**User Manual**

# CHAMPS-BES

*Program for Coupled Heat, Air, Moisture and Pollutant Simulation in Building Envelope Systems*

**Version 1, 2006**

BEESL - Building Energy and Environmental Systems Laboratory
Department of Mechanical and Aerospace Engineering
Syracuse University, NY

John Grunewald • mail:  john.grunewald@gmx.de
Andreas Nicolai • mail:  andreas.nicolai@gmx.net

# Table of Contents

# 1 Introduction

## 1.1 Welcome



This is the user help for the numerical simulation program CHAMPS-BES. This version of the user interface allows 1D, 2D and rotation symmetric 3D modeling of coupled heat and moisture transport.

This help is mainly a program reference. In order to access this help from within the program the user can press F1 in any dialog or view (activate the view first) or press the "Help" buttons. The user can also place the mouse cursor over toolbar buttons to get a hint about what the button is used for.

The CHAMPS-BES overview lists the general steps required to set up a simulation and refers to the appropriate help pages. Small step-by-step tutorials are included as well which explain common tasks needed to set up a simulation in CHAMPS-BES and some advances features.

This documentation is still not quite complete. If the user wants to contribute to it or report mistakes or missing explanations, please contact the authors:

John.Grunewald@gmx.de
Andreas.Nicolai@gmx.net

## 1.2 Authors & Copyright

**CHAMPS-BES Version 1 (2006) is Copyrighted by:**

Building Energy and Environmental Systems Laboratory (BEESL, http://beesl.syr.edu)
Department of Mechanical and Aerospace Engineering
L.C. Smith College of Engineering and Computer Science (http://www.ecs.syr.edu)
Syracuse University (http://www.syr.edu)
149 Link Hall
Syracuse, New York 13244-1240
(Tel. 315-443-2341, Fax 315-443-9099)

```
CHAMPS-BES Solver
Name:        CHAMPS-BES CVODE based solver (executable)
Description: External CHAMPS-BES Solver.
Author(s):   Andreas Nicolai <Andreas.Nicolai@gmx.net>
             John Grunewald <John.Grunewald@gmx.de>
License:     All rights reserved.
```

**CHAMPS-BES**

| | |
|---|---|
| Name: | CHAMPS-BES User Interface |
| Description: | The graphical user interface for creating input files for the CHAMPS-BES Solver. |
| Author(s): | Andreas Nicolai <Andreas.Nicolai@gmx.net> |
| | John Grunewald <John.Grunewald@gmx.de> |
| License: | All rights reserved. |

**<u>Libraries used in the user interface and solver</u>**

**libCC**

| | |
|---|---|
| Name: | Climate component library |
| Description: | Import and export of climate data files |
| Author(s): | Andreas Nicolai <Andreas.Nicolai@gmx.net> |
| | John Grunewald <John.Grunewald@gmx.de> |
| License: | Lesser GNU Public License |

**libIBK**

| | |
|---|---|
| Name: | IBK library |
| Description: | Collection of classes, data types and algorithms for scientific and engineering computer codes |
| Author(s): | Andreas Nicolai <Andreas.Nicolai@gmx.net> |
| | John Grunewald <John.Grunewald@gmx.de> |
| License: | Lesser GNU Public License |

**libDelphinOutputs**

| | |
|---|---|
| Name: | Delphin output file conversion library |
| Description: | Conversion algorithms for CHAMPS-BES output files into input files for different postprocessing tools (TecPlot, DELPHIN4 PostProc, etc.) |
| Author(s): | Andreas Nicolai <Andreas.Nicolai@gmx.net> |
| License: | Lesser GNU Public License |

**libPollutants**

| | |
|---|---|
| Name: | Pollutant and VOC properties |
| Description: | Provides essential properties and calculation algorithms for VOC/pollutant transport and storage to the solver |
| Author(s): | Andreas Nicolai <Andreas.Nicolai@gmx.net> |
| | John Grunewald <John.Grunewald@gmx.de> |
| License: | Lesser GNU Public License |

**libDelphin**

| | |
|---|---|
| Name: | Delphin library |
| Description: | Document Object Model for simulation data |
| Author(s): | Andreas Nicolai <Andreas.Nicolai@gmx.net> |
| | John Grunewald <John.Grunewald@gmx.de> |
| License: | All rights reserved. |

**libDelphinSolver**

| | |
|---|---|
| Name: | Delphin CVODE based solver library. |
| Description: | Solver implementation based on CVODE integrator. |
| Author(s): | Andreas Nicolai <Andreas.Nicolai@gmx.net> |
| | John Grunewald <John.Grunewald@gmx.de> |
| License: | All rights reserved. |

**libSundials**

```
Name:        SUNDIALS
Description: SUNDIALS (SUite of Nonlinear and
             DIfferential/ALgebraic equation Solvers)
Author(s):   Peter Brown, Aaron Collier, Keith Grant,
             Alan Hindmarsh, Steven Lee, Dan Reynolds,
             Radu Serban, Dan Shumaker, Carol Woodward
             Nonlinear Solvers and Differential Equations Project
             Center for Applied Scientific Computing
             Lawrence Livermore National Laboratory
             Livermore, California, USA
License:     Copyright (c) 2002, The Regents of the University of California.
             Produced at the Lawrence Livermore National Laboratory.
             Written by S.D. Cohen, A.C. Hindmarsh, R. Serban,
             D. Shumaker, and A.G. Taylor.
             UCRL-CODE-155951    (CVODE)
             UCRL-CODE-155950    (CVODES)
             UCRL-CODE-155952    (IDA)
             UCRL-CODE-155953    (KINSOL)
             All rights reserved.
```

The CVODE integrator is part of the **sundials** package.

**Plugins and Modules**

**PostProcDLL**
```
Name:        DELPHIN4 Post Processing
Description: The graphical post processing tool for output files
             generated with the DELPHIN4 and CHAMPS-BES Solvers.
Author(s):   Heiko Fechner <Fechner@abkfs2.arch.tu-dresden.de>
License:     All rights reserved.
```

**ProjectConverterDll**
```
Name:        Project conversion tool
Description: A project file converter that provides
             conversation of DELPHIN4 projects into CHAMPS-BES format.
Author(s):   John Grunewald <John.Grunewald@gmx.de>
             Andreas Nicolai <Andreas.Nicolai@gmx.net>
License:     All rights reserved.
```

**MaterialInspectorDLL**
```
Name:        Inspect and view material data
Description: Provides a convenient graphical user interface to
             inspect material properties and functions
Author(s):   Andreas Nicolai <Andreas.Nicolai@gmx.net>
License:     All rights reserved.
```

**OutputFileConverterDLL**
```
Name:        Convert CHAMPS-BES output files to other formats
Description: Provides a convenient graphical user interface to
             convert CHAMPS-BES output files to the native formats
             of post-processing tools and includes conversion
             algorithms for common output presentations
Author(s):   Andreas Nicolai <Andreas.Nicolai@gmx.net>
License:     All rights reserved.
```

## 1.3    About CHAMPS-BES

### *Funding support and ressources*

The CHAMPS-BES (Coupled Heat, Air, Moisture and Pollutant Simulation in Building

Envelope Systems) program has been developed during 2004-2006 with funding support from research grants from U.S. Environmental Protection Agency, U.S. Department of Energy, Syracuse Center of Excellence in Energy and Environmental Systems, EQS-STAR Center/New York State Office of Science, Technology and Academic Research, and Syracuse University.

It is an outcome of a joint effort between BEESL/Syracuse University, U.S.A. and the Institute for Building Climatology/University of Technology Dresden (TUD), Germany. The program has been built upon three foundations:

1.  Delphin 5, a heat and moisture (HAM) simulation program developed by Mr. Andreas Nicolai and Dr. John Grunewald, Building Energy and Environmental Systems Laboratory, Syracuse University.  The program has significant improvement over DELPHIN4 in terms of programming structure and user interface while the core functionality of DELPHIN4 is also available in Delphin 5. Most of the modules and algorithms in DELPHIN4 have been re-implemented in Delphin 5 under the new and improved program structure, which has also been adopted in CHAMPS-BES. The CHAMPS-BES development is based on the Delphin 5 source code dated summer 2004.

2.  DELPHIN4, a heat, air and moisture (HAM) simulation program developed by Dr. John Grunewald and Mr. Heiko Fechner of the Institute of Building Climatology, University of Technology Dresden. Most of the functionality available in DELPHIN4 has, been implemented in CHAMPS-BES.

3.  Research and developmental works conducted at BEESL/Syracuse University including:
    1)  A model of gaseous and adsorbed pollutant transport through building envelopes;
    2)  A one-way coupled model of airflow effects on the heat, moisture and pollutant transport;
    3)  A database of material properties for heat and moisture that include critically reviewed data from TUD, SU and literatures;
    4)  A database of material properties for volatile organic compounds (VOCs);
    5)  Testing and incorporation of an efficient solver for integration of ordinary differential equation systems (CVODE, developed by S.D. Cohen, A.C. Hindmarsh, R. Serbin, D. Shumaker, and A.G. Taylor of the Lawrence Livermore National Laboratory, The Regents of the University of California);
    6)  Small-scale environmental chamber experiments to determine the material properties, and to provide benchmark data for evaluating/validating the simulation model;
    7)  Full-scale experiments using a coupled indoor/outdoor environmental simulator (C-I/O-ES) to provide data for evaluating/validating the model for typical residential and commercial wall assemblies.

## *The core development team*

The CHAMPS-BES core development team in the years 2004 - 2006 included:

1)  Mr. Andreas Nicolai, Ph.D. Candidate and Research Assistant, BEESL/SU: has been the chief developer for the software.  He lead the effort on the software design and implementation, and has authored a majority of the codes in

CHAMPS-BES including the new solver and user interface. He is also the contact point for incorporation of new program modules, and taught other students/contributors how to write proper C++ module for incorporation into the overall CHAMPS-BES program.

2) Mr. Hui Li, Ph.D. Candidate and Research Assistant, BEESL/SU: Lead the effort on developing the one-way coupled airflow and VOC transport model, and implemented the module in C++ in an earlier testing version of CHAMPS-BES as part of his Ph.D. dissertation research. He also lead the effort in the full-scale chamber experiments to provide data for model validation, and conducted sensitivity analysis for CHAMPS-BES.

3) Mr. Mikael Salonvaara, Research Scientist and CHAMPS research Team Leader of BEESL/SU: Managed and coordinated numerical and experimental efforts and provided technical advises to the CHAMPS-BES development, drawing his many years of experience in developing the 3-D LATENITE-VTT program for heat and moisture simulations. He also leads the effort in determining material properties for VOC transport and storage.

4) Dr. John Grunewald, Research Associate Professor, BEESL/SU, and Junior Professor and Principle Researcher of Institute of Building Climatology at Dresden University of Technology: Implemented climate, boundary, field and contact conditions and the new VOC transport module in the CHAMPS-BES program. He also provided and maintains a DELPHIN4 - Delphin 5 / CHAMPS project conversion module and managed the relevant collaborative activities at TUD before joining SU, and provided technical advises to the CHAMPS-BES development, drawing his many years of experience in developing the DELPHIN4 program for HAM simulations.

5) Dr. Jianshun "Jensen" S. Zhang, Professor and Director, BEESL/SU: As the Faculty Advisor and Principle Investigator, he is responsible for the overall scientific integrity and quality of the research program. He also provided technical advises on VOC emissions and transport modeling and experiments, and database development.

## *Further contributors*

Other contributors to the CHAMPS-BES program included:

1) Dr. Mark Bomberg, Research Professor and hygrothermal research Team Leader of BEESL/SU: Provided valuable advises on material characterization and testing for the determination of model parameters in CHAMPS-BES;

2) Mr. Marcin Pazera, Ph.D. Candidates and Research Assistant, BEESL/SU: provided valuable suggestion on material data collection/testing, and standardization for inputting to CHAMPS-BES. He also developed/setup testing methods and apparatus at SU for measuring material properties for moisture transport, and provided bench mark experimental data for model evaluations;

3) Mr. John MacKenzie, M.S. student and Research Assistant, BEESL/SU: setup and conducted experiments to measure heat conductivities of materials under moist conditions, and provided benchmark data for model evaluations;

4) Ms. Jing Xu, Ph.D. student and Research Assistant, BEESL/SU: assisted in collecting and entering the material property data file for VOC;

5) Mr. Heiko Fechner, Researcher of the Institute for Building Climatology at the University of Technology Dresden: shared his experience in developing DELPHIN4, and provided relevant DELPHIN4 program modules/library functions for inclusion in CHAMPS-BES, developed the postprocessing module and assisted

in the post processor implementation in CHAMPS-BES;

6) Mr. Gregor Scheffler, Ph.D. Candidate and Research Assistant of the Institute for Building Climatology at the Technical University of Dresden: provided substantial input to material characterization, developed the material functions generation module and assisted its implementation in CHAMPS-BES.

7) Other faculty and graduate students who conducted beta testing and provided helpful suggestions for new features, functionality or user interface.

### *Contact*

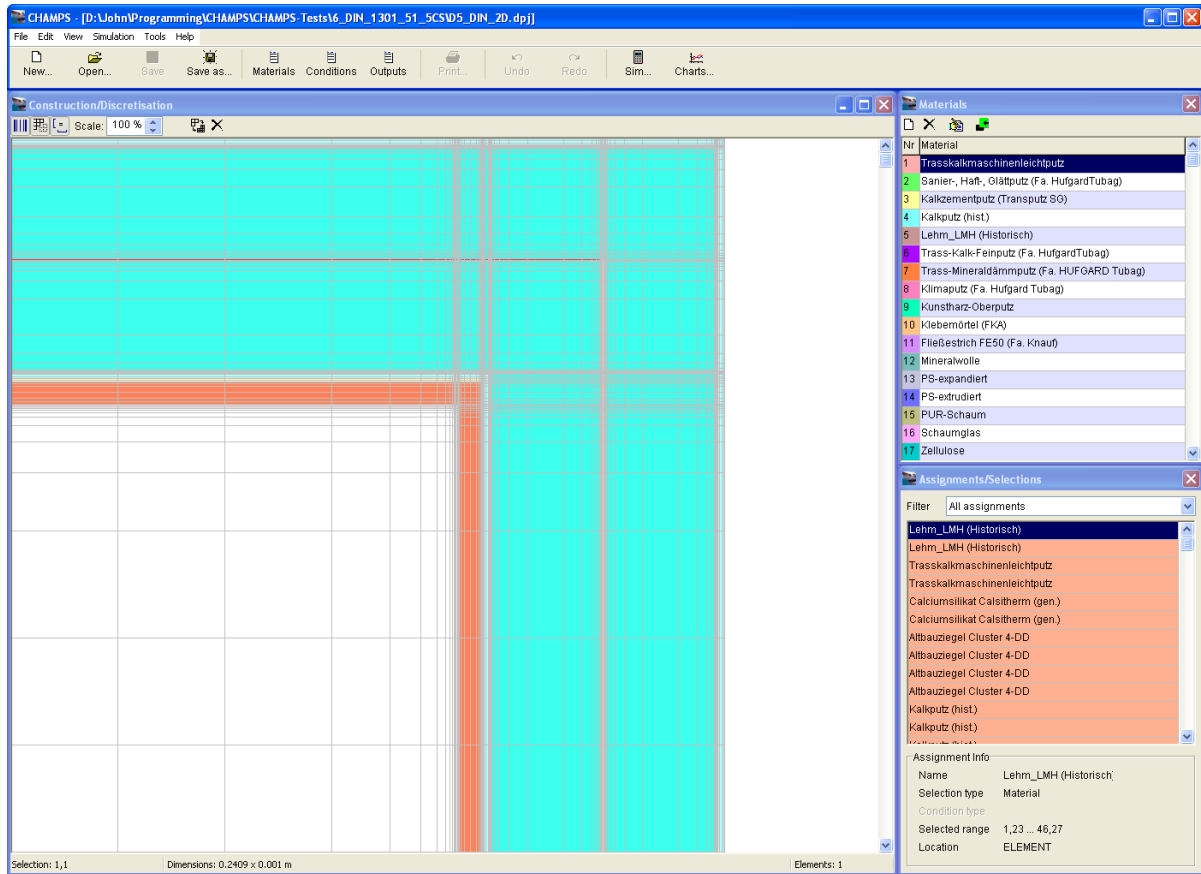For more information, please contact:
Andreas Nicolai (anicolai@syr.edu),
Mikael Salonvaara (hmsalonv@syr.edu),
John Grunewald (jgrunewa@syr.edu), or
Jensen Zhang (jszhang@syr.edu).

August 2006

# 2 Getting started

## 2.1 Overview

This chapter contains an overview of the work with CHAMPS-BES. Here is a screenshot of a typical layout in the CHAMPS-BES User Interface.

**Typical layout of the CHAMPS-BES User Interface.**

Modeling a building envelope or construction detail with CHAMPS-BES is quite simple. Here's a summary of what the user needs to define for a complete project/simulation:

- Geometry/Construction
- Materials
- Initial conditions
- Boundary conditions
- Outputs
- Modeling options
- Solver settings

New CHAMPS-BES projects can be generated either by using pre-defined templates or by setting up a simulation from scratch.

The recommended order of setting up a simulation is given below (the user can use this as a check list):

1. setup construction (Construction/Discretization view)
2. import materials (Materials view and Material Import Dialog)
3. assign materials to construction
4. discretize grid
5. specify initial conditions (Conditions view and Initial Condition Edit Dialog)

6.    specify climate conditions and boundary conditions ([Climate Condition Edit Dialog](#) and [Boundary Condition Edit Dialog](#))
7.    assign conditions to construction
8.    define output grids and formats ([Output Grid Edit Dialog](#) and [Output Format Edit Dialog](#))
9.    define output files ([Outputs view](#) and [Output File Edit Dialog](#))
10.   assign output files to the construction (hereby specifying where outputs are needed)
11.   set modeling options ([Modeling/Simulation Setup Dialog](#))
12.   set solver options if the user wants to tweak the default settings ([Solver Setup Dialog](#))
13.   and finally run the simulation ([Simulation Run Dialog](#))
14.   after simulation is completed, start post-processing

At any point during the modeling process, if the user should by mistake do something the user didn't want to, just use the Undo/Redo feature to correct it.

Alternatively to the built-in output conversion feature, the user can also use the Output File Converter to convert the simulation output to input files for various post processing tools.


## 2.2      Simple step-by-step examples

This chapter contains small tutorials that will teach the user to setup CHAMPS-BES simulation models from scratch.

**[Tutorial 1 - Adsorption experiment](#)**
This tutorial explains how to set up a simple 1D absorption experiment starting from a default project template. In this example a vertical 1D structure is modeled that takes up moisture from the bottom. This tutorial teaches the basic steps necessary to set up a coupled heat and moisture transport simulation with CHAMPS-BES.

### 2.2.1    Tutorial 1 - Water uptake experiment

This tutorial explains how to set up a simple 1D absorption experiment starting from a default project template. In this example a vertical 1D structure is modeled that takes up moisture from the bottom. This tutorial covers the basic steps necessary to set up a coupled heat and moisture transport simulation with CHAMPS-BES.

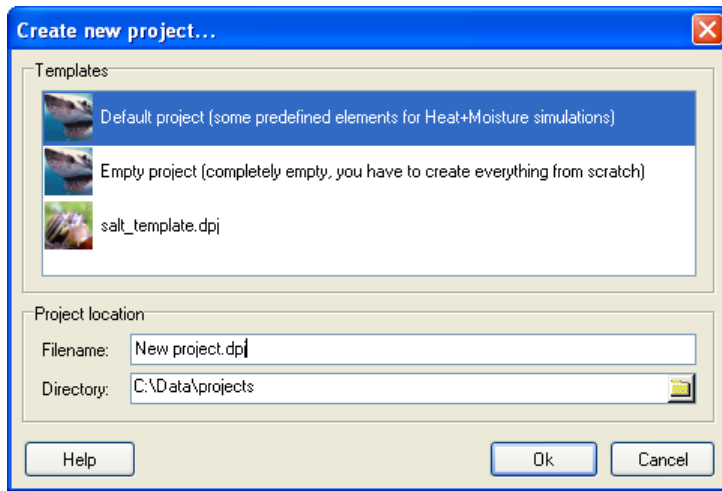#### Creating the project
Start the CHAMPS-BES user interface. Initially all that is displayed is the main menu and toolbar at the top of the screen.
Create a new project using the command: `File->New...` The [New Project Dialog](#) is shown.

**New project dialog.**

Select the default project template (the first), enter a filename (e.g. absorption.dpj), select a project directory and confirm the dialog. Now the user interface shows several views of your empty project. The user may rearrange the views later and adjust the screen layout at will.

### Setting up the grid

Open the context-menu of the construction view (right-click on the empty construction grid) and select `setup construction grid` The Setup new grid dialog is opened. Leave everything like it is and confirm the dialog. This creates a structure with one material layer. Execute the `Change width/height...` command to open the element geometry edit dialog
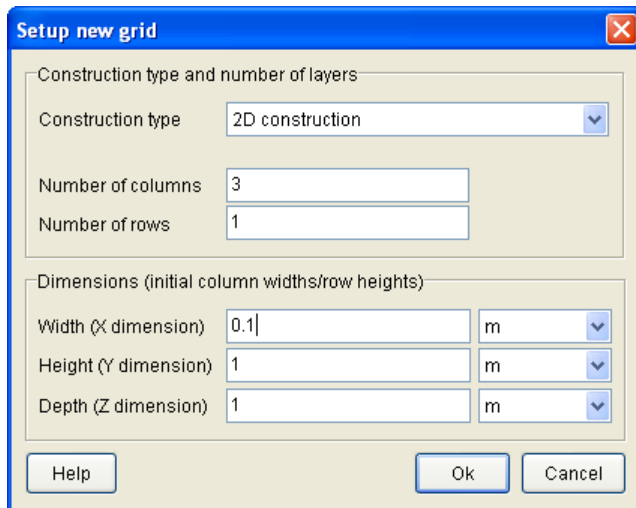


**Edit construction dialog.**

and set the *height* of the grid element/material layer to 100 mm.

