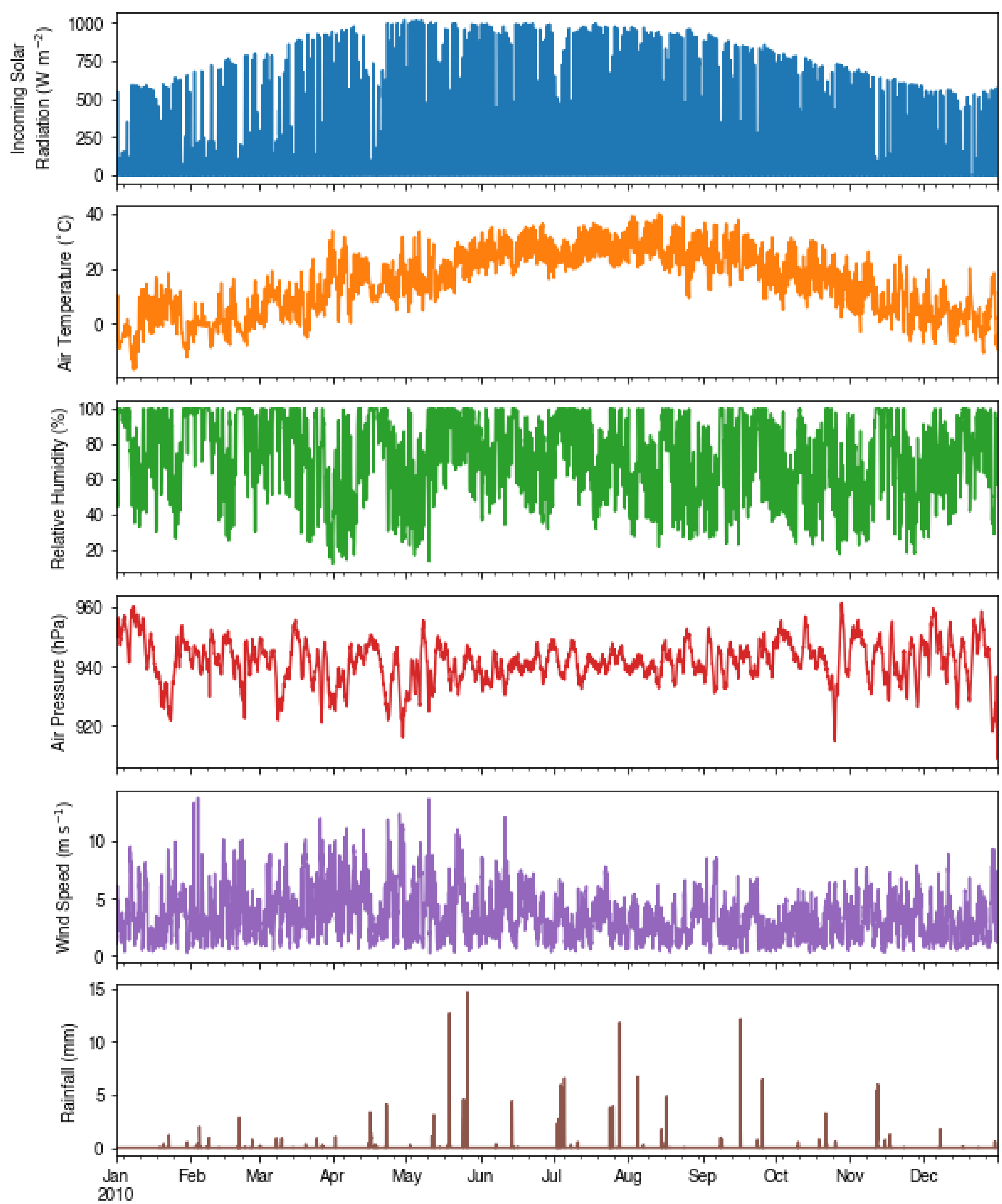
We can examine the forcing data:

[18]:
```
list_var_forcing = [
    "kdown",
    "Tair",
    "RH",
    "pres",
    "U",
    "rain",
]
dict_var_label = {
    "kdown": "Incoming Solar\n Radiation ($ \mathrm{W \ m^{-2}}$)",
    "Tair": "Air Temperature ($^{\circ}}$C)",
    "RH": r"Relative Humidity (%)",
    "pres": "Air Pressure (hPa)",
    "rain": "Rainfall (mm)",
    "U": "Wind Speed (m $\mathrm{s^{-1}}$)",
}
df_plot_forcing_x = (
    df_forcing_amf.loc[:, list_var_forcing].copy().shift(-1).dropna(how="any")
)
df_plot_forcing = df_plot_forcing_x.resample("1h").mean()
df_plot_forcing["rain"] = df_plot_forcing_x["rain"].resample("1h").sum()

axes = df_plot_forcing.plot(subplots=True, figsize=(8, 12), legend=False,)
fig = axes[0].figure
fig.tight_layout()
```

(continues on next page)

(continued from previous page)

```
fig.autofmt_xdate(bottom=0.2, rotation=0, ha="center")
for ax, var in zip(axes, list_var_forcing):
    _ = ax.set_ylabel(dict_var_label[var])
```



Run simulations

Once met-forcing (via `df_forcing_amf`) and initial conditions (via `df_state_amf`) are loaded in, we call `sp.run_supy` to conduct a SUEWS simulation, which will return two `pandas DataFrames`: `df_output` and `df_state_final`.

[19]:
```
df_output, df_state_final = sp.run_supy(df_forcing_amf, df_state_amf)
```

```
2020-07-06 11:24:51,973 - SuPy - INFO - ====================
2020-07-06 11:24:51,974 - SuPy - INFO - Simulation period:
2020-07-06 11:24:51,975 - SuPy - INFO -   Start: 2010-01-01 00:05:00
2020-07-06 11:24:51,975 - SuPy - INFO -   End: 2011-01-01 00:00:00
2020-07-06 11:24:51,976 - SuPy - INFO -
2020-07-06 11:24:51,977 - SuPy - INFO - No. of grids: 1
2020-07-06 11:24:51,977 - SuPy - INFO - SuPy is running in serial mode
2020-07-06 11:25:01,975 - SuPy - INFO - Execution time: 10.0 s
2020-07-06 11:25:01,976 - SuPy - INFO - ====================
```

`df_output`

`df_output` is an ensemble output collection of major SUEWS output groups, including:

- SUEWS: the essential SUEWS output variables
- DailyState: variables of daily state information
- snow: snow output variables (effective when `snowuse = 1` set in `df_state_init`)
- RSL: profile of air temperature, humidity and wind speed within roughness sub-layer.

Detailed description of variables in `df_output` refers to SuPy output

[20]:
```
df_output.columns.levels[0]
```

[20]:
```
Index(['SUEWS', 'snow', 'RSL', 'SOLWEIG', 'DailyState'], dtype='object', name='group')
```

`df_state_final`

`df_state_final` is a `DataFrame` for holding:

1. all model states if `save_state` is set to `True` when calling `sp.run_supy` (`supy` may run significantly slower for a large simulations);
2. or, only the final state if `save_state` is set to `False` (the default setting) in which mode `supy` has a similar performance as the standalone compiled SUEWS executable.

Entries in `df_state_final` have the same data structure as `df_state_init` and can thus be used for other SUEWS simulations staring at the timestamp as in `df_state_final`.

Detailed description of variables in `df_state_final` refers to SuPy output

[21]:
```
df_state_final.T.head()
```

[21]:
```
datetime                 2010-01-01 00:05:00 2011-01-01 00:05:00
grid                                       1                   1
```

(continues on next page)

(continued from previous page)

```
var             ind_dim
ah_min          (0,)                15.0            15.0
                (1,)                15.0            15.0
ah_slope_cooling (0,)                2.7             2.7
                (1,)                 2.7             2.7
ah_slope_heating (0,)                2.7             2.7
```

Examine results

Thanks to the functionality inherited from pandas and other packages under the PyData stack, compared with the standard SUEWS simulation workflow, supy enables more convenient examination of SUEWS results by statistics calculation, resampling, plotting (and many more).

Ouptut structure

df_output is organised with MultiIndex (grid,timestamp) and (group,varaible) as index and columns, respectively.

[22]:
```
df_output.head()
```

[22]:
```
group                    SUEWS                                              \
var                      Kdown  Kup       Ldown         Lup  Tsurf         QN
grid datetime
1    2010-01-01 00:05:00   0.0  0.0  265.492652  305.638434 -1.593 -40.145783
     2010-01-01 00:10:00   0.0  0.0  265.492652  305.638434 -1.593 -40.145783
     2010-01-01 00:15:00   0.0  0.0  265.492652  307.825865 -1.593 -42.333213
     2010-01-01 00:20:00   0.0  0.0  265.492652  307.825865 -1.593 -42.333213
     2010-01-01 00:25:00   0.0  0.0  265.492652  307.825865 -1.593 -42.333213

group                                                          ...  \
var                       QF        QS         QH         QE  ...
grid datetime                                                 ...
1    2010-01-01 00:05:00  0.0 -9.668746 -24.387976   1.284400  ...
     2010-01-01 00:10:00  0.0 -9.424108  -6.676973   1.618190  ...
     2010-01-01 00:15:00  0.0 -0.545992  16.458627  11.833592  ...
     2010-01-01 00:20:00  0.0 -0.536225  15.988621  11.830741  ...
     2010-01-01 00:25:00  0.0 -0.525680  15.537087  11.827934  ...

group                       DailyState                              \
var                      DensSnow_Paved DensSnow_Bldgs DensSnow_EveTr
grid datetime
1    2010-01-01 00:05:00            NaN            NaN            NaN
     2010-01-01 00:10:00            NaN            NaN            NaN
     2010-01-01 00:15:00            NaN            NaN            NaN
     2010-01-01 00:20:00            NaN            NaN            NaN
     2010-01-01 00:25:00            NaN            NaN            NaN

group                                                               \
var                      DensSnow_DecTr DensSnow_Grass DensSnow_BSoil
grid datetime
```

(continues on next page)

(continued from previous page)

```
1    2010-01-01 00:05:00           NaN            NaN            NaN
     2010-01-01 00:10:00           NaN            NaN            NaN
     2010-01-01 00:15:00           NaN            NaN            NaN
     2010-01-01 00:20:00           NaN            NaN            NaN
     2010-01-01 00:25:00           NaN            NaN            NaN

group
var                      DensSnow_Water  a1  a2  a3
grid datetime
1    2010-01-01 00:05:00           NaN NaN NaN NaN
     2010-01-01 00:10:00           NaN NaN NaN NaN
     2010-01-01 00:15:00           NaN NaN NaN NaN
     2010-01-01 00:20:00           NaN NaN NaN NaN
     2010-01-01 00:25:00           NaN NaN NaN NaN

[5 rows x 371 columns]
```

Here we demonstrate several typical scenarios for SUEWS results examination.

The essential SUEWS output collection is extracted as a separate variable for easier processing in the following sections. More advanced slicing techniques are available in pandas documentation.

[23]:
```
grid = df_state_amf.index[0]
df_output_suews = df_output.loc[grid, 'SUEWS']
```

Statistics Calculation

We can use .describe() method for a quick overview of the key surface energy balance budgets.

[24]:
```
df_output_suews.loc[:, ['QN', 'QS', 'QH', 'QE', 'QF']].describe()
```

[24]:
```
var              QN             QS             QH             QE        QF
count  105120.000000  105120.000000  105120.000000  105120.000000  105120.0
mean      118.207887      19.047648      38.349672      62.790798       0.0
std       214.335328      61.955598      85.050755     112.585643       0.0
min      -104.566267     -81.170768    -212.925432     -15.483971       0.0
25%       -33.437969     -23.174678     -15.992876       0.341017       0.0
50%        -1.894385      -2.603727       9.862241       3.042328       0.0
75%       248.960723      52.299898      68.130871      65.272384       0.0
max       749.868243     218.450452     414.514498     559.472107       0.0
```

Plotting

Basic example

Plotting is very straightforward via the .plot method bounded with pandas.DataFrame. Note the usage of loc for to slices of the output DataFrame.

[25]:
```
# a dict for better display variable names
dict_var_disp = {
```

(continues on next page)

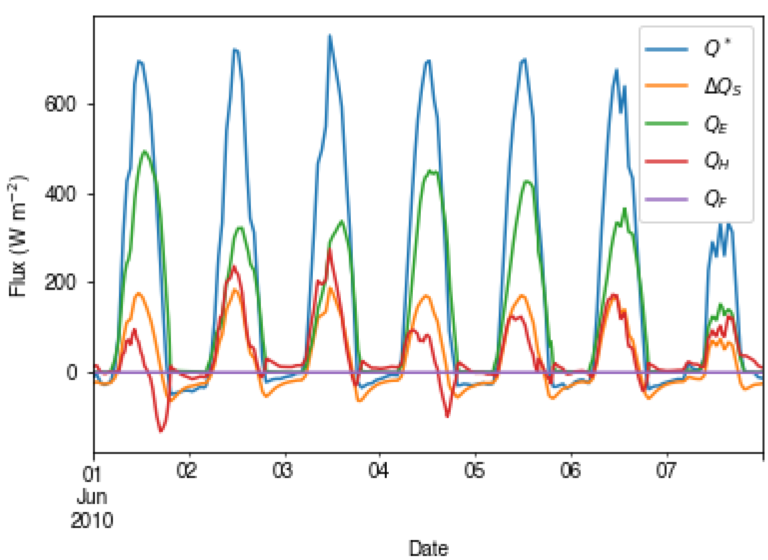
(continued from previous page)

```
    "QN": "$Q^*$",
    "QS": r"$\Delta Q_S$",
    "QE": "$Q_E$",
    "QH": "$Q_H$",
    "QF": "$Q_F$",
    "Kdown": r"$K_{\downarrow}$",
    "Kup": r"$K_{\uparrow}$",
    "Ldown": r"$L_{\downarrow}$",
    "Lup": r"$L_{\uparrow}$",
    "Rain": "$P$",
    "Irr": "$I$",
    "Evap": "$E$",
    "RO": "$R$",
    "TotCh": "$\Delta S$",
}
```

Peek at the simulation results:

[26]:
```
grid = df_state_init.index[0]
```

[27]:
```
ax_output = (
    df_output_suews.loc["2010-06-01":"2010-06-07", ["QN", "QS", "QE", "QH", "QF"]]
    .rename(columns=dict_var_disp)
    .plot()
)
_ = ax_output.set_xlabel("Date")
_ = ax_output.set_ylabel("Flux ($ \mathrm{W \ m^{-2}}$)")
_ = ax_output.legend()
```



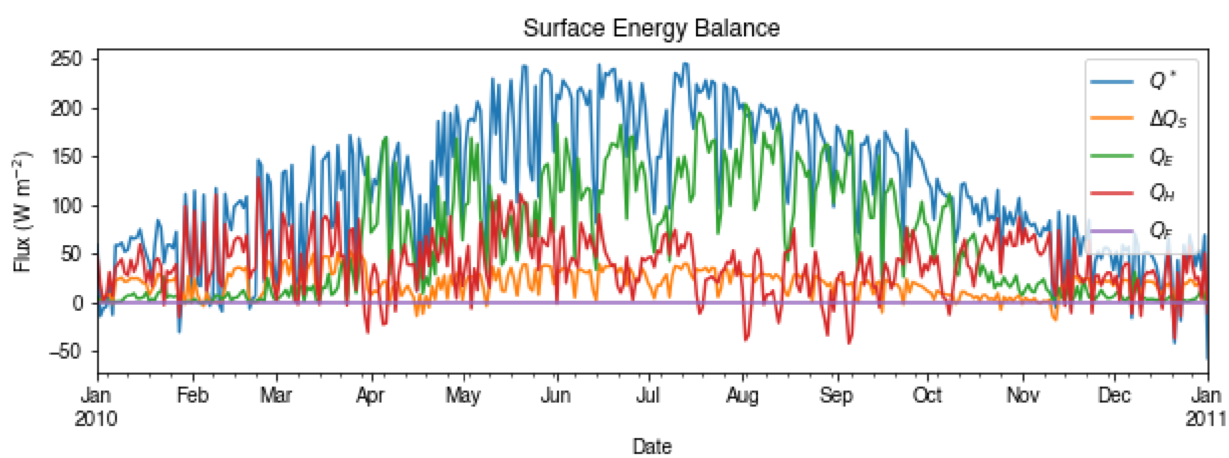
Plotting after resampling

The suggested runtime/simulation frequency of SUEWS is `300 s`, which usually results in a large output and may be over-weighted for storage and analysis. Also, you may feel an apparent slowdown in producing the above figure as a large amount of data were used for the plotting. To slim down the result size for analysis and output, we can `resample` the default output very easily.

[28]:
```
rsmp_1d = df_output_suews.resample("1d")
# daily mean values
df_1d_mean = rsmp_1d.mean()
# daily sum values
df_1d_sum = rsmp_1d.sum()
```

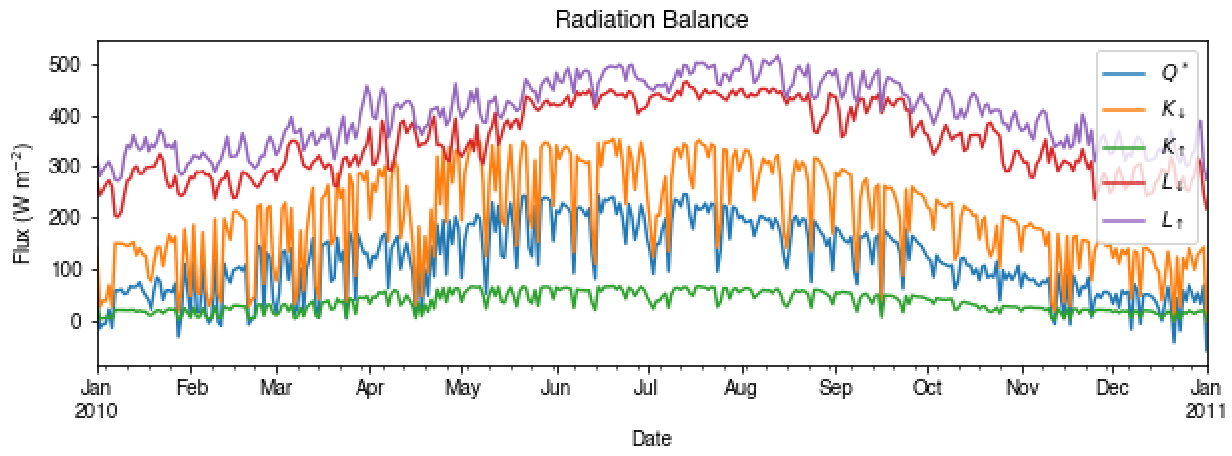
We can then re-examine the above energy balance at hourly scale and plotting will be significantly faster.

[29]:
```
# energy balance
ax_output = (
    df_1d_mean.loc[:, ["QN", "QS", "QE", "QH", "QF"]]
    .rename(columns=dict_var_disp)
    .plot(figsize=(10, 3), title="Surface Energy Balance",)
)
_ = ax_output.set_xlabel("Date")
_ = ax_output.set_ylabel("Flux ($ \mathrm{W \ m^{-2}}$)")
_ = ax_output.legend()
```

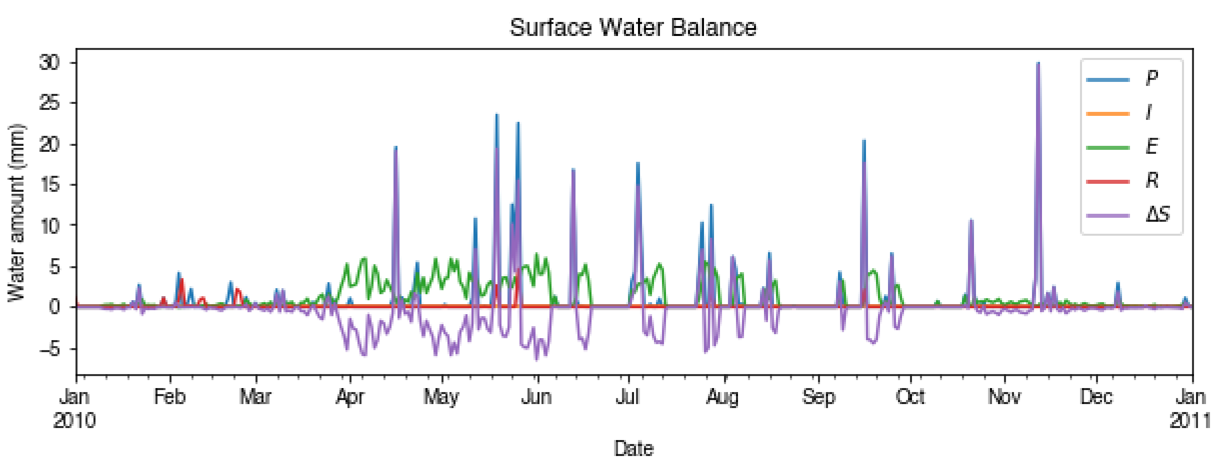


Then we use the hourly results for other analyses.

[30]:
```
# radiation balance
ax_output = (
    df_1d_mean.loc[:, ["QN", "Kdown", "Kup", "Ldown", "Lup"]]
    .rename(columns=dict_var_disp)
    .plot(figsize=(10, 3), title="Radiation Balance",)
)
_ = ax_output.set_xlabel("Date")
_ = ax_output.set_ylabel("Flux ($ \mathrm{W \ m^{-2}}$)")
_ = ax_output.legend()
```



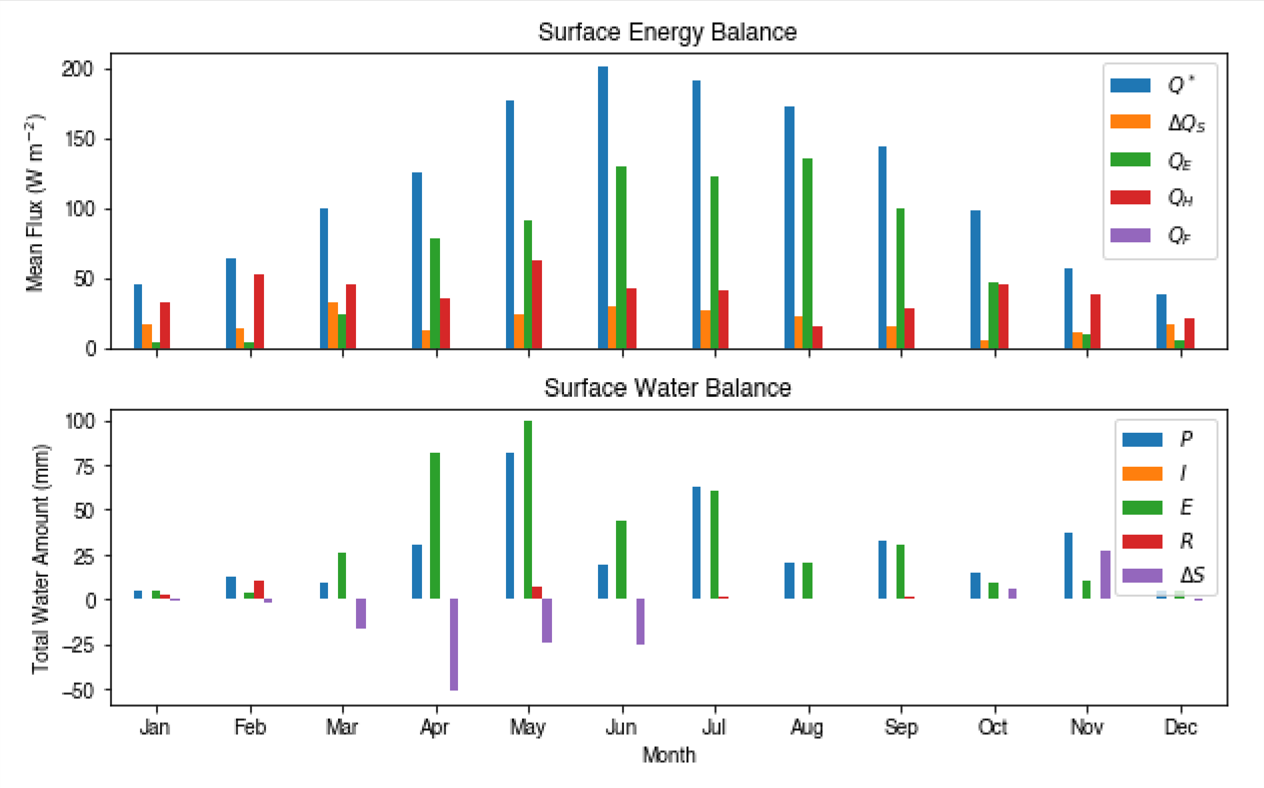
[31]:
```
# water balance
ax_output = (
    df_1d_sum.loc[:, ["Rain", "Irr", "Evap", "RO", "TotCh"]]
    .rename(columns=dict_var_disp)
    .plot(figsize=(10, 3), title="Surface Water Balance",)
)
_ = ax_output.set_xlabel("Date")
_ = ax_output.set_ylabel("Water amount (mm)")
_ = ax_output.legend()
```



Get an overview of partitioning in energy and water balance at monthly scales:

[32]:
```
# get a monthly Resampler
df_plot = df_output_suews.copy()
df_plot.index = df_plot.index.set_names("Month")
rsmp_1M = df_plot.shift(-1).dropna(how="all").resample("1M", kind="period")
# mean values
df_1M_mean = rsmp_1M.mean()
# sum values
df_1M_sum = rsmp_1M.sum()
```

[33]:
```
# month names
name_mon = [x.strftime("%b") for x in rsmp_1M.groups]
# create subplots showing two panels together
fig, axes = plt.subplots(2, 1, sharex=True)
# surface energy balance
_ = (
    df_1M_mean.loc[:, ["QN", "QS", "QE", "QH", "QF"]]
    .rename(columns=dict_var_disp)
    .plot(
        ax=axes[0],  # specify the axis for plotting
        figsize=(10, 6),  # specify figure size
        title="Surface Energy Balance",
        kind="bar",
    )
)
# surface water balance
_ = (
    df_1M_sum.loc[:, ["Rain", "Irr", "Evap", "RO", "TotCh"]]
    .rename(columns=dict_var_disp)
    .plot(
        ax=axes[1],  # specify the axis for plotting
        title="Surface Water Balance",
        kind="bar",
    )
)

# annotations
_ = axes[0].set_ylabel("Mean Flux ($ \mathrm{W \ m^{-2}}$)")
_ = axes[0].legend()
_ = axes[1].set_xlabel("Month")
_ = axes[1].set_ylabel("Total Water Amount (mm)")
_ = axes[1].xaxis.set_ticklabels(name_mon, rotation=0)
_ = axes[1].legend()
```



Save results to external files

The supy output can be saved as `txt` files for further analysis using supy function `save_supy`.

[34]:
```
list_path_save = sp.save_supy(df_output, df_state_final)
```

[35]:
```
for file_out in list_path_save:
    print(file_out.name)
```

```
1_2010_DailyState.txt
1_2010_SUEWS_60.txt
1_2010_snow_60.txt
1_2010_RSL_60.txt
1_2010_SOLWEIG_60.txt
df_state.csv
```

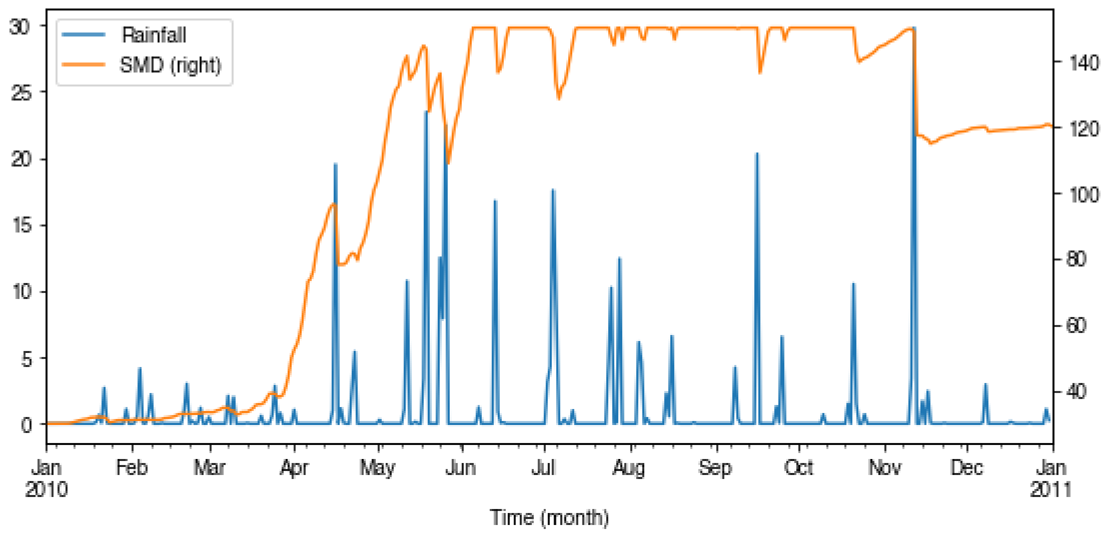
More explorations into simulation results

In this section, we will use the simulation results to explore more features revealed by SuPy/SUEWS simulations but *unavailable in your simple model*.

Dynamics in rainfall and soil moisture deficit (SMD)

[36]:
```
df_dailystate = (
    df_output.loc[grid, "DailyState"].dropna(how="all").resample("1d").mean()
)
```

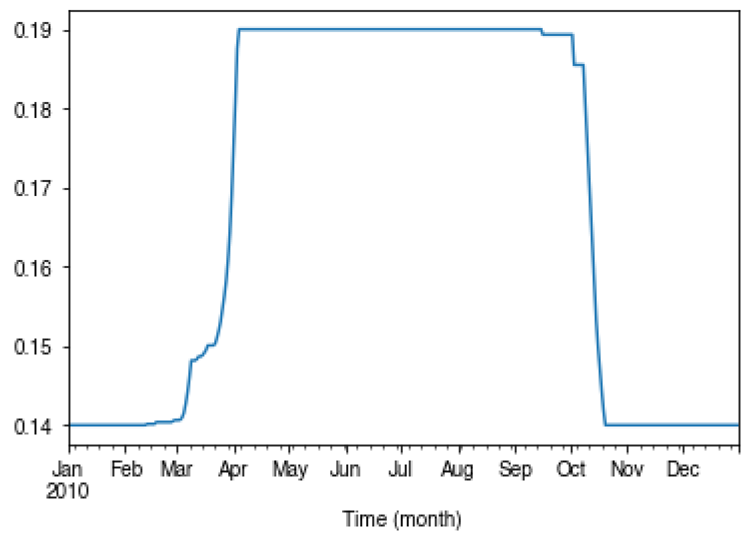
[37]:
```
# daily rainfall
ser_p = df_dailystate.P_day.rename("Rainfall")
ser_smd = df_output_suews.SMD
ser_smd_dmax = ser_smd.resample("1d").max().rename("SMD")

ax = pd.concat([ser_p, ser_smd_dmax], axis=1).plot(secondary_y="SMD", figsize=(9, 4))
_ = ax.set_xlabel("Time (month)")
```



Variability in albedo

How does albedo change over time?