### Import and assign material

Now move over to the materials view and execute the `New...` (material) command (either via context-menu or button). In the Material Import Dialog select a material file such as

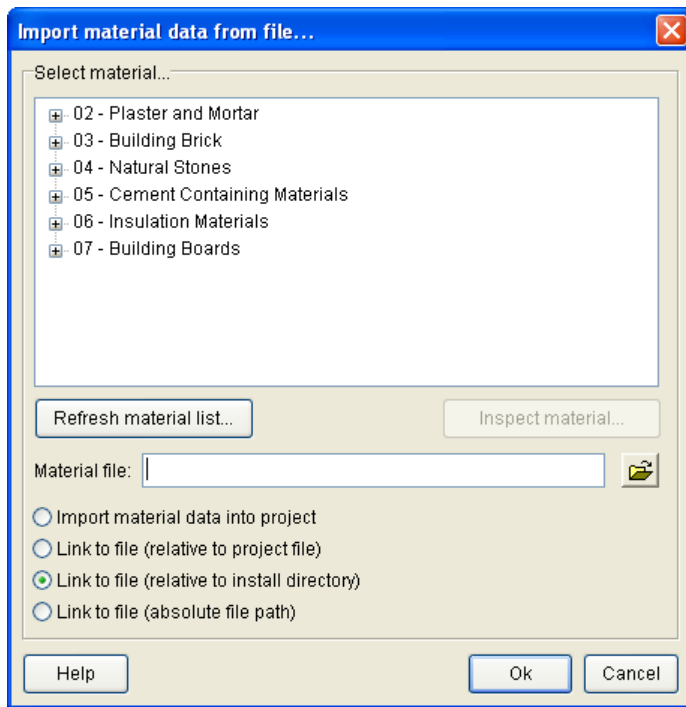the 'Brick Wienerberger', select the radio button 'Import material data into project' and confirm the dialog.



**Material import dialog.**

Depending on your needs the user may want to create a link to the material file instead (refer to the Material Import Dialog documentation for available options).

After importing the material it will show up in the materials list view. In order to assign a material the user first has to select the appropriate cells in the construction view and then use the `Assign material` command, by pressing the green assignment button in the toolbar:
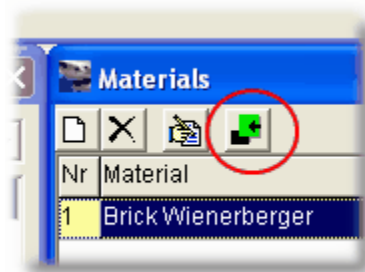


Since the only material layer in the grid is already selected, the user can simply execute the `Assign material` command and the material will be assigned to the grid. The user can now also see an assignment entry in the assignment view.

Hint:   To assign something to the construction grid there is a context-menu.

**Define and assign initial conditions**
For a coupled heat and moisture simulation the user will need an initial condition for the

---

energy balance and one for the moisture balance. For this example we'll create temperature and relative humidity initial conditions. Select the "Initial condition" tab in the conditions view
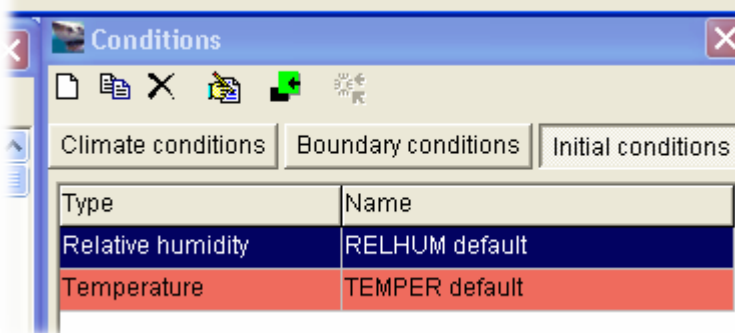


and execute the **New...** command (from context menu or using the toolbar button). In the Initial Condition Edit Dialog type "Initial temperature" as identification name, specify 20°C and confirm the dialog. Then create another initial condition, select "Relative Humidity" as type, name it "Initial RH" and set it to 5 % relative humidity.

Now mark both initial conditions and execute the **Assign Conditions** command (the green button with the arrow again). They will be assigned to the currently selected material layer (actually the only grid element that can be selected in the construction grid right now).

> Hint: If accidentally something was done that the user didn't want to do, simply use the UNDO button to reverse the change.

**Define and assign boundary conditions**

For a very simple model of an absorption experiment, we simply assign a water contact boundary condition. Select the "Boundary condition" tab in the conditions view and create a new boundary condition (command **New...** again). In the Boundary Condition Edit Dialog the user needs to specify several things. Set the identification name to "bottom water contact". Set the type to "Water contact" and leave the kind as "Surface value". For this boundary condition the user also needs to assign a "Water content" climate condition. Click on the "**Create new...**" button in the "Water content" line to open the Climate Condition Edit Dialog. Name the climate condition "water contact" and specify a constant moisture content. For an absorption experiment the user wants to specify the final equilibrium moisture content after the absorption experiment. For the "Brick Wienerberger" that will be 0.319 m3/m3 (use the ***CHAMPS-BES Material Inspector*** to check material properties). Confirm the climate condition dialog to get back to the boundary condition dialog. The user will now see that the newly created climate condition is assigned as source for "Water content" data. Similarly create and assign a climate condition for the water temperature (name it "water temperature" or something like this).

> Hint: After completing the tutorial, try varying the water temperature and see how the moisture and temperature profile is influenced when different water temperatures are used.

The user can now assign the boundary condition (execute the **Assign Conditions**

command). However, this time the user will be prompted to specify a location for the boundary condition. Select the bottom side only (deselect all others) and confirm the assignment dialog.

### Define and assign outputs

The model is now almost ready for calculation. Still, the user needs to define what kind of outputs the user would like to obtain. The default project template, which was used in this tutorial, already contains output grid and format definitions. All the user has to create are some output files that will contain your outputs.

> Hint:   Read the section Output overview to get a better understanding of how the output system of CHAMPS-BES works.

The user can create new output files in the outputs view. These output files define *how* the outputs are done (via an output format), *when* outputs are done (by using a specific output grid) and into which file the data is written. Later one assigns them to the grid and defines *where* they are done (or *where* the data for the outputs is taken from). First create a temperature field output (with command `New...` ).
Select the 'Temperature' output format and select the default output grid. The filename "temp_field.out" is automatically given. Confirm the dialog and create another output file with "Moisture mass" as output format. Finally we want an output file with the total water mass in the structure, so create an output file with "Water mass integral" as output format.

Finally these output files need to be assigned (the simulation will still run without them, but one wouldn't get any results). Select all three output files and assign them (`Assign Output Definition` command).

### Discretization of the structure

After everything has been assigned the structure still needs to be discretized. In the context menu of the construction grid select `Discretize->Discretize selected rows...` to open the Discretization Dialog. For this example we use equidistant spacing, but feel free to try out the variable discretization option. Change the number of elements to 100 or so (more then 50 will usually give very smooth absorption profiles) and confirm the dialog. The discretization is then shown in the construction form. If one doesn't like the result use UNDO and try again.

### Simulation settings

The last thing to do before running the simulation is to specify some modeling options. Open the Modeling/Simulation Setup Dialog using the main menu ( `Simulation->Modeling/Simulation settings...`). For absorption experiments it is recommand to use the Kirchhoff potential as driving potential, select it in the liquid transport potential combo box. Also change the simulation duration to 1 day. After you've confirmed the dialog you are ready to start the simulation.

### Starting a simulation

Open the Simulation Run Dialog by pressing F9 or using the main menu (`Simulation->Run Simulation...`). Select the internal solver. Click on "Run simulation" and in a few seconds the simulation will be completed.
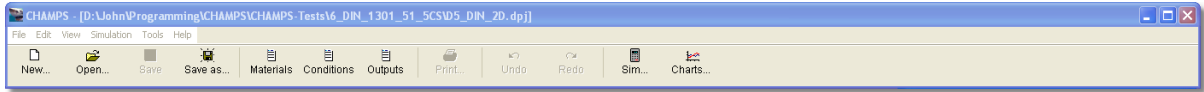
### Post-Processing

After the simulation is complete, you can start the post-processing selection dialog.

# 3 Program reference

## 3.1 Main program window

The main program window of CHAMPS-BES is actually only a small window at the top of the screen with the main menu and the toolbar buttons:



Once a project is opened the corresponding views are shown. The toolbar buttons allow quick access to most frequently used commands from the main menu. The Undo/Redo buttons correspond to the menu entries in the Edit menu and are enabled whenever an undo/redo-action is possible.

## 3.2 Undo/Redo feature

The Undo/Redo feature is included in CHAMPS-BES to simplify the handling of project data and case variations. So it becomes particularly useful if you want to try out, for example, several different grid discretizations. It is also useful as safeguard against accidental key presses (if you delete assignments and just pressed Del once too often).

All actions done in views (and thus directly on the project data) can be undone. After an "undo" you can "redo" the change again. There is no limitation  (except your computer's memory) to the number of undo/redo actions in the program.

> Note:  The Undo-Feature in CHAMPS-BES only works for views. When you change something in a (modal) dialog and confirm the dialog by pressing "Ok", you cannot "Undo" this change with the Undo-button. However, the Discretization-Dialog is an exception and also allows to Undo the changes.

## 3.3 Views

### 3.3.1 Something general about views

Views are used in the CHAMPS-BES user interface to show a subset of the project data. A view, such as the construction view or conditions view always shows the current content in the project file. There are no confirmation or cancel buttons for views. If you close them, you simple remove them from the desktop (they become hidden). You can turn them on or off using the menu commands in the *View* menu or use the key shortcuts (F4 to F8).

Whatever is shown in a view is part of the current project/simulation. Therefore changes you do in a view (e.g. deleting a material or boundary condition) means that the content of the project is directly changed as well. In order to prevent accidental loss of data, CHAMPS-BES includes an Undo/Redo feature.

Views can be arranged on the desktop however you like it. Once you have placed the views in your favourite layout, you can use the menu option `View->Layout->Save view layout` to store the layout. The layout will be read and setup for you when you start

CHAMPS-BES in the future (or if you manually restore the layout using the appropriate menu option).

### 3.3.2 Construction/Discretization view

This view shows the construction you are modeling and allows to:

- specify the geometry and assign materials
- initial conditions, boundary conditions, output formats etc.
- discretize the grid

Also, the construction view can be used to check assignments and definitions. Whenever you click on a material, a condition or output or any assignment in the assignments view, the appropriate assigned region in the construction view will be **marked** by changing the color. This is not to be mistaken with a **Selection**, that is shown in dark blue.
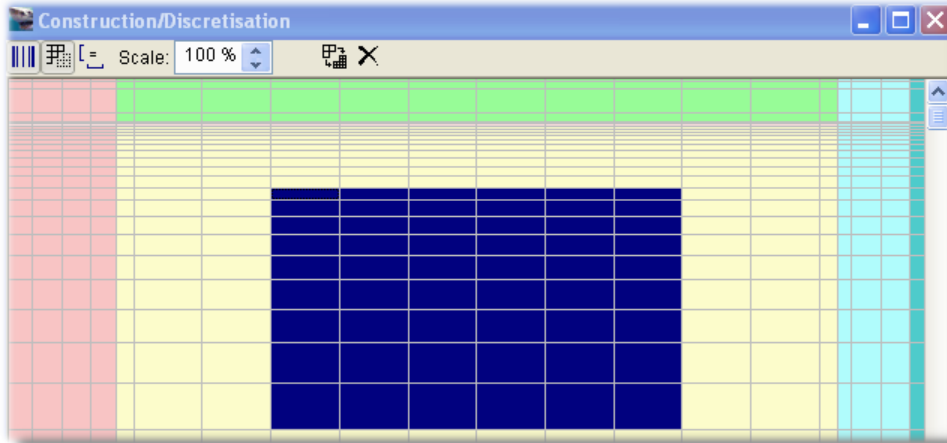


**The blue area in the middle is a selection, the elements on the right with a slightly darker blue than the other elements are highlighted.**

### *The construction grid*
The construction grid shows your current model of the construction. You can select single elements or ranges by clicking and dragging the mouse across. Typical keys like `Home`, `End`, `CTRL+A` also help to quickly select cells in the grid. Whenever you select an element or range of elements you also select the appropriate column(s) and row(s). This is important for all grid related commands like changing width and height or discretization of rows and columns.

In order to define the geometry of your structure you first need to setup the grid and define the widths/heights of your columns and rows. Then you can assign materials. Later material assignments overlay earlier assignments and complex geometries can be defined fairly easy like in the following example:

| grid without material assignments | after assigning material with range: 0 0 2 3 | after assigning VOID with range: 1 1 2 2 | after assigning material with range: 2 2 2 2 |
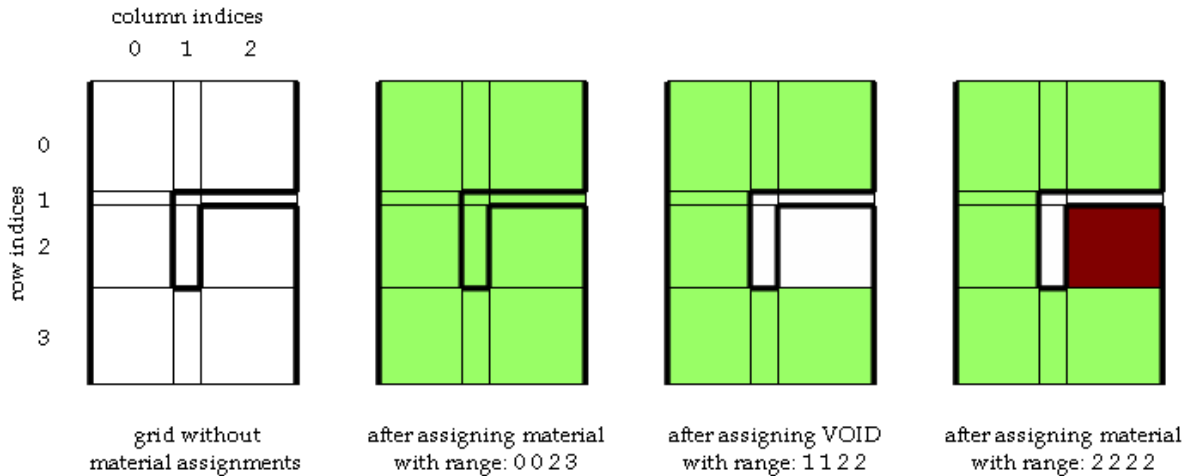
### The construction view toolbar buttons
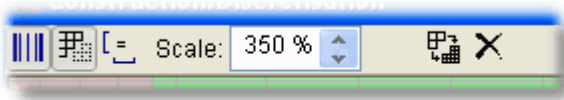


From left to right the toolbar buttons have the following functions:

1. Equidistant drawing on/off - cells are drawn scaled to their widths/heights or all with the same width/height, useful for assigning conditions to small elements at the outside of the grid
2. Show grid lines on/off
3. Same aspect ratio on/off - if turned on, both axis (x and y axis) are scaled with the same scaling factor, if turned off, the axis are scaled independently and fitted into the construction view
4. Scale/Zoom percentage: This option is only active if "Same aspect ratio" is turned on. The view is then drawn to scale based on the given percentage
5. Auto-Discretization: this toolbar button starts the auto-discretization dialog
6. Remove discretization: removes the current discretization

### The construction view context menu

The construction grid has a context-menu (right-click on the construction grid). Several **commands/actions** are available which can be executed from the context-menu.

## *Steps to setup a new construction/geometry for simulation*

When setting up a construction you need to do the following steps:

1. use the command 'Setup construction grid' to initially setup the construction, this opens the Setup new grid dialog
2. specify the dimensions of the different layers using the command 'Change width/height...' which opens the Change dimensions of rows and columns dialog
3. assign materials to specify the geometry of the construction (you may also do this from the Materials view)
4. assign initial and boundary conditions (you may also do this from the Conditions view)

5. assign outputs (you my also do this from the Outputs view)
6. discretize the structure by selecting single rows and colums or whole ranges and executing the command 'Discretize selected columns...' or 'Discretize selected rows...' which will open the Discretization Dialog or use the Auto-Discretization feature

### 3.3.3 Materials view

This view contains a list of all materials included in the project. You can add (import), edit and remove materials. For adding or editing of material definition the Material Import Dialog is used.



**The materials view with the current project materials.**

Hint: Clicking on a material will highlight all cells in the construction view that have this material assigned.

If you want to assign a material to the construction you can either use the context-menu options of the construction view or use the green assignment button or context-menu command from the materials view. For the latter select first a range of elements in the construction view, then select the material you want to assign and execute the assign command (click on the button or use the context menu). You may only assign one material at a time.

Later assignments of materials overwrite earlier assignments. You can use this feature to easily compose complex geometries as in the example for the construction view.

Note: The 'VOID' material cannot be assigned directly from the materials view. You need to use the context-menu of the construction form.

### 3.3.4 Conditions view

This view shows all defined conditions (climate, boundary and initial conditions). You can create new, edit or remove these definitions. Use the toolbar buttons or the context menu to access the available commands. The conditions are colored depending on the condition type to
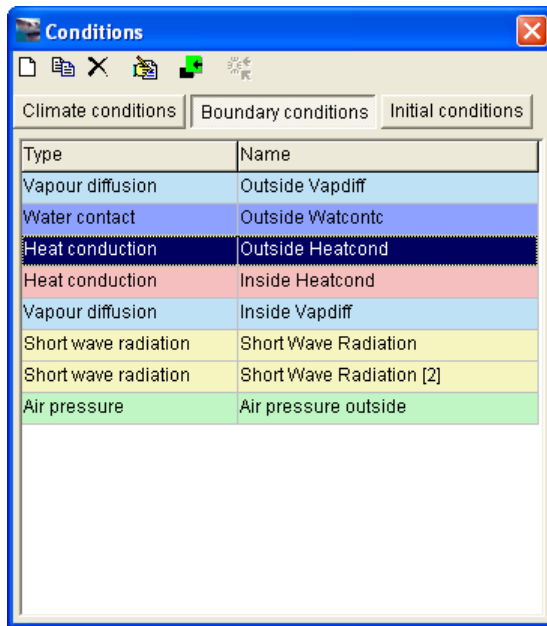
> Hint: Clicking on a boundary or initial condition will highlight the cells in the construction grid, where the condition was assigned to.

The tabs are used to switch between the different conditions lists: climatic conditions, boundary conditions and initial conditions. All lists have context menus. For all commands in these context menus buttons are available in the toolbars. Hold the mouse cursor briefly over a button to get a tool tip that explains what the button/command is used for.
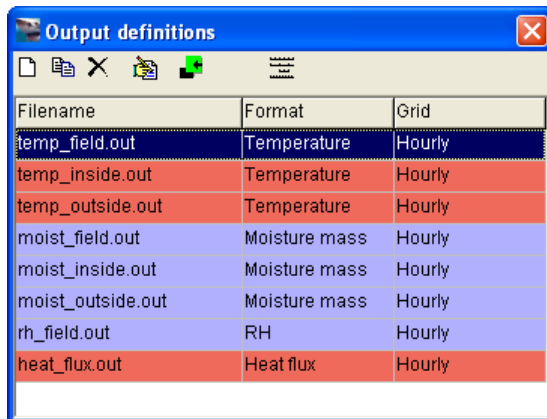
From the conditions view you can open the Climate Condition Edit Dialog, the Boundary Condition Edit Dialog and the Initial Condition Edit Dialog.

If you want to assign a condition to the construction you can either use the context-menu options of the construction view or use the assign button/command from the condition view. First select a range of elements in the construction view, then select the conditions you want to assign and execute the assign command (click on the green button with the arrow or use the context menu).

> Note: You can only assign Boundary Conditions and Initial Conditions, since Climate Conditions are implicitly assigned with each Boundary Condition.

## 3.3.5 Outputs view

This view shows all defined output files. You can create new, edit or remove output file definitions. Use the toolbar buttons or the context menu to access the available commands. You can also open the Output Grid Edit Dialog, using the corresponding toolbar button/context menu option.

If you want to assign an output file to the construction you can either use the context-menu options of the construction view or use the assign button/command from the outputs view. First select a range of elements in the construction view, then select the output file you want to assign and execute the assign command (click on the button or use the context menu).

### 3.3.6    Assignments/Selections view

This view lists all assignments/selections made so far. Assignments basically link definitions (materials, conditions, outputs) with geometry (elements and sides). The filter combo box at the top of the view can be used to select a particular type of assignment.
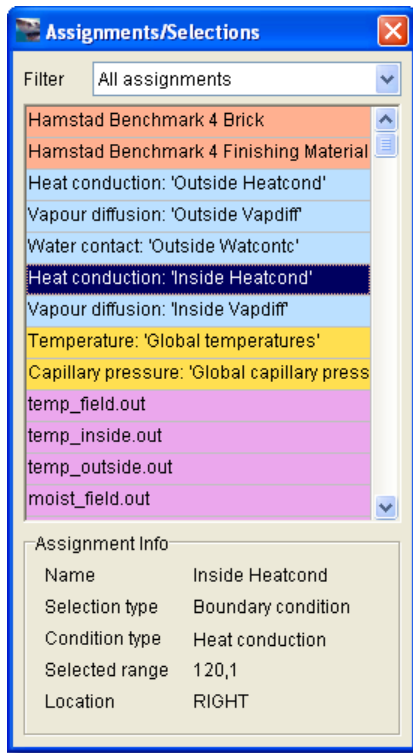
You may not edit an assignment directly. Simple remove the assignment you want to change (via option in context menu or "Delete" key) and create a new assignment instead.