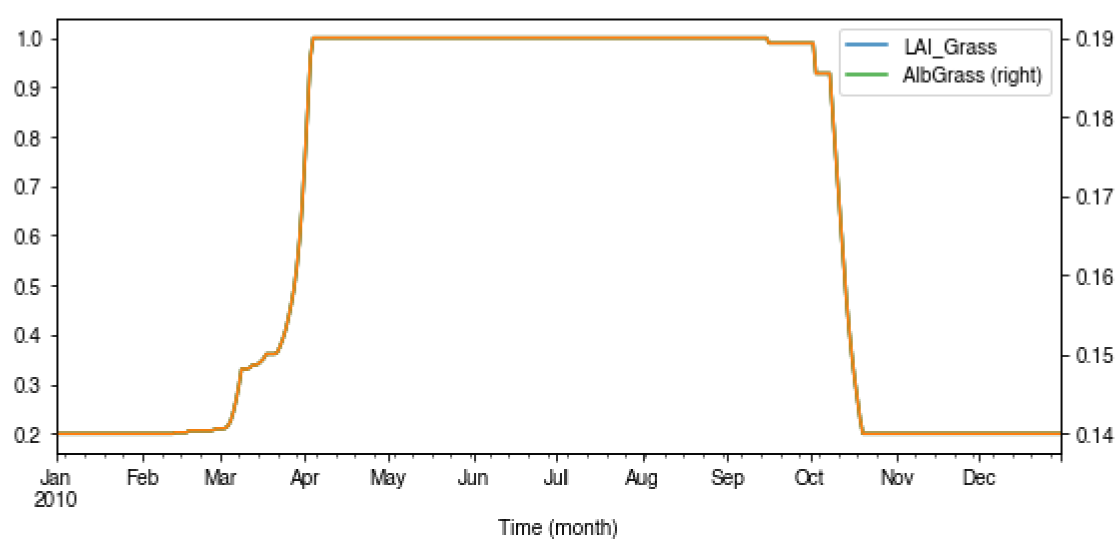
[38]:
```
ser_alb = df_dailystate.AlbGrass
ax = ser_alb.plot()
_ = ax.set_xlabel("Time (month)")
```



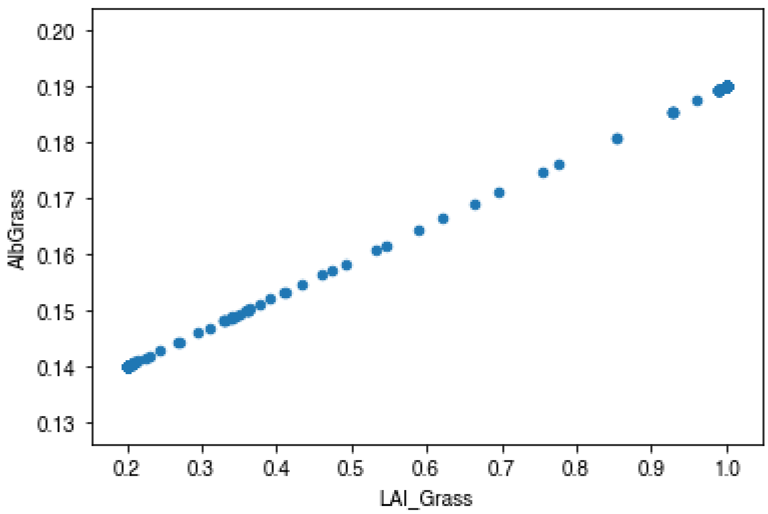
How is albedo associated with vegetation phenology?

[39]:
```
ser_lai = df_dailystate.LAI_Grass
pd.concat([ser_lai, ser_alb], axis=1).plot(secondary_y="AlbGrass", figsize=(9, 4))
ax = ser_lai.plot()
_ = ax.set_xlabel("Time (month)")
```

[39]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8969449978>
```



[40]:
```
ax_alb_lai = df_dailystate[["LAI_Grass", "AlbGrass"]].plot.scatter(
    x="LAI_Grass", y="AlbGrass",
)
ax_alb_lai.set_aspect("auto")
```



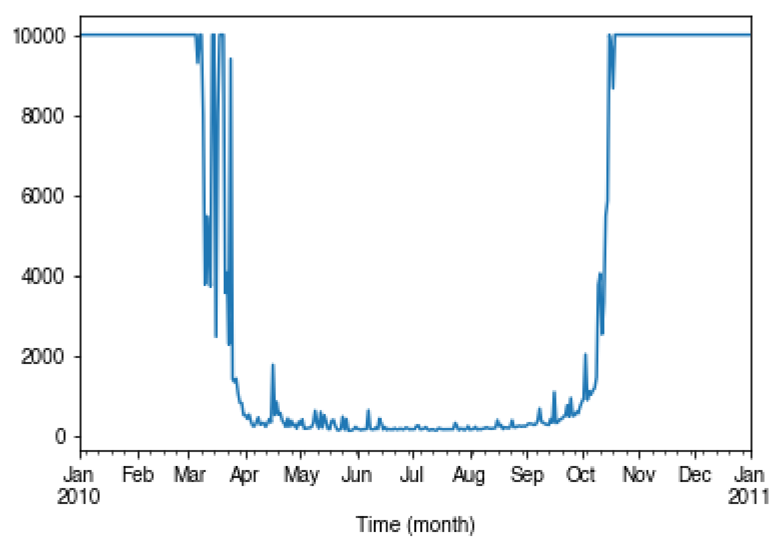
Variability in surface resistance

How does surface resistance vary over time?

[41]:
```
ser_rs = df_output_suews.RS
```

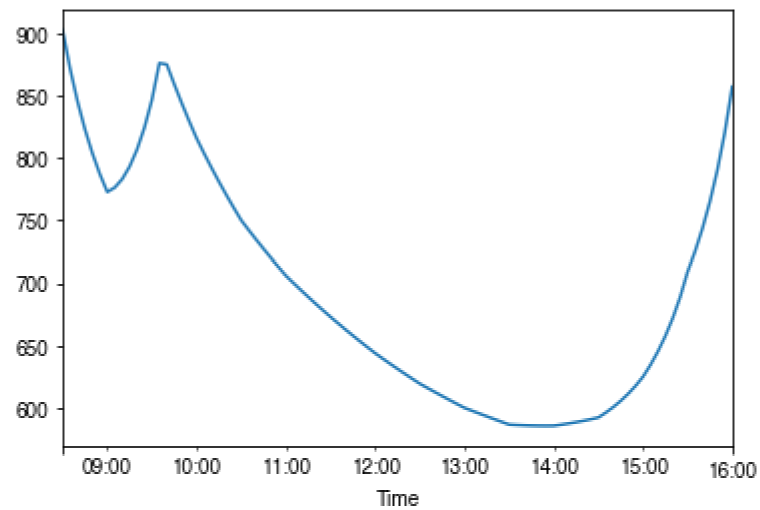
- intra-annual

[42]:
```
ax = ser_rs.resample("1d").median().plot()
_ = ax.set_xlabel("Time (month)")
```

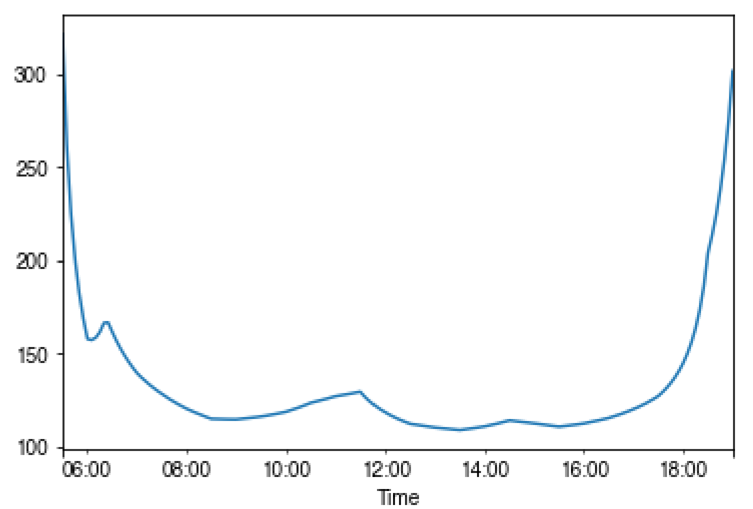


- intra-daily

[43]:
```
# a winter day
ax = ser_rs.loc["2010-01-22"].between_time("0830", "1600").plot()
_ = ax.set_xlabel("Time")
```

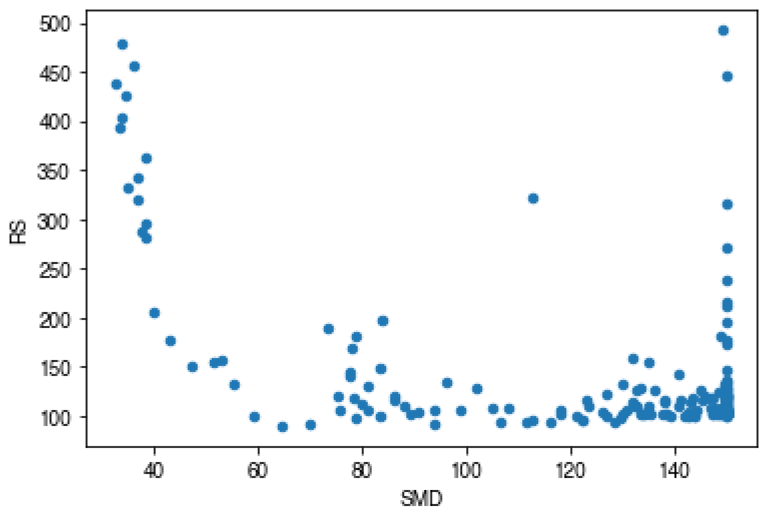


[44]:
```
# a summer day
ax = ser_rs.loc["2010-07-01"].between_time("0530", "1900").plot()
_ = ax.set_xlabel("Time")
```

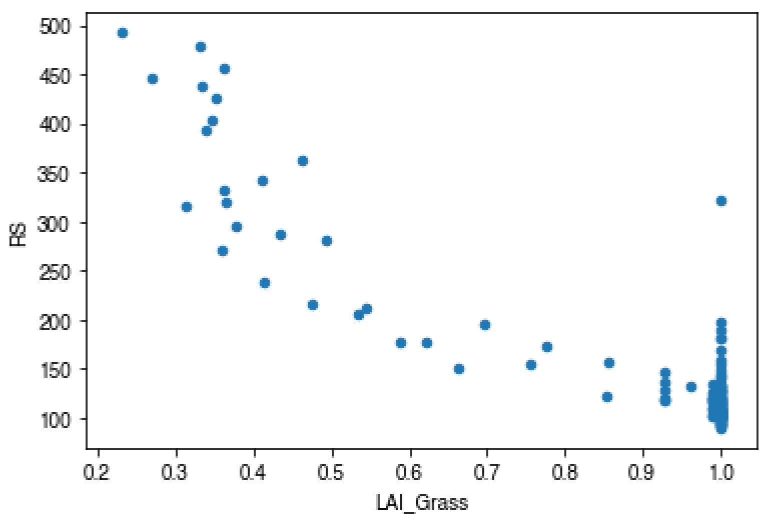


How is surface resistance associated with other surface properties?

[45]:
```
# SMD
ser_smd = df_output_suews.SMD
df_x = (
    pd.concat([ser_smd, ser_rs], axis=1)
    .between_time("1000", "1600")
    .resample("1d")
    .mean()
)
df_x = df_x.loc[df_x.RS < 500]
_ = df_x.plot.scatter(x="SMD", y="RS",)
```



[46]:
```
# LAI
df_x = pd.concat(
    [ser_lai, ser_rs.between_time("1000", "1600").resample("1d").mean()], axis=1
)
df_x = df_x.loc[df_x.RS < 500]
_ = df_x.plot.scatter(x="LAI_Grass", y="RS",)
```



How is surface resistance dependent on meteorological conditions?

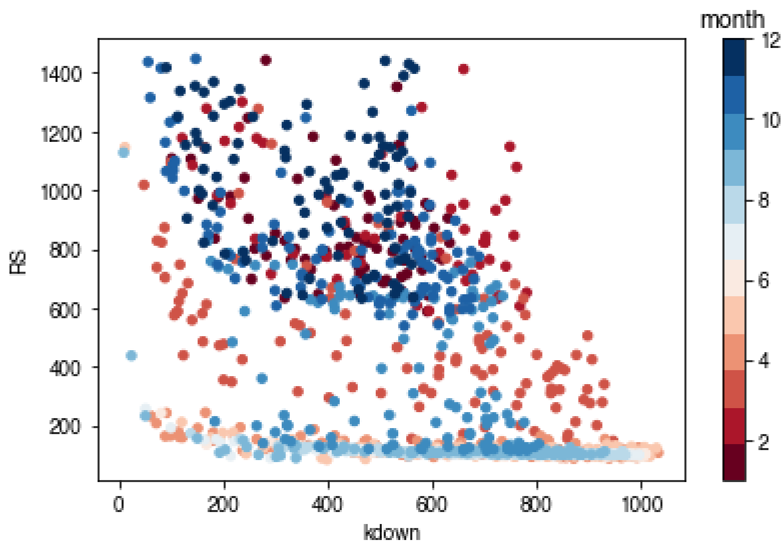
[47]:
```
cmap_sel = plt.cm.get_cmap('RdBu', 12)
```

[48]:
```
# solar radiation
# colour by season
ser_kdown = df_forcing_amf.kdown
df_x = pd.concat([ser_kdown, ser_rs], axis=1).between_time('1000', '1600')
```

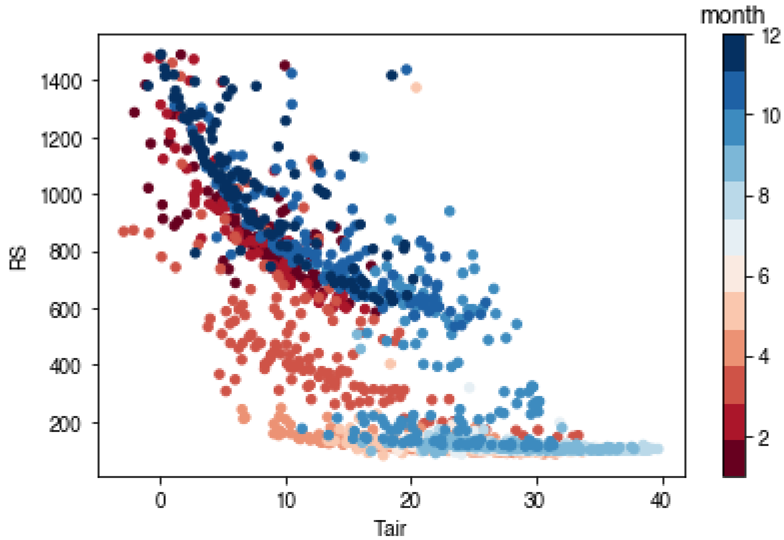
(continues on next page)

(continued from previous page)

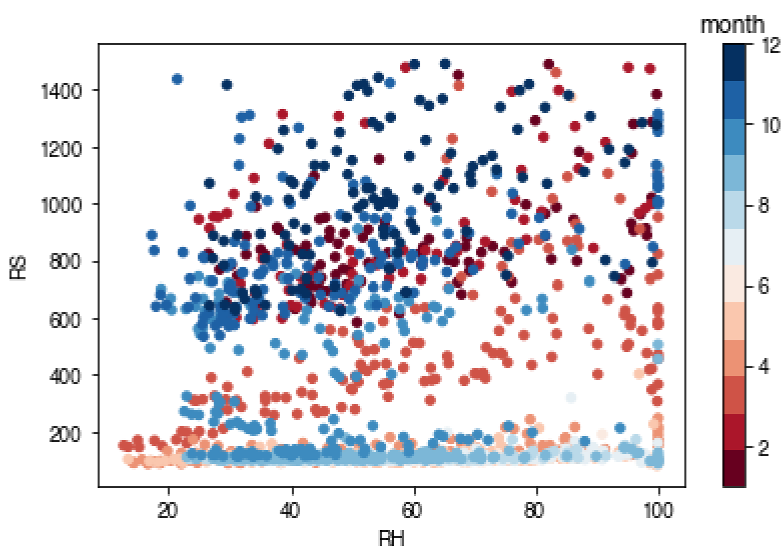
```
df_x = df_x.loc[df_x.RS < 1500]
df_plot = df_x.iloc[::20]
ax = df_plot.plot.scatter(x='kdown',
                          y='RS',
                          c=df_plot.index.month,
                          cmap=cmap_sel,
                          sharex=False)
fig = ax.figure
_ = fig.axes[1].set_title('month')
fig.tight_layout()
```



[49]:
```
# air temperature
ser_ta = df_forcing_amf.Tair
df_x = pd.concat([ser_ta, ser_rs], axis=1).between_time('1000', '1600')
df_x = df_x.loc[df_x.RS < 1500]
df_plot = df_x.iloc[::15]
ax = df_plot.plot.scatter(x='Tair',
                          y='RS',
                          c=df_plot.index.month,
                          cmap=cmap_sel,
                          sharex=False)
fig = ax.figure
_ = fig.axes[1].set_title('month')
fig.tight_layout()
```




```
[50]: # air humidity
ser_rh = df_forcing_amf.RH
df_x = pd.concat([ser_rh, ser_rs], axis=1).between_time('1000', '1600')
df_x = df_x.loc[df_x.RS < 1500]
df_plot = df_x.iloc[::15]
ax = df_plot.plot.scatter(x='RH',
                          y='RS',
                          c=df_plot.index.month,
                          cmap=cmap_sel,
                          sharex=False)
fig = ax.figure
_ = fig.axes[1].set_title('month')
fig.tight_layout()
```



- **Task:**

Based on the above plots showing RS vs. met. conditions, explore these relationships again at the intra-daily scales.

Note:

1. The Anaconda distribution is suggested as the scientific Python 3 environment for its completeness in necessary packages. Please follow the official guide for its installation.
2. Users with less experience in Python are suggested to go through the following section first before using SuPy.

Python 101 before SuPy

Admittedly, this header is somewhat misleading: given the enormity of Python, it's more challenging to get this section *correct* than coding SuPy per se. As such, here a collection of data analysis oriented links to useful Python resources is provided to help novices start using Python and **then** SuPy.

- The gist of Python: a quick introductory blog that covers Python basics for data analysis.
- Jupyter Notebook: Jupyter Notebook provides a powerful notebook-based data analysis environment that SuPy users are strongly encouraged to use. Jupyter notebooks can run in browsers (desktop, mobile) either by easy local configuration or on remote servers with pre-set environments (e.g., Google Colaboratory, Microsoft Azure Notebooks). In addition, Jupyter notebooks allow great shareability by incorporating source code and detailed notes in one place, which helps users to organise their computation work.
  - Installation

    Jupyter notebooks can be installed with pip on any desktop/server system and open .ipynb notebook files locally:

    ```
    python3 -m pip install jupyter -U
    ```

  - Extensions: To empower your Jupyter Notebook environment with better productivity, please check out the Unofficial Jupyter Notebook Extensions. Quick introductory blogs can be found here and here.
- pandas: `pandas` is heavily used in SuPy and thus better understanding of pandas is essential in SuPy workflows.
  - Introductory blogs:
    * Quick dive into Pandas for Data Science: introduction to pandas.
    * Basic Time Series Manipulation with Pandas: pandas-based time series manipulation.
    * Introduction to Data Visualization in Python: plotting using pandas and related libraries.
  - A detailed tutorial in Jupyter Notebooks:
    * Introduction to pandas
    * pandas fundamentals
    * Data Wrangling with pandas

Key IO Data Structures in SuPy

Introduction

The cell below demonstrates a minimal case of SuPy simulation with all key IO data structures included:

[1]:
```
import supy as sp
df_state_init, df_forcing = sp.load_SampleData()
df_output, df_state_final = sp.run_supy(df_forcing, df_state_init)
```

- Input: SuPy requires two `DataFrames` to perform a simulation, which are:
  - `df_state_init`: model initial states;
  - `df_forcing`: forcing data.

  These input data can be loaded either through calling load_SampleData() as shown above or using init_supy. Or, based on the loaded sample `DataFrames`, you can modify the content to create new `DataFrames` for your specific needs.
- Output: The output data by SuPy consists of two `DataFrames`:
  - `df_output`: model output results; this is usually the basis for scientific analysis.
  - `df_state_final`: model final states; any of its entries can be used as a `df_state_init` to start another SuPy simulation.

Input

`df_state_init`: model initial states

[2]:
```
df_state_init.head()
```

[2]:
```
var      ah_min       ah_slope_cooling      ah_slope_heating      ahprof_24hr  \
ind_dim    (0,)  (1,)             (0,) (1,)             (0,) (1,)      (0, 0)
grid
98         15.0  15.0              2.7  2.7              2.7  2.7        0.57

var                                                                ... tair24hr  \
ind_dim (0, 1) (1, 0) (1, 1) (2, 0) (2, 1) (3, 0) (3, 1) (4, 0)   ...    (275,)
grid                                                               ...
98        0.65   0.45   0.49   0.43   0.46    0.4   0.47    0.4   ...    273.15

var                                                                        \
ind_dim  (276,)  (277,)  (278,)  (279,)  (280,)  (281,)  (282,)  (283,)
grid
98       273.15  273.15  273.15  273.15  273.15  273.15  273.15  273.15

var                                     numcapita gridiv
ind_dim  (284,)  (285,)  (286,)  (287,)         0      0
grid
98       273.15  273.15  273.15  273.15    204.58     98

[1 rows x 1200 columns]
```

`df_state_init` is organised with **grids** in **rows** and **their states** in **columns**. The details of all state variables can be found in *the description page*.

Please note the properties are stored as *flattened values* to fit into the tabular format due to the nature of `DataFrame` though they may actually be of higher dimension (e.g. ahprof_24hr with the dimension {24, 2}). To indicate the variable dimensionality of these properties, SuPy use the `ind_dim` level in columns for indices of values:

- `0` for scalars;
- `(ind_dim1, ind_dim2, ...)` for arrays (for a generic sense, vectors are 1D arrays).

Take `ohm_coef` below for example, it has a dimension of {8, 4, 3} according to the description, which implies the actual values used by SuPy in simulations are passed in a layout as an array of the dimension {8, 4, 3}. As such, to get proper values passed in, users should follow the dimensionality requirement to prepare/modify `df_state_init`.

[3]:
```
df_state_init.loc[:,'ohm_coef']
```

[3]:
```
ind_dim  (0, 0, 0)  (0, 0, 1)  (0, 0, 2)  (0, 1, 0)  (0, 1, 1)  (0, 1, 2)  \
grid
98           0.719      0.194      -36.6      0.719      0.194      -36.6

ind_dim  (0, 2, 0)  (0, 2, 1)  (0, 2, 2)  (0, 3, 0)  (0, 3, 1)  (0, 3, 2)  \
grid
98           0.719      0.194      -36.6      0.719      0.194      -36.6

ind_dim  (1, 0, 0)  (1, 0, 1)  (1, 0, 2)  ...  (6, 3, 0)  (6, 3, 1)  \
grid                                      ...
98           0.238      0.427      -16.7  ...        0.5       0.21

ind_dim  (6, 3, 2)  (7, 0, 0)  (7, 0, 1)  (7, 0, 2)  (7, 1, 0)  (7, 1, 1)  \
grid
98           -39.1       0.25        0.6      -30.0       0.25        0.6

ind_dim  (7, 1, 2)  (7, 2, 0)  (7, 2, 1)  (7, 2, 2)  (7, 3, 0)  (7, 3, 1)  \
grid
98           -30.0       0.25        0.6      -30.0       0.25        0.6

ind_dim  (7, 3, 2)
grid
98           -30.0

[1 rows x 96 columns]
```

`df_forcing`: forcing data

`df_forcing` is organised with **temporal records** in **rows** and **forcing variables** in **columns**. The details of all forcing variables can be found in *the description page*.

The missing values can be specified with `-999`s, which are the default NANs accepted by SuPy and its backend SUEWS.

[4]:
```
df_forcing.head()
```

[4]:
```
                       iy  id  it  imin     qn     qh     qe     qs     qf  \
2012-01-01 00:05:00  2012   1   0     5 -999.0 -999.0 -999.0 -999.0 -999.0
2012-01-01 00:10:00  2012   1   0    10 -999.0 -999.0 -999.0 -999.0 -999.0
2012-01-01 00:15:00  2012   1   0    15 -999.0 -999.0 -999.0 -999.0 -999.0
```

(continues on next page)

(continued from previous page)

```
2012-01-01 00:20:00  2012   1   0    20 -999.0 -999.0 -999.0 -999.0 -999.0
2012-01-01 00:25:00  2012   1   0    25 -999.0 -999.0 -999.0 -999.0 -999.0

                         U        RH      Tair       pres  rain     kdown  \
2012-01-01 00:05:00  4.515  85.463333  11.77375  1001.5125   0.0  0.153333
2012-01-01 00:10:00  4.515  85.463333  11.77375  1001.5125   0.0  0.153333
2012-01-01 00:15:00  4.515  85.463333  11.77375  1001.5125   0.0  0.153333
2012-01-01 00:20:00  4.515  85.463333  11.77375  1001.5125   0.0  0.153333
2012-01-01 00:25:00  4.515  85.463333  11.77375  1001.5125   0.0  0.153333

                      snow  ldown   fcld    Wuh   xsmd    lai  kdiff   kdir  \
2012-01-01 00:05:00 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0
2012-01-01 00:10:00 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0
2012-01-01 00:15:00 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0
2012-01-01 00:20:00 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0
2012-01-01 00:25:00 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0 -999.0

                      wdir  isec
2012-01-01 00:05:00 -999.0   0.0
2012-01-01 00:10:00 -999.0   0.0
2012-01-01 00:15:00 -999.0   0.0
2012-01-01 00:20:00 -999.0   0.0
2012-01-01 00:25:00 -999.0   0.0
```

Note:

The index of `df_forcing` **SHOULD BE** strictly of `DatetimeIndex` type if you want create a `df_forcing` for SuPy simulation. The SuPy runtime time-step size is instructed by the `df_forcing` with its index information.

The infomation below indicates SuPy will run at a 5 min (i.e. 300 s) time-step if driven by this specific `df_forcing`:

[5]:
```
freq_forcing=df_forcing.index.freq
freq_forcing
```

[5]:
```
<300 * Seconds>
```

Output

`df_output`: model output results

`df_output` is organised with **temporal records of grids** in **rows** and **output variables of different groups** in **columns**. The details of all forcing variables can be found in *the description page*.

[6]:
```
df_output.head()
```

[6]:
```
group                              SUEWS                                      \
var                                Kdown        Kup        Ldown          Lup
grid datetime
98   2012-01-01 00:05:00  0.153333  0.018279  344.310184  371.986259
     2012-01-01 00:10:00  0.153333  0.018279  344.310184  371.986259
```

(continues on next page)

(continued from previous page)

```
     2012-01-01 00:15:00  0.153333  0.018279  344.310184  371.986259
     2012-01-01 00:20:00  0.153333  0.018279  344.310184  371.986259
     2012-01-01 00:25:00  0.153333  0.018279  344.310184  371.986259

group                                                                \
var                            Tsurf         QN         QF        QS
grid datetime
98   2012-01-01 00:05:00  11.775615 -27.541021  40.574001 -46.53243
     2012-01-01 00:10:00  11.775615 -27.541021  39.724283 -46.53243
     2012-01-01 00:15:00  11.775615 -27.541021  38.874566 -46.53243
     2012-01-01 00:20:00  11.775615 -27.541021  38.024849 -46.53243
     2012-01-01 00:25:00  11.775615 -27.541021  37.175131 -46.53243

group                                                                         \
var                               QH        QE    QHlumps   QElumps   QHresis
grid datetime
98   2012-01-01 00:05:00  62.420064  3.576493  49.732605  9.832804  0.042327
     2012-01-01 00:10:00  61.654096  3.492744  48.980360  9.735333  0.042294
     2012-01-01 00:15:00  60.885968  3.411154  48.228114  9.637861  0.042260
     2012-01-01 00:20:00  60.115745  3.331660  47.475869  9.540389  0.042226
     2012-01-01 00:25:00  59.343488  3.254200  46.723623  9.442917  0.042192

group                               ... DailyState                      \
var                      Rain  Irr  ...  WU_Grass2 WU_Grass3 deltaLAI
grid datetime                       ...
98   2012-01-01 00:05:00  0.0  0.0  ...        NaN       NaN      NaN
     2012-01-01 00:10:00  0.0  0.0  ...        NaN       NaN      NaN
     2012-01-01 00:15:00  0.0  0.0  ...        NaN       NaN      NaN
     2012-01-01 00:20:00  0.0  0.0  ...        NaN       NaN      NaN
     2012-01-01 00:25:00  0.0  0.0  ...        NaN       NaN      NaN

group                                                                 \
var                      LAIlumps AlbSnow DensSnow_Paved DensSnow_Bldgs
grid datetime
98   2012-01-01 00:05:00      NaN     NaN            NaN            NaN
     2012-01-01 00:10:00      NaN     NaN            NaN            NaN
     2012-01-01 00:15:00      NaN     NaN            NaN            NaN
     2012-01-01 00:20:00      NaN     NaN            NaN            NaN
     2012-01-01 00:25:00      NaN     NaN            NaN            NaN

group                                                               \
var                      DensSnow_EveTr DensSnow_DecTr DensSnow_Grass
grid datetime
98   2012-01-01 00:05:00            NaN            NaN            NaN
     2012-01-01 00:10:00            NaN            NaN            NaN
     2012-01-01 00:15:00            NaN            NaN            NaN
     2012-01-01 00:20:00            NaN            NaN            NaN
     2012-01-01 00:25:00            NaN            NaN            NaN

group
var                      DensSnow_BSoil DensSnow_Water  a1  a2  a3
grid datetime
```

(continues on next page)

(continued from previous page)