Note: Clicking on an assignment will highlight the assigned cells in the construction view. However, when the assignment ranges of material assignments overlap, this will result in an unexpected behaviour: clicking on the earlier assignment will highlight all cells of this assignment, even though some of those cells may have been later assigned to a different material and therefore show a different color.

## 3.4 Dialogs

### 3.4.1 Project Information Dialog

In this dialog you can enter some text describing the current project. At the top the dates for creation of the project and last saving are shown.

### 3.4.2 New Project Dialog

In this dialog you can select one of the available project templates that you would like to start with. The dialog offers a choice of default templates (typically a completely empty project and a default project with some typical predefined entries) or user defined templates.
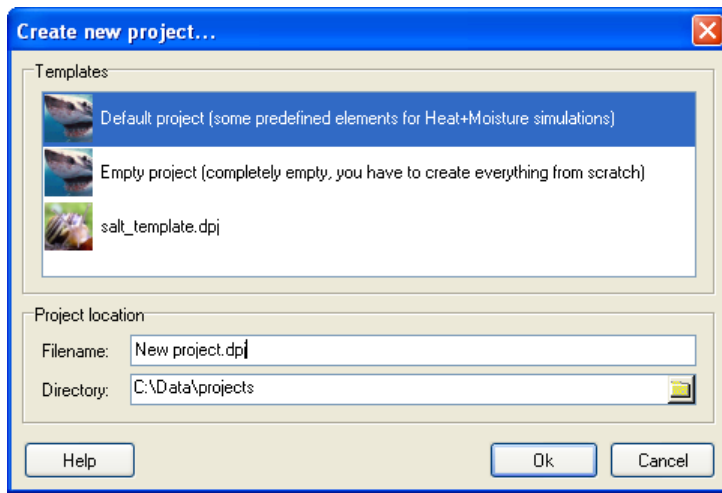
> Hint: To add your own templates, simply save a project file into the **templates** subdirectory in the installation directory.

## 3.4.3 Geometry/Construction

### 3.4.3.1 Setup new grid

This dialog is used to setup the grid for a new construction.
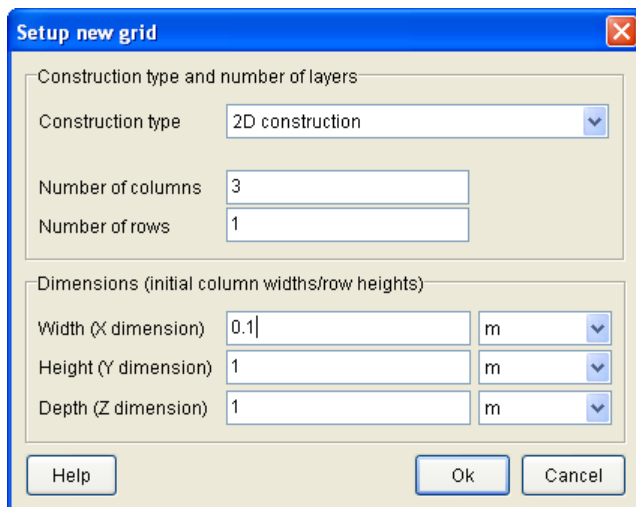


You can specify the number of layers in x and y direction (columns and rows respectively) as well as the initial dimensions of the layers. Once the initial grid is created you can adjust the dimensions of the different layers by using the Change dimensions dialog.

### *Construction type and number of layers*

#### Construction type
You can specify whether it is a planar 2D construction or a 3D rotation symmetric

system. 1D constructions are just a special case of 2D constructions and the solver detects 1D structures automatically (when only one column or row is defined). Gravity always acts in negative y direction.
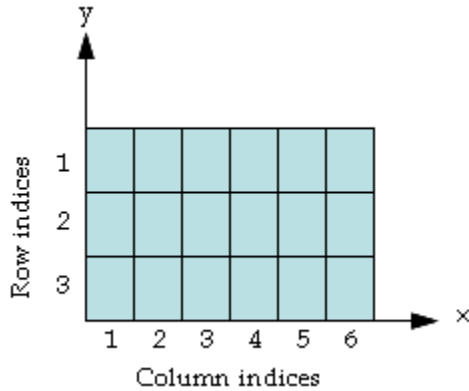


**Definition of coordinate system for plane 2D structures and column/row numbering.**

In rotation symmetric systems the y axis is always the rotation axis. The left-most column is closest to the rotation axis.
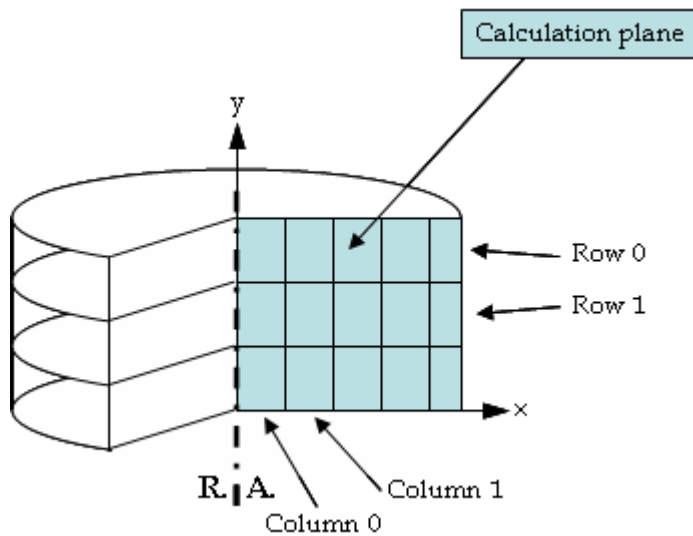


**Rotation symmetric structure setup**

### Number of columns/Number of rows
The number of columns and rows specify how many layers there are in x and y direction. You may later add or remove layers using the discretisation options. However, in this dialog the principle number of layers for the initial grid setup can be specified. If either the number of columns or the number of rows is one, the simulation will automatically switch to 1D calculation mode and use a more efficient solution algorithm.

### Dimensions (initial column widths/row heights)
Here you can enter the initial dimensions of all layers.

Note: The dimension in z direction only affects the output of plane 2D structures and is meaningless for 3D rotation symmetric structures. See section Output overview for details.

### 3.4.3.2 Change dimensions of rows and columns

In this dialog the widths and heights of the selected rows and columns can be changed. Normally you will want to change the dimensions of the layers right after a new grid was created and before the discretization is applied. This dialog is also useful if you want to adjust the dimensions later. Assignments are not affected by a change of dimensions, since the number of columns/rows remains unchanged.

Depending on your selection one or both input fields may be blank. That means that the widths or heights of the selected columns/rows is different.

After you confirm the dialog, all rows and columns in the selection will have the dimensions you have specified. If any of the fields is left blank/empty the dimension remains unchanged.

### 3.4.3.3 Discretization Dialog

This dialog allows to discretize the selected rows and columns using either an equidistant grid or a variable grid. In the dialog either a horizontal or vertical discretization can be done, but not both at the same time. The discretization dialog can also be used to insert a new layer into an existing structure just by discretizing an element (row/column) into 2 and afterwards adjusting the dimensions of the elements.

The total height/width of the selected range will be preserved regardless how the discretization is performed. Assignment ranges will be adjusted.

Note: If more then one row/column is selected for discretization, the total height/width available will be discretized. The original layer geometry will be lost! For example, if you have two columns (0.1 m and 0.2 m) and you discretize them using 10 equidistant layers you will not preserve the layer thickness of 0.1 m or 0.2 m, but instead get three layers which will sum up to 0.09 m and 7 layers which sum up to 0.21 m.

#### *Equidistant discretization*

Equidistant means that all rows/columns (elements) have the same thickness. You have two possibilities:

1. choose the number of elements and let the thickness be calculated automatically
2. enter the thickness of the elements and let the number of elements be calculated

Note: Be aware that depending on the entered/calculated layer thicknesses and potential rounding errors the last element in the range may have to be corrected in order to match the size of the selected range. This leads sometimes to a last element, that is substantially smaller than the others and may cause unwanted numerical difficulties. The dialog will try to detect that and warn if such a situation is encountered.

---

### *Discretization tips*

The discretization commands are quite powerful. You can use them for almost anything that is related to geometry modification that requires changing the number of rows and columns. Some examples are given below:

#### How to add column/row

Select a column/row where you would like to add another column/row besides and select the appropriate discretization command and split the column/row in half (equidistant, 2 columns/rows). Now only the widths/heights have to be adjusted.

#### How to remove a column/row

Select the column/row in question and a neighboring column/row which has the same material (or also no material) assigned. If there is no neighboring column/row with the same material remove material assignments first and assign them back later.
Use the command 'Merge selected columns/rows..." to reduce the selection to a single column/row. Adjust the width/height and you're done.

### 3.4.3.4 Auto-Discretization Dialog

Use this dialog to specify the parameters for the auto-discretization. It is recommended to begin with a coarse grid and refine it later if necessary.
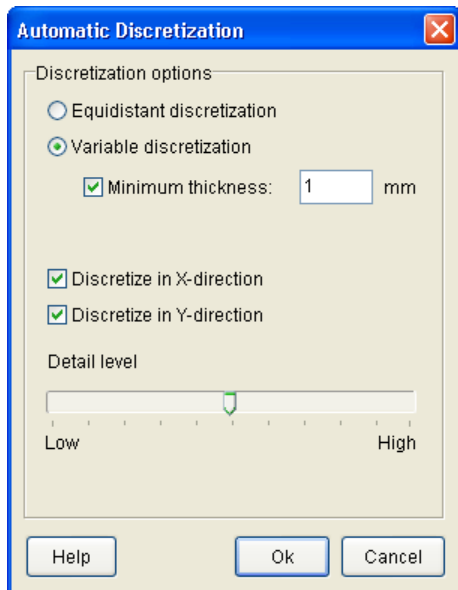


**Auto-Discretization dialog**

You can optionally specify the minimum thickness of the elements when using variable discretization. If you do not specify this thickness, the algorithm calculates it based on the detail level.

### 3.4.3.5 Select Side Dialog

This dialog is used to define which side or sides of an element are used for flux-type assignment. Since you only can select ranges of *elements*, you need to define the actual location where you would like to assign a boundary condition or flux output format.

### *Assigning boundary conditions*

Whenever you assign a boundary condition, either using the *Assign* action in the conditions view or the context menu in the construction view, the location of the boundary you wanted to select is automatically determined and the Select Side Dialog is opened where you can confirm this location.

Note that **boundary conditions** assigned to sides **inside** the structure (with both a left *and* right, or top *and* bottom material element) **are ignored** in the calculation.

### *Assigning flux output formats*

Flux output formats are assigned similar to boundary conditions. But while boundary conditions really are only considered at the interfaces between material and environment, you can also obtain flux outputs from inside the structure. For instance if you are interested in how the flow field is changing in a 2D structure, select the structure and assign all sides (left, right, top and bottom), as a result you will get a 2D flow field with x and y flow vector components. Similarly for 1D structures, you may select a range of elements and obtain flux outputs of several sides.

### 3.4.4 Material Import Dialog

In this dialog you can select and import a material for the simulation. Material data required for simulation is currently stored in material files. Every material file describes a different material. Select a material file, choose one of your import options and confirm the import.

### *Import options*

CHAMPS-BES project files can directly include material data. Alternatively the project file may only link to material files (essentially store the filename) and the material data is read during initialization of the solver. You have several options available:

**Import material data into project**
When selecting this option the material data will be read from file and included permanently into the project file. The advantage is that wherever you copy the project file the material data will always be available. Also future changes in the material data will not affect older projects, so this is the recommended option for simulations that may need to be repeated after a while. However, as a drawback the project files get bigger.

**Link to file (relative to project file)**
The material data is not included into the project and project files stay small. The file name is saved relatively to the project. This option is recommended if your project directory contains a subdirectory with material data which is always copied alongside the project files.

**Link to file (relative to install directory)**
The material data is not included into the project and project files stay small. The file name is saved relatively to the installation directory of CHAMPS-BES. This option is useful if default material data (installed along with CHAMPS-BES) is used in projects.

**Link to file (absolute file path)**
This option include the absolute file path to the material file. It is recommended if material data is sourced from an external data source (i.e. server) or different drive.

If you use any of the link options you must ensure that the material files are available when the solver is run (otherwise it will abort with an error message).

## *Inspect Material*

If the material inspector module is installed, you may use this feature to inspect the properties and material functions of the selected material. Refer to the help file of the material inspector (available via the Help button or F1, when the material inspector window is open).

## 3.4.5 Conditions

### 3.4.5.1 Climate Condition Edit Dialog

In this dialog you can define climatic data used to specify boundary conditions. A climate condition basically defines what kind of climatic condition is present in the environment of the structure whereas a boundary condition defines *how* the climate affects the structure (see Boundary Condition Edit Dialog).
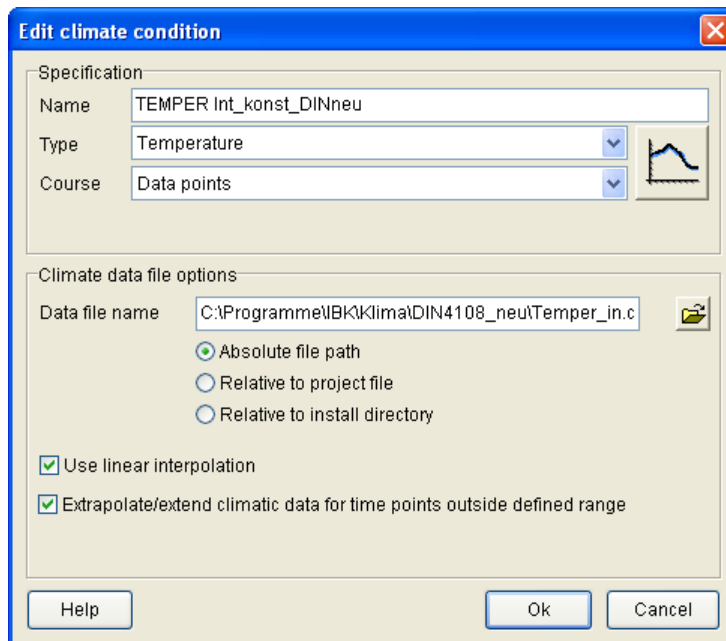


**Climate condition edit dialog.**

The climate condition edit dialog contains following GUI-elements:

**Name**
A name for identification of the climatic data record (content of the dialog). The name must be unique.

**Type**

The user can choose from a number of predefined climatic conditions. See Climate conditions for a table of available conditions.

### Course
Here the user may choose among four choices. Constant condition, sinus wave, double sinus wave and climate data file.

### Parameters
The content of this block depends on the choice of the **course**.

*Constant*
A constant value is used.

*Sinus wave*
You can define a sinusoidal course of the climate condition. Useful for daily or yearly changing conditions.

*Double sinus wave*
You can define two sinusoidal courses which are superimposed. Useful for yearly changing conditions with an overlaying daily cycle.

*Data points*
This is one of the most important climate condition kind since it is used for realistic and measured climate data. At first you need to specify the actual climate data file. These files can be of type CCD (common climate data format) or CLI (CLImate data file). These formats are explained in section Climate data file formats.

Climate data is given as time point - value pairs. Sometimes the solver needs to obtain climate data in between two data points. This can be done by assuming step-wise constant data (e.g. hourly constant data), where always the last time point is taken, or alternatively by linear interpolation between the points. Use the option Use linear interpolation to control this behaviour. The Climate data file formats section gives more information about the  available options.

The shifting value option is used to add an offset to the data values in the climate data file.

Climate data is never embedded into the project file. Instead the filename of the climate data file is stored. Here you have three choices:

Absolute file path
The absolute filename (including driver letter) is stored. This is usefull if climate data always resides in a fixed directory or if climate data files are stored on a different location (e.g. on a server or other drive).

Relative to project file
The relative file path with respect to the project file path is stored. This is useful if climatic data is project specific and is copied along with the project file.

Relative to install directory
The relative file path with respect to the installation directory is stored. This is useful if default climatic data installed along with CHAMPS-BES is used.

---

**3.4.5.2 Boundary Condition Edit Dialog**

In this dialog you can define a boundary condition for a structure. Boundary conditions define *how* a certain climate acts on the surface of the structure. Every boundary condition references one or more climate conditions that are the actual sources of climate data (see Climate Condition Edit Dialog). Once you have defined boundary conditions you can assign them to the boundary elements.
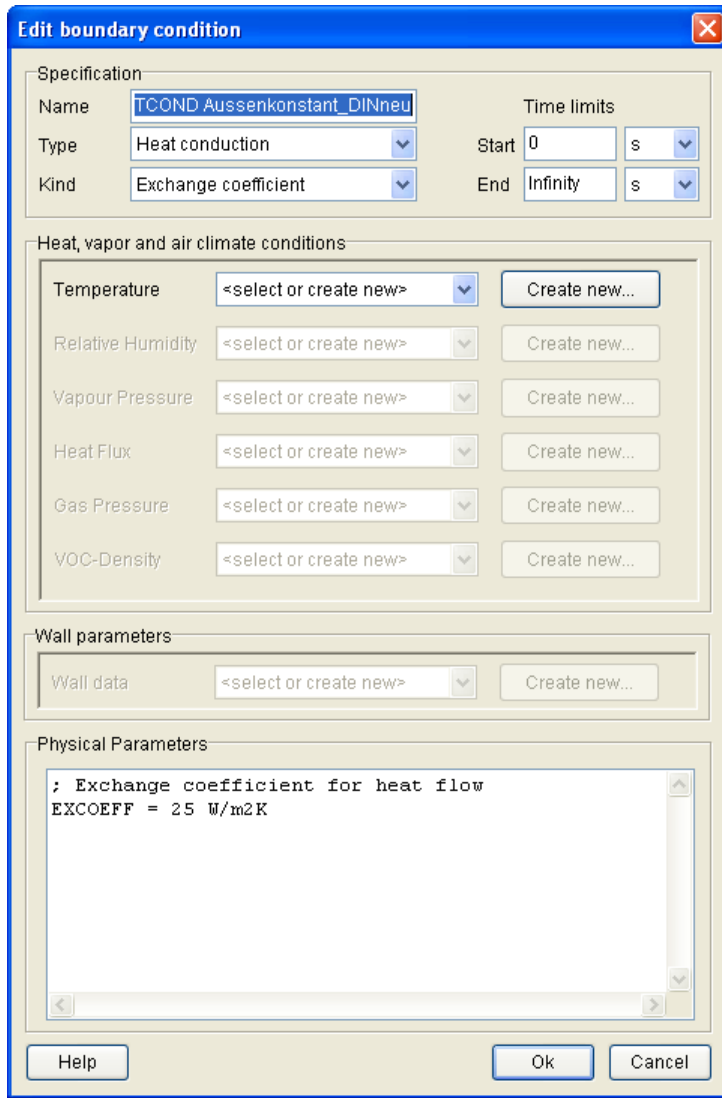


**Boundary condition edit dialog.**

A boundary condition is defined through its type (e.g. heat conduction or vapor diffusion) and its kind (how it is specified). The boundary condition edit dialog contains following GUI-elements:

**Name**
A name for identification of the boundary data record (content of the dialog). The name must be unique.

**Type**

The user can choose from a number of predefined boundary conditions. See Boundary conditions for a table of available conditions.

**Kind**

Here the user may choose among several choices. The choice depends on the chosen **Type** of the boundary condition.

**Climate conditions**

Depending on how the boundary condition is defined, the user needs to provide additional climatic conditions. Note that sometimes you may assign alternative climate conditions or more then actually needed for the BC. The user can select from already defined climate conditions or create a new climate condition if required.

**Wall parameters**

Wall data (orientation, inclination, latitude) are required for some boundary conditions. The wall parameters panel is enabled or disabled accordingly. Similar to climate conditions, the user can select from already defined wall data or create new wall data if required.

**Physical parameters**

Most of the boundary conditions require physical parameters like exchange coefficients or adsorption coefficients. They are specified by plain text using a comment line, a keyword, a value and a physical unit.

### 3.4.5.3   Initial Condition Edit Dialog

In this dialog the user can define or edit an initial condition. Define a unique identification name, select a condition type and enter the initial value. Note that the *combination* of condition type and identification name must be unique within a CHAMPS-BES project. Therefore it is possible to define initial conditions [Global, Temperature] and [Global, Relative Humidity] at the same time. After you defined your initial conditions you need to assign them to the structure.

Depending on the choice of balance equations to be calculated you need to specify initial conditions for all balance equations and in *all* elements.
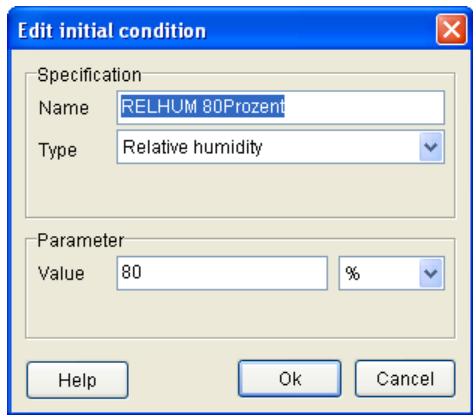
**Initial condition edit dialog.**

An initial condition is defined through its type (e.g. temperature or relative humidity) and its kind (how it is specified). The initial condition edit dialog contains following GUI-elements:

**Name**

A name for identification of the initial data record (content of the dialog). The name must be unique.