```
98   2012-01-01 00:05:00            NaN           NaN NaN NaN NaN
     2012-01-01 00:10:00            NaN           NaN NaN NaN NaN
     2012-01-01 00:15:00            NaN           NaN NaN NaN NaN
     2012-01-01 00:20:00            NaN           NaN NaN NaN NaN
     2012-01-01 00:25:00            NaN           NaN NaN NaN NaN

[5 rows x 218 columns]
```

`df_output` are recorded at the same temporal resolution as `df_forcing`:

[7]:
```
freq_out = df_output.index.levels[1].freq
(freq_out, freq_out == freq_forcing)
```

[7]:
```
(<300 * Seconds>, True)
```

df_state_final: model final states

`df_state_final` has the identical data structure as `df_state_init` except for the extra level `datetime` in index, which stores the temporal information associated with model states. Such structure can facilitate the reuse of it as initial model states for other simulations (e.g., diagnostics of runtime model states with `save_state=True` set in `run_supy`; or simply using it as the initial conditions for future simulations starting at the ending times of previous runs).

The meanings of state variables in `df_state_final` can be found in *the description page*.

[8]:
```
df_state_final.head()
```

[8]:
```
var                      aerodynamicresistancemethod ah_min        \
ind_dim                                            0   (0,)  (1,)
datetime            grid
2012-01-01 00:05:00 98                             2   15.0  15.0
2013-01-01 00:05:00 98                             2   15.0  15.0

var                      ah_slope_cooling      ah_slope_heating        \
ind_dim                              (0,) (1,)              (0,) (1,)
datetime            grid
2012-01-01 00:05:00 98                2.7  2.7               2.7  2.7
2013-01-01 00:05:00 98                2.7  2.7               2.7  2.7

var                      ahprof_24hr                                      \
ind_dim                       (0, 0) (0, 1) (1, 0) (1, 1) (2, 0) (2, 1)
datetime            grid
2012-01-01 00:05:00 98          0.57   0.65   0.45   0.49   0.43   0.46
2013-01-01 00:05:00 98          0.57   0.65   0.45   0.49   0.43   0.46

var                                ... wuprofm_24hr                      \
ind_dim                  (3, 0) (3, 1)  ...      (18, 0) (18, 1) (19, 0)
datetime            grid                ...
2012-01-01 00:05:00 98      0.4   0.47  ...       -999.0  -999.0  -999.0
2013-01-01 00:05:00 98      0.4   0.47  ...       -999.0  -999.0  -999.0

var                                                                       \
ind_dim                  (19, 1) (20, 0) (20, 1) (21, 0) (21, 1) (22, 0)
```

(continues on next page)

(continued from previous page)

```
datetime            grid
2012-01-01 00:05:00 98     -999.0  -999.0  -999.0  -999.0  -999.0  -999.0
2013-01-01 00:05:00 98     -999.0  -999.0  -999.0  -999.0  -999.0  -999.0

var                                                      z  z0m_in zdm_in
ind_dim                  (22, 1) (23, 0) (23, 1)     0      0      0
datetime            grid
2012-01-01 00:05:00 98     -999.0  -999.0  -999.0  49.6    1.9   14.2
2013-01-01 00:05:00 98     -999.0  -999.0  -999.0  49.6    1.9   14.2

[2 rows x 1200 columns]
```

API reference

Top-level Functions

`init_supy`(path_init[, force_reload, check_input])	Initialise supy by loading initial model states.
`load_forcing_grid`(path_runcontrol, grid[, ...])	Load forcing data for a specific grid included in the index of `df_state_init`.
`run_supy`(df_forcing, df_state_init[, ...])	Perform supy simulation.
`save_supy`(df_output, df_state_final[, ...])	Save SuPy run results to files
`load_SampleData`()	Load sample data for quickly starting a demo run.
`show_version`([mode, as_json])	print *SuPy* and `supy_driver` version information.

supy.init_supy

`supy.init_supy`(*path_init: str*, *force_reload=True*, *check_input=False*) → pandas.core.frame.DataFrame

Initialise supy by loading initial model states.

path_init [str]

Path to a file that can initialise SuPy, which can be either of the follows:

- SUEWS RunControl.nml: a namelist file for SUEWS configurations
- SuPy `df_state.csv`: a CSV file including model states produced by a SuPy run via `supy.save_supy()`

force_reload: boolean, optional Flag to force reload all initialisation files by clearing all cached states, with default value `True` (i.e., force reload all files). Note: If the number of simulation grids is large (e.g., > 100), `force_reload=False` is strongly recommended for better performance.

check_input: boolean, optional flag for checking validity of input: `df_forcing` and `df_state_init`. If set to `True`, any detected invalid input will stop SuPy simulation; a `False` flag will bypass such validation and may incur kernel error if any invalid input. *Note: such checking procedure may take some time if the input is large.* (the default is `False`, which bypasses the validation).

df_state_init: pandas.DataFrame Initial model states. See *df_state variables* for details.

1. Use RunControl.nml to initialise SuPy

```
>>> path_init = "~/SUEWS_sims/RunControl.nml"
>>> df_state_init = supy.init_supy(path_init)
```

2. Use `df_state.csv` to initialise SuPy

```
>>> path_init = "~/SuPy_res/df_state_test.csv"
>>> df_state_init = supy.init_supy(path_init)
```

supy.load_forcing_grid

supy.**load_forcing_grid**(*path_runcontrol: str*, *grid: int*, *check_input=False*, *force_reload=True*) → pandas.core.frame.DataFrame

Load forcing data for a specific grid included in the index of `df_state_init`.

path_runcontrol [str] Path to SUEWS RunControl.nml

grid [int] Grid number

check_input [bool, optional] flag for checking validity of input: `df_forcing` and `df_state_init`. If set to `True`, any detected invalid input will stop SuPy simulation; a `False` flag will bypass such validation and may incur kernel error if any invalid input. *Note: such checking procedure may take some time if the input is large.* (the default is `False`, which bypasses the validation).

df_forcing: pandas.DataFrame Forcing data. See *df_forcing variables* for details.

```
>>> path_runcontrol = "~/SUEWS_sims/RunControl.nml"  # a valid path to `RunControl.
↪nml`
>>> df_state_init = supy.init_supy(path_runcontrol) # get `df_state_init`
>>> grid = df_state_init.index[0] # first grid number included in `df_state_init`
>>> df_forcing = supy.load_forcing_grid(path_runcontrol, grid) # get df_forcing
```

supy.run_supy

supy.**run_supy**(*df_forcing: pandas.core.frame.DataFrame*, *df_state_init: pandas.core.frame.DataFrame*, *save_state=False*, *chunk_day=3660*, *logging_level=20*, *check_input=False*, *serial_mode=False*) → Tuple[pandas.core.frame.DataFrame, pandas.core.frame.DataFrame]

Perform supy simulation.

df_forcing [pandas.DataFrame] forcing data for all grids in `df_state_init`.

df_state_init [pandas.DataFrame] initial model states; or a collection of model states with multiple timestamps, whose last temporal record will be used as the initial model states.

save_state [bool, optional] flag for saving model states at each time step, which can be useful in diagnosing model runtime performance or performing a restart run. (the default is False, which instructs supy not to save runtime model states).

chunk_day [int, optional] chunk size (`chunk_day` days) to split simulation periods so memory usage can be reduced. (the default is 3660, which implies ~10-year forcing chunks used in simulations).

logging_level: logging level one of these values [50 (CRITICAL), 40 (ERROR), 30 (WARNING), 20 (INFO), 10 (DEBUG)]. A lower value informs SuPy for more verbose logging info.

check_input [bool, optional] flag for checking validity of input: `df_forcing` and `df_state_init`. If set to `True`, any detected invalid input will stop SuPy simulation; a `False` flag will bypass such validation and may incur kernel error if any invalid input. *Note: such checking procedure may take some time if the input is large.* (the default is `False`, which bypasses the validation).

serial_mode [bool, optional] If set to `True`, SuPy simulation will be conducted in serial mode; a `False` flag will try parallel simulation if possible (Windows not supported, i.e., always serial). (the default is `False`).

df_output, df_state_final [Tuple[pandas.DataFrame, pandas.DataFrame]]

- df_output: *output results*
- df_state_final: *final model states*

```
>>> df_output, df_state_final = supy.run_supy(df_forcing, df_state_init)
```

supy.save_supy

`supy.save_supy`(*df_output: pandas.core.frame.DataFrame*, *df_state_final: pandas.core.frame.DataFrame*, *freq_s: int = 3600*, *site: str = ''*, *path_dir_save: str = PosixPath('.')*, *path_runcontrol: str = None*, *save_tstep=False*, *logging_level=50*, *output_level=1*, *debug=False*) → list

Save SuPy run results to files

df_output [pandas.DataFrame] DataFrame of output

df_state_final [pandas.DataFrame] DataFrame of final model states

freq_s [int, optional] Output frequency in seconds (the default is 3600, which indicates hourly output)

site [str, optional] Site identifier (the default is ‘’, which indicates site identifier will be left empty)

path_dir_save [str, optional] Path to directory to saving the files (the default is Path(‘.’), which indicates the current working directory)

path_runcontrol [str, optional] Path to SUEWS RunControl.nml, which, if set, will be preferably used to derive `freq_s`, `site` and `path_dir_save`. (the default is None, which is unset)

save_tstep [bool, optional] whether to save results in temporal resolution as in simulation (which may result very large files and slow progress), by default False.

logging_level: logging level one of these values [50 (CRITICAL), 40 (ERROR), 30 (WARNING), 20 (INFO), 10 (DEBUG)]. A lower value informs SuPy for more verbose logging info.

output_level [integer, optional] option to determine selection of output variables, by default 1. Notes: 0 for all but snow-related; 1 for all; 2 for a minimal set without land cover specific information.

debug [bool, optional] whether to enable debug mode (e.g., writing out in serial mode, and other debug uses), by default False.

list a list of paths of saved files

1. save results of a supy run to the current working directory with default settings

```
>>> list_path_save = supy.save_supy(df_output, df_state_final)
```

2. save results according to settings in RunControl.nml

```
>>> list_path_save = supy.save_supy(df_output, df_state_final, path_runcontrol=
↪'path/to/RunControl.nml')
```

3. save results of a supy run at resampling frequency of 1800 s (i.e., half-hourly results) under the site code `Test` to a customised location 'path/to/some/dir'

```
>>> list_path_save = supy.save_supy(df_output, df_state_final, freq_s=1800, site=
↪'Test', path_dir_save='path/to/some/dir')
```

supy.load_SampleData

`supy.load_SampleData()` → Tuple[pandas.core.frame.DataFrame, pandas.core.frame.DataFrame]

Load sample data for quickly starting a demo run.

df_state_init, df_forcing: Tuple[pandas.DataFrame, pandas.DataFrame]

- df_state_init: *initial model states*
- df_forcing: *forcing data*

```
>>> df_state_init, df_forcing = supy.load_SampleData()
```

supy.show_version

`supy.show_version`(*mode='simple'*, *as_json=False*)

print *SuPy* and `supy_driver` version information.

Utility Functions

ERA-5 Data Downloader

download_era5(lat_x, lon_x, start, end, ...)	Generate ERA-5 cdsapi-based requests and download data for area of interests.
gen_forcing_era5(lat_x, lon_x, start, end[, ...])	Generate SUEWS forcing files using ERA-5 data.

supy.util.download_era5

`supy.util.download_era5`(*lat_x: float*, *lon_x: float*, *start: str*, *end: str*, *simple_mode: bool*, *dir_save=PosixPath('.')*, *grid=None*, *scale=0*, *logging_level=20*) → dict

Generate ERA-5 cdsapi-based requests and download data for area of interests.

lat_x [float] Latitude of centre at the area of interest.

lon_x [float] Longitude of centre at the area of interest.

start [str] Any datetime-like string that can be parsed by `pandas.daterange()`.

end [str] Any datetime-like string that can be parsed by `pandas.daterange()`.

grid [list, optional] grid size used in CDS request API, by default [0.125, 0.125].

scale [int, optional] scaling factor that determines the area of interest (i.e., `area=grid[0]*scale`), by default 0.

dir_save: Path or path-like string path to directory for saving downloaded ERA5 netCDF files.

logging_level: logging level one of these values [50 (CRITICAL), 40 (ERROR), 30 (WARNING), 20 (INFO), 10 (DEBUG)]. A lower value informs SuPy for more verbose logging info.

dict key: name of downloaded file. value: CDS API request used for downloading the file named by the corresponding key.

This function uses CDS API to download ERA5 data; follow this for configuration first: https://cds.climate.copernicus.eu/api-how-to

supy.util.gen_forcing_era5

`supy.util.gen_forcing_era5`(*lat_x: float*, *lon_x: float*, *start: str*, *end: str*, *dir_save=PosixPath('.')*, *grid=None*, *hgt_agl_diag=100.0*, *scale=0*, *force_download=True*, *simple_mode=True*, *pressure_level=None*, *logging_level=20*) → list

Generate SUEWS forcing files using ERA-5 data.

lat_x [float] Latitude of centre at the area of interest.

lon_x [float] Longitude of centre at the area of interest.

start [str] Any datetime-like string that can be parsed by `pandas.daterange()`.

end [str] Any datetime-like string that can be parsed by `pandas.daterange()`.

dir_save: Path or path-like string path to directory for saving downloaded ERA5 netCDF files.

grid [list, optional] grid size used in CDS request API, by default [0.125, 0.125].

hgt_agl_diag: float height above ground level to diagnose forcing variables, by default 100; the ground level is taken from ERA5 grid altitude.

scale [int, optional] scaling factor that determines the area of interest (i.e., `area=grid[0]*scale`), by default 0

force_download: boolean, optional flag to determine whether to download required ERA5 netCDF files; if `False`, all ERA5-related nc files in `dir_save` will be picked up for generation. by default True.

simple_mode: boolean if use the *simple* mode for diagnosing the forcing variables, by default `True`. In the simple mode, temperature is diagnosed using environmental lapse rate 6.5 K/km and wind speed using MOST under neutral condition. If `False`, MOST with consideration of stability conditions will be used to diagnose forcing variables.

pressure_level: float pressure level to retrieve ERA5 atmospheric data, by default None. If `None`, this option is ignored. If not `None`, calculations implied by `simple_mode` will be skipped: the data at specified pressure level will be used as forcing data and the mean altitude of the pressure level between specified `start` and `end` will be assumed to be the forcing height (i.e., `hgt_agl_diag` will be ignored if set).

logging_level: logging level one of these values [50 (CRITICAL), 40 (ERROR), 30 (WARNING), 20 (INFO), 10 (DEBUG)]. A lower value informs SuPy for more verbose logging info.

List A list of files in SUEWS forcing input format.

1. This function uses CDS API to download ERA5 data; follow this for configuration first: https://cds.climate.copernicus.eu/api-how-to

2. The generated forcing files can be imported using `supy.util.read_forcing` to get simulation-ready `pandas.DataFrame`s.
3. See Section 3.10.2 and 3.10.3 in the reference for details of diagnostics calculation.
4. For `start/end`, it is recommended to use the format `YYYY-MM-DD` to avoid confusion in day/month-first convensions (an upstream known issue due to the `dateutil` behavior)

ECMWF, S. P. (2016). In IFS documentation CY41R2 Part IV: Physical Processes. ECMWF: Reading, UK, 111-113. https://www.ecmwf.int/en/elibrary/16648-part-iv-physical-processes

Typical Meteorological Year

gen_epw(df_output, lat, lon[, tz, path_epw])	Generate an `epw` file of uTMY (urbanised Typical Meteorological Year) using SUEWS simulation results
read_epw(path_epw)	Read in `epw` file as a DataFrame

supy.util.gen_epw

`supy.util.`**`gen_epw`**(*df_output: pandas.core.frame.DataFrame*, *lat*, *lon*, *tz=0*, *path_epw=PosixPath('uTMY.epw')*) → Tuple[pandas.core.frame.DataFrame, str, pathlib.Path]

Generate an `epw` file of uTMY (urbanised Typical Meteorological Year) using SUEWS simulation results

df_output [pd.DataFrame] SUEWS simulation results.

path_epw [Path, optional] Path to store generated epw file, by default Path('./uTMY.epw').

lat: float Latitude of the site, used for calculating solar angle.

lon: float Longitude of the site, used for calculating solar angle.

tz: float time zone represented by time difference from UTC+0 (e.g., 8 for UTC+8), by default 0 (i.e., UTC+0)

df_epw, text_meta, path_epw: Tuple[pd.DataFrame, str, Path]

- df_epw: uTMY result
- text_meta: meta-info text
- path_epw: path to generated `epw` file

supy.util.read_epw

`supy.util.`**`read_epw`**(*path_epw: pathlib.Path*) → pandas.core.frame.DataFrame

Read in `epw` file as a DataFrame

path_epw [Path] path to `epw` file

df_tmy: pd.DataFrame TMY results of `epw` file

Gap Filling

fill_gap_all(ser_to_fill[, freq, ...])	Fill all gaps in a time series using data from neighbouring divisions of 'freq'

supy.util.fill_gap_all

supy.util.**fill_gap_all**(*ser_to_fill: pandas.core.series.Series*, *freq='1D'*, *limit_fill=1*, *thresh_ratio=0.8*) → pandas.core.series.Series

Fill all gaps in a time series using data from neighbouring divisions of 'freq'

ser_to_fill [pd.Series] Time series to gap-fill

freq [str, optional] Frequency to identify gapped divisions, by default '1D'

limit_fill: int, optional Maximum number of consecutive NaNs to fill. Any number less than one means no pre-gap-filling interpolation will be done.

ser_test_filled: pd.Series Gap-filled time series.

010: missing data in division between others with no missing data 01: missing data in division after one with no missing data 10: division with missing data before one with no missing data

OHM

derive_ohm_coef(ser_QS, ser_QN)	A function to linearly fit two independant variables to a dependent one.
sim_ohm(ser_qn, a1, a2, a3)	Calculate QS using OHM (Objective Hysteresis Model).

supy.util.derive_ohm_coef

supy.util.**derive_ohm_coef**(*ser_QS*, *ser_QN*)

A function to linearly fit two independant variables to a dependent one.

ser_QS [pd.Series] The dependent variable QS (Surface heat storage).

ser_QN [pd.Series] The first independent variable (Net all wave radiation).

Tuple a1, a2 coefficients and a3 (intercept)

supy.util.sim_ohm

supy.util.**sim_ohm**(*ser_qn: pandas.core.series.Series*, *a1: float*, *a2: float*, *a3: float*) → pandas.core.series.Series

Calculate QS using OHM (Objective Hysteresis Model).

ser_qn [pd.Series] net all-wave radiation.

a1 [float] a1 of OHM coefficients.

a2 [float] a2 of OHM coefficients.

a3 [float] a3 of OHM coefficients.

pd.Series heat storage flux calculated by OHM.

Surface Conductance

cal_gs_suews(kd, ta_c, rh, pa, smd, lai, ...)	Model surface conductance/resistance using phenology and atmospheric forcing conditions.
cal_gs_obs(qh, qe, ta, rh, pa, ra)	Calculate surface conductance based on observations, notably turbulent fluxes.
calib_g(df_fc_suews, ser_ra, g_max, lai_max, ...)	Calibrate parameters for modelling surface conductance over vegetated surfaces using `LMFIT`.

supy.util.cal_gs_suews

`supy.util.`**`cal_gs_suews`**(*kd*, *ta_c*, *rh*, *pa*, *smd*, *lai*, *g_cst*, *g_max*, *lai_max*, *wp_smd*, *debug=False*)

Model surface conductance/resistance using phenology and atmospheric forcing conditions.

kd [numeric] Incoming solar radiation [W m-2]

ta_c [numeric] Air temperature [degC]

rh [numeric] Relative humidity [%]

pa [numeric] Air pressure [Pa]

smd [numeric] Soil moisture deficit [mm]

lai [numeric] Leaf area index [m2 m-2]

g_cst [size-6 array] Parameters to determine surface conductance/resistance: g_lai (LAI related), g_kd (solar radiation related), g_dq_base (humidity related), g_dq_shape (humidity related), g_ta (air temperature related), g_smd (soil moisture related)

g_max [numeric] Maximum surface conductance [mm s-1]

lai_max [numeric] Maximum LAI [m2 m-2]

wp_smd [numeric] Wilting point indicated as soil moisture deficit [mm]

numeric Modelled surface conductance [mm s-1]

supy.util.cal_gs_obs

`supy.util.`**`cal_gs_obs`**(*qh*, *qe*, *ta*, *rh*, *pa*, *ra*)

Calculate surface conductance based on observations, notably turbulent fluxes.

qh [numeric] Sensible heat flux [W m-2]

qe [numeric] Latent heat flux [W m-2]

ta [numeric] Air temperature [degC]

rh [numeric] Relative humidity [%]

pa [numeric] Air pressure [Pa]

numeric Surface conductance based on observations [mm s-1]

supy.util.calib_g

supy.util.**calib_g**(*df_fc_suews*, *ser_ra*, *g_max*, *lai_max*, *wp_smd*, *method='cobyla'*, *prms_init=None*, *debug=False*)

Calibrate parameters for modelling surface conductance over vegetated surfaces using `LMFIT`.

df_fc_suews [pandas.DataFrame] DataFrame in SuPy forcing format

ser_ra: pandas.Series Series with RA, aerodynamic resistance, [s m-1]

g_max [numeric] Maximum surface conductance [mm s-1]

lai_max [numeric] Maximum LAI [m2 m-2]

wp_smd [numeric] Wilting point indicated as soil moisture deficit [mm]

method: str, optional Method used in minimisation by `lmfit.minimize`: details refer to its `method`.

prms_init: lmfit.Parameters, optional Initial parameters for calibration

debug [bool, optional] Option to output final calibrated `ModelResult`, by default False

dict, or `ModelResult` if `debug==True`

1. dict: {parameter_name -> best_fit_value}
2. `ModelResult`

Note: Parameters for surface conductance: g_lai (LAI related), g2 (solar radiation related), g_dq_base (humidity related), g_dq_shape (humidity related), g_ta (air temperature related), g_smd (soil moisture related)

For calibration validity, turbulent fluxes, QH and QE, in `df_fc_suews` should ONLY be observations, i.e., interpolated values should be avoided. To do so, please place `np.nan` as missing values for QH and QE.

WRF-SUEWS

extract_reclassification(path_nml)	Extract reclassification info from `path_nml` as a DataFrame.
plot_reclassification(path_nml[, path_save, ...])	Produce Sankey Diagram to visualise the reclassification specified in `path_nml`

supy.util.extract_reclassification

supy.util.**extract_reclassification**(*path_nml: str*) → pandas.core.frame.DataFrame

Extract reclassification info from `path_nml` as a DataFrame.

path_nml [str] Path to `namelist.suews`

pd.DataFrame Reclassification DataFrame with rows for WRF land covers while columns for SUEWS.

supy.util.plot_reclassification

supy.util.**plot_reclassification**(*path_nml: str*, *path_save='LC-WRF-SUEWS.png'*, *width=800*, *height=360*, *top=10*, *bottom=10*, *left=280*, *right=130*)

Produce Sankey Diagram to visualise the reclassification specified in `path_nml`

path_nml [str] Path to `namelist.suews`

path_save [str, optional] Path to save Sankey diagram, by default 'LC-WRF-SUEWS.png'

width [int, optional] Width of diagram, by default 800

height [int, optional] Height of diagram, by default 360

top [int, optional] Top margin of diagram, by default 10

bottom [int, optional] Bottom margin of diagram, by default 10

left [int, optional] Left margin of diagram, by default 260

right [int, optional] Right margin of diagram, by default 60

Sankey Diagram Sankey Diagram showing the reclassification.

Plotting

plot_comp(df_var[, scatter_kws, kde_kws, ...])	Produce a scatter plot with linear regression line to compare simulation results and observations.
plot_day_clm(df_var[, fig, ax, show_dif, ...])	Produce a ensemble diurnal climatologies with uncertainties shown in inter-quartile ranges.
plot_rsl(df_output[, var, fig, ax])	Produce a quick plot of RSL results

supy.util.plot_comp

supy.util.**plot_comp**(*df_var*, *scatter_kws={'alpha': 0.1, 'color': 'k', 's': 0.3}*, *kde_kws={'levels': 4, 'shade': True, 'shade_lowest': False}*, *show_pdf=False*, *fig=None*, *ax=None*)

Produce a scatter plot with linear regression line to compare simulation results and observations.

df_var [pd.DataFrame] DataFrame containing variables to plot with datetime as index. Two columns, 'Obs' and 'Sim' for observations and simulation results, respectively, must exist.

scatter_kws: dict keyword arguments passed to `sns.regplot`. By default, `{"alpha": 0.1, "s": 0.3, "color": "k"}`.

show_pdf: boolean if a PDF overlay should be added. By default, `False`.

kde_kws: dict `kde_kws` passed to `sns.kdeplot` when `show_pdf=True`

MPL.figure figure showing 1:1 line plot

supy.util.plot_day_clm

`supy.util.plot_day_clm(df_var, fig=None, ax=None, show_dif=False, col_ref='Obs')`

Produce a ensemble diurnal climatologies with uncertainties shown in inter-quartile ranges.

df_var [pd.DataFrame] DataFrame containing variables to plot with datetime as index.

show_dif: boolean flag to determine if differences against `col_ref` should be plotted.

col_ref: str name of column that is used as reference to show differences instead of original values.

MPL.figure figure showing median lines and IQR in shadings

supy.util.plot_rsl

`supy.util.plot_rsl(df_output, var=None, fig=None, ax=None)`

Produce a quick plot of RSL results

df_output [pandas.DataFrame] SuPy output dataframe with RSL results.

var [str, optional] Varible to plot; must be one of 'U', 'T', or 'q'; or use `None` to plot all; by default None

tuple `(fig,ax)` of plot.

issue If an invalid variable is specified, an issue will be raised.

Roughness Calculation

cal_z0zd(ser_qh, ser_ustar, ser_ta_c, ...[, ...])	Calculates surface roughness and zero plane displacement height.
cal_neutral(ser_qh, ser_ustar, ser_ta_c, ...)	Calculates the rows associated with neutral condition (threshold=0.01)

supy.util.cal_z0zd

`supy.util.cal_z0zd(ser_qh, ser_ustar, ser_ta_c, ser_rh_pct, ser_pres_hpa, ser_ws, z_meas, h_sfc, debug=False)`

Calculates surface roughness and zero plane displacement height. Refer to https://suews-parameters-docs.readthedocs.io/en/latest/steps/roughness-SuPy.html for example

ser_qh: pd.DataFrame sensible heat flux [W/m^2]

ser_ustar: pd.Series friction velocity [m/s]

ser_ta_c: pd.Series air temperature [°C]

ser_rh_pct: pd.Series relative humidity [%]

ser_pres_hpa: pd.Series air pressure [hPa]

z_meas: number measurement height in m

h_sfc: number vegetation height in m

debug [bool, optional] Option to output final calibrated `ModelResult`, by default False

z0 surface roughness length for momentum

zd zero displacement height

supy.util.cal_neutral

`supy.util.`**`cal_neutral`**(*ser_qh*, *ser_ustar*, *ser_ta_c*, *ser_rh_pct*, *ser_pres_hpa*, *ser_ws*, *z_meas*, *h_sfc*)

Calculates the rows associated with neutral condition (threshold=0.01)

ser_qh: pd.DataFrame sensible heat flux [W/m^2]

ser_ustar: pd.Series friction velocity [m/s]

ser_ta_c: pd.Series air temperature [°C]

ser_rh_pct: pd.Series relative humidity [%]

ser_pres_hpa: pd.Series air pressure [hPa]

ser_ws: pd.Series wind speed [m/s]

z_meas measurement height [m]

h_sfc vegetation height [m]

ser_ws_neutral: pd.Series observation time series of WS (Neutral conditions)

ser_ustar_neutral: pd.Series observation time series of u* (Neutral conditions)

Command-Line Tools

suews-run

Run SUEWS simulation using settings in PATH_RUNCONTROL (default: "./RunControl.nml", i.e., the RunControl namelist file in the current directory).

Examples:

1. Run SUEWS simulation using the RunControl namelist file in the current directory:

   $ suews-run -p ./RunControl.nml

2. Run SUEWS simulation using an arbitray RunControl namelist file by specifying the path to the RunControl namelist file:

   $ suews-run -p /path/to/RunControl.nml

```
suews-run [OPTIONS]
```

Options

-p, --path_runcontrol <path_runcontrol>

Path to the RunControl namelist file (default: ./RunControl.nml).

suews-convert

Convert SUEWS input tables from older versions to newer ones (one-way only).

```
suews-convert [OPTIONS]
```

Options

-f, --from <fromVer>

Required Version to convert from

Options 2019b | 2019a | 2018c | 2018b | 2018a | 2017a | 2016a

-t, --to <toVer>

Required Version to convert to

Options 2020a | 2019b | 2019a | 2018c | 2018b | 2018a | 2017a

-i, --input <fromDir>

Required Original directory to convert, which must have the *RunControl.nml* file

-o, --output <toDir>

Required New directory to create for converted tables. Note: the created directory will have the same structure as the origianl one; however, forcing files and output folder won't be includede.

Key Data Structures

`df_state` variables

Note: Data structure of `df_state` is explained here.

aerodynamicresistancemethod

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

ah_min

Description Minimum QF values.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *AHMin_WD*, *AHMin_WE*

`ah_slope_cooling`

Description Cooling slope of QF calculation.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *AHSlope_Cooling_WD*, *AHSlope_Cooling_WE*

`ah_slope_heating`

Description Heating slope of QF calculation.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *AHSlope_Heating_WD*, *AHSlope_Heating_WE*

`ahprof_24hr`

Description Hourly profile values used in energy use calculation.

Dimensionality (24, 2)

Dimensionality Remarks 24: hours of a day

2: {Weekday, Weekend}

SUEWS-related variables *EnergyUseProfWD*, *EnergyUseProfWE*

`air_ext_lw`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`air_ext_sw`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`air_ssa_lw`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`air_ssa_sw`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`alb`

Description Effective surface albedo (middle of the day value) for summertime.

Dimensionality (7,)

Dimensionality Remarks 7: { `Paved`, `Bldgs`, `EveTr`, `DecTr`, `Grass`, `BSoil`, `Water`}

SUEWS-related variables `AlbedoMax`

`albdectr_id`

Description Albedo of deciduous surface `DecTr` on day 0 of run

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `albDecTr0`

`albevetr_id`

Description Albedo of evergreen surface `EveTr` on day 0 of run

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `albEveTr0`

`albgrass_id`

Description Albedo of grass surface `Grass` on day 0 of run

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `albGrass0`

`albmax_dectr`

Description Effective surface albedo (middle of the day value) for summertime.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `AlbedoMax`

`albmax_evetr`

Description Effective surface albedo (middle of the day value) for summertime.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `AlbedoMax`

`albmax_grass`

Description Effective surface albedo (middle of the day value) for summertime.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *AlbedoMax*

`albmin_dectr`

Description Effective surface albedo (middle of the day value) for wintertime (not including snow).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *AlbedoMin*

`albmin_evetr`

Description Effective surface albedo (middle of the day value) for wintertime (not including snow).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *AlbedoMin*

`albmin_grass`

Description Effective surface albedo (middle of the day value) for wintertime (not including snow).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *AlbedoMin*

`alpha_bioco2`

Description The mean apparent ecosystem quantum. Represents the initial slope of the light-response curve.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *alpha*

`alpha_enh_bioco2`

Description Part of the *alpha* coefficient related to the fraction of vegetation.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *alpha_enh*

`alt`

Description Altitude of grids [m].

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Alt*

`baset`

Description Base Temperature for initiating growing degree days (GDD) for leaf growth. [°C]

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *BaseT*

`baset_cooling`

Description Critical cooling temperature.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *TCritic_Cooling_WD*, *TCritic_Cooling_WE*

`baset_hc`

Description Base temperature for heating degree days [°C]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *BaseT_HC*

`baset_heating`

Description Critical heating temperature.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *TCritic_Heating_WD*, *TCritic_Heating_WE*

`basete`

Description Base temperature for initiating sensesance degree days (SDD) for leaf off. [°C]

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *BaseTe*

`basetmethod`

Description Determines method for base temperature used in HDD/CDD calculations.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *BaseTMethod*

`beta_bioco2`

Description The light-saturated gross photosynthesis of the canopy. [umol m^{-2} s^{-1}]

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *beta*

beta_enh_bioco2

Description Part of the *beta* coefficient related to the fraction of vegetation.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *beta_enh*

bldgh

Description Mean building height [m]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *H_Bldgs*

capmax_dec

Description Maximum water storage capacity for upper surfaces (i.e. canopy)

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *StorageMax*

capmin_dec

Description Minimum water storage capacity for upper surfaces (i.e. canopy).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *StorageMin*

chanohm

Description Bulk transfer coefficient for this surface to use in AnOHM [-]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *AnOHM_Ch*

co2pointsource

Description CO2 emission factor [kg km^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *CO2PointSource*

cpanohm

Description Volumetric heat capacity for this surface to use in AnOHM [J m^{-3}]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *AnOHM_Cp*

`crwmax`

Description Maximum water holding capacity of snow [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *CRWMax*

`crwmin`

Description Minimum water holding capacity of snow [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *CRWMin*

`daywat`

Description Irrigation flag: 1 for on and 0 for off.

Dimensionality (7,)

Dimensionality Remarks 7: {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}

SUEWS-related variables *DayWat(1)*, *DayWat(2)*, *DayWat(3)*, *DayWat(4)*, *DayWat(5)*, *DayWat(6)*, *DayWat(7)*

`daywatper`

Description Fraction of properties using irrigation for each day of a week.

Dimensionality (7,)

Dimensionality Remarks 7: {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}

SUEWS-related variables *DayWatPer(1)*, *DayWatPer(2)*, *DayWatPer(3)*, *DayWatPer(4)*, *DayWatPer(5)*, *DayWatPer(6)*, *DayWatPer(7)*

`decidcap_id`

Description Storage capacity of deciduous surface *DecTr* on day 0 of run.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *decidCap0*

`dectreeh`

Description Mean height of deciduous trees [m]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *H_DecTr*

`diagmethod`

Description Defines how near surface diagnostics are calculated.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *DiagMethod*

`diagnose`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`diagqn`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`diagqs`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`drainrt`

Description Drainage rate of bucket for LUMPS [mm h^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *LUMPS_DrRate*

`ef_umolco2perj`

Description Emission factor for fuels used for building heating.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *EF_umolCO2perJ*

`emis`

Description Effective surface emissivity.

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *Emissivity*

`emissionsmethod`

Description Determines method for QF calculation.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *EmissionsMethod*

enddls

Description End of the day light savings [DOY]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *EndDLS*

enef_v_jkm

Description Emission factor for heat [J k m^{-1}].

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *EnEF_v_Jkm*

evapmethod

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

evetreeh

Description Mean height of evergreen trees [m]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *H_EveTr*

faibldg

Description Frontal area index for buildings [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *FAI_Bldgs*

faidectree

Description Frontal area index for deciduous trees [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *FAI_DecTr*

faievetree

Description Frontal area index for evergreen trees [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *FAI_EveTr*

`faut`

Description Fraction of irrigated area that is irrigated using automated systems

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Faut*

`fcef_v_kgkm`

Description CO2 emission factor for weekdays [kg km^{-1}];;CO2 emission factor for weekends [kg km^{-1}]

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *FcEF_v_kgkmWD*, *FcEF_v_kgkmWE*

`flowchange`

Description Difference in input and output flows for water surface [mm h^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *FlowChange*

`frfossilfuel_heat`

Description Fraction of fossil fuels used for building heating [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *FrFossilFuel_Heat*

`frfossilfuel_nonheat`

Description Fraction of fossil fuels used for building energy use [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *FrFossilFuel_NonHeat*

`g1`

Description Related to maximum surface conductance [mm s^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *G1*

`g2`

Description Related to Kdown dependence [W m^{-2}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *G2*

g3

Description Related to VPD dependence [units depend on *gsModel*]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *G3*

g4

Description Related to VPD dependence [units depend on *gsModel*]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *G4*

g5

Description Related to temperature dependence [°C]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *G5*

g6

Description Related to soil moisture dependence [mm^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *G6*

gddfull

Description The growing degree days (GDD) needed for full capacity of the leaf area index (LAI) [°C].

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *GDDFull*

ground_albedo_dir_mult_fact

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`gsmodel`

Description Formulation choice for conductance calculation.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *gsModel*

`h_maintain`

Description water depth to maintain used in automatic irrigation (e.g., ponding water due to flooding irrigation in rice crop-field) [mm].

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *H_maintain*

`humactivity_24hr`

Description Hourly profile values used in human activity calculation.

Dimensionality (24, 2)

Dimensionality Remarks 24: hours of a day

2: {Weekday, Weekend}

SUEWS-related variables *ActivityProfWD*, *ActivityProfWE*

`ie_a`

Description Coefficient for automatic irrigation model.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *Ie_a1*, *Ie_a2*, *Ie_a3*

`ie_end`

Description Day when irrigation ends [DOY]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Ie_end*

`ie_m`

Description Coefficient for manual irrigation model.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *Ie_m1*, *Ie_m2*, *Ie_m3*

`ie_start`

Description Day when irrigation starts [DOY]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Ie_start*

internalwateruse_h

Description Internal water use [mm h^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *InternalWaterUse*

irrfracbldgs

Description Fraction of *Bldgs* that is irrigated [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *IrrFr_Bldgs*

irrfracbsoil

Description Fraction of *BSoil* that is irrigated [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *IrrFr_BSoil*

irrfracdectr

Description Fraction of *DecTr* that is irrigated [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *IrrFr_DecTr*

irrfracevetr

Description Fraction of *EveTr* that is irrigated [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *IrrFr_EveTr*

irrfracgrass

Description Fraction of *Grass* that is irrigated [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *IrrFr_Grass*

irrfracpaved

Description Fraction of *Paved* that is irrigated [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *IrrFr_Paved*

`irrfracwater`

Description Fraction of *Water* that is irrigated [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *IrrFr_Water*

`kkanohm`

Description Thermal conductivity for this surface to use in AnOHM [W m K^{-1}]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *AnOHM_Kk*

`kmax`

Description Maximum incoming shortwave radiation [W m^{-2}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Kmax*

`lai_id`

Description Initial LAI values.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *LAIinitialDecTr*, *LAIinitialEveTr*, *LAIinitialGrass*

`laicalcyes`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`laimax`

Description full leaf-on summertime value

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *LAIMax*

`laimin`

Description leaf-off wintertime value

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *LAIMin*

`laipower`

Description parameters required by LAI calculation.

Dimensionality (4, 3)

Dimensionality Remarks 4: {*LeafGrowthPower1*, *LeafGrowthPower2*, *LeafOffPower1*, *LeafOffPower2*}

3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *LeafGrowthPower1*, *LeafGrowthPower2*, *LeafOffPower1*, *LeafOffPower2*

`laitype`

Description LAI calculation choice.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *LAIEq*

`lat`

Description Latitude [deg].

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *lat*

`lng`

Description longitude [deg]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *lng*

`maxconductance`

Description The maximum conductance of each vegetation or surface type. [mm s^{-1}]

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *MaxConductance*

`maxfcmetab`

Description Maximum (day) CO2 from human metabolism. [W m^{-2}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *MaxFCMetab* `maxqfmetab`

Description Maximum value for human heat emission. [W m^{-2}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *MaxQFMetab*

`min_res_bioco2`

Description Minimum soil respiration rate (for cold-temperature limit) [umol m^{-2} s^{-1}].

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *min_respi*

`minfcmetab`

Description Minimum (night) CO2 from human metabolism. [W m^{-2}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *MinFCMetab*

`minqfmetab`

Description Minimum value for human heat emission. [W m^{-2}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *MinQFMetab*

`n_stream_lw_urban`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`n_stream_sw_urban`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`n_vegetation_region_urban`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

narp_emis_snow

Description Effective surface emissivity.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Emissivity*

narp_trans_site

Description Atmospheric transmissivity for NARP [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *NARP_Trans*

netradiationmethod

Description Determines method for calculation of radiation fluxes.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *NetRadiationMethod*

ohm_coef

Description Coefficients for OHM calculation.

Dimensionality (8, 4, 3)

Dimensionality Remarks 8: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*, one extra land cover type (currently NOT used)}

4: {SummerWet, SummerDry, WinterWet, WinterDry}

3: {a1, a2, a3}

SUEWS-related variables *a1*, *a2*, *a3*

ohm_threshsw

Description Temperature threshold determining whether summer/winter OHM coefficients are applied [°C]

Dimensionality (8,)

Dimensionality Remarks 8: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*, one extra land cover type (currently NOT used)}

SUEWS-related variables *OHMThresh_SW*

ohm_threshwd

Description Soil moisture threshold determining whether wet/dry OHM coefficients are applied [-]

Dimensionality (8,)

Dimensionality Remarks 8: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*, one extra land cover type (currently NOT used)}

SUEWS-related variables *OHMThresh_WD*

ohmincqf

Description Determines whether the storage heat flux calculation uses Q^* or (Q^* +QF).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *OHMIncQF*

pipecapacity

Description Storage capacity of pipes [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *PipeCapacity*

popdensdaytime

Description Daytime population density (i.e. workers, tourists) [people ha^{-1}]

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *PopDensDay*

popdensnighttime

Description Night-time population density (i.e. residents) [people ha^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *PopDensNight*

popprof_24hr

Description Hourly profile values used in dynamic population estimation.

Dimensionality (24, 2)

Dimensionality Remarks 24: hours of a day

2: {Weekday, Weekend}

SUEWS-related variables *PopProfWD*, *PopProfWE*

pormax_dec

Description full leaf-on summertime value Used only for *DecTr* (can affect roughness calculation)

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *PorosityMax*

pormin_dec

Description leaf-off wintertime value Used only for *DecTr* (can affect roughness calculation)

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *PorosityMin*

porosity_id

Description Porosity of deciduous vegetation on day 0 of run.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *porosity0*

preciplimit

Description Temperature limit when precipitation falls as snow [°C]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *PrecipLimSnow*

preciplimitalb

Description Limit for hourly precipitation when the ground is fully covered with snow [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *PrecipLimAlb*

qf0_beu

Description Building energy use [W m^{-2}]

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *QF0_BEU_WD*, *QF0_BEU_WE*

qf_a

Description Base value for QF calculation.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *QF_A_WD*, *QF_A_WE*

qf_b

Description Parameter related to heating degree days.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *QF_B_WD*, *QF_B_WE*

qf_c

Description Parameter related to heating degree days.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *QF_C_WD*, *QF_C_WE*

`radmeltfact`

Description Hourly radiation melt factor of snow [mm W^{-1} h^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *RadMeltFactor*

`raincover`

Description Limit when surface totally covered with water for LUMPS [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *LUMPS_Cover*

`rainmaxres`

Description Maximum water bucket reservoir [mm] Used for LUMPS surface wetness control.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *LUMPS_MaxRes*

`resp_a`

Description Respiration coefficient a.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *resp_a*

`resp_b`

Description Respiration coefficient b - related to air temperature dependency.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *resp_b*

`roughlenheatmethod`

Description Determines method for calculating roughness length for heat.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *RoughLenHeatMethod*

`roughlenmommethod`

Description Determines how aerodynamic roughness length (z0m) and zero displacement height (zdm) are calculated.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *RoughLenMomMethod*

`runofftowater`

Description Fraction of above-ground runoff flowing to water surface during flooding [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *RunoffToWater*

`s1`

Description A parameter related to soil moisture dependence [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *S1*

`s2`

Description A parameter related to soil moisture dependence [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *S2*

`sathydraulicconduct`

Description Hydraulic conductivity for saturated soil [mm s^{-1}]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SatHydraulicCond*

`sddfull`

Description The sensesence degree days (SDD) needed to initiate leaf off. [°C]

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *SDDFull*

`sfr_surf`

Description Surface cover fractions.

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *Fr_Bldgs*, *Fr_Bsoil*, *Fr_DecTr*, *Fr_EveTr*, *Fr_Grass*, *Fr_Paved*, *Fr_Water* `smdmethod`

Description Determines method for calculating soil moisture deficit (SMD).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *SMDMethod*

`snowalb`

Description Initial snow albedo

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *SnowAlb0*

`snowalbmax`

Description Effective surface albedo (middle of the day value) for summertime.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *AlbedoMax*

`snowalbmin`

Description Effective surface albedo (middle of the day value) for wintertime (not including snow).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *AlbedoMin*

`snowdens`

Description Initial snow density of each land cover.

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SnowDensBldgs*, *SnowDensPaved*, *SnowDensDecTr*, *SnowDensEveTr*, *SnowDensGrass*, *SnowDensBSoil*, *SnowDensWater*

`snowdensmax`

Description Maximum snow density [kg m^{-3}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *SnowDensMax*

`snowdensmin`

Description Fresh snow density [kg m^{-3}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *SnowDensMin*

`snowfrac`

Description Initial plan area fraction of snow on each land cover`

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SnowFracBldgs*, *SnowFracPaved*, *SnowFracDecTr*, *SnowFracEveTr*, *SnowFracGrass*, *SnowFracBSoil*, *SnowFracWater*

`snowlimbldg`

Description Limit of the snow water equivalent for snow removal from roads and roofs [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *SnowLimRemove*

`snowlimpaved`

Description Limit of the snow water equivalent for snow removal from roads and roofs [mm]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *SnowLimRemove*

`snowpack`

Description Initial snow water equivalent on each land cover

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SnowPackBldgs*, *SnowPackPaved*, *SnowPackDecTr*, *SnowPackEveTr*, *SnowPackGrass*, *SnowPackBSoil*, *SnowPackWater*

`snowpacklimit`

Description Limit for the snow water equivalent when snow cover starts to be patchy [mm]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SnowLimPatch*

`snowprof_24hr`

Description Hourly profile values used in snow clearing.

Dimensionality (24, 2)

Dimensionality Remarks 24: hours of a day

2: {Weekday, Weekend}

SUEWS-related variables *SnowClearingProfWD*, *SnowClearingProfWE*

snowuse

Description Determines whether the snow part of the model runs.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *SnowUse*

snowwater

Description Initial amount of liquid water in the snow on each land cover

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SnowWaterBldgsState*, *SnowWaterPavedState*, *SnowWaterDecTrState*, *SnowWaterEveTrState*, *SnowWaterGrassState*, *SnowWaterBSoilState*, *SnowWaterWaterState*

soildepth

Description Depth of soil beneath the surface [mm]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SoilDepth*

soilstore_surf

Description Initial water stored in soil beneath *Bldgs* surface [mm];;Initial water stored in soil beneath *Paved* surface [mm];;Initial water stored in soil beneath *DecTr* surface [mm];;Initial water stored in soil beneath *EveTr* surface [mm];;Initial water stored in soil beneath *Grass* surface [mm];;Initial water stored in soil beneath *BSoil* surface [mm]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SoilstoreBldgsState*, *SoilstorePavedState*, *SoilstoreDecTrState*, *SoilstoreEveTrState*, *SoilstoreGrassState*, *SoilstoreBSoilState*

soilstorecap_surf

Description Limit value for *SoilDepth* [mm]

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *SoilStoreCap*

stabilitymethod

Description Defines which atmospheric stability functions are used.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *StabilityMethod*

startdls

Description Start of the day light savings [DOY]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `StartDLS`

state_surf

Description Initial wetness condition on `Bldgs`;;Initial wetness condition on `Paved`;;Initial wetness condition on `DecTr`;;Initial wetness condition on `EveTr`;;Initial wetness condition on `Grass`;;Initial wetness condition on `BSoil`;;Initial wetness condition on `Water`

Dimensionality (7,)

Dimensionality Remarks 7: { `Paved`, `Bldgs`, `EveTr`, `DecTr`, `Grass`, `BSoil`, `Water`}

SUEWS-related variables `BldgsState`, `PavedState`, `DecTrState`, `EveTrState`, `GrassState`, `BSoilState`, `WaterState`

statelimit_surf

Description Upper limit to the surface state. [mm]

Dimensionality (7,)

Dimensionality Remarks 7: { `Paved`, `Bldgs`, `EveTr`, `DecTr`, `Grass`, `BSoil`, `Water`}

SUEWS-related variables `StateLimit`

storageheatmethod

Description Determines method for calculating storage heat flux QS.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `StorageHeatMethod`

storedrainprm

Description Coefficients used in drainage calculation.

Dimensionality (6, 7)

Dimensionality Remarks 6: { `StorageMin`, `DrainageEq`, `DrainageCoef1`, `DrainageCoef2`, `StorageMax`, current storage}

7: { `Paved`, `Bldgs`, `EveTr`, `DecTr`, `Grass`, `BSoil`, `Water`}

SUEWS-related variables `DrainageCoef1`, `DrainageCoef2`, `DrainageEq`, `StorageMax`, `StorageMin`

surfacearea

Description Area of the grid [ha].

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables `SurfaceArea` `sw_dn_direct_frac`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`tau_a`

Description Time constant for snow albedo aging in cold snow [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *tau_a*

`tau_f`

Description Time constant for snow albedo aging in melting snow [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *tau_f*

`tau_r`

Description Time constant for snow density ageing [-]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *tau_r*

`tempmeltfact`

Description Hourly temperature melt factor of snow [mm K^{-1} h^{-1}]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *TempMeltFactor*

`th`

Description Upper air temperature limit [°C]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *TH*

`theta_bioco2`

Description The convexity of the curve at light saturation.

Dimensionality (3,)

Dimensionality Remarks 3: { *EveTr*, *DecTr*, *Grass*}

SUEWS-related variables *theta*

timezone

Description Time zone [h] for site relative to UTC (east is positive). This should be set according to the times given in the meteorological forcing file(s).

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Timezone*

tl

Description Lower air temperature limit [°C]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *TL*

trafficrate

Description Traffic rate used for CO2 flux calculation.

Dimensionality (2,)

Dimensionality Remarks 2: {Weekday, Weekend}

SUEWS-related variables *TrafficRate_WD*, *TrafficRate_WE*

trafficunits

Description Units for the traffic rate for the study area. Not used in v2018a.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *TrafficUnits*

traffprof_24hr

Description Hourly profile values used in traffic activity calculation.

Dimensionality (24, 2)

Dimensionality Remarks 24: hours of a day

2: {Weekday, Weekend}

SUEWS-related variables *TraffProfWD*, *TraffProfWE*

tstep

Description Specifies the model time step [s].

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *Tstep*

use_sw_direct_albedo

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`veg_contact_fraction_const`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`veg_fsd_const`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`veg_ssa_lw`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`veg_ssa_sw`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`veg_type`

Description Internal use. Please DO NOT modify

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables None

`waterdist`

Description Fraction of water redistribution

Dimensionality (8, 6)

Dimensionality Remarks 8: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*, one extra land cover type (currently NOT used)}

6: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*}

SUEWS-related variables *ToBSoil*, *ToBldgs*, *ToDecTr*, *ToEveTr*, *ToGrass*, *ToPaved*, *ToRunoff*, *ToSoilStore*, *ToWater* `waterusemethod`

Description Defines how external water use is calculated.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *WaterUseMethod*

`wetthresh_surf`

Description Depth of water which determines whether evaporation occurs from a partially wet or completely wet surface [mm].

Dimensionality (7,)

Dimensionality Remarks 7: { *Paved*, *Bldgs*, *EveTr*, *DecTr*, *Grass*, *BSoil*, *Water*}

SUEWS-related variables *WetThreshold*

`wuprofa_24hr`

Description Hourly profile values used in automatic irrigation.

Dimensionality (24, 2)

Dimensionality Remarks 24: hours of a day

2: {Weekday, Weekend}

SUEWS-related variables *WaterUseProfAutoWD*, *WaterUseProfAutoWE*

`wuprofm_24hr`

Description Hourly profile values used in manual irrigation.

Dimensionality (24, 2)

Dimensionality Remarks 24: hours of a day

2: {Weekday, Weekend}

SUEWS-related variables *WaterUseProfManuWD*, *WaterUseProfManuWE*

`z`

Description Measurement height [m] for all atmospheric forcing variables set in *SSss_YYYY_data_tt.txt*.

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *z*

`z0m_in`

Description Roughness length for momentum [m]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *z0*

zdm_in

Description Zero-plane displacement [m]

Dimensionality 0

Dimensionality Remarks Scalar