**Type**

The user can choose from a number of predefined initial conditions. See Initial conditions for a table of available conditions.

**Parameter**

The parameter is specified by value and a physical unit.

## 3.4.6    Outputs

### 3.4.6.1    Output overview

Outputs in CHAMPS-BES can be very flexibly defined. This is done with the following definitions:

- **Output grids** define the output time points (when and how many outputs are made)
- **Output formats** define which quantity is monitored and how the data is written to file
- **Output files** finally define the actual output files that are created and combine a specific output format and grid

Last but not least you need to define which elements/sides you would like to monitor. Any combination of selections is possible by assigning output file definitions to the discretized construction.

Normally output grids and formats need to be defined only once, thus allowing fast creation of output file definitions.

### 3.4.6.2 Output Grid Edit Dialog

In this dialog you can define an output grid. Output grids are used to define time periods during your simulation (see simulation time frame) in which outputs are made. You can define as many output grids as necessary. Usually you will have one or more global output grids, running from start to end of the simulation, but with different output time steps.

Also, every output grid can have several intervals. For instance, if you would like to monitor a variable more closely at the begin of the simulation but less frequently (e.g. to save diskspace and speed up postprocessing) at the end of the simulation.

The following example shows two output grids with several intervals each:
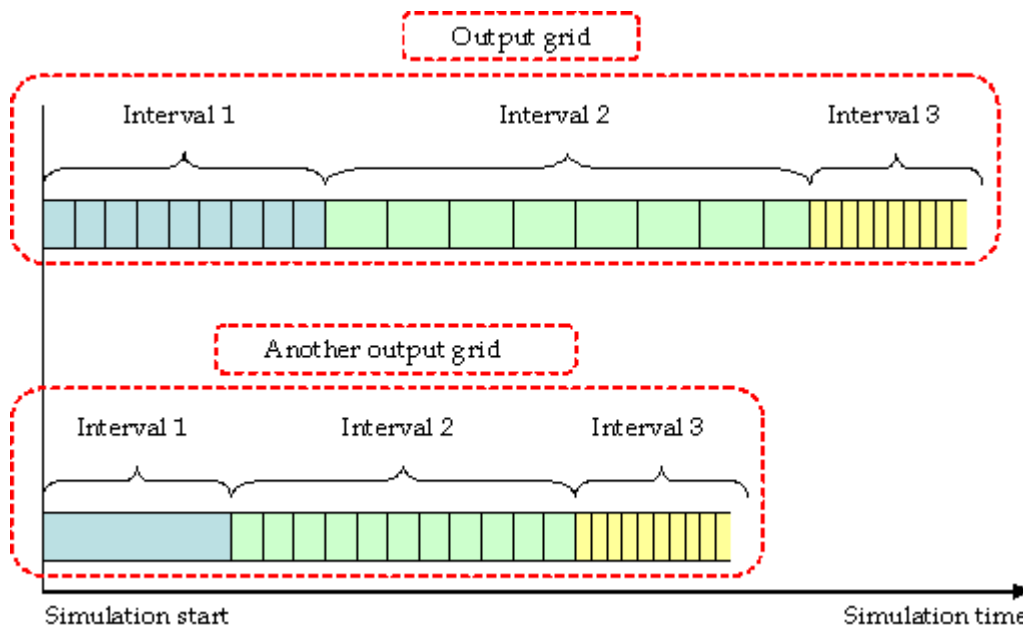


Each output interval is defined via its duration and output step size. The number of outputs is then calculated. If the last output interval should not be restricted in time and continue until the simulation ends, a duration of 0 (zero) must be entered.

For instance, if you would like to have outputs at regular time points (e.g. hourly values) you only need one output interval. Enter 0 as duration (infinite interval) and "1 h" as duration.

### 3.4.6.3 Output Format Edit Dialog

In this dialog you can define output formats. Output formats define, *which quantity* is monitored and *how* the values are recorded and in *which format* they are stored.

#### *Name*

The name of an output format definition must be unique.

## *Kind and format of output data*

You can obtain outputs for field variables and flux variables:

**Field variables** are all state variables and quantities that are defined for each element in the construction.
**Fluxes** are monitored over sides of elements (between elements or over element boundaries to the environment).

Select either field or flux outputs and depending on the selected type you have the choice of different quantities you may want to monitor.

**What format is used if multiple elements/sides are selected?**
If several elements/sides become monitored using this output format you can select one of three options. You can write:

Single values for each element or side
For each selected element or side a value is stored in the output file. Use this option if you, for instance, want to monitor temperature fields or the distribution of moisture throughout a structure

Averaged values of elements or sides
A weighted average of the quantity in all selected elements/sides is calculated and stored as single value. Use this option if you, for instance, want to monitor the mean temperature or relative humidity in a section of the structure.

Integrated values in space (multiply unit with m3)
The quantity in all selected elements/sides is integrated (i.e. quantity x volume for each element) and stored as single value. Use this option for monitoring accumulation of a quantity in the structure (or parts of it). Typically you may want to monitor the total moisture mass in critical regions. Integration in space results in a modified unit. It gets multiplied with m3 (cubic meters).

**What to do with intermediate values?**
If the output time steps are larger then calculation time steps, intermediate values are calculated for all monitored quantities. You have now three options to process these intermediate values:

Ignore, write values as calculated at output times
If selected, all intermediately calculated values are ignored. This is the default and mostly used setting.

Average values in time (mean value in output step)
If selected, all values calculated since the last output step are averaged (using a weighted average over time) and the resulting value is considered representative for the whole last output interval. Use this if you have many calculation time steps in between your output time points and you don't want to miss any significant effects (e.g. you have

Integrate values in time (multiply unit with s)
If selected, all values calculated since the last output step are integrated in time (i.e. quantity at calculation time point x time step taken). Use this to option to obtain integral quantities, e.g. to monitor the total heat loss through a wall.

NOTE: As of now the time types "Average" and "Integral" are not completely implemented into the solver yet!

**Output units**
You can further select the units used in CHAMPS-BES output files. Note that the units of the monitored quantities do not incorporate unit changes due to integration in space or time.

**Example**
Assume you would like to monitor the total moisture mass in a part of the structure. The output format would be a "FIELD" format, monitored quantity would be "Total moisture mass", and the space type would be "Integral". As value unit you would select "kg/m3", but keep in mind that the integration will actually yield a total moisture mass in "kg"!

### 3.4.6.4   Output File Edit Dialog

In this dialog you can define output files.

#### *Filename*

The filename of an output file definition must be unique (same filenames would result in the solver attempting to write into the same output file). Also, the filename of on output format must not contain any absolute paths or directories. The extension .out is automatically appended to the filename if not included already.

#### *Format*

Select here the output format to be used for the file.

#### *Output Grid*

Select here the output grid you would like to use for this output.

## 3.4.7   Modeling/Simulation Setup Dialog

This dialog is used to set the main modeling options and setup the simulation. Refer to the CHAMPS Modeling section for details of the physical model.

#### *Balance equations*
You can select any combination of balance equations given.

Energy Balance
If you select energy balance only the solver will only calculate heat transfer, assuming a dry material. Use this to precalculate/estimate transport in large 2D structures because it is much faster then fully coupled simulations.

Moisture Balance
If you select moisture balance only the solver will calculate isothermal moisture transport. The uniform temperature inside the construction is assumed to be 20°C.

Energy and Moisture Balance
If you select both balance equations the solver will calculate fully coupled heat and moisture transport (this is the default option).


## *Liquid Transport Modeling*

Several options are available to specify the modeling of liquid moisture transport in the structure. By default, transport within materials is calculated using a Kirchhoff-Potential function and liquid flow over interfaces is calculated using liquid conductivity and a liquid pressure gradient.

### Preferred Driving Potential for Liquid Moisture Flow

Typically, the liquid transport function for a material will be either liquid conductivity or moisture diffusivity. In the case that both functions are given, you can select a preferred function to be used in the simulation. When the material is initialized, the transport function is numerically integrated to obtain a Kirchhoff-Potential function, which is then used to calculate liquid fluxes within materials.

Note: When a Kirchhoff-Potential function is already specified in the material, this setting is ignored and the given function is taken directly.

Liquid pressure gradient
This driving potential is physically correct, continuous over material boundaries and allows calculation under non-isobaric conditions.

Moisture content gradient
When the moisture content gradient is selected as driving potential the liquid fluxes are calculated using moisture diffusivities. The moisture diffusivity is never directly used to calculate the liquid fluxes. For internal flows the Kirchhoff-Potential is used (integrated from the moisture diffusivity) and for fluxes over material interfaces/boundaries a liquid conductivity is used that is calculated by a finite difference quotient from the Kirchhoff-Potential.

### Always use liquid conductivity

If this option is checked, the liquid conductivity will also be used for calculation of internal fluxes. The Kirchhoff-Potential is then ignored. The default is off.

### Gravity

You can choose to include gravity effects in your calculation (for liquid transport and boyancy effects). The default is off.


## *Simulation time*

These settings control the duration of the simulation and how the simulation time frame is related to real time (needed for syncronization with climate data). For most simulations you only need to specify the duration.

### Simulation Time Frame

Inside the solver the "simulation time" is defined. It runs from $t = 0$ (i.e. the begin of the simulation) until $t = t\_end$. A simulation time year has always 365 days, so leap years are excluded. That is the reason why you cannot select the 29th of February as a start date.

The simulation time frame can be located anywhere in the real time, so you may chose to let the simulation start on 1st of September, 12 am (midnight) and let it run for 6 months (0.5 a). If you have climate data defined for a year (starting on 1st of January and running for 365 days) the solver automatically translates the simulation time (t = 0 at 1st of September) into real time date and picks the climate entry.

### 3.4.8   Solver Setup Dialog

This dialog is used to set the solver options.

#### *General solver options*
These options are used by all solvers that can be run from the CHAMPS-BES user interface.

<u>Initial time step</u>
This is the time step the simulation is started with. The default value is 10e-3.

<u>Maximum time step</u>
You can limit the maximum time step to control the truncation error or stability of the solver. The default value is 30 minutes.

<u>Lower time step limit</u>
If the solver reduces time steps due to convergence problems or because the local error exceeds the tolerances, this time step may become sometimes very small. This result in an apparent "freezing" of the solver and is considered an unnormal behaviour. You can set a lower time step limit that, if the actual time step becomes smaller then this limit, causes the solver to abort. The default value is 10e-7.

#### *CVODE specific parameters*
The parameters in this section are specific for CVODE based solvers (the only one currently supported by the CHAMPS-BES user interface).

**Tolerances**
You can control the tolerances used by the solver in various ways. The solver uses absolute and relative tolerances in the local error test. It is possible to use a single absolute tolerance for all balance equations ("Global absolute tolerance"). However, since energy and moisture mass are given in values of very different orders of magnitude, you may also specify absolute tolerances for each balance equation alone ("Equation specific absolute tolerance").

All tolerance values have one in common. The lower the values, the faster calculates the solver. This is especially important for 2D simulations. Recommended procedure is to start with large tolerances (AbsTol: 1e-3, RelTol: 5e-4) and increase them, if convergence failures or over/undershot values are calculated (see numeric problems).

<u>AbsTol: Global</u>
This is the absolute tolerance used for all balance equations. The default value is 1e-4.

<u>AbsTol: Internal energy</u>
This is the absolute tolerance used for the energy balance equation alone. The default

value is 1.

AbsTol: Moisture mass
This is the absolute tolerance used for the moisture mass balance equation alone. The default value is 1e-4.

Relative tolerance
This is the relative tolerance used for all balance equations in the local error test. The relative tolerance is most critical for stability of solution and can also significantly increase/decrease calculation times. The default value is 1e-4.


## *Other options*
Several other options can be used to control the behaviour of the solver.

Iterations for HM decomposition
During the calculation of the divergences in all control volumes, the intensive state variables must be calculated (this is called decomposition). This is an iterative algorithm where the mass of water vapour is calculated with increasing accuracy. If you require very accurate calculation in the low moisture content region, set the number of iterations to 5. If you set the number to 1 (the default), the first approximation is kept. Note that increasing the number of iterations significantly slows down the simulation and for most problems one iteration already yields good results.

Maximum order
CVODE is a variable-order variable-step multistep solver and controls its order based on time step, local error estimates and other criteria (see CVODE user guide). The maximum order that can be used by CVODE is 5. You may, however, set an upper limit for the order that CVODE may choose. This may help solving problems when the polynomial approximation of higher orders causes the solver to "swing" (see numeric problems). The default maximum order is 5.

Maximum steps
Set here the maximum number of steps the solver is allowed to take between output time points. If the solver takes very small time steps and needs a long time to reach the next output time point you can enforce an early return from the integration function and obtain intermediate outputs. The default is 10000.

Use "one step" mode
CVODE can be run in one step mode. If you select this, the CVODE solver will take one internal step and return control to the CHAMPS-BES solver which can perform additional operations (like integration of intermediate values) using the same time steps as chosen by the CVODE solver. This feature is mainly used for special applications and debugging. The default is "off".

Write monitors file
If you enable this the solver will write diagnostic output and solver statistics like: number of iterations, number of Jacobian calculatinos, number of matrix decompositions, number of calls to f, time steps used etc. If selected these diagnostic outputs are written at every output time step. The default is off.

Write "one step" monitors
You can only select this option if "one step mode" and "write monitors files" is enabled.

In addition to writing the diagnostics at output time points, these outputs are now written at each internal solver step. This is a feature to find problems with simulations, that "don't calculate" or only very slowly. The default is off.

Optimized Jacoby generation
This option is only available for the calculation of coupled heat and moisture. If checked, a user defined and optimized algorithm for generating the Jacobian will be used instead of the default implementation provided by the CVODE solver. This may greatly increase simulation speed. The default is off (since it only applies to HM simulations).

## 3.4.9    Simulation Run Dialog

In this dialog you can set the final run options and start a simulation.

### *Solver types*
Currently you can use either the internal CVODE solver or call the external solver executable. The button "Solver parameters" opens the [Solver Setup Dialog](#) where you can adjust solver/numerics options.

If the external solver aborts and the simulation window disappears either consult the error logfiles or run the internal solver to see the error message.

### *Run options*
Here you can adjust some options that are not part of the simulation data and are used to control the behaviour of the solver.

Verbose level
Depending on the verbose level, the solver prints more or less informative (and diagnostic) output.

Test init
If this option is checked the solver runs through the initialization and stops afterwards. Use this to test a simulation setup without actually starting it.

Restart
If you stopped a lengthy calculation (by aborting the solver) you may restart it with this option.

Wait
After every output step the solver takes (or every step if run in one-step-mode) a command prompt is shown. Press Enter to continue calculation to next step. Type in commands to alter solver options (not fully implemented yet). Type the command 'quit' to stop calculation without aborting it.

### *Command line*
If you select the external solver, this command line corresponds to the run options you have selected. You can copy this command line into a batch file and run the solver later. Substitute $(INSTALL_DIR) with the directory the solver executable is installed to.

### 3.4.10  Post-Processing Start Dialog

This dialog shows a list of all created output files. You can select one or more output files and start one of the available post-processing programs. Right now TecPlot and the DELPHIN4 Post-Processing are supported. However, some output formats are only supported by one post-processing tool.

Once you selected the files and clicked one of the post-processing tool buttons, the native output files are converted to a format usable by the post-processing tools and displayed accordingly. If you require better control over the output file conversion, use the external ***CHAMPS-BES Output File Converter*** tool.

# 4     CHAMPS Modeling

## *General introductory remarks*

CHAMPS is an acronym for coupled heat, air, moisture, pollutant simulation or alternatively for heat, air, moisture, pollutants and salt transport, which is the underlying physical model of the CHAMPS-BES code.

CHAMPS modeling comprises the description of fluxes in the calculation domain or in the field (between volume elements including material interfaces) and at the boundary (between volume elements and exterior or interior rooms) by physical models. Also included are models for  storage processes like adsorption, desorption and release.

The description of field flux and storage is usually done by a set of balance equations. These balance equations are of parabolic type and there is no closed analytical solution for this problem. The transport coefficients depend usually highly on the variables of state. They can vary over several orders of magnitude, as for example the liquid water conductivity. Therefore, a numerical solution is the only choice to solve a coupled CHAMPS problem. The numerical solution is done by semi-discretization in space (using a finite/control volume method) and subsequent integration in time.

By defining a set of balance equations one establishes a framework for the solution of a variety of CHAMPS problems. To solve the problem, also the definition of conditions (initial conditions, boundary conditions etc.) is required. Lastly the numerical solution is applied to solve the system of equations.

## 4.1     Balance equations

## *Notation rules*

The CHAMPS notation rules will follow the notation styles already used in literature by major authors (BEAR & BACHMAT, JISCHA …).
1)  Uppercase characters for absolute quantities (i.e. entropy, enthalpy, momentum, energy),  exception: mass
2)  Lowercase characters for specific quantities (per mass unit)

3) Volume-related quantities are denoted as densities (i.e. internal energy density, mass density)
4) Superscripts denote the related quantity, subscripts the respective phase (or all phases in case of a Representative Elementary Volume – REV), superscript can be omitted in case of an intrinsic density of the phase itself
5) Concentrations are always related to the mass of the respective phase (intrinsic)
6) Flux generally relates to REAs (Representative Elementary Areas), REA subscript omitted
7) Examples:

| | | | |
|---|---|---|---|
| $m$ | **absolute** mass | kg | |
| $U$ | **absolute** internal energy | J | |
| $u$ | **specific** internal energy | J/kg | (related to the mass of a phase) |
| $\rho_g^{m_v}$ | **intrinsic** vapor mass density | kg/m$^3$ | (vapor mass per volume of gas phase) |
| $\rho_{REV}^{m_v}$ | **volumetric** vapor mass density | kg/m$^3$ | (vapor mass per REV) |
| $\rho_g$ | **intrinsic** gas mass density | kg/m$^3$ | (gas mass per volume of gas phase) |
| $\rho_{REV}^{m_g}$ | **volumetric** gas mass density | kg/m$^3$ | (gas mass per REV) |
| $c_g^{m_v}$ | **intrinsic** vapor mass concentration | kg/kg | (vapor mass per mass of gas phase) |
| $c_l^{M_s}$ | **intrinsic** molal salt concentration | mol/kg | (salt amount per mass of liquid phase) |
| $c_l^{m_{voc}}$ | **intrinsic** VOC mass concentration | kg/kg | (VOC mass per mass of liquid phase) |
| $\dot{j}_{conv}^{m_g}$ | convective gas mass flux | kg/m$^2$s | (gas mass flux per REA) |
| $\dot{j}_{diff}^{m_v}$ | diffusive vapor mass flux | kg/m$^2$s | (vapor mass flux per REA) |

### 4.1.1   Moisture mass balance

#### *Assumptions*
- No ice
- Evaporation equilibrium (Kelvin-equation)
- Diffusive liquid water mass flow, dispersive liquid water and dispersive water vapor mass flow is negligible
- Pressure equilibrium between all phases (this leads to the capillary pressure as well defined quantity and to the water retention characteristics)
- Distortions of the solid material matrix are negligible

#### *Moisture mass balance can be written as*

$$\frac{\partial}{\partial t}\rho_{REV}^{m_{w+v}} = -\frac{\partial}{\partial x}\left[ j_{conv}^{m_w} + j_{conv}^{m_v} + j_{diff}^{m_v} \right] + \sigma_{REV}^{m_{w+v}}$$

| | | |
|---|---|---|
| $\rho_{REV}^{m_{w+v}}$ | Moisture (liquid+vapor) density in reference volume | $kg/m^3$ |
| $j_{conv}^{m_w}$ | Convective liquid (capillary) water flux | $kg/m^2 s$ |
| $j_{conv}^{m_v}$ | Convective water vapor flux | $kg/m^2 s$ |
| $j_{diff}^{m_v}$ | Diffusive water vapor flux | $kg/m^2 s$ |
| $\sigma_{REV}^{m_{w+v}}$ | Moisture sources/sinks in reference volume | $kg/m^3 s$ |

## Convective liquid (capillary) water flux

$$j_{conv}^{m_w} = c_l^{m_w} j_{conv}^{m_l}$$

| | | |
|---|---|---|
| $c_l^{m_w} = 1 - c_l^{m_{voc}}$ | Water mass concentration in the liquid phase | $kg/kg$ |
| $c_l^{m_{voc}}$ | VOC mass concentration in the liquid phase | $kg/kg$ |
| $j_{conv}^{m_l} = -K_l\left[ \frac{\partial p_l}{\partial x} + \rho_l g \right]$ | Convective flux of liquid phase | $kg/m^2 s$ |
| $K_l$ | Liquid water conductivity | $s$ |
| $p_l$ | Liquid water pressure | $Pa$ |
| $\rho_l$ | Intrinsic density of liquid phase | $kg/m^3$ |
| $g$ | Gravity constant | $m/s^2$ |

Note the difference between $m_l$ (mass of liquid phase) and $m_w$ (mass of pure liquid water). Calculation of the liquid phase conductivity from liquid water conductivity can be done by multiplication with concentration and temperature dependent viscosity ratio.

$$K_l(\theta_l, T, c_l^{m_{voc}}) = \frac{\left[ \eta_{voc}(T) - \eta_w(T) \right]c_l^{m_{voc}} + \eta_w(T)}{\eta_w(T_{ref})} \cdot K_l(\theta_l, T_{ref}, c_l^{m_{voc}} = 0)$$

| | |
|---|---|
| $\eta_w(T)$ | Viscosity of pure liquid water |
| $\eta_{voc}(T)$ | Viscosity of VOC |
| $K_l(\theta_l, T_{ref}, c_l^{m_{voc}} = 0)$ | Measured liquid water conductivity under reference conditions |

## Convective water vapor flux

$$j_{conv}^{m_v} = c_g^{m_v} j_{conv}^{m_g}$$

$$c_g^{m_v} = \frac{p_v}{p_a + p_v} \cdot \frac{R_a}{R_v}$$      *Water vapor mass concentration in the gas phase*      *kg/kg*

$p_v$      *Water vapor pressure in gas phase*      *Pa*

$p_a$      *Air pressure in gas phase*      *Pa*

$R_a$      *Gas constant of air*      *J/kgK*

$R_v$      *Gas constant of water vapor*      *J/kgK*

$j_{conv}^{m_g}$      *Convective flux of the gas phase (defined below)*      *$kg/m^2s$*

### *Diffusive water vapor flux*

$$j_{diff}^{m_v} = -\frac{D_{v,air}(T)}{\mu R_v T} f(\theta_g) \frac{\partial p_v}{\partial x}$$

$D_{v,air}(T)$      *Vapor diffusivity in free air (SCHIRMERs equation)*      *$m^2/s$*

$\mu$      *Vapor diffusion resistance factor*      *—*

     *(accounts for tortuosity and reduction of diffusion cross section)*

$f(\theta_g)$      *Function of volume fraction of gas phase*      *—*

     *(accounts for moisture dependency, $f(\theta_g) = \theta_g$ in many cases)*

$\theta_g = \theta_{por} - \theta_l$      *Volume fraction of gas phase*      *$m^3/m^3$*

$\theta_{por}$      *Porosity of material*      *$m^3/m^3$*

$\theta_l$      *Volume fraction of the liquid phase*      *$m^3/m^3$*

### *Moisture sources and sinks*

No sources and sinks implemented yet.