SUEWS-related variables *zd*

df_forcing variables

Note: Data structure of df_forcing is explained here.

RH

Description Relative Humidity [%] (measurement height (*z*) is needed in *SUEWS_SiteSelect.txt*)

Tair

Description Air temperature [°C] (measurement height (*z*) is needed in *SUEWS_SiteSelect.txt*)

U

Description Wind speed [m s-1] (measurement height (*z*) is needed in *SUEWS_SiteSelect.txt*)

Wuh

Description External water use [m^3]

fcld

Description Cloud fraction [tenths]

id

Description Day of year [DOY]

imin

Description Minute [M]

isec

Description Second [S]

it

Description Hour [H]

iy

Description Year [YYYY]

kdiff

Description Diffuse radiation [W m^{-2}] **Recommended in this version.** if *SOLWEIGUse* = 1

kdir

Description Direct radiation [W m^{-2}] **Recommended in this version.** if *SOLWEIGUse* = 1

kdown

Description Incoming shortwave radiation [W m^{-2}] Must be > 0 W m^{-2}.

lai

Description Observed leaf area index [m^{-2} m^{-2}]

ldown

Description Incoming longwave radiation [W m^{-2}]

pres

Description Barometric pressure [kPa] (measurement height (*z*) is needed in *SUEWS_SiteSelect.txt*)

qe

Description Latent heat flux [W m^{-2}]

qf

Description Anthropogenic heat flux [W m^{-2}]

qh

Description Sensible heat flux [W m^{-2}]

qn

Description Net all-wave radiation [W m^{-2}] (Required if `NetRadiationMethod` = 0.)

qs

Description Storage heat flux [W m^{-2}]

rain

Description Rainfall [mm] (measurement height (*z*) is needed in *SUEWS_SiteSelect.txt*)

snow

Description Snow cover fraction (0 – 1) [-] (Required if `SnowUse` = 1)

wdir

Description Wind direction [°] **Not available in this version.**

xsmd

Description Observed soil moisture [m^3 m^{-3}] or [kg kg^{-1}]

`df_output` variables

Note: Data structure of `df_output` is explained here.

AddWater

Description Additional water flow received from other grids [mm]

Group SUEWS

`AlbBulk`

Description Bulk albedo [-]

Group SUEWS

`AlbDecTr`

Description Albedo of deciduous trees [-]

Group DailyState

`AlbEveTr`

Description Albedo of evergreen trees [-]

Group DailyState

`AlbGrass`

Description Albedo of grass [-]

Group DailyState

`AlbSnow`

Description Snow albedo [-]

Group DailyState

`AlbSnow`

Description Snow albedo [-]

Group SUEWS

`Azimuth`

Description Solar azimuth angle [°]

Group SUEWS

`CI`

Description clearness index for Ldown (Lindberg et al. 2008)

Group BEERS

`DLHrs`

Description Day length [h]

Group DailyState

`DaysSR`

Description Days since rain [days]

Group DailyState

`DecidCap`

Description Moisture storage capacity of deciduous trees [mm]

Group DailyState

DensSnow_BSoil

Description Snow density – bare soil surface [kg m^{-3}]

Group DailyState

DensSnow_BSoil

Description Snow density - bare soil surface [kg m^{-3}]

Group DailyState

DensSnow_BSoil

Description Snow density – bare soil surface [kg m^{-3}]

Group snow

DensSnow_BSoil

Description Snow density - bare soil surface [kg m^{-3}]

Group snow

DensSnow_Bldgs

Description Snow density – building surface [kg m^{-3}]

Group snow

DensSnow_Bldgs

Description Snow density - building surface [kg m^{-3}]

Group DailyState

DensSnow_Bldgs

Description Snow density - building surface [kg m^{-3}]

Group snow

DensSnow_Bldgs

Description Snow density – building surface [kg m^{-3}]

Group DailyState

DensSnow_DecTr

Description Snow density – deciduous surface [kg m^{-3}]

Group snow

DensSnow_DecTr

Description Snow density - deciduous surface [kg m^{-3}]

Group DailyState

DensSnow_DecTr

Description Snow density – deciduous surface [kg m^{-3}]

Group DailyState

`DensSnow_DecTr`

Description Snow density - deciduous surface [kg m^{-3}]

Group snow

`DensSnow_EveTr`

Description Snow density - evergreen surface [kg m^{-3}]

Group snow

`DensSnow_EveTr`

Description Snow density – evergreen surface [kg m^{-3}]

Group snow

`DensSnow_EveTr`

Description Snow density - evergreen surface [kg m^{-3}]

Group DailyState

`DensSnow_EveTr`

Description Snow density – evergreen surface [kg m^{-3}]

Group DailyState

`DensSnow_Grass`

Description Snow density - grass surface [kg m^{-3}]

Group DailyState

`DensSnow_Grass`

Description Snow density – grass surface [kg m^{-3}]

Group DailyState

`DensSnow_Grass`

Description Snow density - grass surface [kg m^{-3}]

Group snow

`DensSnow_Grass`

Description Snow density – grass surface [kg m^{-3}]

Group snow

`DensSnow_Paved`

Description Snow density - paved surface [kg m^{-3}]

Group DailyState

`DensSnow_Paved`

Description Snow density – paved surface [kg m^{-3}]

Group snow

`DensSnow_Paved`

Description Snow density - paved surface [kg m^{-3}]

Group snow

`DensSnow_Paved`

Description Snow density – paved surface [kg m^{-3}]

Group DailyState

`DensSnow_Water`

Description Snow density - water surface [kg m^{-3}]

Group snow

`DensSnow_Water`

Description Snow density – water surface [kg m^{-3}]

Group DailyState

`DensSnow_Water`

Description Snow density - water surface [kg m^{-3}]

Group DailyState

`DensSnow_Water`

Description Snow density – water surface [kg m^{-3}]

Group snow

`DiffuseRad`

Description Diffuse shortwave radiation

Group BEERS

`DirectRad`

Description Direct shortwave radiation

Group BEERS

`Drainage`

Description Drainage [mm]

Group SUEWS

`Evap`

Description Evaporation [mm]

Group SUEWS

`Fc`

Description CO2 flux [umol m^{-2} s^{-1}]

Group SUEWS

`FcBuild`

Description CO2 flux from buildings [umol m^{-2} s^{-1}]

Group SUEWS

`FcMetab`

Description CO2 flux from metabolism [umol m^{-2} s^{-1}]

Group SUEWS

`FcPhoto`

Description CO2 flux from photosynthesis [umol m^{-2} s^{-1}]

Group SUEWS

`FcPoint`

Description CO2 flux from point source [umol m^{-2} s^{-1}]

Group SUEWS

`FcRespi`

Description CO2 flux from respiration [umol m^{-2} s^{-1}]

Group SUEWS

`FcTraff`

Description CO2 flux from traffic [umol m^{-2} s^{-1}]

Group SUEWS

`Fcld`

Description Cloud fraction [-]

Group SUEWS

`FlowCh`

Description Additional flow into water body [mm]

Group SUEWS

`GDD_DecTr`

Description Growing degree days for deciduous tree [°C d]

Group DailyState

`GDD_EveTr`

Description Growing degree days for evergreen tree [°C d]

Group DailyState

`GDD_Grass`

Description Growing degree days for grass [°C d]

Group DailyState

`GlobalRad`

Description Input Kdn

Group BEERS

`HDD1_h`

Description Heating degree days [°C d]

Group DailyState

`HDD2_c`

Description Cooling degree days [°C d]

Group DailyState

`HDD3_Tmean`

Description Average daily air temperature in forcing data [°C]

Group DailyState

`HDD4_T5d`

Description 5-day running-mean air temperature in forcing data [°C]

Group DailyState

`I0`

Description theoretical value of maximum incoming solar radiation

Group BEERS

`Irr`

Description Irrigation [mm]

Group SUEWS

`Kdown`

Description Incoming shortwave radiation [W m^{-2}]

Group SUEWS

`Kdown2d`

Description Incoming shortwave radiation at POI

Group BEERS

`Keast`

Description Shortwave radiation from east at POI

Group BEERS

`Knorth`

Description Shortwave radiation from north at POI

Group BEERS

`Ksouth`

Description Shortwave radiation from south at POI

Group BEERS

`Kup`

Description Outgoing shortwave radiation [W m^{-2}]

Group SUEWS

`Kup`

Description Outgoing shortwave radiation [W m^{-2}]

Group SPARTACUS

`Kup2d`

Description Outgoing shortwave radiation at POI

Group BEERS

`Kwest`

Description Shortwave radiation from west at POI

Group BEERS

`LAI`

Description Leaf area index [m 2 m^{-2}]

Group SUEWS

`LAI_DecTr`

Description Leaf area index of deciduous trees [m^{-2} m^{-2}]

Group DailyState

`LAI_EveTr`

Description Leaf area index of evergreen trees [m^{-2} m^{-2}]

Group DailyState

`LAI_Grass`

Description Leaf area index of grass [m^{-2} m^{-2}]

Group DailyState

`LAIlumps`

Description Leaf area index used in LUMPS (normalised 0-1) [-]

Group DailyState

`Ldown`

Description Incoming longwave radiation [W m^{-2}]

Group SUEWS

`Ldown2d`

Description Incoming longwave radiation at POI

Group BEERS

`Least`

Description Longwave radiation from east at POI

Group BEERS

`Lnorth`

Description Longwave radiation from north at POI

Group BEERS

`Lob`

Description Obukhov length [m]

Group SUEWS

`Lsouth`

Description Longwave radiation from south at POI

Group BEERS

`Lup`

Description Outgoing longwave radiation [W m^{-2}]

Group SUEWS

`Lup`

Description Outgoing longwave radiation [W m^{-2}]

Group SPARTACUS

`Lup2d`

Description Outgoing longwave radiation at POI

Group BEERS

`Lwest`

Description Longwave radiation from west at POI

Group BEERS

`MeltWStore`

Description Meltwater store [mm]

Group SUEWS

`MeltWater`

Description Meltwater [mm]

Group SUEWS

`MwStore_BSoil`

Description Melt water store – bare soil surface [mm]

Group snow

`MwStore_Bldgs`

Description Melt water store – building surface [mm]

Group snow

`MwStore_DecTr`

Description Melt water store – deciduous surface [mm]

Group snow

`MwStore_EveTr`

Description Melt water store – evergreen surface [mm]

Group snow

`MwStore_Grass`

Description Melt water store – grass surface [mm]

Group snow

`MwStore_Paved`

Description Melt water store – paved surface [mm]

Group snow

`MwStore_Water`

Description Melt water store – water surface [mm]

Group snow

`Mw_BSoil`

Description Meltwater – bare soil surface [mm h^{-1}]

Group snow

`Mw_Bldgs`

Description Meltwater – building surface [mm h^{-1}]

Group snow

`Mw_DecTr`

Description Meltwater – deciduous surface [mm h^{-1}]

Group snow

`Mw_EveTr`

Description Meltwater – evergreen surface [mm h^{-1}]

Group snow

`Mw_Grass`

Description Meltwater – grass surface [mm h^{-1} 1]

Group snow

`Mw_Paved`

Description Meltwater – paved surface [mm h^{-1}]

Group snow

`Mw_Water`

Description Meltwater – water surface [mm h^{-1}]

Group snow

`NWtrState`

Description Surface wetness state (for non-water surfaces) [mm]

Group SUEWS

`P_day`

Description Daily total precipitation [mm]

Group DailyState

`Porosity`

Description Porosity of deciduous trees [-]

Group DailyState

`Q2`

Description Air specific humidity at 2 m agl [g kg^{-1}]

Group SUEWS

`QE`

Description Latent heat flux (calculated using SUEWS) [W m^{-2}]

Group SUEWS

`QElumps`

Description Latent heat flux (calculated using LUMPS) [W m^{-2}]

Group SUEWS

`QF`

Description Anthropogenic heat flux [W m^{-2}]

Group SUEWS

`QH`

Description Sensible heat flux (calculated using SUEWS) [W m^{-2}]

Group SUEWS

`QHlumps`

Description Sensible heat flux (calculated using LUMPS) [W m^{-2}]

Group SUEWS

`QHresis`

Description Sensible heat flux (calculated using resistance method) [W m^{-2}]

Group SUEWS

`QM`

Description Snow-related heat exchange [W m^{-2}]

Group SUEWS

`QMFreeze`

Description Internal energy change [W m^{-2}]

Group SUEWS

`QMRain`

Description Heat released by rain on snow [W m^{-2}]

Group SUEWS

`QN`

Description Net all-wave radiation [W m^{-2}]

Group SUEWS

`QNSnow`

Description Net all-wave radiation for snow area [W m^{-2}]

Group SUEWS

`QNSnowFr`

Description Net all-wave radiation for snow-free area [W m^{-2}]

Group SUEWS

`QS`

Description Storage heat flux [W m^{-2}]

Group SUEWS

`Qa_BSoil`

Description Advective heat – bare soil surface [W m^{-2}]

Group snow

`Qa_Bldgs`

Description Advective heat – building surface [W m^{-2}]

Group snow

Qa_DecTr

Description Advective heat – deciduous surface [W m^{-2}]

Group snow

Qa_EveTr

Description Advective heat – evergreen surface [W m^{-2}]

Group snow

Qa_Grass

Description Advective heat – grass surface [W m^{-2}]

Group snow

Qa_Paved

Description Advective heat – paved surface [W m^{-2}]

Group snow

Qa_Water

Description Advective heat – water surface [W m^{-2}]

Group snow

QmFr_BSoil

Description Heat related to freezing of surface store – bare soil surface [W m^{-2}]

Group snow

QmFr_Bldgs

Description Heat related to freezing of surface store – building surface [W m^{-2}]

Group snow

QmFr_DecTr

Description Heat related to freezing of surface store – deciduous surface [W m^{-2}]

Group snow

QmFr_EveTr

Description Heat related to freezing of surface store – evergreen surface [W m^{-2}]

Group snow

QmFr_Grass

Description Heat related to freezing of surface store – grass surface [W m^{-2}]

Group snow

QmFr_Paved

Description Heat related to freezing of surface store – paved surface [W m^{-2}]

Group snow

`QmFr_Water`

Description Heat related to freezing of surface store – water [W m^{-2}]

Group snow

`Qm_BSoil`

Description Snowmelt-related heat – bare soil surface [W m^{-2}]

Group snow

`Qm_Bldgs`

Description Snowmelt-related heat – building surface [W m^{-2}]

Group snow

`Qm_DecTr`

Description Snowmelt-related heat – deciduous surface [W m^{-2}]

Group snow

`Qm_EveTr`

Description Snowmelt-related heat – evergreen surface [W m^{-2}]

Group snow

`Qm_Grass`

Description Snowmelt-related heat – grass surface [W m^{-2}]

Group snow

`Qm_Paved`

Description Snowmelt-related heat – paved surface [W m^{-2}]

Group snow

`Qm_Water`

Description Snowmelt-related heat – water surface [W m^{-2}]

Group snow

`RA`

Description Aerodynamic resistance [s m^{-1}]

Group SUEWS

`RA`

Description Aerodynamic resistance [s m^{-1}]

Group debug

`RH2`

Description Relative humidity at 2 m agl [%]

Group SUEWS

`RO`

Description Runoff [mm]

Group SUEWS

`ROImp`

Description Above ground runoff over impervious surfaces [mm]

Group SUEWS

`ROPipe`

Description Runoff to pipes [mm]

Group SUEWS

`ROSoil`

Description Runoff to soil (sub-surface) [mm]

Group SUEWS

`ROVeg`

Description Above ground runoff over vegetated surfaces [mm]

Group SUEWS

`ROWater`

Description Runoff for water body [mm]

Group SUEWS

`RS`

Description Surface resistance [s m^{-1}]

Group SUEWS

`RS`

Description Surface resistance [s m^{-1}]

Group debug

`Rain`

Description Rain [mm]

Group SUEWS

`RainSn_BSoil`

Description Rain on snow – bare soil surface [mm]

Group snow

`RainSn_Bldgs`

Description Rain on snow – building surface [mm]

Group snow

`RainSn_DecTr`

Description Rain on snow – deciduous surface [mm]

Group snow

`RainSn_EveTr`

Description Rain on snow – evergreen surface [mm]

Group snow

`RainSn_Grass`

Description Rain on snow – grass surface [mm]

Group snow

`RainSn_Paved`

Description Rain on snow – paved surface [mm]

Group snow

`RainSn_Water`

Description Rain on snow – water surface [mm]

Group snow

`SDD_DecTr`

Description Senescence degree days for deciduous tree [°C d]

Group DailyState

`SDD_EveTr`

Description Senescence degree days for evergreen tree [°C d]

Group DailyState

`SDD_Grass`

Description Senescence degree days for grass [°C d]

Group DailyState

`SMD`

Description Soil moisture deficit [mm]

Group SUEWS

`SMDBSoil`

Description Soil moisture deficit for bare soil surface [mm]

Group SUEWS

`SMDBldgs`

Description Soil moisture deficit for building surface [mm]

Group SUEWS

`SMDDecTr`

Description Soil moisture deficit for deciduous surface [mm]

Group SUEWS

`SMDEveTr`

Description Soil moisture deficit for evergreen surface [mm]

Group SUEWS

`SMDGrass`

Description Soil moisture deficit for grass surface [mm]

Group SUEWS

`SMDPaved`

Description Soil moisture deficit for paved surface [mm]

Group SUEWS

`SWE`

Description Snow water equivalent [mm]

Group SUEWS

`SWE_BSoil`

Description Snow water equivalent – bare soil surface [mm]

Group snow

`SWE_Bldgs`

Description Snow water equivalent – building surface [mm]

Group snow

`SWE_DecTr`

Description Snow water equivalent – deciduous surface [mm]

Group snow

`SWE_EveTr`

Description Snow water equivalent – evergreen surface [mm]

Group snow

`SWE_Grass`

Description Snow water equivalent – grass surface [mm]

Group snow

`SWE_Paved`

Description Snow water equivalent – paved surface [mm]

Group snow

`SWE_Water`

Description Snow water equivalent – water surface [mm]

Group snow

`Sd_BSoil`

Description Snow depth – bare soil surface [mm]

Group snow

`Sd_Bldgs`

Description Snow depth – building surface [mm]

Group snow

`Sd_DecTr`

Description Snow depth – deciduous surface [mm]

Group snow

`Sd_EveTr`

Description Snow depth – evergreen surface [mm]

Group snow

`Sd_Grass`

Description Snow depth – grass surface [mm]

Group snow

`Sd_Paved`

Description Snow depth – paved surface [mm]

Group snow

`Sd_Water`

Description Snow depth – water surface [mm]

Group snow

`SnowCh`

Description Change in snow pack [mm]

Group SUEWS

`SnowRBldgs`

Description Snow removed from building surface [mm]

Group SUEWS

`SnowRPaved`

Description Snow removed from paved surface [mm]

Group SUEWS

`StBSoil`

Description Surface wetness state for bare soil surface [mm]

Group SUEWS

`StBldgs`

Description Surface wetness state for building surface [mm]

Group SUEWS

`StDecTr`

Description Surface wetness state for deciduous tree surface [mm]

Group SUEWS

`StEveTr`

Description Surface wetness state for evergreen tree surface [mm]

Group SUEWS

`StGrass`

Description Surface wetness state for grass surface [mm]

Group SUEWS

`StPaved`

Description Surface wetness state for paved surface [mm]

Group SUEWS

`StWater`

Description Surface wetness state for water surface [mm]

Group SUEWS

`State`

Description Surface wetness state [mm]

Group SUEWS

`SurfCh`

Description Change in surface moisture store [mm]

Group SUEWS

`T2`

Description Air temperature at 2 m agl [°C]

Group SUEWS

`T_1`

Description Air temperature at level 1 [°C]

Group RSL

`T_10`

Description Air temperature at level 10 [°C]

Group RSL

`T_11`

Description Air temperature at level 11 [°C]

Group RSL

`T_12`

Description Air temperature at level 12 [°C]

Group RSL

`T_13`

Description Air temperature at level 13 [°C]

Group RSL

`T_14`

Description Air temperature at level 14 [°C]

Group RSL

`T_15`

Description Air temperature at level 15 [°C]

Group RSL

`T_16`

Description Air temperature at level 16 [°C]

Group RSL

`T_17`

Description Air temperature at level 17 [°C]

Group RSL

`T_18`

Description Air temperature at level 18 [°C]

Group RSL

`T_19`

Description Air temperature at level 19 [°C]

Group RSL

`T_2`

Description Air temperature at level 2 [°C]

Group RSL

`T_20`

Description Air temperature at level 20 [°C]

Group RSL

`T_21`

Description Air temperature at level 21 [°C]

Group RSL

`T_22`

Description Air temperature at level 22 [°C]

Group RSL

`T_23`

Description Air temperature at level 23 [°C]

Group RSL

`T_24`

Description Air temperature at level 24 [°C]

Group RSL

`T_25`

Description Air temperature at level 25 [°C]

Group RSL

`T_26`

Description Air temperature at level 26 [°C]

Group RSL

`T_27`

Description Air temperature at level 27 [°C]

Group RSL

`T_28`

Description Air temperature at level 28 [°C]

Group RSL

`T_29`

Description Air temperature at level 29 [°C]

Group RSL

`T_3`

Description Air temperature at level 3 [°C]

Group RSL

T_30

Description Air temperature at level 30 [°C]

Group RSL

T_4

Description Air temperature at level 4 [°C]

Group RSL

T_5

Description Air temperature at level 5 [°C]

Group RSL

T_6

Description Air temperature at level 6 [°C]

Group RSL

T_7

Description Air temperature at level 7 [°C]

Group RSL

T_8

Description Air temperature at level 8 [°C]

Group RSL

T_9

Description Air temperature at level 9 [°C]

Group RSL

Ta

Description Air temperature

Group BEERS

Tg

Description Surface temperature

Group BEERS

Tmax

Description Daily maximum temperature in forcing data [°C]

Group DailyState

Tmin

Description Daily minimum temperature in forcing data [°C]

Group DailyState

`Tmrt`

Description Mean Radiant Temperature

Group BEERS

`TotCh`

Description Change in surface and soil moisture stores [mm]

Group SUEWS

`Ts`

Description Skin temperature [°C]

Group SUEWS

`Tsnow_BSoil`

Description Snow surface temperature – bare soil surface [°C]

Group snow

`Tsnow_Bldgs`

Description Snow surface temperature – building surface [°C]

Group snow

`Tsnow_DecTr`

Description Snow surface temperature – deciduous surface [°C]

Group snow

`Tsnow_EveTr`

Description Snow surface temperature – evergreen surface [°C]

Group snow

`Tsnow_Grass`

Description Snow surface temperature – grass surface [°C]

Group snow

`Tsnow_Paved`

Description Snow surface temperature – paved surface [°C]

Group snow

`Tsnow_Water`

Description Snow surface temperature – water surface [°C]

Group snow

`Tsurf`

Description Bulk surface temperature [°C]

Group SUEWS

U10

Description Wind speed at 10 m agl [m s^{-1}]

Group SUEWS

U_1

Description Wind speed at level 1 [m s^{-1}]

Group RSL

U_10

Description Wind speed at level 10 [m s^{-1}]

Group RSL

U_11

Description Wind speed at level 11 [m s^{-1}]

Group RSL

U_12

Description Wind speed at level 12 [m s^{-1}]

Group RSL

U_13

Description Wind speed at level 13 [m s^{-1}]

Group RSL

U_14

Description Wind speed at level 14 [m s^{-1}]

Group RSL

U_15

Description Wind speed at level 15 [m s^{-1}]

Group RSL

U_16

Description Wind speed at level 16 [m s^{-1}]

Group RSL

U_17

Description Wind speed at level 17 [m s^{-1}]

Group RSL

U_18

Description Wind speed at level 18 [m s^{-1}]

Group RSL

`U_19`

Description Wind speed at level 19 [m s^{-1}]

Group RSL

`U_2`

Description Wind speed at level 2 [m s^{-1}]

Group RSL

`U_20`

Description Wind speed at level 20 [m s^{-1}]

Group RSL

`U_21`

Description Wind speed at level 21 [m s^{-1}]

Group RSL

`U_22`

Description Wind speed at level 22 [m s^{-1}]

Group RSL

`U_23`

Description Wind speed at level 23 [m s^{-1}]

Group RSL

`U_24`

Description Wind speed at level 24 [m s^{-1}]

Group RSL

`U_25`

Description Wind speed at level 25 [m s^{-1}]

Group RSL

`U_26`

Description Wind speed at level 26 [m s^{-1}]

Group RSL

`U_27`

Description Wind speed at level 27 [m s^{-1}]

Group RSL

`U_28`

Description Wind speed at level 28 [m s^{-1}]

Group RSL

`U_29`

Description Wind speed at level 29 [m s^{-1}]

Group RSL

`U_3`

Description Wind speed at level 3 [m s^{-1}]

Group RSL

`U_30`

Description Wind speed at level 30 [m s^{-1}]

Group RSL

`U_4`

Description Wind speed at level 4 [m s^{-1}]

Group RSL

`U_5`

Description Wind speed at level 5 [m s^{-1}]

Group RSL

`U_6`

Description Wind speed at level 6 [m s^{-1}]

Group RSL

`U_7`

Description Wind speed at level 7 [m s^{-1}]

Group RSL

`U_8`

Description Wind speed at level 8 [m s^{-1}]

Group RSL

`U_9`

Description Wind speed at level 9 [m s^{-1}]

Group RSL

`WUDecTr`

Description Water use for irrigation of deciduous trees [mm]

Group SUEWS

`WUEveTr`

Description Water use for irrigation of evergreen trees [mm]

Group SUEWS

`WUGrass`

Description Water use for irrigation of grass [mm]

Group SUEWS

`WUInt`

Description Internal water use [mm]

Group SUEWS

`WU_DecTr1`

Description Total water use for deciduous trees [mm]

Group DailyState

`WU_DecTr2`

Description Automatic water use for deciduous trees [mm]

Group DailyState

`WU_DecTr3`

Description Manual water use for deciduous trees [mm]

Group DailyState

`WU_EveTr1`

Description Total water use for evergreen trees [mm]

Group DailyState

`WU_EveTr2`

Description Automatic water use for evergreen trees [mm]

Group DailyState

`WU_EveTr3`

Description Manual water use for evergreen trees [mm]

Group DailyState

`WU_Grass1`

Description Total water use for grass [mm]

Group DailyState

`WU_Grass2`

Description Automatic water use for grass [mm]

Group DailyState

`WU_Grass3`

Description Manual water use for grass [mm]

Group DailyState

`Zenith`

Description Solar zenith angle [°]

Group SUEWS

`a1`

Description OHM cofficient a1 - [-]

Group DailyState

`a2`

Description OHM cofficient a2 [W m^{-2} h^{-1}]

Group DailyState

`a3`

Description OHM cofficient a3 - [W m^{-2}]

Group DailyState

`altitude`

Description Altitude angle of the Sun

Group BEERS

`azimuth`

Description Azimuth angle of the Sun

Group BEERS

`deltaLAI`

Description Change in leaf area index (normalised 0-1) [-]

Group DailyState

`frMelt_BSoil`

Description Amount of freezing melt water – bare soil surface [mm]

Group snow

`frMelt_Bldgs`

Description Amount of freezing melt water – building surface [mm]

Group snow

`frMelt_DecTr`

Description Amount of freezing melt water – deciduous surface [mm]

Group snow

`frMelt_EveTr`

Description Amount of freezing melt water – evergreen surface [mm]

Group snow

`frMelt_Grass`

Description Amount of freezing melt water – grass surface [mm]

Group snow

`frMelt_Paved`

Description Amount of freezing melt water – paved surface [mm]

Group snow

`frMelt_Water`

Description Amount of freezing melt water – water surface [mm]

Group snow

`fr_Bldgs`

Description Fraction of snow – building surface [-]

Group snow

`fr_DecTr`

Description Fraction of snow – deciduous surface [-]

Group snow

`fr_EveTr`

Description Fraction of snow – evergreen surface [-]

Group snow

`fr_Grass`

Description Fraction of snow – grass surface [-]

Group snow

`fr_Paved`

Description Fraction of snow – paved surface [-]

Group snow

`kup_BSoilSnow`

Description Reflected shortwave radiation – bare soil surface [W m^{-2}]

Group snow

`kup_BldgsSnow`

Description Reflected shortwave radiation – building surface [W m^{-2}]

Group snow

`kup_DecTrSnow`

Description Reflected shortwave radiation – deciduous surface [W m^{-2}]

Group snow

`kup_EveTrSnow`

Description Reflected shortwave radiation – evergreen surface [W m^{-2}]

Group snow

`kup_GrassSnow`

Description Reflected shortwave radiation – grass surface [W m^{-2}]

Group snow

`kup_PavedSnow`

Description Reflected shortwave radiation – paved surface [W m^{-2}]

Group snow

`kup_WaterSnow`

Description Reflected shortwave radiation – water surface [W m^{-2}]

Group snow

`q_1`

Description Specific humidity at level 1 [g kg^{-1}]

Group RSL

`q_10`

Description Specific humidity at level 10 [g kg^{-1}]

Group RSL

`q_11`

Description Specific humidity at level 11 [g kg^{-1}]

Group RSL

`q_12`

Description Specific humidity at level 12 [g kg^{-1}]

Group RSL

`q_13`

Description Specific humidity at level 13 [g kg^{-1}]

Group RSL

`q_14`

Description Specific humidity at level 14 [g kg^{-1}]

Group RSL

`q_15`

Description Specific humidity at level 15 [g kg^{-1}]

Group RSL

q_16

Description Specific humidity at level 16 [g kg^{-1}]

Group RSL

q_17

Description Specific humidity at level 17 [g kg^{-1}]

Group RSL

q_18

Description Specific humidity at level 18 [g kg^{-1}]

Group RSL

q_19

Description Specific humidity at level 19 [g kg^{-1}]

Group RSL

q_2

Description Specific humidity at level 2 [g kg^{-1}]

Group RSL

q_20

Description Specific humidity at level 20 [g kg^{-1}]

Group RSL

q_21

Description Specific humidity at level 21 [g kg^{-1}]

Group RSL

q_22

Description Specific humidity at level 22 [g kg^{-1}]

Group RSL

q_23

Description Specific humidity at level 23 [g kg^{-1}]

Group RSL

q_24

Description Specific humidity at level 24 [g kg^{-1}]

Group RSL

q_25

Description Specific humidity at level 25 [g kg^{-1}]

Group RSL

q_26

Description Specific humidity at level 26 [g kg^{-1}]

Group RSL

q_27

Description Specific humidity at level 27 [g kg^{-1}]

Group RSL

q_28

Description Specific humidity at level 28 [g kg^{-1}]

Group RSL

q_29

Description Specific humidity at level 29 [g kg^{-1}]

Group RSL

q_3

Description Specific humidity at level 3 [g kg^{-1}]

Group RSL

q_30

Description Specific humidity at level 30 [g kg^{-1}]

Group RSL

q_4

Description Specific humidity at level 4 [g kg^{-1}]

Group RSL

q_5

Description Specific humidity at level 5 [g kg^{-1}]

Group RSL

q_6

Description Specific humidity at level 6 [g kg^{-1}]

Group RSL

q_7

Description Specific humidity at level 7 [g kg^{-1}]

Group RSL

q_8

Description Specific humidity at level 8 [g kg^{-1}]

Group RSL

`q_9`

Description Specific humidity at level 9 [g kg^{-1}]

Group RSL

`z0m`

Description Roughness length for momentum [m]

Group SUEWS

`z_1`

Description Height at level 1 [m]

Group RSL

`z_10`

Description Height at level 10 [m]

Group RSL

`z_11`

Description Height at level 11 [m]

Group RSL

`z_12`

Description Height at level 12 [m]

Group RSL

`z_13`

Description Height at level 13 [m]

Group RSL

`z_14`

Description Height at level 14 [m]

Group RSL

`z_15`

Description Height at level 15 [m]

Group RSL

`z_16`

Description Height at level 16 [m]

Group RSL

`z_17`

Description Height at level 17 [m]

Group RSL

`z_18`

Description Height at level 18 [m]

Group RSL

`z_19`

Description Height at level 19 [m]

Group RSL

`z_2`

Description Height at level 2 [m]

Group RSL

`z_20`

Description Height at level 20 [m]

Group RSL

`z_21`

Description Height at level 21 [m]

Group RSL

`z_22`

Description Height at level 22 [m]

Group RSL

`z_23`

Description Height at level 23 [m]

Group RSL

`z_24`

Description Height at level 24 [m]

Group RSL

`z_25`

Description Height at level 25 [m]

Group RSL

`z_26`

Description Height at level 26 [m]

Group RSL

`z_27`

Description Height at level 27 [m]

Group RSL

`z_28`

Description Height at level 28 [m]

Group RSL

`z_29`

Description Height at level 29 [m]

Group RSL

`z_3`

Description Height at level 3 [m]

Group RSL

`z_30`

Description Height at level 30 [m]

Group RSL

`z_4`

Description Height at level 4 [m]

Group RSL

`z_5`

Description Height at level 5 [m]

Group RSL

`z_6`

Description Height at level 6 [m]

Group RSL

`z_7`

Description Height at level 7 [m]

Group RSL

`z_8`

Description Height at level 8 [m]

Group RSL

`z_9`

Description Height at level 9 [m]

Group RSL

`zdm`

Description Zero-plane displacement height [m]

Group SUEWS

FAQ

Contents

I cannot install SuPy following the docs, what is wrong there?

please check if your environment meets the following requirements:

1. Operating system (OS):

   a. is it 64 bit? only 64 bit systems are supported.

   b. is your OS up to date? only recent desktop systems are supported:

      - Windows 10 and above
      - macOS 10.13 and above
      - Linux: no restriction; If SuPy cannot run on your specific Linux distribution, please report it to us.

   You can get the OS information with the following code:

```
import platform
platform.platform()
```

2. Python interpreter:

   a. is your Python interpreter 64 bit?

   Check running mode with the following code:

```
import struct
struct.calcsize('P')*8
```

   b. is your Python version above 3.5?

   Check version info with the following code:

```
import sys
sys.version
```

If your environment doesn't meet the requirement by SuPy, please use a proper environment; otherwise, **`please report your issue`__**.

How do I know which version of SuPy I am using?

Use the following code:

```
import supy
supy.show_version()
```

Note: *show_version* is only available after v2019.5.28.

A `kernel may have died` exception happened, where did I go wrong?

The issue is highly likely due to invalid input to SuPy and SUEWS kernel. We are trying to avoid such exceptions, but unfortunately they might happen in some edge cases.

Please report such issues to us with your input files for debugging. Thanks!

How can I upgrade SuPy to an up-to-date version?

Run the following code in your terminal:

```
python3 -m pip install supy --upgrade
```

How to deal with `KeyError` when trying to load initial model states?

Please see **:issue:`28`**

Version History

Version 2022.9.22

- **New**

  Added experimental support SPARTACUS module.

- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  None.

- **Known issue**

  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2021.11.22

- **New**

  None.

- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  1. Fixed an issue in incorrect pressure unit in *gen_forcing_era5* for `pressure` mode. (Thansk to @XiaoxiongXie for fixing via **:PR:`#39`**)

- **Known issue**

  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2021.11.20

- **New**

  1. Added option `pressure_level` in *gen_forcing_era5*.

- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  1. Fixed an issue in generating ERA5 forcing due to xarray update in `merge`.

- **Known issue**

  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2021.7.22

- **New**

  None.

- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  1. Fixed an issue in loading parameter table caused by recent update of `pandas` to `1.3.x`.
  2. Fixed an issue in ERA5 download due to renaming of `orography` to `geopotential`.

- **Known issue**

  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2021.5.26

- **New**

  1. Update supy-driver to `2021a` iteration.

- **Improvement**

  1. a new method for calculating roughness length for momentum and displacement height (`roughlenmommethod=4`) based on plan area index as illustrated in figure 1a of GO99.

- **Changes**

  None.

- **Fix**

  None.

- **Known issue**

  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2020.11.3

- **New**
  1. Update supy-driver to `2020b` iteration.
  2. Add function for plotting RSL variables *supy.util.plot_rsl*.
- **Improvement**
  1. The RSL related functions are more robust in dealing with broader urban morphology settings.
  2. Internal changes to conform with recent upgrades in `pandas`.
- **Changes**

  None.
- **Fix**

  1. Fix an issue in `supy.util.read_forcing` that improper resampling could be conducted if input temporal resolution is the same as the desirable resampling time step `tstep_mod`.
- **Known issue**
  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2020.5.29

- **New**
  1. Update supy-driver to `2020a` iteration.
  2. Add function for plotting RSL variables *supy.util.plot_rsl*.
- **Improvement**

  None.
- **Changes**

  None.
- **Fix**
  1. Fix the humidity variable in ERA5-based forcing generation.
  2. Fix the impact study tutorial.
- **Known issue**
  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2020.2.2

- **New**
  1. A checker to validate input `DataFrame`s. See option `check_input` in *run_supy*.
  2. Utilities to generate forcing data using ERA-5 data. See *download_era5* and *gen_forcing_era5*.
- **Improvement**
  1. Improved performance of the parallel mode.
- **Changes**

  None.
- **Fix**

  None.
- **Known issue**
  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Simulation in parallel mode is NOT supported on Windows due to system limitation.

Version 2019.8.29

- **New**
  1. added WRF-SUEWS related functions.
  2. added diagnostics of canyon profiles.
- **Improvement**

  None.
- **Changes**
  1. synchronised with v2019a interface: minimum supy_driver v2019a2.
- **Fix**

  None.
- **Known issue**
  1. ESTM is not supported yet.
  2. BLUEWS, a CBL modules in SUEWS, is not supported yet.
  3. Performance in parallel mode can be worse than serial mode sometimes due to heavy (de)-serialisation loads.

Version 2019.7.17

- **New**
  1. added OHM related functions.
  2. added surface conductance related functions.
- **Improvement**

  None.
- **Changes**

  None.
- **Fix**
  1. Fixed a bug in unit conversion for TMY data generation.
- **Known issue**

  ESTM is not supported yet.

Version 2019.6.8

- **New**

  None.
- **Improvement**

  None.
- **Changes**

  None.
- **Fix**
  1. Fixed a bug in rescaling Kdown when loading forcing data.
- **Known issue**

  ESTM is not supported yet.

Version 2019.5.28

Spring house cleaning with long-await command line tools (more on the way!).

- **New**
  1. Added version info function: `show_version`.
  2. Added command line tools:
  - *suews-run*: SuPy wrapper to mimic SUEWS-binary-based simulation.
  - *suews-convert*: convert input tables from older versions to newer ones (one-way only).
- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  1. Fixed a bug in writing out multi-grid output files caused by incorrect dropping of temporal information by pandas .

- **Known issue**

  ESTM is not supported yet.

Version 2019.4.29

Parallel run.

- **New**

  Added support for parallel run on the fly.

- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  None.

- **Known issue**

  None

Version 2019.4.17

UMEP compatibility tweaks.

- **New**

  None.

- **Improvement**

  None.

- **Changes**

  Error messages: problems.txt will be written out in addition to the console error message similarly as SUEWS binary.

- **Fix**

  Incorrect caching of input libraries.

- **Known issue**

  None

Version 2019.4.15

ERA-5 download.

- **New**

  Added experimental support for downloading and processing ERA-5 data to force supy simulations.

- **Improvement**

  Improved compatibility with earlier `pandas` version in resampling output.

- **Changes**

  None.

- **Fix**

  None.

- **Known issue**

  None

Version 2019.3.21

TMY generation.

- **New**

  Added preliminary support for generating TMY dataset with SuPy output.

- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  None.

- **Known issue**

  None

Version 2019.3.14

This release improved memory usage.

- **New**

  None.

- **Improvement**

  Optimised memory consumption for longterm simulations.

- **Changes**

  None.

- **Fix**

  None.

- **Known issue**

  None

Version 2019.2.25

This release dropped support for Python 3.5 and below.

- **New**

  None.

- **Improvement**

  None.

- **Changes**

  Dropped support for Python 3.5 and below.

- **Fix**

  None.

- **Known issue**

  None

Version 2019.2.24

This release added the ability to save output files.

- **New**

  1. Added support to save output files. See: *supy.save_supy()*
  2. Added support to initialise SuPy from saved `df_state.csv`. See: *supy.init_supy()*

- **Improvement**

  None.

- **Changes**

  None.

- **Fix**

  None.

- **Known issue**

  None

Version 2019.2.19

This is a release that improved the exception handling due to fatal error in `supy_driver`.

- **New**

  Added support to handle python kernel crash caused by fatal error in `supy_driver` kernel; so python kernel won't crash any more even `supy_driver` is stopped.

- **Improvement**

  None.

- **Changes**

  None

- **Fix**

  None.

- **Known issue**

  None

Version 2019.2.8

This is a release that fixes recent bugs found in SUEWS that may lead to abnormal simulation results of storage heat flux, in particular when *SnowUse* is enabled (i.e., `snowuse=1`).

- **New**

  None.

- **Improvement**

  Improved the performance in loading initial model state from a large number of grids (>1k)

- **Changes**

  Updated `SampleRun` dataset by: 1. setting surface fractions (`sfr`) to a more realistic value based on London KCL case; 2. enabling snow module (`snowuse=1`).

- **Fix**

  1. Fixed a bug in the calculation of storage heat flux.
  2. Fixed a bug in loading `popdens` for calculating anthropogenic heat flux.

- **Known issue**

  None

Version 2019.1.1 (preview release, 01 Jan 2019)

- **New**
  1. Slimmed the output groups by excluding unsupported *ESTM* results
  2. SuPy documentation
     - Key IO data structures documented:
     - *df_output variables* (**:issue:`9`**)
     - *df_state variables* (**:issue:`8`**)
     - *df_forcing variables* (**:issue:`7`**)
     - Tutorial of parallel SuPy simulations for impact studies
- **Improvement**
  1. Improved calculation of OHM-related radiation terms
- **Changes**

  None.
- **Fix**

  None
- **Known issue**

  None

Version 2018.12.15 (internal test release in December 2018) ————————————=====

- **New**
  1. Preview release of SuPy based on the computation kernel of SUEWS 2018b
- **Improvement**
  1. Improved calculation of OHM-related radiation terms
- **Changes**

  None.
- **Fix**

  None
- **Known issue**
  1. The heat storage modules AnOHM and ESTM are not supported yet.

7.6.2 SUEWS in UMEP

SUEWS can be run as a standalone model but also can be used within UMEP. There are numerous tools included within UMEP to help a user get started. The SUEWS (Simple) within UMEP is a fast way to start using SUEWS.

The version of SUEWS within UMEP is the complete model. Thus all options that are listed in this manual are available to the user. In the UMEP SUEWS (Simple) runs all options are set to values to allow initial exploration of the model behaviour.

- **Pre-Processor**
  - **Meteorological Data**
    - **Prepare Existing Data** Transforms meteorological data into UMEP format
    - **Download data (WATCH)** Prepare meteorological dataset from *WATCH*
  - **Spatial Data**
    - **Spatial Data Downloader** Plugin for retrieving geodata from online services suitable for various UMEP related tools - LCZ Converter Conversion from Local Climate Zones (LCZs) in the WUDAPT database into SUEWS input data
  - **Urban land cover**
    - **Land Cover Reclassifier** Reclassifies a grid into UMEP format land cover grid. Land surface models
    - **Land Cover Fraction (Point)** Land cover fractions estimates from a land cover grid based on a specific point in space
    - **Land Cover Fraction (Grid)** Land cover fractions estimates from a land cover grid based on a polygon grid
  - **Urban Morphology**
    - **Morphometric Calculator (Point)** Morphometric parameters from a DSM based on a specific point in space
    - **Morphometric Calculator (Grid)** Morphometric parameters estimated from a DSM based on a polygon grid
    - **Source Area Model (Point)** Source area calculated from a DSM based on a specific point in space.
  - **SUEWS input data**
    - **SUEWS Prepare** Preprocessing and preparing input data for the SUEWS model
- **Processor**
  - **Anthropogenic Heat (Q_F)**
    - **LQF** Spatial variations anthropogenic heat release for urban areas
    - **GQF** Anthropogenic Heat (Q_F).
  - **Urban Energy Balance**
    - **SUEWS (Simple)** Urban Energy and Water Balance.
    - **SUEWS (Advanced)** Urban Energy and Water Balance.
- **Post-Processor**
  - **Urban Energy Balance**

      * **SUEWS analyser** Plugin for plotting and statistical analysis of model results from SUEWS simple and SUEWS advanced

  - **Benchmark**

      * **Benchmark System** For statistical analysis of model results, such as SUEWS

7.6.3 Differences between SUEWS and LUMPS

The largest difference between LUMPS and SUEWS is that the latter simulates the urban water balance in detail while LUMPS takes a simpler approach for the sensible and latent heat fluxes and the water balance ("water bucket"). The calculation of evaporation/latent heat in SUEWS is more biophysically based. Due to its simplicity, LUMPS requires less parameters in order to run. SUEWS gives turbulent heat fluxes calculated with both models as an output.

Similarities and differences between LUMPS and SUEWS.

	LUMPS	SUEWS	
Net all-wave radiation (Q*)	Input or NARP	Input or NARP	
Storage heat flux (QS)	Input or from OHM	Input or from OHM	
Anthropogenic heat flux (QF)	Input or calculated	Input or calculated	
Latent heat (QE)	DeBruin and Holtslag (1982)	Penman-Monteith equation2	
Sensible heat flux (QH)	DeBruin and Holtslag (1982)	Residual from available energy minus QE	
Water balance	No water balance included	Running water balance of canopy and water balance of soil	
Soil moisture	Not considered	Modelled	
Surface wetness	Simple water bucket model	Running water balance	
Irrigation	Only fraction of surface area that is irrigated	Input or calculated with a simple model	
Surface cover	Buildings, paved, vegetation	Buildings, paved, coniferous and deciduous trees/shrubs, irrigated and unirrigated grass	

7.6.4 Differences between SUEWS and FRAISE

FRAISE, Flux Ratio – Active Index Surface Exchange scheme, provides an estimate of mean midday (±3 h around solar noon) energy partitioning from information on the surface characteristics and estimates of the mean midday incoming radiative energy and anthropogenic heat release. Please refer to Loridan and Grimmond [2012] for further details.

Topic	FRAISE	LUMPS	SUEWS
Complexity	Simplest	Moderate	More complex
Software provided	R code	Windows exe (written in Fortran)	Windows exe (written in Fortran) - other versions available
Applicable period	Midday (within 3 h of solar noon) Calculates active surface	hourly	5 min-hourly-annu al
Unique features	Calculates active surface and fluxes	Radiation and energy balances	Radiation, energy and water balance (includes LUMPS)

7.7 Tutorials

7.7.1 SUEWS

To help users getting started with SUEWS, the community is working on setting up tutorials and instructions for different parts of SUEWS and related tool.

The tutorials are available are found in the table below.

Note: the following tutorials are hosted on a separate website including other UMEP related tutorials.

Topic	Application
IntroductionToSuews	Energy, water and radiation fluxes for one location
SUEWSAdvanced	Energy, water and radiation fluxes for one location
SUEWSSpatial	Energy, water and radiation fluxes for a spatial grid
SUEWSWUDAPT	Making use of WUDAPT local climate zones in SUEWS

7.7.2 SuPy

For Python users, a Python package *SuPy* with SUEWS as the calculation kernel is available to conduct SUEWS simulations. SuPy tutorials are provided at its tutorial site.

7.8 Benchmark Report

Since *v2018a*, SUEWS is benchmarked against observations for assessment of model performance. A site based benchmark report generation system is introduced in *v2018c* to produce detailed reports for testing sites; the number of sites is expanding and more cases will be added as they are benchmarked.

Each report includes the following parts:

1. **Overall performance**:

1. Performance Score: Large scores indicate better performance. The scores are calculated according to weighted averages of statistics for selected benchmark variables.
2. Detailed Statistics: Grids are coloured based relative performance between different versions: a **greener** grid indicates better performance in the chosen variable using the specific release whereas a **redder** one shows poorer performance; and those with **gray** backgrounds indicate the same performance across different releases.

2. **Cross-comparison in model variables between releases**:

1. Detailed statistics tables: statistics for each variable.
2. Pair plots: comparison in simulation results between different version-pairs.
3. Time series plots: comparison in simulated monthly climatologies of diurnal cycles of each variable between different version-pairs.

The latest benchmark reports are available at the SUEWS Benchmark site.

7.9 Notation

F Frontal area index

QS Storage heat flux

BLUEWS Boundary Layer part of SUEWS

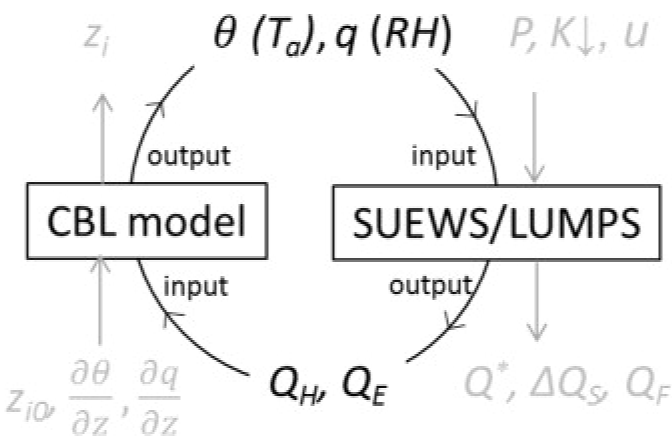


Fig. 7.3: Relation between BLUEWS and SUEWS

CDD Cooling degree days

GDD Growing degree days

HDD Heating degree days

CBL Convective boundary layer

DEM Digital Elevation Model

DSM Digital surface model

DTM Digital Terrain Model

ESTM Element Surface Temperature Method [Offerle *et al.*, 2005]

L↓ Incoming longwave radiation

LAI Leaf area index

LUMPS Local-scale Urban Meteorological Parameterization Scheme [Loridan *et al.*, 2011]

NARP Net All-wave Radiation Parameterization [Loridan *et al.*, 2011, Offerle *et al.*, 2003]

OHM Objective Hysteresis Model [Grimmond and Oke, 1999, Grimmond and Oke, 2002, Grimmond *et al.*, 1991]

Q* Net all-wave radiation

QE Latent heat flux

QF Anthropogenic heat flux

QH Sensible heat flux

SOLWEIG The solar and longwave environmental irradiance geometry model [Lindberg and Grimmond, 2011, Lindberg *et al.*, 2008]

SVF Sky view factor

Potential temperature

tt Time step of data

UMEP Urban Multi-scale Environmental Predictor

WATCH The WATCH project has produced a large number of data sets which should be of considerable use in regional and global studies of climate and water. see WATCH webpage

zi Convective boundary layer height

7.10 Contributing Guide

Note: This guide is heavily inspired by the excellent work by the xarray project: much appreciated!

Warning: This guide is incomplete and under construction: information here might be INCORRECT.

We welcome all contributions – bug reports/fixes, documentation corrctions/improments, enhancements, and ideas – as long as they apply to the SUEWS domain, please follow these guides:

7.10.1 Bug reports and enhancement requests

where to report a bug?

how to report a bug? what to be included? - version info - MWE (minimal working example) to reproduce the issue

7.10.2 Documentation Guide

7.10.3 Development Guide

Note: If you are interested in contributing to the code please open a new discussion in the UMEP Community to illustrate your proposal: we are happy to collaborate in an open development mode.

Essential pre-requisites

compliation

git

testing

Code guidelines

If you are interested in contributing to the code please contact Sue Grimmond.

Coding

1. Core physics and calculation schemes of SUEWS are written in Fortran 90
2. Code is hosted in GitHub as private repository
3. Variables
   - Names should be defined at least in one place in the code – ideally when defined
   - Implicit None should be used in all subroutines
   - Variable name should include units. e.g. Temp_C, Temp_K
   - Output variable attributes should be provided in the TYPE structure defined in the ctrl_output module as follows:

```
: TYPE varAttr
: CHARACTER(len = 15) :: header ! short name in headers
: CHARACTER(len = 12) :: unit   ! unit
: CHARACTER(len = 14) :: fmt    ! output format
: CHARACTER(len = 50) :: longNm ! long name for detailed description
: CHARACTER(len = 1)  :: aggreg ! aggregation method
: CHARACTER(len = 10) :: group  ! group: datetime, default, ESTM, Snow,␣
↪etc.
: INTEGER             :: level  ! output priority level: 0 for highest␣
↪(defualt output)
: END TYPE varAttr
```

4. Code should be written generally
5. Data set for testing should be provided
6. Demonstration that the model performance has improved when new code has been added or that any deterioration is warranted.
7. Additional requirements for modelling need to be indicated in the manual
8. All code should be commented in the program (with initials of who made the changes – name specified somewhere and institution)
9. The references used in the code and in the equations will be collected to a webpage
10. Current developments that are being actively worked on

Testing

1. The testing of SUEWS is done using Python 3
2. The following tests are done for each release of SUEWS:
1. Working status of *all physics schemes*
2. Year-grid looping logic
3. Identity of output results with internal test dataset

Please use pre-defined `make test` option to check if your code can pass all tests or not. If not, the correctness of added code should be justified with caution.

Preparation of SUEWS Manual

1. The SUEWS manual is written in reStructuredText (aka rst) with a Sphinx flavour
2. The SUEWS manual is hosted by readthedocs.org
3. CSV tables used in following pages are automatically generated from the *Description* field in *Input Options* by each build, so **DON'T** manually edit them as your edits will be swiped automatically:

- *SUEWS_AnthropogenicEmission.txt*
- *SUEWS_BiogenCO2.txt*
- *SUEWS_Conductance.txt*
- *SUEWS_Irrigation.txt*
- *SUEWS_NonVeg.txt*
- *SUEWS_OHMCoefficients.txt*
- *SUEWS_Profiles.txt*
- *SUEWS_SiteSelect.txt*
- *SUEWS_Snow.txt*
- *SUEWS_Soil.txt*
- *SUEWS_Veg.txt*
- *SUEWS_Water.txt*
- *SUEWS_WithinGridWaterDist.txt*

F2PY tips

This includes several **DON'T**'s that have never been mentioned by F2PY docs:

1. DON'T mix comments as lines into argument list of Fortran subroutines/functions:

   DONT:

   ```
   subroutine(&
   ! DONT DO this
   args&
   )
   ```

   OK:

```
subroutine(&
args& ! OK this way
)
```

2. DON'T end a subroutine as ENDSUBROUTINE. Instead, leave a space in between to form END SUBROUTINE. Otherwise, the subroutines won't be correctly parsed and picked up by F2PY.

7.11 API

This link redirects to the SUEWS API site, which provides documentation of SUEWS source code automatically generated by Doxygen.

SUEWS developers are strongly suggested to use the API site as the main reference for understanding SUEWS source code.

7.12 Acknowledgements

7.12.1 Contributors

Name	Affiliation	Contributions	Versions	Remarks
Prof Sue Grimmond	University of Reading, UK; prior: Indiana University, USA, King's College London, UK, University of British Columbia, Canada	OHM, Evaporation-Interception, Resistances, NARP, irrigation, anthropogenic heat, etc	v2011b – onwards	Team Leader
Dr Ting Sun	University of Reading, UK	AnOHM; Documentation system; WRF-SUEWS coupling; SuPy (python wrapper of SUEWS)	v2017b – onwards	Current Lead Developer
Dr Leena Järvi	University of Helsinki, Finland	Snow-related physics; Anthropogenic emission calculation, CO2	v2011b – v2019a	Lead Developer of v2011b – v2014b
Dr Helen Ward	University of Reading, UK	OHM improvement; Resistance calculation; Anthropogenic heat calculation	v2016a - v2017b	Lead Developer of v2016a - v2017
Dr Fredrik Lindberg	Göteborg University, Sweden	UMEP-related work, NARP, ESTM	v2011b – owards	Lead Developer of UMEP
Dr Lewis Blunn	University of Reading, UK	SUEWS-SPARTACUS coupling; RSL improvement	v2021a	Major contributor to SUEWS-SPARTACUS coupling
Dr Hamidreza Omidvar	University of Reading, UK	WRF-SUEWS coupling; Documentation system	v2018c – v2019a	Major contributor to WRF(v4.0)-SUEWS(v2018c) coupling
Minttu P. Havu	University of Helsinki, Finland	CO2	v2018c – v2019a	
Dr Zhenkun Li	Shanghai Climate Centre, China	WRF-SUEWS coupling	v2018b – v2018c	Major contributor to WRF(v3.9)-SUEWS(v2018b) coupling
Yihao Tang	University of Reading, UK	Stability, air temperature	v2018b - v2018c	
Dr Shiho Onomura	Göteborg University, Sweden	BLUEWS, ESTM	v2016a	
Dr Thomas Loridan	King's College London, UK	NARP	v2011a	
Dr Brian Offerle	Indiana University, USA	ESTM, NARP	v2011a	

7.12.2 Dependency Libraries

Note: We gratefully acknowledge the libraries/code that SUEWS uses as dependency and greatly appreciate their developers for the excellent work. Please let us know if any inapproriate use of these code and we will remove/modify the related parts accordingly.

Library	Remarks
datetime-fortran	date and time related processsing
minpack	AnOHM-related sinusoidal curve fitting
Recursive Fortran 95 quicksort routine	netCDF output for QGIS-compliant grid layout
Fortran Strings Module by Dr George Benthien	string processing

7.12.3 Funding

Note: The following grants are acknowledged for their contribution to **model development (D)** and/or **supportive observations (O)**.

Funder	Project	D , O
ERC Synergy	urbisphere 855005	D,O
NERC	APEx	D
NERC	COSMA NE/S005889/	D
UKRI	GCRF Urban Disaster Risk Hub	D
Newton/Met Office	CSSP-China (AJYG-DX4P1V HRC,AJYF-2GLAMK EUN, others)	D, O
NERC	ClearfLo Clean Air for London NE/H003231/1	O
NERC/Belmont	TRUC NE/L008971/1, G8MUREFU3FP-2201-075	D, O
EPSRC	LoHCool Low carbon climate-responsive Heating and Cooling of Cities EP/N009797/1	D
NERC	Independent Research Fellowship	D
NSF	BCS-0095284, ATM-0710631, BCS-0221105	D, O
EPSRC	Data Assimilation for the REsilient City (DARE) EP/P002331/1	O
Royal Society/Newton	Mobility funding	O
H2020	UrbanFluxes (637519)	D, O
EUf7	BRIDGE (211345)	D, O
EUf7	emBRACE (283201)	D, O
University of Reading	Sue Grimmond	O, D
KCL	Sue Grimmond	O
EPSRC	EP/I00159X/1 EP/I00159X/2 Materials Innovation Hub: Connecting Materials Culture to Materials Science	O
NERC	Field Spectroscopy Facility (FSF) 616.1110 Investigating the Urban Energy Balance of London	O
EUf7	MEGAPOLI 212520	D
NERC	Airborne Remote Sensing Facility & Field Spectroscopy Facility (GB08/19)	O
CFCAS	Environmental Prediction for Canadian Cities	D, O

7.13 Version History

Warning: Information here is ONLY for developers.

7.13.1 Version 2021a (in development)

- **Improvement**
  1. Added a new `RoughLenMomMethod` (4) to calculate roughness and displacement height as a function of plan area index and effective height of roughness elements following the ensemble mean of Fig 1a in [Grimmond and Oke, 1999]
  2. Coupled SPARCATUS into SUEWS for detailed modelling of radiation balance.
  3. Added a new option `DiagMethod` in *RunControl.nml* to control the output of radiation balance.
- **Changes**
  1. TO ADD
- **Fix**
  1. fixed a bug in radiation scheme: observed incoming longwave radiation cannot be used.
- **Known issues**
  1. Wind direction is not currently downscaled so non -999 values will cause an error.

7.13.2 Version 2020a (released on 14 May 2020)

Note: In a future release, we will **ONLY** deliver SUEWS along with SuPy as a command line tool *suews-run*: release of standalone SUEWS binaries **will be stopped** to ease our maintenance load and to facilitate rapid developments. Users will need to have Python 3.6+ to install SuPy:

```
python3 -m pip install -U supy
```

However, as the source code of SUEWS are public, users can feel free to compile standalone binaries for platforms of their own interests.

- **Improvement**
  1. A ponding water scheme is added in the automatic irrigation calculation; useful when a certain depth of ponding water to maintain in irrigation (e.g., flooding irrigation in rice crop-field).
  2. Irrigation fraction can be specified for all surfaces (previously only available for vegetated surfaces)
  3. A U-shape approach for calculating HDD/CDD is introduced to account for a wide comfort zone between heating and cooling critical temperatures.
- **Changes**
  1. A new `RoughLenHeatMethod` option `5`: adaptively choose option `1` for fully pervious surface or `2` otherwise (if any impervious surface exists).
  2. A new column `H_maintain` is added in *SUEWS_Irrigation.txt* to set ponding water depth.
  3. New columns to specify irrigation fractions for non-vegetated surfaces in *SUEWS_SiteSelect.txt*.

4. A new scheme option `BaseTMethod` in *RunControl.nml* to set calculation scheme for HDD/CDD.

- **Fix**

  NONE.

- **Known issues**

  1. Wind direction is not currently downscaled so non -999 values will cause an error.

7.13.3 Version 2019a (released on 15 November 2019)

Download page (under assets)

- **Improvement**

  1. An anthropogenic emission module is added. Module details refer to Järvi et al. (2019) [Järvi *et al.*, 2019].
  2. A canyon profile module is added. Module details refer to Theeuwes et al. (2019) [Theeuwes *et al.*, 2019].

- **Changes**

  1. Input file `SUEWS_AnthropogenicHeat.txt` is renamed to `SUEWS_AnthropogenicEmission.txt` with new parameters added: `MinFCMetab`, `MaxFCMetab`, `FrPDDwe`, `FcEF_v_kgkmWD` and `FcEF_v_kgkmWE`.
  2. BLUEWS has been recovered; set `CBLUse` to use it.
  3. Removed features:

  – SOLWEIG: fully removed from code.

  – netCDF: fully removed as this is very infrequently used; users who need this are suggested to use *SuPy* with help from `pandas` and xarray to save results in netCDF more elegantly.

- **Fix**

  1. Fixed a bug in LAI calculation for longterm runs.
  2. Fixed a bug in net all-wave radiation differential calculation for OHM.
  3. Fixed a bug in GDD/SDD calculation that different vegetative land covers could unexpectedly affect each other.
  4. Fixed water redistribution bug in snow module.

- **Known issues**

  1. Wind direction is not currently downscaled so non -999 values will cause an error.

7.13.4 Version 2018c (released on 21 February 2019)

Download page (under assets)

- **Improvement**

  1. SuPy (SUEWS in Python): a Python-enhanced wrapper of SUEWS, which can facilitate a more fluent workflow of SUEWS-centred urban climate research. More details refer to SuPy documentation site.
  2. Improved benchmark report: More testing sites are added thanks to an automated benchmark report system.

- **Changes**

  None.

- **Fix**

1. Fixed a bug in LAI calculation for longterm runs.
2. Fixed a bug in net all-wave radiation differential calculation for OHM.
3. Fixed water redistribution bug in snow module.

- **Known issues**
  1. BLUEWS is disabled
  2. Observed soil moisture can not be used as an input
  3. Wind direction is not currently downscaled so non -999 values will cause an error.

7.13.5 Version 2018b (released on 17 December 2018)

Download page (under assets)

- **Improvement**
  1. Improved calculation of OHM-related radiation terms:

     The temporal difference term `dQ*/dt` is now calculated using the time-step-weighted `dQ*` of previous time step instead of a series of `Q*` values from previous time steps, which improves the usage of memory and allows time-step-varying simulations (needed by WRF-SUEWS coupling).
- **Changes**

  None.
- **Fix**
  1. Fixed a bug in picking up external water use from meteorological forcing file.
- **Known issues**
  1. BLUEWS is disabled
  2. Observed soil moisture can not be used as an input
  3. Wind direction is not currently downscaled so non -999 values will cause an error.

7.13.6 Version 2018a (released on 2 August 2018)

- **New**
  1. Many under-the-hood improvements:
     - Added explicit interface intent for confusion-less coupling between SUEWS modules
     - Restructured layout of physics schemes for better modularity
     - Improved the alignment in output txt files
  2. New `readthedocs.org`-based documentation system
  3. Added *SUEWS input converter* for conversion of input files between versions
  4. Added *Benchmark Report* for recent releases.
- **Improvement**
  1. Improved the near surface diagnostics scheme (T2, Q2, U10)
  2. Improved skin temperature calculation (Ts)

- **Changes**
  1. *StabilityMethod*: recommended option is change from 2 to 3 as options other than 3 have been noticed with numerical issues under several scenarios, which will be fixed in the next release.
  2. Model run - changes in selections moved from *SUEWS_SiteSelect.txt* to `SUEWS_AnthropogenicHeat.txt`: *EnergyUseProfWD*, *EnergyUseProfWE*, *ActivityProfWD*, *ActivityProfWE*.
  3. *BiogenCO2Code* is added to *SUEWS_Veg.txt* for looking up biogenic characteristics in the new *SUEWS_BiogenCO2.txt* file.
  4. `TraifficRate` and `BuildEnergyUse` in *SUEWS_SiteSelect.txt* are expanded to allow weekday and weekend values: *TrafficRate_WD*, *TrafficRate_WE*, *QF0_BEU_WD*, *QF0_BEU_WE*.
  5. `AnthropCO2Method` is removed from *RunControl.nml*.
  6. `AnthropHeatMethod` is renamed to *EmissionsMethod*.
  7. `AHMin`, `AHSlope` and `TCritic` are expanded to allow weekday and weekend values by adding `_WD` and `_WE` as suffix, of which `AHSlope` and `TCritic` are also expanded to allow cooling and heating settings.
- **Known issues**
  1. BLUEWS is disabled
  2. Observed soil moisture can not be used as an input
  3. Wind direction is not currently downscaled so non -999 values will cause an error.

7.13.7 Version 2017b (released on 2 August 2017)

`PDF Manual for v2017b`

1. Surface-level diagnostics: T2 (air temperature at 2 m agl), Q2 (air specific humidity at 2 m agl) and U10 (wind speed at 10 m agl) added as default output.
2. Output in netCDF format. Please note this feature is **NOT** enabled in the public release due to the dependency of netCDF library. Assistance in enabling this feature may be requested to the development team via SUEWS mail list.
3. Edits to the manual.
4. New capabilities being developed, including two new options for calculating storage heat flux (AnOHM, ESTM) and modelling of carbon dioxide fluxes. These are currently under development and **should not be used** in Version 2017b.
5. Known issues
   1. BLUEWS parameters need to be checked
   2. Observed soil moisture can not be used as an input
   3. Wind direction is not currently downscaled so non -999 values will cause an error.

7.13.8 Version 2017a (released on 1 Feb 2017)

1. Changes to input file formats (including RunControl.nml and InitialConditions files) to facilitate setting up and running the model. Met forcing files no longer need two rows of -9 at the end to indicate the end of the file.
2. Changes to output file formats (now option to write out only a subset of variables, rather than all variables).
3. SUEWS can now disaggregate forcing files to the model time-step and aggregate output at the model time-step to lower resolution. This removes the need for the python wrapper used with previous versions.
4. InitialConditions format and requirements changed. A single file can now be provided for multiple grids. SUEWS will approximate most (but not all) of the required initial conditions if values are unknown. (However, if detailed information about the initial conditions is known, this can still be provided to and used by SUEWS.)
5. Leaf area index calculations now use parameters provided for each vegetated surface (previously only the deciduous tree LAI development parameters were applied to all vegetated surfaces).
6. For compatibility with GIS, **the sign convention for longitude has been changed**. Now negative values are to the west, positive values are to the east. Note this appears to have been incorrectly coded in previous versions (but may not necessarily have been problematic).
7. Storage heat flux calculation adapted for shorter (sub-hourly) model time-step: hysteresis calculation now based on running means over the previous hour.
8. Improved error handling, including separate files for serious errors (problems.txt) and less critical issues (warnings.txt).
9. Edits to the manual.
10. New capabilities being developed, including two new options for calculating storage heat flux (AnOHM, ESTM) and modelling of carbon dioxide fluxes. These are currently under development and **should not be used** in Version 2017a.

7.13.9 Version 2016a (released on 21 June 2016)

`PDF Manual for v2016a`

1. Major changes to the input file formats to facilitate the running of multiple grids and multiple years. Surface characteristics are provided in *SUEWS_SiteSelect.txt* and other input files are cross-referenced via codes or profile types.
2. The surface types have been altered:
   - Previously, grass surfaces were entered separately as irrigated grass and unirrigated grass surfaces, whilst the 'unmanaged' land cover fraction was assumed by the model to behave as unirrigated grass. There is now a single surface type for grass (total for irrigated plus unirrigated) and a new bare soil surface type.
   - The proportion of irrigated vegetation must now be specified for grass, evergreen trees and deciduous trees individually.
3. The entire model now runs at a time step specified by the user. Note that 5 min is strongly recommended. (Previously only the water balance calculations were done at 5 min with the energy balance calculations at 60 min).
4. Surface conductance now depends on the soil moisture under the vegetated surfaces only (rather than the total soil moisture for the whole study area as previously).
5. Albedo of evergreen trees and grass surfaces can now change with leaf area index as was previously possible for deciduous trees only.
6. New suggestions in Troubleshooting section.

7. Edits to the manual.
8. CBL model included.
9. SUEWS has been incorporated into UMEP

7.13.10 Version 2014b (released on 8 October 2014)

`PDF Manual for v2014b`

These affect the run configuration if previously run with older versions of the model:

1. New input of three additional columns in the Meteorological input file (diffusive and direct solar radiation, and wind direction)
2. Change of input variables in InitialConditions.nml file. Note we now refer to CT as ET (ie. Evergreen trees rather than coniferous trees)
3. In GridConnectionsYYYY.txt, the site names should now be without the underscore (e.g `Sm` and not `Sm_`)

Other issues:

1. Number of grid areas that can be modelled (for one grid, one year 120; for one grid two years 80)
2. Comment about Time interval of input data
3. Bug fix: Column headers corrected in 5 min file
4. Bug fix: Surface state 60 min file - corrected to give the last 5 min of the hour (rather than cumulating through the hour)
5. Bug fix: units in the Horizontal soil water transfer
6. ErrorHints: More have been added to the problems.txt file.
7. Manual: new section on running the model appropriately
8. Manual: notation table updated
9. Possibility to add snow accumulation and melt: new paper

Järvi L, Grimmond CSB, Taka M, Nordbo A, Setälä H, and Strachan IB Version 2014: Development of the Surface Urban Energy and Water balance Scheme (SUEWS) for cold climate cities, Geosci. Model Dev. 7, 1691-1711, doi:10.5194/gmd-7-1691-Version 2014.

7.13.11 Version 2014a.1 (released 26 February 2014)

1. Please see the large number of changes made in the Version 2014a release.
2. This is a minor change to address installing the software.
3. Minor updates to the manual

7.13.12 Version 2014a (released on 21 February 2014)

1. Bug fix: External irrigation is calculated as combined from automatic and manual irrigation and during precipitation events the manual irrigation is reduced to 60% of the calculated values. In previous version of the model, the irrigation was in all cases taken 60% of the calculated value, but now this has been fixed.
2. In previous versions of the model, irrigation was only allowed on the irrigated grass surface type. Now, irrigation is also allowed on evergreen and deciduous trees/shrubs surfaces. These are not however treated as separate surfaces, but the amount of irrigation is evenly distributed to the whole surface type in the modelled area. The amount of water is calculated using same equation as for grass surface (equation 5 in Järvi et al. Version 2011), and the fraction of irrigated trees/shrubs (relative to the area of tree/shrubs surface) is set in the gis file (See Table 4.11: SSss_YYYY.gis)
3. In the current version of the model, the user is able to adjust the leaf-on and leaf-off lengths in the FunctionalTypes. nml file. In addition, user can choose whether to use temperature dependent functions or combination of temperature and day length (advised to be used at high-latitudes)
4. In the gis-file, there is a new variable Alt that is the area altitude above sea level. If not known exactly use an approximate value.
5. Snow removal profile has been added to the HourlyProfileSSss_YYYY.txt. Not yet used!
6. Model time interval has been changed from minutes to seconds. Preferred interval is 3600 seconds (1 hour)
7. Manual correction: input variable Soil moisture said soil moisture deficit in the manual – word removed
8. Multiple compiled versions of SUEWS released. There are now users in Apple, Linux and Windows environments. So we will now release compiled versions for more operating systems (section 3).
9. There are some changes in the output file columns so please, check the respective table of each used output file.
10. Bug fix: with very small amount of vegetation in an area – impacted Phenology for LUMPS

7.13.13 Version 2013a

1. Radiation selection bug fixed
2. Aerodynamic resistance – when very low - no longer reverts to neutral (which caused a large jump) – but stays low
3. Irrigation day of week fixed
4. New error messages
5. min file – now includes a decimal time column – see Section 5.4 – Table 5.3

7.13.14 Version 2012b

1. Error message generated if all the data are not available for the surface resistance calculations
2. Error message generated if wind data are below zero plane displacement height.
3. All error messages now written to 'Problem.txt' rather than embedded in an ErrorFile. Note some errors will be written and the program will continue others will stop the program.
4. Default variables removed (see below). Model will stop if any data are problematic. File should be checked to ensure that reasonable data are being used. If an error occurs when there should not be one let us know as it may mean we have made the limits too restrictive.

Contents no longer used File defaultFcld=0.1 defaultPres=1013 defaultRH=50 defaultT=10 defaultU=3 RunControl.nml

- Just delete lines from file
- Values you had were likely different from these example value shown here

7.13.15 Version 2012a

1. Improved error messages when an error is encountered. Error message will generally be written to the screen and to the file 'problems.txt'
2. Format of all input files have changed.
3. New excel spreadsheet and R programme to help prepare required data files. (Not required)
4. Format of coef flux (OHM) input files have changed.
   - This allows for clearer identification for users of the coefficients that are actually to be used
   - This requires an additional file with coefficients. These do not need to be adjusted but new coefficients can be added. We would appreciate receiving additional coefficients so they can be included in future releases – Please email Sue.
5. Storage heat flux (OHM) coefficients can be changed by
   - time of year (summer, winter)
   - surface wetness state
6. New files are written: DailyState.txt
   - Provides the status of variables that are updated on a daily or basis or a snapshot at the end of each day.
7. Surface Types
   - Clarification of surface types has been made. See GIS and OHM related files

7.13.16 Version 2011b

1. Storage heat flux (Qs) and anthropogenic heat flux (QF) can be set to be 0 W m^{-2}
2. Calculation of hydraulic conductivity in soil has been improved and HydraulicConduct in SUEWSInput.nml is replaced with name SatHydraulicConduct
3. Following removed from HeaderInput.nml
   - HydraulicConduct
   - GrassFractionIrrigated
   - PavedFractionIrrigated
   - TreeFractionIrrigated

The lower three are now determined from the water use behaviour used in SUEWS

1. Following added to HeaderInput.nml
   - SatHydraulicConduct
   - defaultQf
   - defaultQs
2. If Qs and QF are not calculated in the model but are given as an input, the missing data is replaced with the default values.

3. Added to SAHP input file

   - AHDIUPRF – diurnal profile used if EmissionsMethod = 1

Version 2012a this became obsolete OHM file (SSss_YYYY.ohm)

7.14 Parameterisations and sub-models within SUEWS

7.14.1 Net all-wave radiation, Q*

There are several options for modelling or using observed radiation components depending on the data available. As a minimum, SUEWS requires incoming shortwave radiation to be provided.

1. Observed net all-wave radiation can be provided as input instead of being calculated by the model.
2. Observed incoming shortwave and incoming longwave components can be provided as input, instead of incoming longwave being calculated by the model.
3. Other data can be provided as input, such as cloud fraction (see options in *RunControl.nml*).
4. **NARP** (Net All-wave Radiation Parameterization) [Loridan *et al.*, 2011, Offerle *et al.*, 2003] scheme calculates outgoing shortwave and incoming and outgoing longwave radiation components based on incoming shortwave radiation, temperature, relative humidity and surface characteristics (albedo, emissivity).
5. *SPARTACUS-Surface (SS)* computes the 3D interaction of shortwave and longwave radiation with complex surface canopies, including vegetated and urban canopies (with or without vegetation). More details can be found in the *SPARTACUS-Surface (SS)* section.

7.14.2 Anthropogenic heat flux, Q_F

1. Two simple anthropogenic heat flux sub-models exist within SUEWS:
   - Järvi *et al.* [2011] approach, based on heating and cooling degree days and population density (allows distinction between weekdays and weekends).
   - Loridan *et al.* [2011] approach, based on a linear piece-wise relation with air temperature.
2. Pre-calculated values can be supplied with the meteorological forcing data, either derived from knowledge of the study site, or obtained from other models, for example:
   - **LUCY** [Allen *et al.*, 2010, Lindberg *et al.*, 2013]. A new version has been now included in UMEP. To distinguish it is referred to as LQF
   - **GreaterQF** [Iamarino *et al.*, 2011]. A new version has been now included in UMEP. To distinguish it is referred to as GQF

7.14.3 Storage heat flux, Q_S

1. Three sub-models are available to estimate the storage heat flux:
   - **OHM** (Objective Hysteresis Model) [Grimmond and Oke, 1999, Grimmond and Oke, 2002, Grimmond *et al.*, 1991]. Storage heat heat flux is calculated using empirically-fitted relations with net all-wave radiation and the rate of change in net all-wave radiation.
   - **AnOHM** (Analytical Objective Hysteresis Model) [Sun *et al.*, 2017]. OHM approach using analytically-derived coefficients. **Not recommended in this version.**

  - **ESTM** (Element Surface Temperature Method) [Offerle *et al.*, 2005]. Heat transfer through urban facets (roof, wall, road, interior) is calculated from surface temperature measurements and knowledge of material properties. **Not recommended in this version.**

2. Alternatively, 'observed' storage heat flux can be supplied with the meteorological forcing data.

7.14.4 Turbulent heat fluxes, Q_H and Q_E

1. **LUMPS** (Local-scale Urban Meteorological Parameterization Scheme) [Grimmond and Oke, 2002] provides a simple means of estimating sensible and latent heat fluxes based on the proportion of vegetation in the study area.
2. **SUEWS** adopts a more biophysical approach to calculate the latent heat flux; the sensible heat flux is then calculated as the residual of the energy balance. The initial estimate of stability is based on the LUMPS calculations of sensible and latent heat flux. Future versions will have alternative sensible heat and storage heat flux options.

Sensible and latent heat fluxes from both LUMPS and SUEWS are provided in the *Output files*. Whether the turbulent heat fluxes are calculated using LUMPS or SUEWS can have a major impact on the results. For SUEWS, an appropriate surface conductance parameterisation is also critical [Järvi *et al.*, 2011] [Ward *et al.*, 2016]. For more details see `Differences_between_SUEWS_LUMPS_and_FRAISE`.

7.14.5 Water balance

The running water balance at each time step is based on the urban water balance model of Grimmond *et al.* [1986] and urban evaporation-interception scheme of Grimmond and Oke [1991].

- Precipitation is a required variable in the meteorological forcing file.
- Irrigation can be modelled [Järvi *et al.*, 2011] or observed values can be provided if data are available.
- Drainage equations and coefficients to use must be specified in the input files.
- Soil moisture can be calculated by the model.
- Runoff is permitted:
  - between surface types within each model grid
  - between model grids (**Not available in this version.**)
  - to deep soil
  - to pipes.

7.14.6 Snowmelt

The snowmelt model is described in Järvi *et al.* [2014]. Changes since v2016a: 1) previously all surface states could freeze in 1-h time step, now the freezing surface state is calculated similarly as melt water and can freeze within the snow pack. 2) Snowmelt-related coefficients have also slightly changed (see *SUEWS_Snow.txt*).

7.14.7 Convective boundary layer

A convective boundary layer (CBL) slab model [Cleugh and Grimmond, 2001] calculates the CBL height, temperature and humidity during daytime [Onomura *et al.*, 2015].

7.14.8 Wind, Temperature and Humidity Profiles in the Roughness Sublayer

A dignostic RSL scheme for calculating the wind, temperature and humidity profiles in the roughness sublayer is implemented in 2020a following Harman and Finnigan [2007], Harman and Finnigan [2008] and Theeuwes *et al.* [2019]. An recent application of this RSL scheme can be found in Tang *et al.* [2021].

The diagnostic profiles are outputed in 30 uneven levels between the ground and forcing height, which are divided into two groups:

- One group of levels are evenly distributed within the urban canopy layer characterised by mean height of roughness elements (e.g. buildings, trees, etc.) z_H, which determines the number of layers within urban canopy n_{can}:

$$n_{can} = \begin{cases} 3 & \text{if } z_H \leq 2 \text{ m} \\ 10 & \text{if } 2 \text{ m} z_H \leq 10 \text{ m} \\ 15 & \text{if } z_H 10 \text{ m} \end{cases}$$

- The other levels are evenly distributed between the urban canopy layer top and forcing height.

Note: All the diagnostic profiles (wind speed, temperature and humidity) are calculated from the forcing data down into the canopy. Therefore it is assumed that the forcing temperature and humidity are above the blending height.

Common near-surface diagnostics:

- T2: air temperature at 2 m agl
- Q2: air specific humidity at 2 m agl
- RH2: air relative humidity at 2 m agl
- U10: wind speed at 10 m agl

are calculated by the *RSL scheme* by interpolating RSL profile results to the corresponding diagnostic heights.

7.14.9 SPARTACUS-Surface (SS)

Warning: This module is highly experimental and not yet fully tested: description here is not yet complete, either. Please refer to the original SPARTACUS-Surface page for more details, which may differ from the coupled version in SUEWS described below due to possibly different implementations.

Note: Future Work

- New SUEWS input table containing SPARTACUS profiles
- Add check for consistency of SUEWS and SS surface fractions
- Include snow

Introduction to SS

The SPARTACUS-Surface module computes the 3D interaction of shortwave and longwave radiation with complex surface canopies, including vegetated and urban canopies (with or without vegetation).

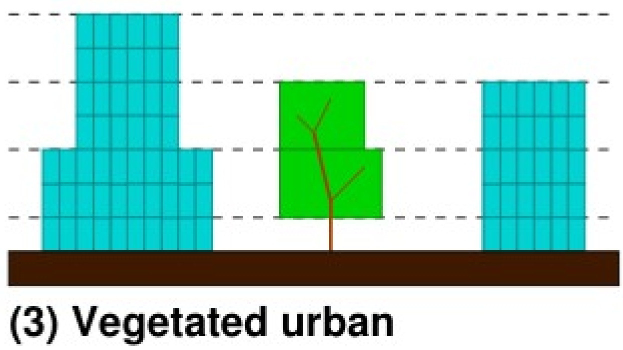


Fig. 7.4: Multi-layer structure (horizontal dashed lines) used in SS to characterise differences in the canopy (Cyan building, Green – vegetation). Source: SPARTACUS-Surface GH page

It uses a multi-layer description of the canopy (Fig. 7.4), with a statistical description of the horizontal distribution of trees and buildings. Assumptions include:

- Trees are randomly distributed.
- Wall-to-wall separation distances follow an exponential probability distribution.
- From a statistical representation of separation distances one can determine the probabilities of light being intercepted by trees, walls and the ground.

In the tree canopy (i.e. between buildings) there are two or three regions (based on user choice) (Fig. 7.5): clear-air and either one vegetated region or two vegetated regions of equal fractional cover but different extinction coefficient. Assumptions include:

- The rate of exchange of radiation between the clear and vegetated parts of a layer are assumed to be proportional to the length of the interface between them.
- Likewise for the rate of interception of radiation by building walls.

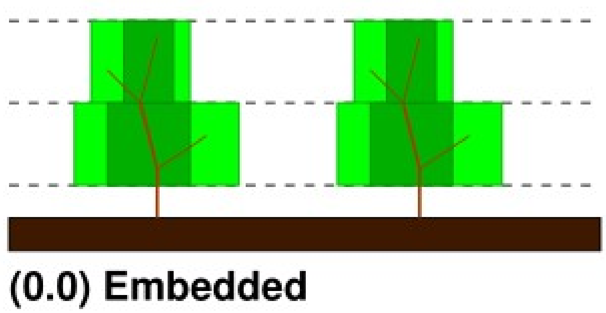


Fig. 7.5: Areas between trees. Source: SPARTACUS-Surface GH page

Each time light is intercepted it can undergo diffuse or specular reflection, be absorbed or be transmitted (as diffuse radiation). The probabilities for buildings and the ground are determined by albedos and emissivities, and for trees are determined by extinction coefficients and single scattering albedos.

SUEWS-SS Implementation

- Maximum of 15 vertical layers.
- Building and tree fractions, building and tree dimensions, building albedo and emissivity, and diffuse versus specular reflection, can be treated as vertically heterogenous or uniform with height depending on parameter choices.
- As tree fraction increases towards 1 it is assumed that the tree crown merges when calculating tree perimeters.
- Representing horizontal heterogeneity in the tree crowns is optional. When represented it is assumed that heterogeneity in leaf area index is between the core and periphery of the tree, not between trees.
- When calculating building perimeters it is assumed that buildings do not touch (analogous to crown shyness) as building fraction increases towards 1.
- Vegetation extinction coefficients (calculated from leaf area index, LAI) are assumed to be the same in all vegetated layers.
- Building facet and ground temperatures are equal to SUEWS TSfc_C (i.e.surface temperature) **[#estm_coupling]_**.
- Leaf temperatures are equal to SUEWS temp_C (i.e. air temperature within the canopy) **[#rsl_layers]_**.
- Ground albedo and emissivity are an area weighted average of SUEWS paved, grass, bare soil and water values.
- Inputs from SUEWS: `sfr`, `zenith_deg`, `TSfc_C`, `avKdn`, *`ldown`*, `temp_c`, `alb_next`, *`emis`*, `LAI_id`.
- SS specific input parameters: read in from *SUEWS_SPARTACUS.nml*.
- Outputs used by SUEWS: alb_spc, emis_spc, lw_emission_spc.
- Although the radiation is calculated in multiple vertical layers within SS it is only the upwelling top-of-canopy fluxes: `alb_spc*avKdn`, `(emis_spc)*ldown`, and `lw_emission_spc` that are used by SUEWS.
- Output variables (including multi-layer ones) are in SUEWS-SS output file `SSss_YYYY_SPARTACUS.txt`. **[#ss_output]_**

RSL and SS Canopy Representation Comparison

- The RSL has 30 levels but when the average building height is <2 m, < 12 m and > 12 m there are 3, 10 and 15 evenly spaced layers in the canopy.
- The remaining levels are evenly spaced up to the forcing level (Fig. 7.6).
- The buildings are assumed to be uniform height.

A maximum of 15 layers are used by SS (`vertial_layers_SS-RSL`), with the top of the highest layer at the tallest building height. The layer heights are user defined and there is no limit on maximum building height. The buildings are allowed to vary in height.

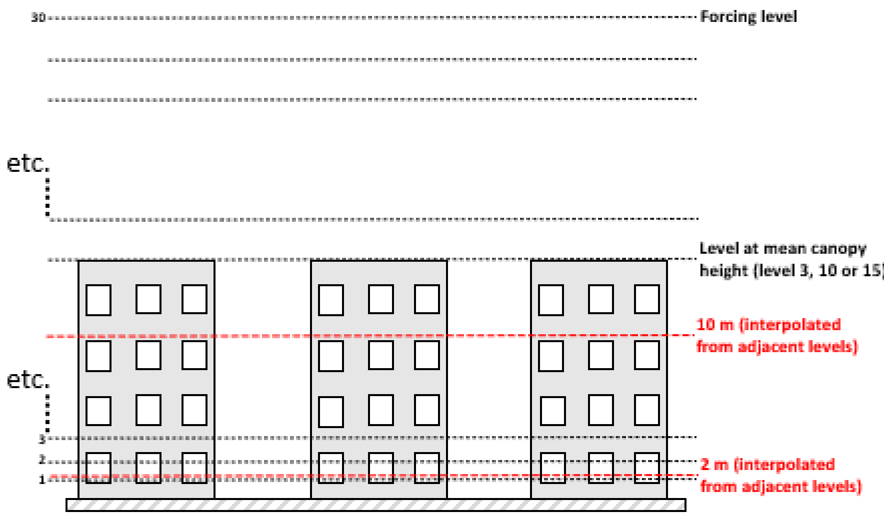


Fig. 7.6: SUEWS-RSL module assumes the RSL has 30 layers that are spread between the canopy and within the atmosphere above

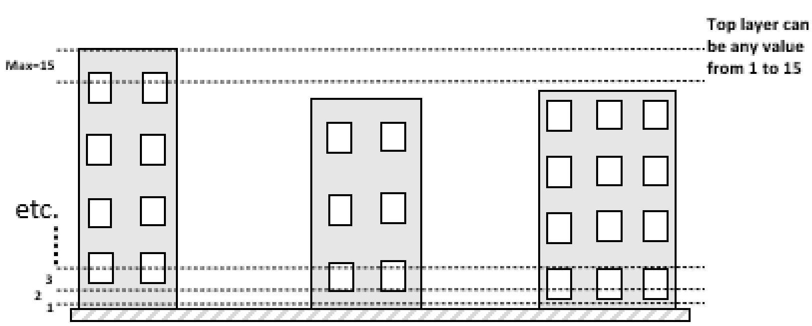


Fig. 7.7: Vertical layers used by SS

How to use SUEWS-SS

Inputs

To run SUEWS-SS the SS specific files that need to be modified are:

- *RunControl.nml* (see `NetRadiationMethod`)
- *SUEWS_SPARTACUS.nml*

Note: Non-SS specific SUEWS input file parameters also need to have appropriate values. For example, LAI, albedos and emissivities are used by SUEWS-SS as explained in *More background information*.

Outputs

See *SSss_YYYY_SPARTACUS_TT.txt*.

More background information

Vegetation single scattering albedo (SSA)

The **shortwave** broadband SSA is equal to the sum of the broadband reflectance R and broadband transmittance T [Yang *et al.*, 2020]. Given reflectance r and transmittance t spectra the SSA is calculated to modify equation

$$\mathrm{SSA} = \frac{\int_{\sim 400\ \mathrm{nm}}^{\sim 2200\ \mathrm{nm}} r \times S \mathrm{d}}{\int_{\sim 400\ \mathrm{nm}}^{\sim 2200\ \mathrm{nm}} S \mathrm{d}} + \frac{\int_{\sim 400\ \mathrm{nm}}^{\sim 2200\ \mathrm{nm}} t \times S \mathrm{d}}{\int_{\sim 400\ \mathrm{nm}}^{\sim 2200\ \mathrm{nm}} S \mathrm{d}}$$

where S clear-sky surface spectrum **:numfig:`rami5`**.

The integrals are performed between 400 nm and 2200 nm because this is the spectral range that RAMI5[5] Järvselja birch stand forest spectra are available. This is a reasonable approximation since it is where the majority of incoming SW energy resides (as seen from the clear-sky surface spectrum in Fig. 6).

Users can use the default value of 0.46, from RAMI5 Järvselja birch stand forest tree types or calculate their own SSA (Fig. 7.8). There are more tree R and T profiles here[5],

The **longwave** broadband SSA could be calculated in the same way but with the integral over the thermal infra-red (8-14 m), S replaced with the Plank function at Earth surface temperature, and r and t for the spectra for the thermal infra-red. The approximation that R + T = 2R can be made. r for different materials is available at https://speclib.jpl.nasa.gov/library. The peak in the thermal infra-red is ~10 m. Based on inspection of r profiles for several tree species SSA=0.06 is the default value.

Building albedo and emissivity

Use broadband values in Table C.1 of Kotthaus *et al.* [2014]. Full spectra can be found in the spectral library documentation.

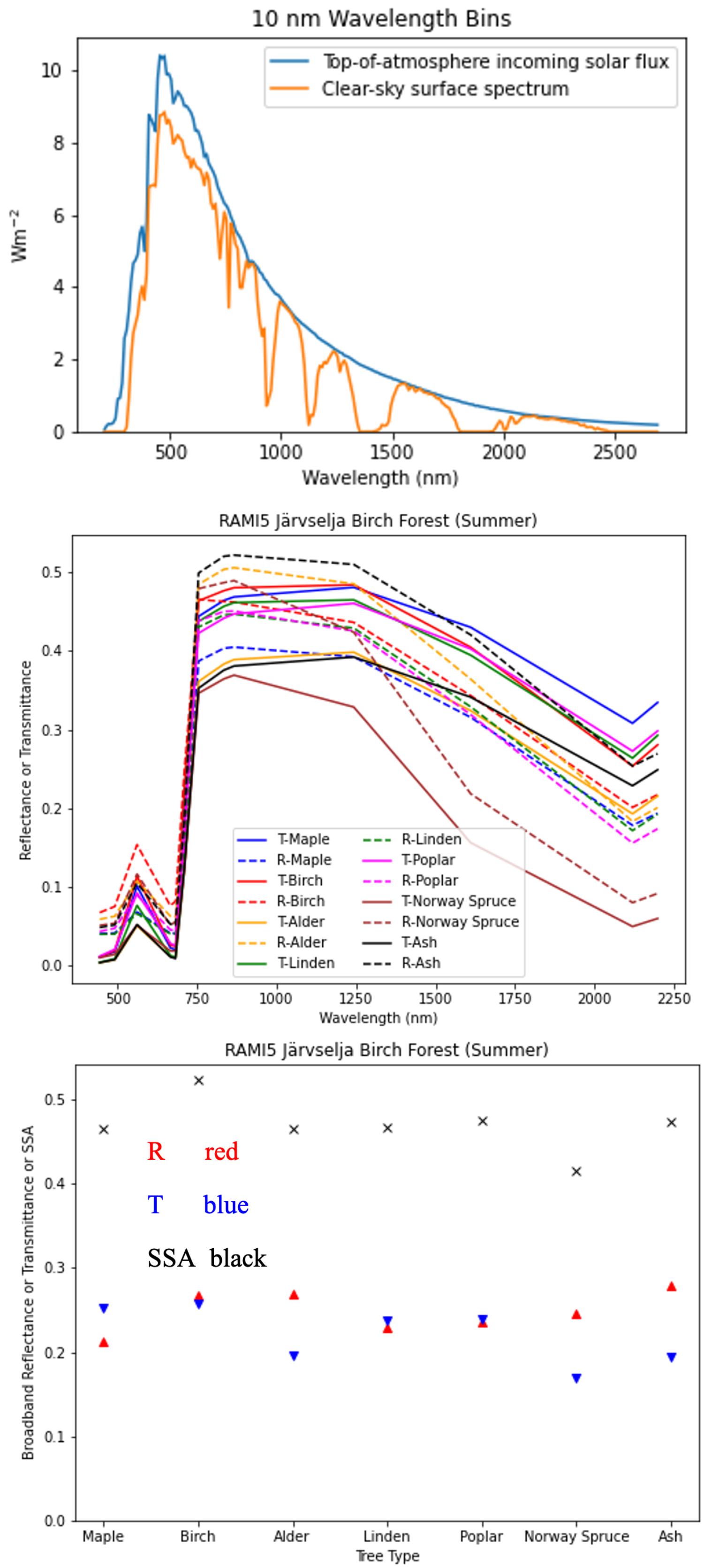


Fig. 7.8: RAMI5[5] data used to calculate R, T, and SSA, and R, T, and SSA values: (a) top-of-atmosphere incoming solar flux and clear-sky surface spectrum [Hogan and Matricardi, 2020] (b) RAMI5 r and t spectra, and (c) calculated broadband R, T, and SSA values.

Ground albedo and emissivity

In SUEWS-SS this is calculated as:

```
((1)*sfr(PavSurf)+(5)*sfr(GrassSurf)+(6)*sfr(BSoilSurf)+(7)*sfr(WaterSurf))/␣
↪(sfr(PavSurf) + sfr(GrassSurf) + sfr(BSoilSurf) + sfr(WaterSurf))
```

where is either the ground albedo or emissivity.

 values for the surfaces should be set by specifying surface codes in *SUEWS_SiteSelect.txt*. Codes should correspond to existing appropriate surfaces in *SUEWS_NonVeg.txt* and *SUEWS_NonVeg.txt*. Alternatively, new surfaces can be made in *SUEWS_NonVeg.txt* and *SUEWS_NonVeg.txt* with values obtained for example from the spectral library.

Consistency of SUEWS and SS parameters

SUEWS building and tree (evergreen+deciduous) fractions in *SUEWS_SiteSelect.txt* should be consistent with the *SUEWS_SPARTACUS.nml* `building_frac` and `veg_frac` of the lowest model layer.

Leaf area index (LAI)

The total vertically integrated LAI provided by SUEWS is used in SS to determine the LAI and vegetation extinction coefficient in each layer. Surface codes in *SUEWS_SiteSelect.txt* should correspond to appropriate LAI values in *SUEWS_Veg.txt*.

7.15 SUEWS-related Publications

Note:

1. If you have papers to add to this list please let us and others know via the email list.
2. The following list is sorted in a reversed chronological order.

- Tang, Yihao, Sun, Ting, Luo, Zhiwen, Omidvar, Hamidreza, Theeuwes, Natalie, Xie, Xiaoxiong, Xiong, Jie, Yao, Runming, and Grimmond, Sue. Urban meteorological forcing data for building energy simulations. *Building and Environment*, 204:108088, October 2021. doi:10.1016/j.buildenv.2021.108088.
- Theeuwes, Natalie E., Ronda, Reinder J., Harman, Ian N., Christen, Andreas, and Grimmond, C. Sue B. Parametrizing horizontally-averaged wind and temperature profiles in the urban roughness sublayer. *Boundary-Layer Meteorol*, 173(3):321–348, September 2019. doi:10.1007/s10546-019-00472-1.
- Järvi, Leena, Havu, Minttu, Ward, Helen C., Bellucco, Veronica, McFadden, Joseph P., Toivonen, Tuuli, Heikinheimo, Vuokko, Kolari, Pasi, Riikonen, Anu, and Grimmond, C. Sue B. Spatial modeling of Local-Scale biogenic and anthropogenic carbon dioxide emissions in Helsinki. *J. Geophys. Res. Atmos.*, 124(15):8363–8384, August 2019. doi:10.1029/2018jd029576.
- Sun, Ting and Grimmond, Sue. A python-enhanced urban land surface model SuPy (SUEWS in python, v2019.2): Development, deployment and demonstration. *Geosci. Model Dev.*, 12(7):2781–2795, July 2019. doi:10.5194/gmd-12-2781-2019.
- Kokkonen, Tom V., Grimmond, Sue, Murto, Sonja, Liu, Huizhi, Sundström, Anu-Maija, and Järvi, Leena. Simulation of the radiative effect of haze on the urban hydrological cycle using reanalysis data in Beijing. *Atmos. Chem. Phys.*, 19(10):7001–7017, May 2019. doi:10.5194/acp-19-7001-2019.

- Kokkonen, T. V., Grimmond, C. S. B., Christen, A., Oke, T. R., and Järvi, L. Changes to the water balance over a century of urban development in two neighborhoods: Vancouver, Canada. *Water Resour. Res.*, 54(9):6625–6642, September 2018. doi:10.1029/2017wr022445.
- Kokkonen, T.V., Grimmond, C.S.B., Räty, O., Ward, H.C., Christen, A., Oke, T.R., Kotthaus, S., and Järvi, L. Sensitivity of surface urban energy and water balance scheme (SUEWS) to downscaling of reanalysis forcing data. *Urban Clim.*, 23:36–52, March 2018. ICUC9: The 9th International Conference on Urban Climate. doi:10.1016/j.uclim.2017.05.001.
- Ao, Xiangyu, Grimmond, C. S. B., Ward, H. C., Gabey, A. M., Tan, Jianguo, Yang, Xiu-Qun, Liu, Dongwei, Zhi, Xing, Liu, Hongya, and Zhang, Ning. Evaluation of the surface urban energy and water balance scheme (SUEWS) at a dense urban site in shanghai: Sensitivity to anthropogenic heat and irrigation. *J. Hydrometeorol.*, 19(12):1983–2005, December 2018. doi:10.1175/jhm-d-18-0057.1.
- Ward, H.C. and Grimmond, C.S.B. Assessing the impact of changes in surface cover, human behaviour and climate on energy partitioning across greater London. *Landscape Urban Plan.*, 165:142–161, September 2017. doi:10.1016/j.landurbplan.2017.04.001.
- Sun, Ting, Wang, Zhi-Hua, Oechel, Walter C., and Grimmond, Sue. The analytical objective hysteresis model (AnOHM v1.0): Methodology to determine bulk storage heat flux coefficients. *Geosci. Model Dev.*, 10(7):2875–2890, July 2017. doi:10.5194/gmd-10-2875-2017.
- Järvi, L., Grimmond, C. S. B., McFadden, J. P., Christen, A., Strachan, I. B., Taka, M., Warsta, L., and Heimann, M. Warming effects on the urban hydrology in cold climate regions. *Sci Rep*, 7(1):5833, July 2017. doi:10.1038/s41598-017-05733-y.
- Ward, H.C., Kotthaus, S., Järvi, L., and Grimmond, C.S.B. Surface urban energy and water balance scheme (SUEWS): Development and evaluation at two UK sites. *Urban Clim.*, 18:1–32, December 2016. doi:10.1016/j.uclim.2016.05.001.
- Ao, Xiangyu, Grimmond, C. S. B., Liu, Dongwei, Han, Zhihui, Hu, Ping, Wang, Yadong, Zhen, Xinrong, and Tan, Jianguo. Radiation fluxes in a business district of shanghai, china. *J. Appl. Meteorol. Clim.*, 55(11):2451–2468, November 2016. doi:10.1175/jamc-d-16-0082.1.
- Onomura, S., Grimmond, C.S.B., Lindberg, F., Holmer, B., and Thorsson, S. Meteorological forcing data for urban outdoor thermal comfort models from a coupled convective boundary layer and surface energy balance scheme. *Urban Clim.*, 11:1–23, March 2015. doi:10.1016/j.uclim.2014.11.001.
- Järvi, L., Grimmond, C. S. B., Taka, M., Nordbo, A., Setälä, H., and Strachan, I. B. Development of the surface urban energy and water balance scheme (SUEWS) for cold climate cities. *Geosci. Model Dev.*, 7(4):1691–1711, August 2014. doi:10.5194/gmd-7-1691-2014.
- Lindberg, F., Grimmond, C.S.B., Yogeswaran, N., Kotthaus, S., and Allen, L. Impact of city changes and weather on anthropogenic heat flux in Europe 1995–2015. *Urban Clim.*, 4:1–15, July 2013. doi:10.1016/j.uclim.2013.03.002.
- Loridan, Thomas and Grimmond, C. S. B. Characterization of energy flux partitioning in urban environments: Links with surface seasonal properties. *J. Appl. Meteorol. Clim.*, 51(2):219–241, February 2012. doi:10.1175/jamc-d-11-038.1.
- Iamarino, Mario, Beevers, Sean, and Grimmond, C. S. B. High-resolution (space, time) anthropogenic heat emissions: London 1970-2025. *Int. J. Climatol.*, 32(11):1754–1767, July 2011. doi:10.1002/joc.2390.
- Järvi, L., Grimmond, C.S.B., and Christen, A. The surface urban energy and water balance scheme (SUEWS): Evaluation in Los Angeles and Vancouver. *J. Hydrol.*, 411(3-4):219–237, December 2011. doi:10.1016/j.jhydrol.2011.10.001.
- Loridan, Thomas, Grimmond, C. S. B., Offerle, Brian D., Young, Duick T., Smith, Thomas E. L., Järvi, Leena, and Lindberg, Fredrik. Local-scale urban meteorological parameterization scheme (LUMPS): Longwave radiation parameterization and seasonality-related developments. *J. Appl. Meteorol. Clim.*, 50(1):185–202, January 2011. doi:10.1175/2010jamc2474.1.

- Lindberg, Fredrik and Grimmond, C. S. B. The influence of vegetation and building morphology on shadow patterns and mean radiant temperatures in urban areas: Model development and evaluation. *Theor Appl Climatol*, 105(3-4):311–323, January 2011. doi:10.1007/s00704-010-0382-8.
- Allen, L., Lindberg, F., and Grimmond, C. S. B. Global to city scale urban anthropogenic heat flux: Model and variability. *Int. J. Climatol.*, 31(13):1990–2005, September 2010. doi:10.1002/joc.2210.
- Lindberg, Fredrik, Holmer, Björn, and Thorsson, Sofia. SOLWEIG 1.0 – modelling spatial variations of 3D radiant fluxes and mean radiant temperature in complex urban settings. *Int J Biometeorol*, 52(7):697–713, June 2008. doi:10.1007/s00484-008-0162-7.
- Offerle, B., Grimmond, C. S. B., and Fortuniak, K. Heat storage and anthropogenic heat flux in relation to the energy balance of a central European city centre. *Int. J. Climatol.*, 25(10):1405–1419, August 2005. doi:10.1002/joc.1198.
- Offerle, B., Grimmond, C. S. B., and Oke, T. R. Parameterization of net all-wave radiation for urban areas. *J. Appl. Meteor.*, 42(8):1157–1173, August 2003. doi:10.1175/1520-0450(2003)042<1157:ponarf>2.0.co;2.
- Grimmond, C. S. B. and Oke, T. R. Turbulent heat fluxes in urban areas: Observations and a local-scale urban meteorological parameterization scheme (LUMPS). *J. Appl. Meteor.*, 41(7):792–810, July 2002. doi:10.1175/1520-0450(2002)041<0792:thfiua>2.0.co;2.
- Grimmond, C. S. B. and Oke, T. R. Heat storage in urban areas: Local-scale observations and evaluation of a simple model. *J. Appl. Meteor.*, 38(7):922–940, July 1999. doi:10.1175/1520-0450(1999)038<0922:hsiual>2.0.co;2.
- Grimmond, C. S. B. and Oke, T. R. An evapotranspiration-interception model for urban areas. *Water Resour. Res.*, 27(7):1739–1755, July 1991. doi:10.1029/91wr00557.
- Grimmond, C.S.B., Cleugh, H.A., and Oke, T.R. An objective urban heat storage model and its comparison with other schemes. *Atmospheric Environment. Part B. Urban Atmosphere*, 25(3):311–326, January 1991. doi:10.1016/0957-1272(91)90003-w.
- Grimmond, C. S. B., Oke, T. R., and Steyn, D. G. Urban water balance: 1. a model for daily totals. *Water Resour. Res.*, 22(10):1397–1403, September 1986. doi:10.1029/wr022i010p01397.

7.16 References

References

[Allen et al. 2010] Allen, L., Lindberg, F., and Grimmond, C. S. B. Global to city scale urban anthropogenic heat flux: Model and variability. *Int. J. Climatol.*, 31(13):1990–2005, September 2010. doi:10.1002/joc.2210.

[Anandakumar 1999] Anandakumar, K. A study on the partition of net radiation into heat fluxes on a dry asphalt surface. *Atmos. Environ.*, 33(24-25):3911–3918, October 1999. doi:10.1016/s1352-2310(99)00133-8.

[Asaeda and Ca 1993] Asaeda, Takashi and Ca, Vu Thanh. The subsurface transport of heat and moisture and its effect on the environment: A numerical model. *Boundary-Layer Meteorol*, 65(1-2):159–179, July 1993. doi:10.1007/bf00708822.

[Auer 1974] Auer, August H. The rain versus snow threshold temperatures. *Weatherwise*, 27(2):67–67, April 1974. doi:10.1080/00431672.1974.9931684.

[Bellucco et al. 2017] Bellucco, Veronica, Marras, Serena, Grimmond, C. Susan B., Järvi, Leena, Sirca, Costantino, and Spano, Donatella. Modelling the biogenic CO 2 exchange in urban and non-urban ecosystems through the assessment of light-response curve parameters. *Agr. Forest Meteorol.*, 236:113–122, April 2017. doi:10.1016/j.agrformet.2016.12.011.

[Berrizbeitia et al. 2020] Berrizbeitia, Saioa Etxebarria, Jadraque Gago, Eulalia, and Muneer, Tariq. Empirical Models for the Estimation of Solar Sky-Diffuse Radiation. A Review and Experimental Analysis. *Energies*, 13(3):701, Feb 2020. doi:10.3390/en13030701.

[Best and Grimmond 2014] Best, M.J. and Grimmond, C.S.B. Importance of initial state and atmospheric conditions for urban land surface models' performance. *Urban Clim.*, 10:387–406, December 2014. doi:10.1016/j.uclim.2013.10.006.

[Breuer et al. 2003] Breuer, Lutz, Eckhardt, Klaus, and Frede, Hans-Georg. Plant parameter values for models in temperate climates. *Ecol. Model.*, 169(2-3):237–293, November 2003. doi:10.1016/s0304-3800(03)00274-6.

[Businger et al. 1971] Businger, J. A., Wyngaard, J. C., Izumi, Y., and Bradley, E. F. Flux-profile relationships in the atmospheric surface layer. *J. Atmos. Sci.*, 28(2):181–189, March 1971. doi:10.1175/1520-0469(1971)028<0181:fprita>2.0.co;2.

[Calder and Wright 1986] Calder, I. R. and Wright, I. R. Gamma ray attenuation studies of interception from sitka spruce: Some evidence for an additional transport mechanism. *Water Resour. Res.*, 22(3):409–417, March 1986. doi:10.1029/wr022i003p00409.

[Campbell and Norman 1998] Campbell, Gaylon S. and Norman, John M. *Wind*, chapter Wind, pages 63–75. Springer New York, New York, NY, 1998. doi:10.1007/978-1-4612-1626-1_5.

[Cleugh and Grimmond 2001] Cleugh, H. A. and Grimmond, C. S. B. Modelling regional scale surface energy exchanges and cbl growth in a heterogeneous, urban-rural landscape. *Bound.-Layer Meteorol.*, 98(1):1–31, January 2001. doi:10.1023/a:1018798928158.

[Davies and Hollis 1981] Davies, Hilary and Hollis, T. Measurements of rainfall-runoff volume relationships and water balance for roofs and roads. In *Second International Conference on Urban Storm Drainage*, volume 434, 441. 1981.

[Doll et al. 1985] Doll, Dennis, Ching, J. K. S., and Kaneshiro, Jack. Parameterization of subsurface heating for soil and concrete using net radiation data. *Boundary-Layer Meteorol*, 32(4):351–372, August 1985. doi:10.1007/bf00122000.

[Dyer 1974] Dyer, A. J. A review of flux-profile relationships. *Boundary-Layer Meteorol*, 7(3):363–372, November 1974. doi:10.1007/bf00240838.

[Falk and Niemczynowicz 1978] Falk, J. and Niemczynowicz, J. Characteristics of the above ground runoff in sewered catchments. In *Proc. Internat. Conf. on Urban Storm Drainage*, 159–171. John Wiley and Sons, New York, 1978.

[Flanagan et al. 2002] Flanagan, Lawrence B., Wever, Linda A., and Carlson, Peter J. Seasonal and interannual variation in carbon dioxide exchange and carbon balance in a northern temperate grassland. *Global Change Biol.*, 8(7):599–615, June 2002. doi:10.1046/j.1365-2486.2002.00491.x.

[Fuchs and Hadas 1972] Fuchs, Marcel and Hadas, Amos. The heat flux density in a non-homogeneous bare loessial soil. *Boundary-Layer Meteorol*, 3(2):191–200, December 1972. doi:10.1007/bf02033918.

[Grimmond 1992] Grimmond, C. S. B. The suburban energy balance: Methodological considerations and results for a mid-latitude west coast city under winter and spring conditions. *Int. J. Climatol.*, 12(5):481–497, July 1992. doi:10.1002/joc.3370120506.

[Grimmond and Oke 1991] Grimmond, C. S. B. and Oke, T. R. An evapotranspiration-interception model for urban areas. *Water Resour. Res.*, 27(7):1739–1755, July 1991. doi:10.1029/91wr00557.

[Grimmond and Oke 1999] Grimmond, C. S. B. and Oke, T. R. Heat storage in urban areas: Local-scale observations and evaluation of a simple model. *J. Appl. Meteor.*, 38(7):922–940, July 1999. doi:10.1175/1520-0450(1999)038<0922:hsiual>2.0.co;2.

[Grimmond and Oke 2002] Grimmond, C. S. B. and Oke, T. R. Turbulent heat fluxes in urban areas: Observations and a local-scale urban meteorological parameterization scheme (LUMPS). *J. Appl. Meteor.*, 41(7):792–810, July 2002. doi:10.1175/1520-0450(2002)041<0792:thfiua>2.0.co;2.

[Grimmond et al. 1986] Grimmond, C. S. B., Oke, T. R., and Steyn, D. G. Urban water balance: 1. a model for daily totals. *Water Resour. Res.*, 22(10):1397–1403, September 1986. doi:10.1029/wr022i010p01397.

[Grimmond et al. 1991] Grimmond, C.S.B., Cleugh, H.A., and Oke, T.R. An objective urban heat storage model and its comparison with other schemes. *Atmospheric Environment. Part B. Urban Atmosphere*, 25(3):311–326, January 1991. doi:10.1016/0957-1272(91)90003-w.

[Halldin et al. 1979] Halldin, S., Grip, H., and Perttu, K. Model for energy exchange of a pine forest canopy. In *Comparison of Forest Water and Energy Exchange Models*, pages 59–75. Elsevier, 1979. doi:10.1016/b978-0-444-41844-9.50012-3.

[Harman and Finnigan 2007] Harman, Ian N. and Finnigan, John J. A simple unified theory for flow in the canopy and roughness sublayer. *Boundary-Layer Meteorol*, 123(2):339–363, March 2007. doi:10.1007/s10546-006-9145-6.

[Harman and Finnigan 2008] Harman, Ian N. and Finnigan, John J. Scalar concentration profiles in the canopy and roughness sublayer. *Boundary-Layer Meteorol*, 129(3):323–351, October 2008. doi:10.1007/s10546-008-9328-4.

[Hogan 2019] Hogan, Robin J. Flexible Treatment of Radiative Transfer in Complex Urban Canopies for Use in Weather and Climate Models. *Boundary-Layer Meteorol.*, 173(1):53–78, Oct 2019. doi:10.1007/s10546-019-00457-0.

[Hogan and Matricardi 2020] Hogan, Robin J. and Matricardi, Marco. Evaluating and improving the treatment of gases in radiation schemes: the Correlated K-Distribution Model Intercomparison Project (CKDMIP). *Geosci. Model Dev.*, 13(12):6501–6521, Dec 2020. doi:10.5194/gmd-13-6501-2020.

[Hogan et al. 2018] Hogan, Robin J., Quaife, Tristan, and Braghiere, Renato. Fast matrix treatment of 3-D radiative transfer in vegetation canopies: SPARTACUS-Vegetation 1.1. *Geosci. Model Dev.*, 11(1):339–350, Jan 2018. doi:10.5194/gmd-11-339-2018.

[Högström 1988] Högström, Ulf. Non-dimensional wind and temperature profiles in the atmospheric surface layer: A re-evaluation. *Boundary-Layer Meteorol*, 42(1-2):55–78, January 1988. doi:10.1007/bf00119875.

[Iamarino et al. 2011] Iamarino, Mario, Beevers, Sean, and Grimmond, C. S. B. High-resolution (space, time) anthropogenic heat emissions: London 1970-2025. *Int. J. Climatol.*, 32(11):1754–1767, July 2011. doi:10.1002/joc.2390.

[Jarvis 1976] Jarvis, P. G. The interpretation of the variations in leaf water potential and stomatal conductance found in canopies in the field. *Phil. Trans. R. Soc. Lond. B*, 273(927):593–610, February 1976. doi:10.1098/rstb.1976.0035.

[Järvi et al. 2014] Järvi, L., Grimmond, C. S. B., Taka, M., Nordbo, A., Setälä, H., and Strachan, I. B. Development of the surface urban energy and water balance scheme (SUEWS) for cold climate cities. *Geosci. Model Dev.*, 7(4):1691–1711, August 2014. doi:10.5194/gmd-7-1691-2014.

[Järvi et al. 2011] Järvi, L., Grimmond, C.S.B., and Christen, A. The surface urban energy and water balance scheme (SUEWS): Evaluation in Los Angeles and Vancouver. *J. Hydrol.*, 411(3-4):219–237, December 2011. doi:10.1016/j.jhydrol.2011.10.001.

[Järvi et al. 2012] Järvi, L., Nordbo, A., Junninen, H., Riikonen, A., Moilanen, J., Nikinmaa, E., and Vesala, T. Seasonal and annual variation of carbon dioxide surface fluxes in helsinki, finland, in 2006–2010. *Atmos. Chem. Phys.*, 12(18):8475–8489, September 2012. doi:10.5194/acp-12-8475-2012.

[Järvi et al. 2019] Järvi, Leena, Havu, Minttu, Ward, Helen C., Bellucco, Veronica, McFadden, Joseph P., Toivonen, Tuuli, Heikinheimo, Vuokko, Kolari, Pasi, Riikonen, Anu, and Grimmond, C. Sue B. Spatial modeling of Local-Scale biogenic and anthropogenic carbon dioxide emissions in Helsinki. *J. Geophys. Res. Atmos.*, 124(15):8363–8384, August 2019. doi:10.1029/2018jd029576.

[Kanda et al. 2007] Kanda, M., Kanega, M., Kawai, T., Moriwaki, R., and Sugawara, H. Roughness lengths for momentum and heat derived from outdoor urban scale models. *J. Appl. Meteorol. Clim.*, 46(7):1067–1079, July 2007. doi:10.1175/jam2500.1.

[Kawai et al. 2009] Kawai, Toru, Ridwan, Mohammad Kholid, and Kanda, Manabu. Evaluation of the simple urban energy balance model using selected data from 1-yr flux observations at two cities. *J. Appl. Meteorol. Clim.*, 48(4):693–715, April 2009. doi:10.1175/2008jamc1891.1.

[Kent et al. 2017a] Kent, Christoph W., Grimmond, Sue, Barlow, Janet, Gatey, David, Kotthaus, Simone, Lindberg, Fredrik, and Halios, Christos H. Evaluation of urban local-scale aerodynamic parameters: Implications for the vertical profile of wind speed and for source areas. *Boundary-Layer Meteorol*, 164(2):183–213, April 2017. doi:10.1007/s10546-017-0248-z.

[Kent et al. 2017b] Kent, Christoph W., Grimmond, Sue, and Gatey, David. Aerodynamic roughness parameters in cities: Inclusion of vegetation. *J. Wind Eng. Ind. Aerod.*, 169:168–176, October 2017. doi:10.1016/j.jweia.2017.07.016.

[Kokkonen et al. 2018a] Kokkonen, T. V., Grimmond, C. S. B., Christen, A., Oke, T. R., and Järvi, L. Changes to the water balance over a century of urban development in two neighborhoods: Vancouver, Canada. *Water Resour. Res.*, 54(9):6625–6642, September 2018. doi:10.1029/2017wr022445.

[Kokkonen et al. 2018b] Kokkonen, T.V., Grimmond, C.S.B., Räty, O., Ward, H.C., Christen, A., Oke, T.R., Kotthaus, S., and Järvi, L. Sensitivity of surface urban energy and water balance scheme (SUEWS) to downscaling of reanalysis forcing data. *Urban Clim.*, 23:36–52, March 2018. ICUC9: The 9th International Conference on Urban Climate. doi:10.1016/j.uclim.2017.05.001.

[Konarska et al. 2013] Konarska, Janina, Lindberg, Fredrik, Larsson, Annika, Thorsson, Sofia, and Holmer, Björn. Transmissivity of solar radiation through crowns of single urban trees—application for outdoor thermal comfort modelling. *Theor Appl Climatol*, 117(3-4):363–376, September 2013. doi:10.1007/s00704-013-1000-3.

[Kotthaus et al. 2014] Kotthaus, Simone, Smith, Thomas E. L., Wooster, Martin J., and Grimmond, C. S. B. Derivation of an urban materials spectral library through emittance and reflectance spectroscopy. *ISPRS J. Photogramm. Remote Sens.*, 94:194–212, Aug 2014. doi:10.1016/j.isprsjprs.2014.05.005.

[Lindberg et al. 2013] Lindberg, F., Grimmond, C.S.B., Yogeswaran, N., Kotthaus, S., and Allen, L. Impact of city changes and weather on anthropogenic heat flux in Europe 1995–2015. *Urban Clim.*, 4:1–15, July 2013. doi:10.1016/j.uclim.2013.03.002.

[Lindberg and Grimmond 2011] Lindberg, Fredrik and Grimmond, C. S. B. The influence of vegetation and building morphology on shadow patterns and mean radiant temperatures in urban areas: Model development and evaluation. *Theor Appl Climatol*, 105(3-4):311–323, January 2011. doi:10.1007/s00704-010-0382-8.

[Lindberg et al. 2008] Lindberg, Fredrik, Holmer, Björn, and Thorsson, Sofia. SOLWEIG 1.0 – modelling spatial variations of 3D radiant fluxes and mean radiant temperature in complex urban settings. *Int J Biometeorol*, 52(7):697–713, June 2008. doi:10.1007/s00484-008-0162-7.

[Loridan and Grimmond 2012] Loridan, Thomas and Grimmond, C. S. B. Characterization of energy flux partitioning in urban environments: Links with surface seasonal properties. *J. Appl. Meteorol. Clim.*, 51(2):219–241, February 2012. doi:10.1175/jamc-d-11-038.1.

[Loridan et al. 2011] Loridan, Thomas, Grimmond, C. S. B., Offerle, Brian D., Young, Duick T., Smith, Thomas E. L., Järvi, Leena, and Lindberg, Fredrik. Local-scale urban meteorological parameterization scheme (LUMPS): Longwave radiation parameterization and seasonality-related developments. *J. Appl. Meteorol. Clim.*, 50(1):185–202, January 2011. doi:10.1175/2010jamc2474.1.

[Macdonald et al. 1998] Macdonald, R.W., Griffiths, R.F., and Hall, D.J. An improved method for the estimation of surface roughness of obstacle arrays. *Atmos. Environ.*, 32(11):1857–1864, June 1998. doi:10.1016/s1352-2310(97)00403-2.

[McCaughey 1985] McCaughey, J. H. Energy balance storage terms in a mature mixed forest at petawawa, Ontario ? a case study. *Boundary-Layer Meteorol*, 31(1):89–101, January 1985. doi:10.1007/bf00120036.

[Meyn and Oke 2009] Meyn, Stephanie K. and Oke, T.R. Heat fluxes through roofs and their relevance to estimates of urban heat storage. *Energ. Buildings*, 41(7):745–752, July 2009. doi:10.1016/j.enbuild.2009.02.005.

[NARITA et al. 1984] NARITA, Ken-ichi, SEKINE, Takeshi, and TOKUOKA, Toshikazu. Thermal properties of urban surface materials. *Geogr. Rev. Jpn, Ser. A, Chirigaku Hyoron*, 57(9):639–651, September 1984. doi:10.4157/grj1984a.57.9_639.

[Novak 1981] Novak, Michael David. *The moisture and thermal regimes of a bare soil in the lower Fraser Valley during spring*. PhD thesis, University of British Columbia, 1981. doi:10.14288/1.0095282.

[Nunez and Oke 1977] Nunez, M. and Oke, T. R. The energy balance of an urban canyon. *J. Appl. Meteor.*, 16(1):11–19, January 1977. doi:10.1175/1520-0450(1977)016<0011:teboau>2.0.co;2.

[Offerle et al. 2005] Offerle, B., Grimmond, C. S. B., and Fortuniak, K. Heat storage and anthropogenic heat flux in relation to the energy balance of a central European city centre. *Int. J. Climatol.*, 25(10):1405–1419, August 2005. doi:10.1002/joc.1198.

[Offerle et al. 2003] Offerle, B., Grimmond, C. S. B., and Oke, T. R. Parameterization of net all-wave radiation for urban areas. *J. Appl. Meteor.*, 42(8):1157–1173, August 2003. doi:10.1175/1520-0450(2003)042<1157:ponarf>2.0.co;2.

[Oke 2002] Oke, T. R. *Boundary Layer Climates*. Routledge, September 2002. ISBN 9781134951345. doi:10.4324/9780203407219.

[Onomura et al. 2015] Onomura, S., Grimmond, C.S.B., Lindberg, F., Holmer, B., and Thorsson, S. Meteorological forcing data for urban outdoor thermal comfort models from a coupled convective boundary layer and surface energy balance scheme. *Urban Clim.*, 11:1–23, March 2015. doi:10.1016/j.uclim.2014.11.001.

[Reindl et al. 1990] Reindl, D.T., Beckman, W.A., and Duffie, J.A. Diffuse fraction correlations. *Sol. Energy*, 45(1):1–7, 1990. doi:10.1016/0038-092x(90)90060-p.

[Ruimy et al. 1995] Ruimy, A., Jarvis, P.G., Baldocchi, D.D., and Saugier, B. CO2 fluxes over plant canopies and solar radiation: A review. In *Advances in Ecological Research*, pages 1–68. Elsevier, 1995. doi:10.1016/s0065-2504(08)60063-x.

[Sailor and Lu 2004] Sailor, David J. and Lu, Lu. A top–down methodology for developing diurnal and seasonal anthropogenic heating profiles for urban areas. *Atmos. Environ.*, 38(17):2737–2748, June 2004. doi:10.1016/j.atmosenv.2004.01.034.

[Sailor and Vasireddy 2006] Sailor, David J. and Vasireddy, Chittaranjan. Correcting aggregate energy consumption data to account for variability in local weather. *Environ. Modell. Softw.*, 21(5):733–738, May 2006. doi:10.1016/j.envsoft.2005.08.001.

[Schmid 2000] Schmid, H. Measurements of CO2 and energy fluxes over a mixed hardwood forest in the mid-western united states. *Agr. Forest Meteorol.*, 103(4):357–374, July 2000. doi:10.1016/s0168-1923(00)00140-4.

[South et al. 1998] South, Catherine, Susan, C., Grimmond, B., and Wolfe, Charlotte P. Evapotranspiration rates from wetlands with different disturbance histories: Indiana dunes national lakeshore. *Wetlands*, 18(2):216–229, June 1998. doi:10.1007/bf03161657.

[Sun et al. 2017] Sun, Ting, Wang, Zhi-Hua, Oechel, Walter C., and Grimmond, Sue. The analytical objective hysteresis model (AnOHM v1.0): Methodology to determine bulk storage heat flux coefficients. *Geosci. Model Dev.*, 10(7):2875–2890, July 2017. doi:10.5194/gmd-10-2875-2017.

[Taesler 1980] Taesler, Roger. *Studies of the development and thermal structure of the urban boundary layer in uppsala. Part 1: Experimental program and Part 2: Data analysis and results. Rep. 61*. Uppsala, Sweden : Uppsala Universitetet. Uppsala, Sweden : Uppsala Universitetet, 1980. ISBN 221015643.

[Tang et al. 2021] Tang, Yihao, Sun, Ting, Luo, Zhiwen, Omidvar, Hamidreza, Theeuwes, Natalie, Xie, Xiaoxiong, Xiong, Jie, Yao, Runming, and Grimmond, Sue. Urban meteorological forcing data for building energy simulations. *Building and Environment*, 204:108088, October 2021. doi:10.1016/j.buildenv.2021.108088.

[Theeuwes et al. 2019] Theeuwes, Natalie E., Ronda, Reinder J., Harman, Ian N., Christen, Andreas, and Grimmond, C. Sue B. Parametrizing horizontally-averaged wind and temperature profiles in the urban roughness sublayer. *Boundary-Layer Meteorol*, 173(3):321–348, September 2019. doi:10.1007/s10546-019-00472-1.

[Van Ulden and Holtslag 1985] Van Ulden, A. P. and Holtslag, A. A. M. Estimation of atmospheric boundary layer parameters for diffusion applications. *J. Climate Appl. Meteor.*, 24(11):1196–1207, November 1985. doi:10.1175/1520-0450(1985)024<1196:eoablp>2.0.co;2.

[Voogt and Grimmond 2000] Voogt, J. A. and Grimmond, C. S. B. Modeling surface sensible heat flux using surface radiative temperatures in a simple urban area. *J. Appl. Meteorol.*, 39(10):1679–1699, October 2000. doi:10.1175/1520-0450-39.10.1679.

[Ward et al. 2016] Ward, H.C., Kotthaus, S., Järvi, L., and Grimmond, C.S.B. Surface urban energy and water balance scheme (SUEWS): Development and evaluation at two UK sites. *Urban Clim.*, 18:1–32, December 2016. doi:10.1016/j.uclim.2016.05.001.

[Yang et al. 2020] Yang, Bin, He, Yuhao, and Chen, Wei. A simple method for estimation of leaf dry matter content in fresh leaves using leaf scattering albedo. *Global Ecol. Conserv.*, 23:e01201, Sep 2020. doi:10.1016/j.gecco.2020.e01201.

[Yap 1973] Yap, David Hamilton. *Sensible heat fluxes measured in and near Vancouver, B.C.* PhD thesis, University of British Columbia, 1973. doi:10.14288/1.0101037.

[Yoshida et al. 1990] Yoshida, Atsumasa, Tominaga, Kazuhide, and Watatani, Shigeru. Field measurements on energy balance of an urban canyon in the summer season. *Energ. Buildings*, 15(3-4):417–423, January 1990. doi:10.1016/0378-7788(90)90016-c.

INDEX

Symbols

A

B

C

D

E

F

G

K

L

M

N

O

P

Q

T

U