## 4.1.2    Air mass balance

### *Assumptions*
- Dry air consists of O2, N2, and other minor components, excluding VOC and H2O (water vapor)
- The gas phase consists of dry air, water vapor and VOC.
- The convective flow of the gas phase is considered separately from heat, moisture and pollutant transport, i.e. heat, moisture and pollutant transport has no effect on airflow but buoyancy
- Only laminar gas flow is considered

• Kinetic energy of moving phases is negligible
• Gas constant of dry air can be used for moist air

### The air mass balance reads as

$$\frac{\partial}{\partial t}\rho_{REV}^{m_a} = -\frac{\partial}{\partial x}\left[j_{conv}^{m_a}\right] + \sigma_{REV}^{m_a}$$

| | | |
|---|---|---|
| $\rho_{REV}^{m_a}$ | Air mass density in reference volume | $kg/m^3$ |
| $j_{conv}^{m_a}$ | Convective air mass flux | $kg/m^2 s$ |
| $\sigma_{REV}^{m_a}$ | Air sources/sinks in reference volume | $kg/m^3 s$ |

### Convective air mass flux

$$j_{conv}^{m_a} = c_g^{m_a} j_{conv}^{m_g}$$

| | | |
|---|---|---|
| $c_g^{m_a} = \dfrac{p_a}{p_a + p_v} \cdot \dfrac{R_v}{R_a}$ | Air mass concentration in the gas phase | $kg/kg$ |
| $j_{conv}^{m_g} = -K_g\left[\dfrac{\partial p_g}{\partial x} + \rho_g g\right]$ | Convective flux of the gas phase | $kg/m^2 s$ |
| $K_g$ | Gas permeability of the material | $s$ |
| $p_g = p_a + p_v$ | Gas pressure | $Pa$ |
| $\rho_g = \dfrac{p_g}{R_a T}$ | Intrinsic density of gas phase | $kg/kg$ |

### Air sources and sinks

No sources and sinks implemented yet.

## 4.1.3   VOC mass balance

### Assumptions
• VOC adsorption takes place in competition to water adsorption. Since water is always present we assume adsorption of a VOC-water mixture   (lumped consideration of dissolved / adsorbed phases).
• Thermodynamic VOC equilibrium (temperatures, pressures, chemical potentials) between gas phase and dissolved / adsorbed phase
• Gas phase VOC diffusion can be treated as Fickian diffusion process. It includes effusion in micro pores.
• Dispersive VOC mass flows are negligible
• No chemical reaction

### The VOC mass balance can be written as

$$\frac{\partial}{\partial t}\rho_{REV}^{m_{voc,l+g}} = -\frac{\partial}{\partial x}\left[ j_{conv}^{m_{voc,l}} + j_{conv}^{m_{voc,g}} + j_{diff}^{m_{voc,g}} \right] + \sigma_{REV}^{m_{voc,l+g}}$$

| | | |
|---|---|---|
| $\rho_{REV}^{m_{voc,l+g}}$ | VOC (liquid+vapor) density in reference volume | $kg/m^3$ |
| $j_{conv}^{m_{voc,l}}$ | Convective liquid (dissolved+adsorbed) VOC flux* | $kg/m^2 s$ |
| $j_{conv}^{m_{voc,g}}$ | Convective gas phase VOC flux | $kg/m^2 s$ |
| $j_{diff}^{m_{voc,g}}$ | Diffusive gas phase VOC flux | $kg/m^2 s$ |
| $\sigma_{REV}^{m_{voc,l+g}}$ | VOC sources/sinks in reference volume | $kg/m^3 s$ |

*not yet implemented*

### Convective liquid (dissolved / adsorbed) VOC flux

$$j_{conv}^{m_{voc,l}} = c_l^{m_{voc}} j_{conv}^{m_l}$$

| | | |
|---|---|---|
| $c_l^{m_{voc}} = \dfrac{\rho_l^{m_{voc}}}{\rho_l}$ | VOC mass concentration in the liquid phase | $kg/m^3$ |

### Surface diffusion of adsorbed phase VOC flux

Differentiation between adsorbed surface layers and adsorption / condensation in small pores is hardly possible. Therefore, this flux has been included in the convective liquid (dissolved / adsorbed) VOC flux.

### Diffusive dissolved phase VOC flux

Since we always assume a VOC-water mixture (of different concentration), this is already described by the convective liquid (dissolved / adsorbed) VOC flux. Note scaling of the liquid phase conductivity by concentration and temperature dependent viscosity.

### Convective gas phase VOC flux

$$j_{conv}^{m_{voc,g}} = c_g^{m_{voc}} j_{conv}^{m_g}$$

| | | |
|---|---|---|
| $c_g^{m_{voc}} = \dfrac{\rho_g^{m_{voc}}}{\rho_g}$ | VOC mass concentration in the gas phase | $kg/m^3$ |

### Diffusive gas phase VOC flux

$$j_{diff}^{m_{voc,g}} = -\frac{D_{voc}^{air}(T)}{\mu_{voc} R_{voc} T} f(\theta_g) \frac{\partial p_{voc}}{\partial x}$$

| | | |
|---|---|---|
| $D_{voc}^{air}(T)$ | VOC diffusivity in free air | $m^2/s$ |
| $\mu_{voc} = \mu \cdot k_{voc}$ | VOC diffusion resistance factor | — |
| | ($\mu$ accounts for tortuosity and reduction of diffusion cross section) | |
| $k_{voc}$ | Correction factor to be determined for material-VOC combinations | |
| $f(\theta_g)$ | Function of volume fraction of gas phase | — |
| | (accounts for moisture dependency, $f(\theta_g) = \theta_g$ in many cases) | |
| $p_{voc}$ | VOC pressure in gas phase | Pa |

## VOC sources and sinks

No sources and sinks implemented yet.

### 4.1.4  Internal energy balance

### Assumptions
• Temperature equilibrium between all phases
• The difference between internal energy U and enthalpy H may be neglected

### The energy balance is given by

$$\frac{\partial}{\partial t} \rho_{REV}^U = -\frac{\partial}{\partial x}\left[ j_{diff}^Q + u_l j_{conv}^{m_l} + u_g j_{conv}^{m_g} + h_v j_{diff}^{m_v} + h_{voc,g} j_{diff}^{m_{voc,g}} \right] + \sigma_{REV}^U$$

| | | |
|---|---|---|
| $\rho_{REV}^U$ | Internal energy density in reference volume | $J/m^3$ |
| $j_{diff}^Q$ | Heat conduction | $W/m^2$ |
| $\sigma_{REV}^U$ | Energy sources/sinks in reference volume | $W/m^3$ |

### Heat conduction flow

$$j_{diff}^Q = -\lambda \frac{\partial T}{\partial x}$$

| | | |
|---|---|---|
| $\lambda$ | Thermal conductivity of material | $W/mK$ |
| $T$ | Temperature | $K$ |

### Total internal energy density

$$\rho_{REV}^{U} = \rho_s \theta_s u_s + \rho_l \theta_l u_l + \rho_g \theta_g u_g$$

*Expressions for specific internal energy of solid matrix*

| | | |
|---|---|---|
| $\rho_s \theta_s = \rho_m$ | Bulk density of material | $kg/m^3$ |
| $u_s(T) = c_{T,m} \cdot (T - T_0)$ | Specific internal energy of material | J/kg |
| $c_{T,m}$ | Specific heat capacity of material | J/kgK |

*Expressions for specific internal energy of liquid phase*

| | | |
|---|---|---|
| $u_l(T) = c_l^{m_w} u_w(T) + c_l^{m_{voc}} u_{voc,l}(T)$ | Specific internal energy of liquid phase | J/kg |
| $u_w(T) = c_{T,w} \cdot (T - T_0)$ | Specific internal energy of liquid water | J/kg |
| $c_{T,w}$ | Specific heat capacity of liquid water | J/kgK |
| $u_{voc,l}(T) = c_{T,voc,l} \cdot (T - T_0)$ | Specific internal energy of liquid VOC | J/kg |
| $c_{T,voc,l}$ | Specific heat capacity of liquid VOC | J/kgK |

*Expressions for specific internal energy of gas phase*

| | | |
|---|---|---|
| $u_g(T) = c_g^{m_v} h_v(T) + c_g^{m_a} u_a(T) + c_g^{m_{voc}} h_{voc,g}(T)$ | Specific internal energy of gas phase | J/kg |
| $h_v(T) = c_{T,v} \cdot (T - T_0) + h_v^{evap}(T_0)$ | Specific enthalpy of water vapor | J/kg |
| $c_{T,v}$ | Specific heat capacity of water vapor | J/kgK |
| $h_v^{evap}(T_0)$ | Specific evaporation enthalpy of water vapor | J/kg |
| $u_a(T) = c_{T,a} \cdot (T - T_0)$ | Specific internal energy of dry air | J/kg |
| $c_{T,a}$ | Specific heat capacity of dry air | J/kgK |
| $h_{voc,g}(T) = c_{T,voc,g} \cdot (T - T_0) + h_{voc,g}^{evap}(T_0)$ | Specific enthalpy of gaseous VOC | J/kg |
| $c_{T,voc,g}$ | Specific heat capacity of gaseous VOC | J/kgK |

### Energy sources and sinks

No sources and sinks implemented yet.

## 4.2 Condition types

The condition types for a simulation problem are described by specification data records generated by help of dialogs and stored in the project file. The defined condition types of CHAMPS-BES are listed in the table below.

| Condition type | Explanation |
|---|---|
| Climate Conditions | Time dependent quantities (e.g. course of temperature or relative humidity) that can be used for closer specification of the boundary conditions or field conditions. Climate conditions ARE NOT assigned to the construction setup. |
| Boundary Conditions | Model selection (kind), parameters, time limits and assigned climate conditions to specify fluxes over the boundary of the problem. Boundary conditions MUST be assigned to the construction setup. Zero-flux (e.g. adiabate boundary) is specified by omitting the boundary conditions assignment. |
| Initial Conditions | Initial conditions of the problem; usually single values and physical units. Specification of initial conditions is required for each balance equation to be solved. Initial conditions MUST be assigned to the construction setup. |
| Field Conditions | Sources and sinks of the problem. Standard sources and sinks depend on time and/or climate conditions only. Special sources and sinks require information from the field solution of the problem; they depend on state variables. Field conditions MUST be assigned to the construction setup. |
| Contact Conditions | Additional resistance values at material layer interfaces. Can be used to model materials without remarkable storage capacity but with high transport resistance without discretization (e.g. vapor barriers). Contact conditions MUST be assigned to the construction setup. |

**Condition types of CHAMPS-BES**

The user can specify condition data records by using predefined climate, boundary, initial, field and contact conditions. The predefined conditions are available via type selection boxes in the GUI-dialogs of CHAMPS-BES.

It is possible to generate more condition data records than needed for a specific problem. Unused data records (not assigned to the construction setup) are ignored by CHAMPS-BES.

## 4.2.1   Climate conditions

The climate conditions specify time dependent loads with regard to boundary conditions or field conditions. The evolution with time of the climate conditions can be described by periodic functions or by numerical values initialized from climatic data files. The user can define climate condition data records using the Climate Condition Edit Dialog. Table 2 shows the predefined climatic conditions, explanations, limits and SI-units.

| Keyword | Explanation | Limits (recommended) | SI-Unit |
|---------|-------------|----------------------|---------|
| TEMPER | Temperature | -20 … 60 | °C |
| RELHUM | Relative humidity | 0 … 100 | % |
| VAPPRESS | Water vapor pressure | 0 … | Pa |
| NORRAIN | Rain flux density normal to the wall surface | 0 … | $l/m^2s$ |
| HORRAIN | Rain flux density on a horizontal plane | 0 … | $l/m^2s$ |
| WINDDIR | Wind direction, North=0 Deg, East=90 Deg, … | 0 … 360 | Deg |
| WINDVEL | Wind velocity | 0 … | m/s |
| SHWRAD | Short wave radiation (balance) | 0 … | $W/m^2$ |
| DIRRAD | Direct sun radiation (short wave component) | 0 … | $W/m^2$ |
| DIFRAD | Diffuse sun radiation (short wave component) | 0 … | $W/m^2$ |
| LOWRAD | Long wave radiation (balance) | | $W/m^2$ |
| LWEMIS | Heat emission from surrounding ground (long wave component) | … 0 | $W/m^2$ |
| SKYRAD | Atmospheric counter radiation (long wave component) | 0 … | $W/m^2$ |
| CLOUDCOV | Cloud covering (alternative to long wave radiation) | 0 … 1 | - |
| PREHEAD | Pressure head | | m |
| WATFLUX | Imposed moisture flux | | $kg/m^2s$ |
| HEATFLUX | Imposed heat flux | | $W/m^2$ |
| SALTCONT | Salt solution (concentration as molality) | 0 … | mol/kg |
| GASPRESS | Gas (air) pressure | | Pa |
| VOCDENSG | Intrinsic VOC density in gas phase | 0 … | $kg/m^3$ |

**Table 2: Climate conditions of CHAMPS-BES**

The **Keyword** is used to identify the respective data record in the CHAMPS-BES project file. The **Limits** are recommended values. There is no automatic check by the program. Possible physical units are the **SI-Unit** or any convertible unit.

### 4.2.2    Boundary conditions

The boundary conditions describe model selections, parameters, time limits and assigned climate conditions. The user can define boundary condition data records using the Boundary Condition Edit Dialog. Table 3 shows the predefined boundary conditions, explanations, and related fluxes according to the balance equations.

| Balance equation | Keyword | Explanation | Related flux |
|---|---|---|---|
| Moisture mass | VAPDIFF | Vapor diffusion | Mass and enthalpy (latent heat) of the diffusive water vapor flux |
| | RAIN | Wind driven rain | Mass and internal energy of the liquid water flux |
| | WATCONTC | Water contact | |
| Air mass | AIRPRESS | Air convection using air pressure difference | Mass and internal energy of the convective air flux<br>Mass and internal energy of the convective water vapor flux (if applicable)<br>Mass of the convective VOC flux (if applicable) |
| VOC mass | VOCDIFF | Volatile Organic Compound (VOC) diffusion | Mass of the diffusive VOC flux |
| Internal energy | HEATCOND | Heat conduction | Heat conduction by temperature difference |
| | SHWRAD | Short wave sun radiation | Short wave radiation flux as balance of direct and diffuse sun radiation |
| | LOWRAD | Long wave heat radiation | Long wave radiation flux as balance of long wave emission and atmospheric counter radiation |

**Table 3: Boundary conditions of CHAMPS-BES**

The **Keyword** is used to identify the respective data record in the CHAMPS-BES project file. The **Related flux** are calculated at the assignment locations using the selected model (**Type** and **Kind**), their parameters and climate conditions. Boundary conditions act ADDITIVE, i.e. assignment of RAIN and VAPDIFF to the same boundary location would result in a total moisture flux consisting of a liquid and a vapor part. For the air convection condition, the convective flux of water vapor and VOCs is only calculated if applicable, i.e. the respective balance equation is switched on. The decision about definition and assignment of physically meaningful combinations of boundary conditions lies in the responsibility of the user.

Currently the following boundary condition **types** and **kinds** are implemented:

## *Vapor diffusion*

Defines a vapor diffusion boundary condition (driven by vapor pressure gradient). The user can *either* assign Temperature and Relative Humidity or provide a boundary Vapor Pressure climatic condition. If you specify all three, the Vapor Pressure climatic condition is ignored.

| Kind | Parameters | Physical units / default values | Climate data |
|------|-----------|-------------------------------|--------------|
| Exchange coefficient | Water vapor exchange coefficient | EXCOEFF = 3e-08 s/m | Temperature and Relative humidity or Water vapor pressure of ambient air |

Exchange coefficient

$$j_{diff}^{m_v} = \beta_v \cdot (p_v^{elem} - p_v^{air})$$ 

Mass of diffusive water vapor flux — kg/m²s

$$j_{diff}^{H_v} = h_v(T_v) \cdot j_{diff}^{m_v}$$ 

Enthalpy of diffusive water vapor flux — J/m²s

---

$\beta_v$ — Water vapor exchange coefficient — s/m

$p_v^{elem}$ — Water vapor pressure in boundary element — Pa

$p_v^{air} = p_{v,sat}(T^{air}) \cdot \varphi^{air}$ — Water vapor pressure of ambient air — Pa

$T^{air}$ — Temperature of ambient air — K

$\varphi^{air}$ — Relative humidity of ambient air — -

$h_v(T_v)$ — Specific enthalpy of water vapor — J/kgK

$T_v = T^{air}$ if $j_{diff}^{m_v} > 0$; $T^{elem}$ else — Temperature of vapor flux (depends on direction) — K

## *Rain*

Defines a Wind Driven Rain (WDR) boundary condition. Two kinds are available: In case of **Imposed flux** the rain flux density normal to the wall surface has been calculated be the user (no WDR calculation by CHAMPS-BES necessary). Using the **Standard rain model**, the rain flux density normal to the wall surface is calculated from the Rain flux density on a horizontal plane, the Wind direction and the Wind velocity (WDR calculation) using the wall parameters Orientation and Inclination.

There are different WDR models available in the literature. For more information see appendix Wind driven rain modeling.

Knowing the rain flux density normal to the wall surface the liquid water flux over the boundary can be calculated. No further parameter is needed. The maximum liquid water flux over the boundary is determined by multiplication of the liquid water conductivity of the material with a capillary pressure gradient calculated from the difference of the moisture content in the surface volume element to its saturation value (immediate surface saturation is assumed in case of rain penetration).

| Kind | Parameters | Physical units / default values | Climate data |
|---|---|---|---|
| Imposed flux | Wall parameters<br><br>Rain exposure coefficient (accounts for sheltering) | Orientation, Inclination<br><br>EXPCOEFF = 1 - | Rain flux density normal to the wall surface<br>Temperature of ambient air<br>Relative humidity of ambient air |
| Standard rain model | Minimum rain temperature (below rain falls as snow)<br>Minimum normal rain intensity (lower values are neglected) | MINRTEMP = -2 C<br><br>MINRFLUX = 0 l/m2s | Rain flux density on a horizontal plane<br>Wind direction of ambient air<br>Wind velocity of ambient air<br>Temperature of ambient air<br>Relative humidity of ambient air |

Imposed flux

$$j_{conv}^{m_w} = \min(j_{imp}^{m_w}, j_{max}^{m_w})$$  Mass of convective liquid water flux  kg/m$^2$s

$$j_{conv}^{U_w} = u_w(T_w) \cdot j_{conv}^{m_w}$$  Internal energy of convective liquid water flux  J/m$^2$s

$j_{imp}^{m_w}$  Imposed water flux  kg/m$^2$s

$$j_{max}^{m_w} = \psi(\theta_{cap}^{elem}) - \psi(\theta_l^{elem})$$  Maximum water flux  kg/m$^2$s

$\psi(\theta_{cap}^{elem})$  Kirchhoff potential at capillary saturation  kg/m$^2$s

$\psi(\theta_l^{elem})$  Kirchhoff potential in boundary element  kg/m$^2$s

$\theta_{cap}^{elem}$  Capillary saturation  m$^3$/m$^3$

$\theta_l^{elem}$  Water content in boundary element  m$^3$/m$^3$

$u_w(T_w)$  Specific internal energy of liquid water  J/kgK

$T_w$  Temperature of imposed water flux  K

Standard rain model

$$j_{conv}^{m_w} = \min(j_{rain,nor}^{m_w}, j_{max}^{m_w})$$  Mass of convective liquid water flux  kg/m$^2$s

$$j_{conv}^{U_w} = u_w(T_{rain}) \cdot j_{conv}^{m_w}$$  Internal energy of convective liquid water flux  J/m$^2$s

$$j_{rain,nor}^{m_w} = k_{wind} \cdot k_{rain} \cdot j_{rain,hor}^{m_w}$$  Rain flux density normal to the wall surface  kg/m$^2$s

$k_{wind}$  Wind coefficient  -

$k_{rain}$  Rain exposure coefficient  -

$j_{rain,hor}^{m_w}$  Rain flux density on a horizontal plane  kg/m$^2$s

$u_w(T_{rain})$  Specific internal energy of rain  J/kgK

$T_{rain}$  Temperature of rain water  K

Standard rain model

$$\beta_{wind} = \begin{cases} \left| \alpha_{wall} - \alpha_{wind} \right| & if \quad \left| \alpha_{wall} - \alpha_{wind} \right| \le \pi \\ 2\pi - \left| \alpha_{wall} - \alpha_{wind} \right| & else \end{cases}$$    Wind angle    Deg

$$k_{wind} = \begin{cases} 0 & if \quad \left( \beta_{wind} \ge \dfrac{\pi}{2} \right) \vee \left( v_{wind} \le 0 \right); \quad else \\ \dfrac{\cos(\beta_{wind})}{\sqrt{1 + 1141 \cdot \sqrt{\dfrac{3600 \cdot j_{Rain,hor}^{m_w}}{v_{wind}^4}}}} \cdot \exp\left( -\dfrac{12}{5 \cdot \sqrt[4]{3600 \cdot j_{Rain,hor}^{m_w}}} \right) \end{cases}$$    Wind coefficient    -

$$T_{rain} = \left( \varphi^{env} \right)^{0.1247} (109.8 + T^{env}) - 109.8$$    Temperature of rain water    K

$\alpha_{wall}$    Wall orientation    Deg

$\alpha_{wind}$    Wind direction    Deg

$v_{wind}$    Wind velocity    m/s

## *Water contact*

A boundary condition defining direct water contact. Two kinds are available: In case of **Imposed flux** the moisture flux through the boundary is defined directly. In case of **Surface value** the water pressure value at the surface of the boundary element is fixed taking into account the liquid water conductivity between the surface and the element's center point (similar approach as for rain calculation). The pressure head specifies the pressure of the water in terms of a height (0 m = untensioned water). Ground water pressure can be simulated using a positive pressure head. In both cases, the temperature is needed to calculate the internal energy of the water flux. If omitted, 20°C is assumed as water temperature.

| Kind | Parameters | Physical units / default values | Climate data |
|------|-----------|--------------------------------|-------------|
| Surface value | no parameters | no parameters | Pressure head of water<br>Temperature at wall surface |
| Imposed flux | | | Imposed liquid water flux<br>Temperature of water flux |

Surface value

$$j_{conv}^{m_w} = \psi(\theta_l^{surf}) - \psi(\theta_l^{elem})$$    Mass of convective liquid water flux    kg/m$^2$s

$$j_{conv}^{U_w} = u_w(T^{elem}) \cdot j_{conv}^{m_w}$$    Internal energy of convective liquid water flux    J/m$^2$s

| | | |
|---|---|---|
| $\psi(\theta_l^{surf})$ | Kirchhoff potential at wall surface | kg/m$^2$s |
| $\psi(\theta_l^{elem})$ | Kirchhoff potential in boundary element | kg/m$^2$s |
| $\theta_l^{surf} = \theta_l(\rho_w \cdot g \cdot h^{surf})$ | Water content at surface | m$^3$/m$^3$ |
| $\theta_l^{elem}$ | Water content in boundary element | m$^3$/m$^3$ |
| $\rho_w$ | Liquid water mass density | kg/m$^3$ |
| $g$ | Gravity constant | m/s$^2$ |
| $h^{surf}$ | Pressure head of water | m |

Imposed flux

$$j_{conv}^{m_w} = j_{imp}^{m_w}$$    Mass of convective liquid water flux    kg/m$^2$s

$$j_{conv}^{U_w} = u_w(T_w) \cdot j_{conv}^{m_w}$$    Internal energy of convective liquid water flux    J/m$^2$s

| | | |
|---|---|---|
| $j_{imp}^{m_w}$ | Imposed liquid water flux | kg/m$^2$s |
| $u_w(T_w)$ | Specific internal energy of liquid water | J/kgK |
| $T_w = T^{wat}$ if $j_{conv}^{m_w} > 0$; $T^{elem}$ else | Temperature of water flux (depends on flux direction) | K |

## *Air pressure*

Defines an air convection boundary condition. The convective gas flux is calculated from an air pressure difference multiplied with an Air convection exchange coefficient. The convective flux of water vapor and VOCs is only calculated if applicable. Then, climate data for calculation of the vapor pressure and VOC density are necessary.

| Kind | Parameters | Physical units / default values | Climate data |
|---|---|---|---|
| Exchange coefficient | Air convection exchange coefficient | EXCOEFF = 0.1 s/m | Temperature of ambient air Relative humidity of ambient air Gas (air) pressure of ambient air VOC density of ambient air (if applicable) |

Exchange coefficient

$$j_{conv}^{m_a} = \beta_a \cdot (p_a^{elem} - p_a^{air})$$      Mass of convective air flux      kg/m²s

$$j_{conv}^{m_v} = c_g^{m_v} \cdot j_{conv}^{m_a}$$      Mass of convective water vapor flux      kg/m²s

$$j_{conv}^{m_{voc}} = c_g^{m_{voc}} \cdot j_{conv}^{m_a}$$      Mass of convective VOC flux      kg/m²s

$$j_{conv}^{U_a} = u_a(T_a) \cdot j_{conv}^{m_a}$$      Enthalpy of convective air flux      J/m²s

$$j_{conv}^{U_v} = u_v(T_a) \cdot j_{conv}^{m_v}$$      Enthalpy of convective water vapor flux      J/m²s

---

| | | |
|---|---|---|
| $\beta_a$ | Air exchange coefficient | s/m |
| $p_a^{elem}, p_a^{air}$ | Air pressure in boundary element, of ambient air | Pa |
| $c_g^{m_v}(\varphi_a, T_a), c_g^{m_{voc}}(\rho_{voc})$ | Mass concentration of vapor, of VOC | kg/kg |
| $u_a(T_a), u_v(T_a)$ | Specific internal energy of air, of water vapor | J/kgK |
| $\varphi_a = \varphi^{air}$   if $j_{conv}^{m_a} > 0$;   $\varphi^{elem}$ else | Relative humidity of air flux (depends on direction) | % |
| $T_a = T^{air}$   if $j_{conv}^{m_a} > 0$;   $T^{elem}$ else | Temperature of air flux (depends on direction) | K |
| $\rho_{voc} = \rho_{voc}^{air}$ if $j_{conv}^{m_a} > 0$; $\rho_{voc}^{elem}$ else | VOC-density of air flux (depends on direction) | kg/m³ |

## VOC diffusion

Accounts for diffusion of VOCs over the boundary. The diffusive VOC flux is calculated from a difference in VOC density (mass per volume of gas phase) multiplied with the exchange coefficient for VOC diffusion.

| Kind | Parameters | Physical units / default values | Climate data |
|---|---|---|---|
| Exchange coefficient | Exchange coefficient for VOC diffusion | EXCOEFF = 1e-08 m/s | VOC density of ambient air |

Exchange coefficient

$$j_{diff}^{m_{voc}} = \gamma_{voc} \cdot (\rho_{voc}^{elem} - \rho_{voc}^{air})$$      Mass of diffusive VOC flux      kg/m²s

---

| | | |
|---|---|---|
| $\gamma_{voc}$ | VOC exchange coefficient | m/s |
| $\rho_{voc}^{elem}$ | Intrinsic VOC density in boundary element | kg/m³ |
| $\rho_{voc}^{air}$ | Intrinsic VOC density of ambient air | kg/m³ |

## Heat conduction

A heat conduction boundary condition. Three kinds are available: In case of **Surface value** the value at the surface of the boundary element is fixed. Using **Imposed flux** the

heat flux through the boundary is defined. In case of **Exchange coefficient** the heat conduction is defined via exchange coefficient.

| Kind | Parameters | Physical units / default values | Climate data |
|------|-----------|--------------------------------|--------------|
| Surface value | no parameters | no parameters | Temperature at wall surface |
| Imposed flux | | | Imposed heat flux |
| Exchange coefficient | Exchange coefficient for heat flow | EXCOEFF = 25 W/m2K | Temperature of ambient air |

Surface value

$$j_{diff}^{Q} = 1000 \cdot (T^{elem} - T^{surf})$$   Heat flux   W/m$^2$

$T^{elem}$   Temperature in boundary element   K

$T^{surf}$   Temperature at wall surface   K

Imposed flux

$$j_{diff}^{Q} = j_{imp}^{Q}$$   Heat flux   W/m$^2$

$j_{imp}^{Q}$   Imposed heat flux   W/m$^2$

Exchange coefficient

$$j_{diff}^{Q} = \alpha \cdot (T^{elem} - T^{air})$$   Heat flux   W/m$^2$

$\alpha$   Heat exchange coefficient   W/m$^2$K

$T^{air}$   Temperature of ambient air   K

## *Short wave radiation*

Defines a short wave sun radiation boundary condition. Two kinds are available: In case of **Imposed flux** the radiation flux density normal to the wall surface has been calculated be the user (no radiation calculation by CHAMPS-BES necessary). Using the **Direct sun radiation model**, the radiation flux density normal to the wall surface is calculated from the day time, the Direct sun radiation and Diffuse sun radiation components (radiation calculation) using the wall parameters Orientation, Inclination and Latitude.

There are different radiation models available in the literature. For more information see appendix Radiation modeling.

| Kind | Parameters | Physical units / default values | Climate data |
|------|-----------|-------------------------------|--------------|
| Imposed flux | Wall parameters | Orientation, Inclination, Latitude | Short wave radiation (balance) |
| Direct sun radiation model | Absorption coefficient of the building surface Reflection coefficient of the surrounding ground (albedo) | SURABSOR = 0.6 - GRDREFLE = 0.9 - | Direct sun radiation Diffuse sun radiation |

Imposed flux

$$j_{swrad}^{Q} = \alpha_{sw} \cdot j_{imp}^{Q}$$

Short wave radiation flux — W/m²

$\alpha_{sw}$ — Absorption coefficient for short wave radiation — -

$j_{imp}^{Q}$ — Imposed radiation flux — W/m²

Direct sun radiation model

$$j_{swrad}^{Q} = \alpha_{sw} \cdot (j_{dir,n}^{Q} + j_{diff,n}^{Q})$$

Short wave radiation flux — W/m²

$j_{dir,n}^{Q} = j_{dir,h}^{Q} \cdot f_{dir}(\alpha_{wall}, \beta_{wall}, l_{geo}, t_{day})$ — Direct radiation flux normal to wall surface — W/m²

$j_{diff,n}^{Q} = j_{diff,h}^{Q} \cdot \cos(\beta_{wall}/2)^2 + \alpha_{albedo} \cdot j_{glob,h}^{Q} \cdot \sin(\beta_{wall}/2)^2$ — Diffuse radiation flux normal to wall surface — W/m²

$j_{glob,h}^{Q} = j_{dir,h}^{Q} + j_{diff,h}^{Q}$ — Global radiation flux on a horizontal plane — W/m²

$j_{dir,h}^{Q}$ — Direct radiation flux on a horizontal plane — W/m²

$j_{diff,h}^{Q}$ — Diffuse radiation flux on a horizontal plane — W/m²

$f_{dir}(\alpha_{wall}, \beta_{wall}, l_{geo}, t_{day})$ — Direct radiation factor — -

$\alpha_{wall}$ — Wall orientation — Deg

$\beta_{wall}$ — Wall inclination — Deg

$l_{geo}$ — Geographic latitude of location — Deg

$t_{day}$ — Day number — -

$\alpha_{albedo}$ — Ground reflection coefficient (albedo) — -

## *Long wave radiation*

Defines a long wave heat radiation boundary condition. Three kinds are available: In case of **Imposed flux** the radiation flux density normal to the wall surface has been calculated be the user (no radiation calculation by CHAMPS-BES necessary). Using the **Long wave components**, the radiation flux density normal to the wall surface is calculated from the Heat emission of the building and the Atmospheric counter radiation (radiation calculation) using the wall parameter Inclination. In case of a **Boltzmann calculation** the radiation flux is calculated basing on Boltzmann's law. The surface temperature is used for the long wave emission and the air temperature is used instead

of the sky temperature for calculating the Atmospheric counter radiation. Except for imposed flux, the Emission coefficient of the surrounding ground is taken into account additionally.

There are different radiation models available in the literature. For more information see appendix Radiation modeling.

| Kind | Parameters | Physical units / default values | Climate data |
|---|---|---|---|
| Imposed flux | Emission coefficient of the building surface | SUREMISS = 0.9 - | Long wave radiation (balance) |
| Long wave components | Wall parameters | Inclination | Heat emission from surrounding ground |
| | Emission coefficient of the surrounding ground | GRDEMISS = 0.9 - | Atmospheric counter radiation |
| Boltzmann calculation | Emission coefficient of the building surface | SUREMISS = 0.9 - | Temperature of ambient air Relative humidity of ambient air Cloud covering |

Imposed flux

$$j_{lwrad}^{Q} = \varepsilon_{lw} \cdot j_{imp}^{Q}$$

| | Long wave radiation flux | W/m² |
|---|---|---|

$\varepsilon_{lw}$ — Emission coefficient for long wave radiation — -

$j_{imp}^{Q}$ — Imposed radiation flux — W/m²

Long wave components

$$j_{lwrad}^{Q} = f_{sky} \cdot (j_{sky}^{Q} - \sigma \cdot T_{surf}^{4}) + f_{grd} \cdot (j_{grd}^{Q} - \sigma \cdot T_{surf}^{4})$$

| | Long wave radiation flux | W/m² |
|---|---|---|

$j_{sky}^{Q}$ — Atmospheric counter radiation flux — W/m²

$j_{grd}^{Q}$ — Ground emission radiation flux — W/m²

$\sigma$ — Stefan Boltzmann constant — W/m²K⁴

$T_{surf} = T^{elem}$ — Temperature of building surface — K

$f_{sky} = \cos(\beta_{wall} / 2)^{2} \cdot \varepsilon_{lw,surf}$ — Sky radiation factor — -

$f_{grd} = \sin(\beta_{wall} / 2)^{2} \cdot \dfrac{1}{\dfrac{1}{\varepsilon_{lw,surf}} + \dfrac{1}{\varepsilon_{lw,grd}} - 1}$ — Ground radiation factor — -

$\beta_{wall}$ — Wall inclination — Deg

$\varepsilon_{lw,surf}$ — Long wave emission coefficient of building surface — -

$\varepsilon_{lw,grd}$ — Surrounding ground emission coefficient — -

Boltzmann calculation

$$j_{lwrad}^{Q} = f_{sky} \cdot \sigma \cdot (T_{sky}^{4} - T_{surf}^{4})$$
$$+ f_{grd} \cdot \sigma \cdot (T_{grd}^{4} - T_{surf}^{4})$$

Long wave radiation flux          W/m²

$$T_{sky} = T^{air} \cdot \sqrt[4]{f_{RH} \cdot f_{CC}}$$   Sky temperature          K

$$T_{grd} = T^{air}$$   Ground temperature          K

$$T_{surf} = T^{elem}$$   Temperature of building surface          K

$$f_{RH} = 0.82 - 0.25 \cdot 10^{-0.001333224 \cdot p_{v}^{air}}$$   Relative humidity factor          -

$$f_{CC} = 1 + 0.2 \cdot B^{2}$$   Cloud covering factor          -

$$p_{v}^{air}$$   Water vapor pressure in air          Pa

$$B$$   Cloud covering          -

## 4.2.3   Initial conditions

The initial conditions define the initial values of the problem. The user can define initial condition data records using the Initial Condition Edit Dialog. Table 4 shows the predefined initial conditions, explanations, limits and SI-units.

A definition of an initial condition is required for each balance equation to be solved. For the moisture balance, there are four alternatives: Liquid moisture content, Total moisture mass, Relative humidity and Capillary pressure.

| Balance equation | Keyword | Explanation | Limits | SI-Unit |
|---|---|---|---|---|
| Moisture mass | MSTCONT | Liquid moisture content | 0 ... 1 | $m^3/m^3$ |
| | MSTMASS | Total moisture mass | 0 ... 1000 | $kg/m^3$ |
| | RELHUM | Relative humidity | 0 ... 100 | % |
| | CAPPRESS | Capillary pressure | | Pa |
| Air mass | GASPRESS | Air (gas) pressure | | Pa |
| VOC mass | VOCMASS | Total VOC mass | 0 ... | $kg/m^3$ |
| Internal energy | TEMPER | Temperature | -20 ... 60 | °C |

**Table 4: Initial conditions of CHAMPS-BES**

The **Keyword** is used to identify the respective data record in the CHAMPS-BES project file. The **Limits** are recommended values. There is an automatic check by the program. Warnings are printed to the error log file in case of necessary corrections. Possible physical units are the **SI-Unit** or any convertible unit.

### 4.2.4 Field conditions

No field conditions defined yet.

### 4.2.5 Contact conditions

No contact conditions defined yet.

## 4.3 Numerical solution

### *Introduction*

The implementation of the physical models for CHAMPS leads usually to diffusion-convection-reaction equations which are partial differential equations of parabolic type. Depending on the user's problem settings the set of balance equations can appear in coupled manner (as usually for hygrothermal problems) or they can be decoupled (e.g., single solution of the heat balance). The numerical solution applied to the problem doesn't depend on the status of equation coupling; it rather matters the degree of freedom in physical modeling. The chosen numerical method allows a maximum of flexibility in terms of choice of transport potentials and consideration of physical dependencies.

Various numerical solution techniques exist for coupled nonlinear systems of ODEs and have been successfully employed for solving the coupled heat and moisture transport. However, finding a correct and also quite importantly an efficient solution scheme for such a system of equations, particularly with the tight coupling between the equations, remains a challenge. Hence, integral techniques such as the finite element method (FEM) and control volume  method (CVM) became more popular and are now used in state-of-the-art models.

Here, for the numerical solution of the governing system of balance equations the CVM is applied. For orthogonal, equidistant grids the control volume method yields exactly the same discretized equations as the finite difference method (FDM). However, it can as easily be applied to unstructured grids. Advantages of the CVM are:

* usable for structured and unstructured grids,
* accurate formulation of fuxes over CV boundaries possible (mass conservation),
* relatively simple numerical scheme resulting from discretized equations.

The limitation to rectangular volume elements facilitates pre- and postprocessing of data considerably and the solution is for practical problem in building engineering more as satisfying.

The partial differential equations of parabolic type are transformed into a system of ordinary differential equations (ODEs) by semi-discretization of the spatial coordinates of the problem. This is essential for an independent selection of primary state variables (the unknowns) and driving potentials. This advantageous two-step approach allows a flexible and fast numerical solution that can be achieved with reasonable working effort and being open for future extensions.

The system of ODEs describes the temporal changes of all unknowns of the problem. The

integration in time is carried out with help of the CVODE integrator that is part of the SUNDIALS package (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers), Copyright (c) 2002 by The Regents of the University of California; Produced at the Lawrence Livermore National Laboratory; Written by A.C. Hindmarsh et al. The CVODE package is available as source code in C-language and it comprises a selection of well elaborated and tested methods for numerical solution of ODEs.

## 4.3.1  Description of solution method

### *Formulation a general balance equation*

It is advantageous to introduce a generalized formulation of the balance equations for demonstration of the major steps in numerical solution. A general extensive quantity *E* is introduced that stands for all quantities a conservation law can be applied to. This can be, for example, any mass component, internal energy, linear momentum or entropy.

In real applications, the unknowns of the system are chosen. In case of CHAMPS problems, these are the masses of liquid water, water vapor, dry air and a pollutant component. In addition, the internal energy of the system is balanced. Eliminating the phase transmutation term (evaporation and condensation) the balances liquid water and water vapor are added to a total moisture balance.

The balances for dry air mass and VOC mass can be treated separately. Here a 1-2-1 solution is applied: First, the air mass balance is solved that can be done under quasi-steady state conditions. Then, the coupled heat and moisture balances are solved and subsequently, the VOC mass balance is solved. This approach was developed to find an optimal solution in terms of accuracy versus performance. A partial decoupling, particularly of the air mass balance and the heat and moisture balances, increases the numerical performance drastically.

**General form of balance equations (parabolic differential equations)**

$$\boxed{\frac{\partial \rho_{REV}^{E}}{\partial t} = \frac{\partial}{\partial x}\left(j^{E}\right) + \sigma_{REV}^{E}}$$

$E =$ *general extensive quantity*

*e.g.* $m$      *mass*

$\rho_{REV}^{E}$   *Volumetric E-density*      $U$      *internal energy*

$j^{E}$   *E-flux density (related to REA)*      $I$      *linear momentum*

$\sigma_{REV}^{E}$   *Volumetric E-production density (source)*      $S$      *entropy*

     ...**in case of CHAMPS:**      $m_{w+v}$      *liquid water and vapor mass*

$m_{a}$      *dry air mass*

$m_{w}$      *VOC mass*

$U$      *internal energy*

## *Mathematical formulation*

To illustrate the procedure of applying the CVM to obtain a discretized equation, we consider the general balance equation derived above. Basically, the numerical solution of the governing equation system consists of four main steps:

1.  The differential equation is rearranged and integrated over the control volume. This step comprises transformation of the balance equations into a system integral equations by integration over an arbitrary volume.
2.  Applying the Gauss-Green theorem for the second term transforms the integral divergence of the flux to a surface integral.
3.  The third step is the semi-discretization of the spatial coordinates of the problem. The discretization of the flux terms can be done depending on the problem using central differences or upstream methods. The surface integral is then transformed into a sum of the flux over all surfaces k of the control volume.
4.  Finally, the ODE-system is integrated in time by using the numerical solver package CVODE which is available from LLNL software libraries. CVODE a generic solver with a BDF implementation and achieves very good simulation performance with acceptable programming effort. The CVODE solver is part of the SUNDIALS system (Hindmarsh et al. , 2005) and is a variable-order, variable-step multi-step solver.

1. Integration over arbitrary volume (REV-subscript omitted)

$$\int_V \frac{\partial \rho^E}{\partial t} dV = \int_V \frac{\partial}{\partial x}(j^E) dV + \int_V \sigma^E dV$$

2. Transformation of r.h.s. by Gaussian Theorem

$$\int_V \frac{\partial \rho^E}{\partial t} dV = \int_A j^E \vec{n} dA + \int_V \sigma^E dV$$

3. Semi-discretization in space (transformation in a system of ODEs)

$$\frac{d\rho_i^E}{dt} \Delta V_i = \sum_k j_{i,k}^E \vec{n} \Delta A_{i,k} + \sigma_i^E \Delta V_i$$

$i = 1 \dots N$ *(number of volume elements)*
$k = 1 \dots S_i$ *(number of sides per element)*

System of **O**rdinary **D**ifferential **E**quations (ODEs) describing the temporal change of primary state variables in each volume element

Example: $\quad E = \{m_{w+v}, m_a, m_{voc}, U\}$

$$N = 1000$$

would result in 4000 ODEs to be solved.

4. Integration in time using a mathematical solver

$$\rho_i^E(t) = \int_0^t \left( \frac{1}{\Delta V_i} \sum_k j_{i,k}^E \vec{n} \Delta A_{i,k} + \sigma_i^E \right) \cdot dt$$

CVODE (Hindmarsh@LLNL) solving

$$\dot{Y} = f(Y, t)$$

## *Solver interaction*

The CVODE integrator applies a generic multi-step/multi-value method including Newton-Iteration for ordinary differential equations. This method requires interaction with the calling program (where the solver is embedded) for establishing a time integration loop. A second dependency exists through the necessity to generate a
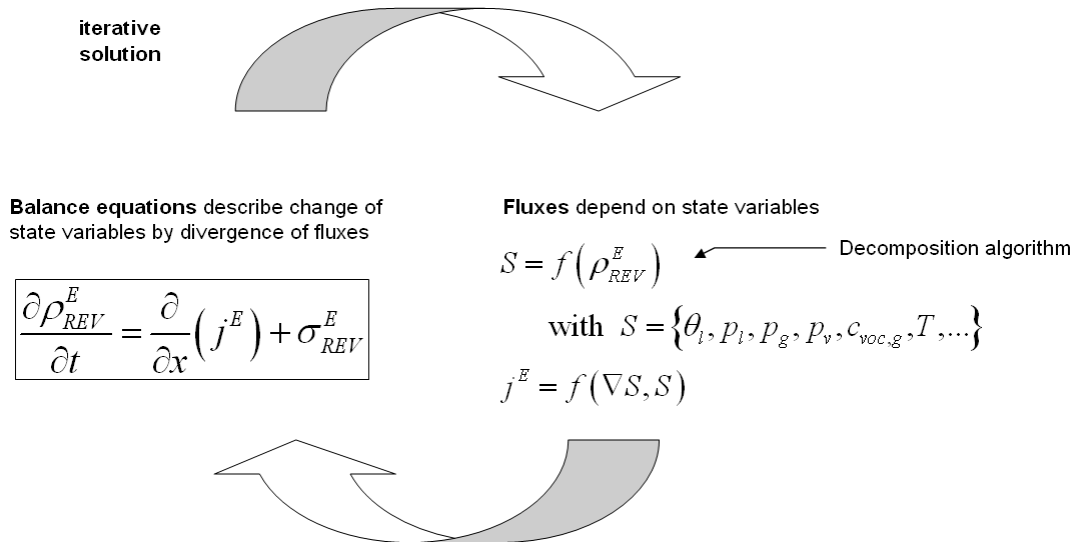
Jacobian matrix that describes the answer of the system on local disturbances.

The time frame for integration is set up by the calling program (here the CHAMPS-BES-solver). It processes the user input for the desired total simulation time and the time intervals for output of the state variables and derived quantities. The CVODE solver is called for each time interval between outputs (either in NORMAL or in SINGLE STEP mode). CVODE performs the time integration internally by using shorter time steps depending on the current convergence criteria.

To estimate the temporal changes of the unknowns in the system, CVODE calculates a numerical approximation of the Jacobian matrix. It applies local disturbances to the system calling a user-supplied function f(...). This f-function is used to calculate and return the divergence in flux depending on the current state variables and the time. For calculation of the flux, the primary state variables (as total volumetric mass densities and internal energy density) must be transformed into secondary state variables (derived quantities as volumetric fraction of the liquid and gas phases, pressures, concentrations, temperature, ...). This step is called decomposition algorithm which forms an important part of the numerical solution since basic decisions on the physics involved are made in this implementation.

The decomposition algorithm (described below in more detail) and the consecutive calculation of the transport coefficients and the gradients of the driving potentials allow the determine the flux and their divergence. The procedure is applied iteratively until convergence is reached for the current time step.

**Numerical solution (Semi-discretisation, Finite Control Volume Method - CVM)**



**iterative solution**

**Balance equations** describe change of state variables by divergence of fluxes

$$\frac{\partial \rho_{REV}^E}{\partial t} = \frac{\partial}{\partial x}\left(j^E\right) + \sigma_{REV}^E$$

**Fluxes** depend on state variables

$$S = f\left(\rho_{REV}^E\right)$$

Decomposition algorithm

$$\text{with } S = \left\{\theta_l, p_l, p_g, p_v, c_{voc,g}, T, ...\right\}$$

$$j^E = f\left(\nabla S, S\right)$$

## 4.3.2 Independent CHAMPS fluxes

### *Definition of independent CHAMPS flux expressions*

In order to derive a closely solvable system of equations the number of flux expression must be reduced. Considering convection and diffusion processes of all involved components (water vapor, liquid water, dry air and VOC) in all fluid phases (liquid and gaseous) one arrives at nine flux expressions in total. Not all of them are independent. The convective flux of water vapor and VOC depend both on the convective flux of the gas phase, for example. Eliminating both a new flux is introduced and the number of flux expressions is reduced by one.

For all nine related CHAMPS flux expressions defined below seven of them can be eliminated introducing three new ones which results in five independent flux expressions. For elimination the mass concentrations in the liquid and gas phases as well as the specific internal energy and enthalpy of the the liquid and gas phases must be known. The calculation of the concentrations, specific internal energy and enthalpy can be done either directly together with the flux calculation or in the frame of the decomposition algorithms described below.

## Convective and diffusive fluxes

$$j^E = j^E_{conv} + j^E_{diff} \quad \text{total } E\text{-flux}$$

$$j^E_{conv} \quad \text{convective } E\text{-flux}$$

$$j^E_{diff} \quad \text{diffusive } E\text{-flux}$$

**Independent CHAMPS-fluxes**

1. water vapor diffusion ($kg/m^2s$)
2. voc diffusion ($kg/m^2s$)
3. convective gas flux ($kg/m^2s$)
4. convective liquid water flux ($kg/m^2s$)
5. heat conduction ($W/m^2$)

## in case of CHAMPS ...

moisture mass
$$\frac{\partial}{\partial t}\rho^{m_{w+v}}_{REV} = \frac{\partial}{\partial x}\left[ j^{m_w}_{conv} + j^{m_v}_{diff} + j^{m_v}_{conv} \right] + \sigma^{m_{w+v}}_{REV} \qquad (1)$$

air mass
$$\frac{\partial}{\partial t}\rho^{m_a}_{REV} = \frac{\partial}{\partial x}\left[ j^{m_a}_{conv} \right] + \sigma^{m_a}_{REV}$$

pollutant mass
$$\frac{\partial}{\partial t}\rho^{m_{voc,l+g}}_{REV} = \frac{\partial}{\partial x}\left[ j^{m_{voc,g}}_{conv} + j^{m_{voc,g}}_{diff} + j^{m_{voc,l}}_{conv} \right] + \sigma^{m_{voc}}_{REV} \qquad (2)$$

internal energy
$$\frac{\partial}{\partial t}\rho^U_{REV} = \frac{\partial}{\partial x}\left[ j^U_{conv} + j^U_{diff} \right] + \sigma^U_{REV}$$

**Elimination of dependent fluxes**

9 introduced by balance equations
− 7 eliminated
+ 3 newly introduced
= 5 independent

$$j_{conv}^{m_v} = c^{m_v} \, \underline{j_{conv}^{m_g}}$$

$$j_{conv}^{m_a} = c^{m_a} \, j_{conv}^{m_g}$$

$$j_{conv}^{m_{voc,g}} = c^{m_{voc,g}} \, j_{conv}^{m_g}$$

③

mass concentrations in gas phase (kg/kg$_{gas\ phase}$)

$$j_{conv}^{m_w} = c^{m_w} \, \underline{j_{conv}^{m_l}}$$

$$j_{conv}^{m_{voc,l}} = c^{m_{voc,l}} \, j_{conv}^{m_l}$$

④

mass concentrations in liquid phase (kg/kg$_{liq\ phase}$)

$$j_{conv}^{U} = u_g \, j_{conv}^{m_g} + u_l \, j_{conv}^{m_l}$$

specific internal energy of gas and liquid phases (J/kg)

$$j_{diff}^{U} = \underline{j_{diff}^{Q}} + h_v \, j_{diff}^{m_v} + h_{voc,g} \, j_{diff}^{m_{voc,g}}$$

⑤

specific enthalpy of water vapor and gaseous voc (J/kg)

## 4.3.3 Decomposition algorithms

### *Physical models to establish the CHAMPS decomposition*

The decomposition algorithm as introduced above forms an essential part of the numerical solution. It should be described in very detail to clarify the physics involved and the mathematical equation implemented. The decomposition algorithm starts basically with input from CVODE to the user's f-function that comprises the current solution of the unknowns of the CHAMPS-system:
1. total volumetric moisture mass density
2. volumetric dry air mass density
3. volumetric pollutant mass density
4. total internal energy density

Solution for unknown variables from solver

$$\rho_{REV}^{E} = \left\{ \rho_{REV}^{m_{w+v}}, \rho_{REV}^{m_a}, \rho_{REV}^{m_{voc}}, \rho_{REV}^{U} \right\}$$

The decomposition algorithms for the heat and moisture balances, the air balance and the pollutant balance are applied sequentially. First the moisture mass density is decomposed into state variables as written below. This requires an iterative calculation since the vapor density in the reference volume depends on temperature and moisture content. Its initial estimate is set to zero. After a few iterations it converges normally.

Knowing the temperature, the pressures and the volumetric contents of the phases the

hygrothermal driving forces and transport coefficients can be determined. Driving forces are the gradients of liquid water pressure, water vapor pressure and temperature. For the liquid flux a Kirchhoff potential function is calculated as described below.

The equations used below imply some simplifications. For the temperature calculation, for example, minor heat storage terms as the contribution of the water vapor (except phase transmutation enthalpy) and VOC are neglected. The calculation of the logarithmic capillary pressure of the system makes use of the reverse moisture retention function which is a material function to be determined experimentally. Further functions involved are Kelvin equation for calculation of the relative humidity and the temperature dependent water vapor saturation pressure function.

## Decomposition algorithm for heat and moisture

Iterative calculation of secondary state variables

| | | |
|---|---|---|
| $\rho_{REV}^{m_v} = 0$ | Initial estimate of volumetric vapor density | $kg/m^3$ |
| $\rho_{REV}^{m_w} = \rho_{REV}^{m_{w+v}} - \rho_{REV}^{m_v}$ | Volumetric liquid water density | $kg/m^3$ |
| $\theta_l = \dfrac{\rho_{REV}^{m_w}}{\rho_l}$ | Volume fraction of liquid phase | $m^3/m^3$ |
| $T = \dfrac{\rho_{REV}^{U} - h_v \rho_{REV}^{m_v}}{c_{T,m} + c_{T,w}\rho_{REV}^{m_w} + c_{T,v}\rho_{REV}^{m_v}} - T_0$ | Temperature | $K$ |
| $pC = RMRC(\theta_l)$ | Logarithmic capillary pressure calculated from Reverse Moisture Retention Curve (material function) | - |
| $p_c = -10^{pC}$ | Capillary pressure | $Pa$ |
| $\varphi = e^{\frac{p_c}{\rho_l R_v T}}$ | Relative humidity | - |
| $p_v = \varphi \cdot p_{v,sat}(T)$ | Water vapor pressure | $Pa$ |
| $\theta_g = \theta_{por} - \theta_l$ | Volume fraction of gas phase | $m^3/m^3$ |
| $\rho_{REV}^{m_v} = \dfrac{p_v}{R_v T}\theta_g$ | Corrected volumetric vapor density | $kg/m^3$ |

Calculation of further state variables and related transport coefficients

| | | |
|---|---|---|
| $p_l = -p_c$ | Pressure in liquid phase | $Pa$ |
| $\psi_l = \psi_l(\theta_l)$ | Kirchhoff potential for liquid water flux | $kg/ms$ |
| $K_v = K_v(\theta_l)$ | Water vapor conductivity | $s$ |
| $\lambda = \lambda(\theta_l)$ | Thermal conductivity | $W/mK$ |

$$\frac{\partial \psi_l}{\partial x} = K_l(\theta_l)\ \frac{\partial p_c}{\partial x}$$

*Definition of the Kirchhoff potential*      $kg/m^2 s$

$$\psi_l(\theta_l) = \int_0^{\theta_l} \left( K_l(\theta_l)\frac{\partial p_c}{\partial \theta_l} \right) \cdot d\theta_l$$

*Calculation of the Kirchhoff potential*      $kg/ms$

The decomposition algorithms for the air mass and pollutant mass balances are straight forward procedures. They make use of already calculated variables of the heat and moisture decomposition. For the calculation of the intrinsic VOC density a reverse proportionality of the partition coefficient to the temperature dependent saturation density in the gas phase is assumed. Since the partition coefficient is known (measured) at reference temperature the ratio of saturation densities is used to correct for the actual temperature.

### Decomposition algorithm for air mass balance

Calculation of secondary state variables

$$p_a = \frac{\rho_{REV}^{m_a}}{R_a T}\frac{1}{\theta_g}$$

*Air pressure*      $Pa$

Calculation of further state variables and related transport coefficients

$$K_g = K_g(\theta_l)$$

*Air permeability*      $s$

### Decomposition algorithm for VOC mass balance

Calculation of secondary state variables

$$\rho_g^{m_{voc,g}} = \frac{\rho_{REV}^{m_{voc,l+g}}}{k_{part}^{voc}}\frac{\rho_{g,sat}^{m_{voc,g}}(T)}{\rho_{g,sat}^{m_{voc,g}}(T_{ref})}$$

*Intrinsic VOC density in gas phase*      $kg/m^3$

$$p_{voc} = \rho_g^{m_{voc,g}} R_{voc} T$$

*VOC pressure in gas phase*      $Pa$

Calculation of further state variables and related transport coefficients

$$K_{voc} = K_{voc}(T,\theta_l)$$

*VOC permeability*      $s$

## 4.3.4    Fixing numerical problems

When modeling coupled heat and moisture problems you may sometimes run into problems or difficulties resulting from the complexity of the physical problem to be solved. This chapter will address some of the likely causes of unexpected results or errors in the solver.

NOTE: If the solver fails to run at all, in about 99% of the cases the error is either in the project file or material data. If the solver produces unrealistic results the reason is most likely an error in the input data or modeling assumptions.

### Simulation warnings

TODO : Add detailed descriptions of possible errors and causes...

Overshot/Undershot warning at begin of simulation (for instance for adsorption/desorption simulations)
The calculated divergences are too large. Try reducing the initial time step.

Overshot/Undershot warning during simulation (steep gradients, e.g. in driving rain simulations).
The calculated divergences are too large. Try different relative/absolute tolerances.

### Simulation aborts

Try one or more of the following things:
- reduce the relative tolerance
- reduce maximum time step gradually, e.g. from 1h, to 10 min, to 1 min, to 1 s etc. and see if the problem persists
- turn on diagnostic output (monitor file, verbose level 4 etc.)

### Numerical Issues

Simulation runs very slow.
Try different relative/absolute tolerances. Ensure smooth material functions. Enable interpolation for climate data.
(many different reasons can slow down the solver, a single solution cannot be given).

# 5 Appendix

## 5.1 Wind driven rain modeling

### Introduction

The development of the method for determination of the catch ratios and its experimental validation was done by Bert Blocken in his PhD work and Jan Carmeliet, KU Leuven. For more detailed information please refer to the publication below. Here, a short description of the numerical method of Wind Driven Rain (WDR) modeling and its implementation in the CHAMPS code is provided.

# Spatial and temporal distribution of driving rain on a low-rise building

Bert Blocken[†] and Jan Carmeliet[‡]

*Department of Civil Engineering, Laboratory of Building Physics, Katholieke Universiteit Leuven, Kasteelpark Arenberg 51, 3001 Leuven, Belgium*

Wind driven rain modeling provides a practical numerical method to determine both the spatial and temporal distribution of driving rain on buildings. It is based on an existing numerical simulation technique and uses the building geometry and climatic data at the building site as input. The method is applied to determine the 3D spatial and temporal distribution of wind-driven rain on the building facade.

## 5.1.1 Determination of catch ratios

### *Definitions and parameters*

The joint occurrence of wind and rain causes obliquity of the rain intensity vector. In general, driving rain intensity refers to this oblique rain vector. From the viewpoint of the interaction between rain and vertical building facades, the term driving rain intensity takes on the narrower meaning of component of the rain vector causing rain flux through a vertical plane. The other component of the rain vector, that causes rain flux through a horizontal plane, is termed (horizontal) rainfall intensity. The quantities used to describe the driving rain load are the specific catch ratio, related to the raindrop diameter, and the catch ratio, related to the entire spectrum of raindrop diameters.

Rain intensity vector
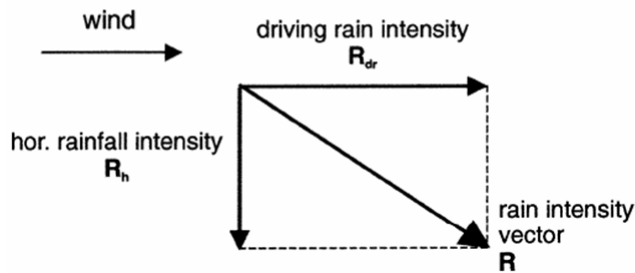


Specific catch ratio, Catch ratio

$$\eta_d(t) = \frac{R_{dr}(d, t)}{R_h(d, t)}, \qquad \eta(t) = \frac{R_{dr}(t)}{R_h(t)}$$

$R_{dr}(d, t)$     specific driving rain intensity

$R_h(d, t)$     specific unobstructed horizontal rainfall intensity for raindrops with diameter $d$.

$R_{dr}(t)$, $R_h(t)$    refer to the same quantities but integrated over all raindrop diameters.

## *CFD-Simulation technique for determination of catch ratios*

CFD-Simulation technique for determination of catch ratios is a steady-state technique, comprising four steps:
> *(1) The steady-state wind flow pattern around the building is calculated using a CFD code.*
> *(2) Raindrop trajectories are obtained by injecting raindrops of different sizes in the calculated wind flow pattern and solving their equations of motion.*
> *(3) The specific catch ratio is determined based on the calculated raindrop trajectories.*
> *(4) The catch ratio is calculated from the specific catch ratio and the size distribution of raindrops.*

The calculation of the specific catch ratio and catch ratio is performed for a selected number of zones on the building envelope. For each zone, the same procedure is employed. In a steady-state wind flow pattern - thus neglecting turbulent dispersion of raindrops - raindrop trajectories of diameter ending on the corner points of the zone form a stream tube.

Steady-state simulation technique

(1)  Wind flow pattern around the building using a
     CFD code

CFD

(2)  Raindrop trajectories obtained by injecting
     raindrops of different sizes

$R_h(d)$

(3)  Specific catch ratio (for specific sizes) based on
     the calculated raindrop trajectories

$A_h(d)$

Stream tube bounded by
raindrop trajectories

$$\eta_d^n(d) = \frac{R_{dr}(d)}{R_h(d)} = \frac{A_h(d)}{A_f}$$

$R_{dr}(d)$

$A_f$

for one single value of wind speed and
rainfall intensity.

(4)  Catch ratio calculated from the
     raindrop size distribution

Raindrop size distribution
through a horizontal plane

$$\eta^n = \int_d f_h(d)\eta_d^n(d)\,\mathrm{d}d$$

parameter rainfall intensity

valid for warm front rain
(no thunder storm)
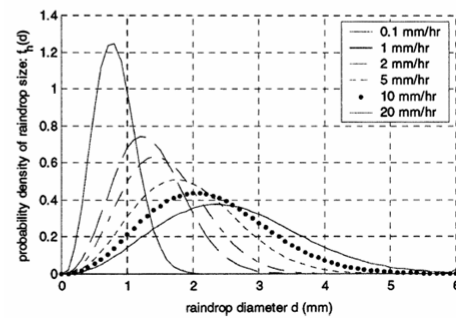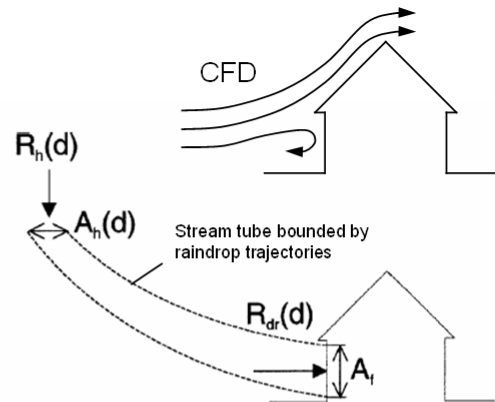


$A_f$ is the area of the zone on the building envelope where the specific catch ratio is to be determined, and $A_h(d)$ is the area of the horizontal plane bounded by the injection positions of the raindrops of diameter $d$ ending on the corner points of $A_f$.

This plane is located in the undisturbed wind flow. Its location must allow the raindrops injected at that position to reach their terminal velocity of fall (vertical) and the wind velocity (horizontal) before entering the flow pattern disturbed by the presence of the building. The catch ratio for a zone is obtained by multiplying the specific catch ratio for each raindrop diameter for this zone with the fraction of these drops in the spell (raindrop size distribution) and integrating over all raindrop diameters.

## *Calculation of rain drop size distribution through a horizontal plane*

For calculation of the fraction of liquid water in the air with raindrops of diameter less than $d$, the formula of Best (1950) is adopted.

Raindrop size distribution *in the air*

The formula of Best (1950) is adopted, based
on the extent of the study carried out:

• wide bibliographical survey

• measurements for a large number of rain events

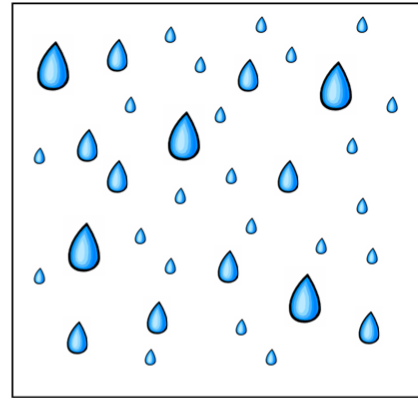$$F(d) = 1 - \exp\left(-\left(\frac{d}{a}\right)^n\right), \quad a = AR_h^p, \quad f(d) = \frac{\mathrm{d}F}{\mathrm{d}d}$$



F(d)  fraction of liquid water *in the air* with raindrops of
      diameter less than *d*

Parameters:
$A = 1.30$
$n = 2.25$
$p = 0.232$

Best, A.C. (1950), "The size distribution of raindrops", *Q. J. Roy. Meteor. Soc.*, **76**, 16-36.

Due to the variation of the terminal velocity of fall of a raindrop with size, the raindrop
size distribution *in the air* differs from the raindrop size distribution *through a horizontal
plane*. The former can be converted to the latter by multiplying with the raindrop
terminal velocity of fall. The terminal velocity of fall
for raindrops as measured by Gunn and Kinzer is depicted below along with the fitted
curve.

Raindrop size distribution *through a horizontal plane*
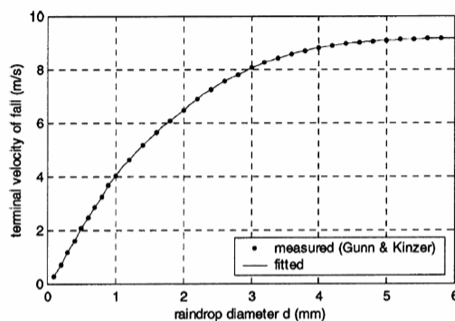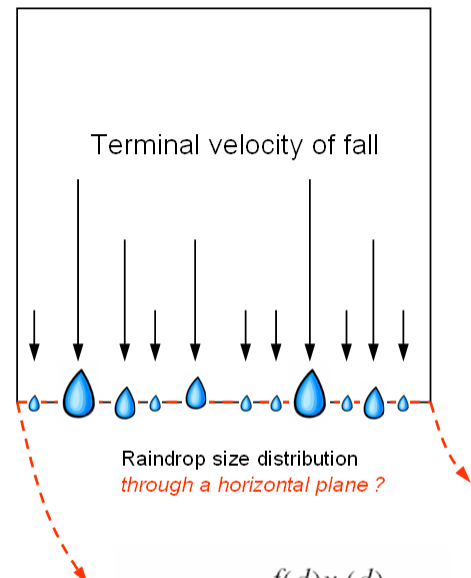


Terminal velocity =
f(Raindrop size)

$v_1$ , $d_1$

$v_2$ , $d_2$

$v_3$ , $d_3$

…

Terminal velocity of fall

wind

stream tube

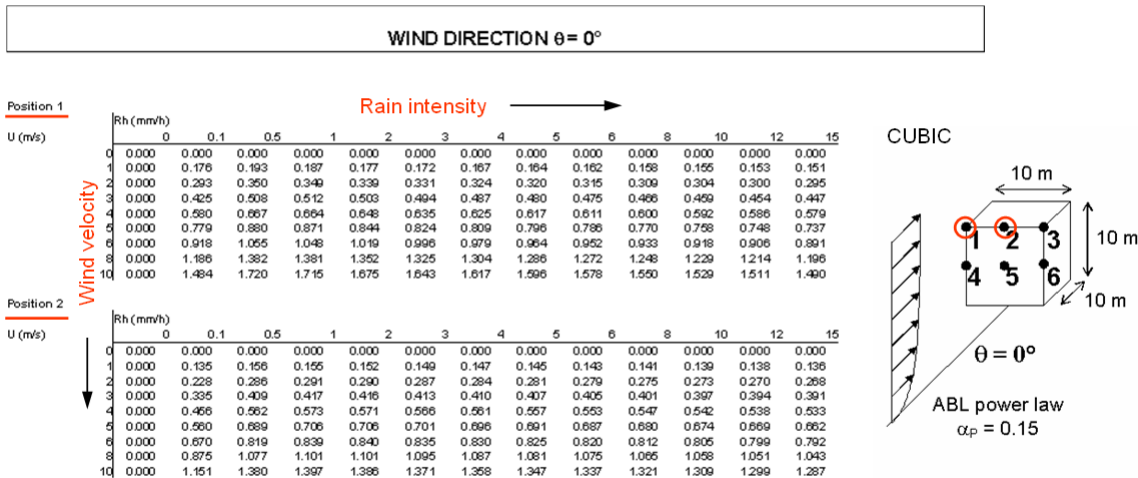Raindrop size distribution
*through a horizontal plane ?*

Conversion of raindrop size distribution *in the air* to that
*through a horizontal plane* by weighting with the
raindrop terminal velocity of fall

$$f_h(d) = \frac{f(d)v_t(d)}{\int\limits_d f(d)v_t(d)\mathrm{d}d}$$

## Calculation of catch ratio tables

Taking into account the rain modeling considerations, a steady-state simulation technique is used to estimate both the spatial and temporal distribution of driving rain on buildings for transient rain events. The spatial distribution of driving rain is obtained by determining the driving rain load at each position of the building envelope. The temporal distribution results from performing these calculations for user defined discrete time steps. For a given building geometry, position on the building envelope and wind direction, the catch ratio is a function of wind speed, rainfall intensity and raindrop size distribution. When the raindrop size distribution of Best is adopted, a one to one relation exists between rainfall intensity and the raindrop spectrum, causing the wind speed and rainfall intensity values to unambiguously define a numerical catch ratio.

The steady-state simulation technique can be used to calculate the numerical catch ratio under steady-state conditions, i.e., for one single value of wind speed and rainfall intensity. To determine the numerical catch ratio for a transient rain event, with fluctuating wind speed and rainfall intensity, this event is partitioned into a number of equidistant time steps, each of which is considered steady state.

Example

**WIND DIRECTION θ = 0°**

Position 1 — Rain intensity — CUBIC

$U$ (m/s) / $Rh$ (mm/h):

| U \ Rh | 0 | 0.1 | 0.5 | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 10 | 12 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 1 | 0.000 | 0.176 | 0.193 | 0.187 | 0.177 | 0.172 | 0.167 | 0.164 | 0.162 | 0.158 | 0.155 | 0.153 | 0.151 |
| 2 | 0.000 | 0.293 | 0.350 | 0.349 | 0.339 | 0.331 | 0.324 | 0.320 | 0.315 | 0.309 | 0.304 | 0.300 | 0.295 |
| 3 | 0.000 | 0.425 | 0.508 | 0.512 | 0.503 | 0.494 | 0.487 | 0.480 | 0.475 | 0.466 | 0.459 | 0.454 | 0.447 |
| 4 | 0.000 | 0.580 | 0.667 | 0.664 | 0.648 | 0.635 | 0.625 | 0.617 | 0.611 | 0.600 | 0.592 | 0.586 | 0.579 |
| 5 | 0.000 | 0.779 | 0.880 | 0.871 | 0.844 | 0.824 | 0.809 | 0.796 | 0.786 | 0.770 | 0.758 | 0.748 | 0.737 |
| 6 | 0.000 | 0.918 | 1.055 | 1.048 | 1.019 | 0.996 | 0.979 | 0.964 | 0.952 | 0.933 | 0.918 | 0.906 | 0.891 |
| 8 | 0.000 | 1.186 | 1.382 | 1.381 | 1.352 | 1.325 | 1.304 | 1.286 | 1.272 | 1.248 | 1.229 | 1.214 | 1.196 |
| 10 | 0.000 | 1.484 | 1.720 | 1.715 | 1.675 | 1.643 | 1.617 | 1.596 | 1.578 | 1.550 | 1.529 | 1.511 | 1.490 |

Position 2

$U$ (m/s) / $Rh$ (mm/h):

| U \ Rh | 0 | 0.1 | 0.5 | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 10 | 12 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 1 | 0.000 | 0.135 | 0.156 | 0.155 | 0.152 | 0.149 | 0.147 | 0.145 | 0.143 | 0.141 | 0.139 | 0.138 | 0.136 |
| 2 | 0.000 | 0.228 | 0.286 | 0.291 | 0.290 | 0.287 | 0.284 | 0.281 | 0.279 | 0.275 | 0.273 | 0.270 | 0.268 |
| 3 | 0.000 | 0.335 | 0.409 | 0.417 | 0.416 | 0.413 | 0.410 | 0.407 | 0.405 | 0.401 | 0.397 | 0.394 | 0.391 |
| 4 | 0.000 | 0.456 | 0.562 | 0.573 | 0.571 | 0.566 | 0.561 | 0.557 | 0.553 | 0.547 | 0.542 | 0.538 | 0.533 |
| 5 | 0.000 | 0.560 | 0.689 | 0.706 | 0.706 | 0.701 | 0.696 | 0.691 | 0.687 | 0.680 | 0.674 | 0.669 | 0.662 |
| 6 | 0.000 | 0.670 | 0.819 | 0.839 | 0.840 | 0.835 | 0.830 | 0.825 | 0.820 | 0.812 | 0.805 | 0.799 | 0.792 |
| 8 | 0.000 | 0.875 | 1.077 | 1.101 | 1.101 | 1.095 | 1.087 | 1.081 | 1.075 | 1.065 | 1.058 | 1.051 | 1.043 |
| 10 | 0.000 | 1.151 | 1.380 | 1.397 | 1.386 | 1.371 | 1.358 | 1.347 | 1.337 | 1.321 | 1.309 | 1.299 | 1.287 |

10 m · 10 m · 10 m

θ = 0°

ABL power law
$\alpha_P = 0.15$

... more positions and wind directions

The catch ratio is a function of space and time:

$$\eta = f(BType, Pos, \theta, U, R_h)$$

| $BType$ | building geometry (including environment topology) |
|---|---|
| $Pos$ | position on the building envelope |
| $\theta$ | undisturbed wind direction in 10m height |
| $U$ | undisturbed wind speed in 10m height |
| $R_h$ | (horizontal) rainfall intensity |
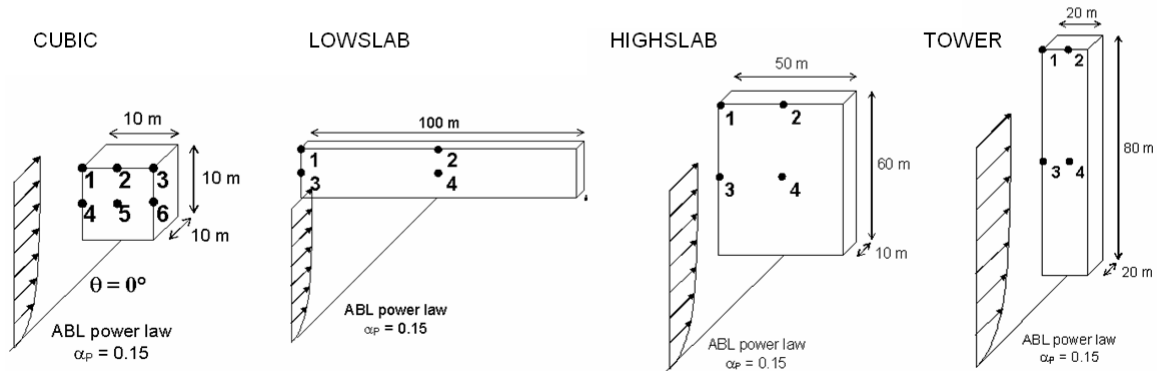


## 5.1.2 WDR-implementation in simulation codes

### *Usage of catch ratio tables in CHAMPS-BES simulations*

To reduce the computational expense, the time consuming steady-state technique will only be employed for a limited number of wind speed and rainfall intensity couples, while intermediate results will be obtained by linear interpolation. Interpolation in a 2D $U$-$R_h$ space provided by an area-weighted computation yields two catch ratio values (for two wind directions). Then, interpolation between these catch ratio values is performed to find the value for the wind direction. Nonlinear interpolation method for wind directions should be discussed!!

Interpolation in a 2D $U$-$R_h$ space provided by an area-weighted computation

Rain intensity $\longrightarrow$

$R_h$

Wind velocity

$\eta_{k,l}$     $\eta_{k,l+1}$

$A_{k,l}$

$U$

$\eta = f(U, R_h)$
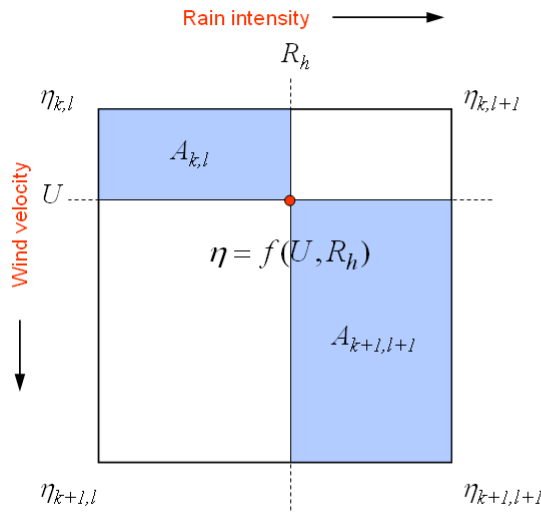
$A_{k+1,l+1}$

$\eta_{k+1,l}$     $\eta_{k+1,l+1}$

Numerical catch ratio

$$\eta = \frac{1}{A}(\eta_{k,l} \cdot A_{k+1,l+1}$$

$$+\eta_{k,l+1} \cdot A_{k+1,l}$$

$$+\eta_{k+1,l} \cdot A_{k,l+1}$$

$$+\eta_{k+1,l+1} \cdot A_{k,l})$$

$$A = (U_{k+1} - U_k) \cdot (R_{l+1} - R_l)$$

$$A_{k,l} \ldots A_{k+1,l+1} \rightarrow accordingly$$

This yields two catch ratio values (for two wind directions). Then, interpolation between these catch ratio values is performed to find the value for the wind direction.

Nonlinear interpolation method for wind directions to be discussed!!

Since values for the horizontal rain intensity are given by discrete data points only, the temporal interpolation should be discussed, too. Step function or linear interpolation is usually used for interpolation of climate file data given on hourly basis. As shown below, the interpolation method has no influence on the total rain amount. Linear interpolation is recommended because avoiding jumps in boundary conditions result in better numerical performance of the simulation.

Assume a given horizontal rain intensity data file ...



Have time scale and interpolation method influence on WDR-calculation?

----- Step function

—— Linear interpolation
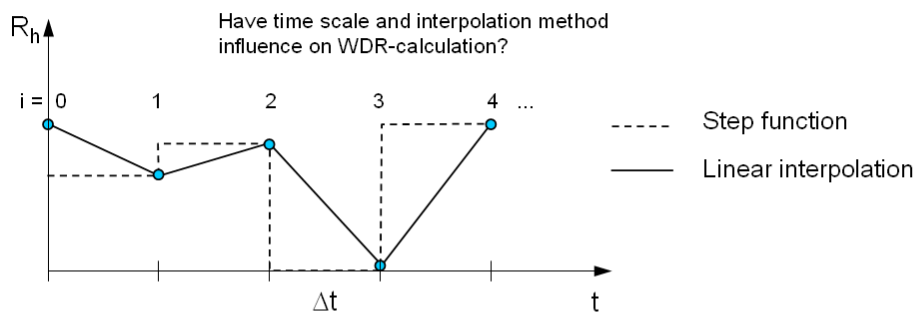
Total rain amount ...

$$S^{(step)} = \Delta t \cdot \sum_{i=1}^{n} R_{h,i}$$

$$S^{(lin)} = \underbrace{\Delta t \cdot \sum_{i=1}^{n} R_{h,i}}_{S^{(step)}} + \underbrace{\Delta t \cdot \sum_{i=1}^{n} \frac{R_{h,i-1} - R_{h,i}}{2}}_{\Delta t \frac{R_{h,0} - R_{h,n}}{2}}$$

Conclusions

- Interpolation method has no influence
- Equidistant time steps
- Same time scale for $R_h$, $\eta$ and $R_{dr}$

with    $R_{h,0} = R_{h,n}$

we get   $S^{(step)} = S^{(lin)}$

## *Numerical modeling of water uptake by rain events*

By application of the modeling considerations above, the driving rain load normal to the wall surface can be calculated as function of time. The real water uptake flux can be different due to different saturation degrees of the material and its reduced ability the take up the rain water. Rain water not penetrating the wall is considered to run off.

The maximum water uptake flux is calculated from the material parameters (liquid water conductivity or diffusivity) and a gradient in capillary pressure or moisture content, respectively. For sake of convenience, a surface transfer coefficient can be introduced as shown below. The water uptake flux can be determined by comparison of the maximum water uptake flux and the driving rain intensity normal to the wall.

Maximum water uptake

Driving rain
intensity $R_{dr,nor}$

transfer
coefficient

$\gamma_K, \gamma_D$

Surface
water
content

Water content $\theta_l$

$\theta_{surf} = \theta_{sat}$

Maximum water uptake

$R_{max}$

$\Delta x / 2$
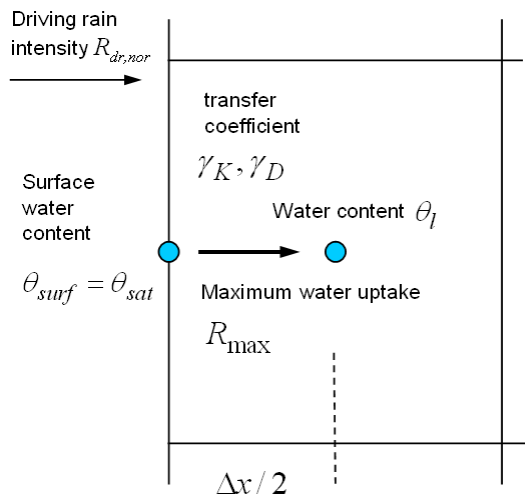
Boundary volume element of a wall construction

Approaching volume saturation the water
uptake ability of the wall is reduced.

1. Immediate saturation is assumed at surface in case of rain
events. The surface moisture value $\theta_{surf}$ is **not** part of the
numerical solution.

2. A transfer coefficient in the boundary element is
calculated from the liquid water conductivity of the material.

$$\gamma_K = 2 K_l(\theta_l)/\Delta x \qquad \text{or} \qquad \gamma_D = 2\rho_l D_l(\theta_l)/\Delta x$$

(conductivity approach)          (diffusivity approach)

3. The maximum influx is calculated from the difference in
capillary pressure (surface - boundary element). By
definition, the capillary pressure at saturation is zero.

$$R_{max} = \gamma_K \cdot p_c(\theta_l) \qquad \text{or} \qquad R_{max} = \gamma_D \cdot (\theta_{sat} - \theta_l)$$

(conductivity approach)          (diffusivity approach)

$$R_{uptake} = Min(R_{dr,nor}, R_{max})$$

# 5.2    Radiation modeling

### *Physical background*

The intensity of radiation (energy density of radiation - physical unit is W/m$^3$) is a
function of its wave length and the temperature. The energy density  of radiation
increases with higher temperatures and it shows a distinctive maximum between 1 ... 3 $\mu$
m that shifts toward lower wave length. This dependency is called Max Planck's radiation
law of black bodies. As an practical approach in building physics, one differentiates
between short wave radiation and long wave radiation. Short wave radiation comprises
the visible sun radiation and a certain range above (0.2 ... 3 $\mu$m). The long wave
radiation (3 ..30 $\mu$m) is commonly denoted as heat radiation.

The integration of Max Planck's radiation law of black bodies over the wave length
delivers a radiation flux density as function of temperature to the power of four. This
dependency is called Stefan Boltzmann's radiation law; its proportionality factor is the
Boltzmann constant. Knowing the surface temperature of bodies, it is then possible to
establish radiation balances.
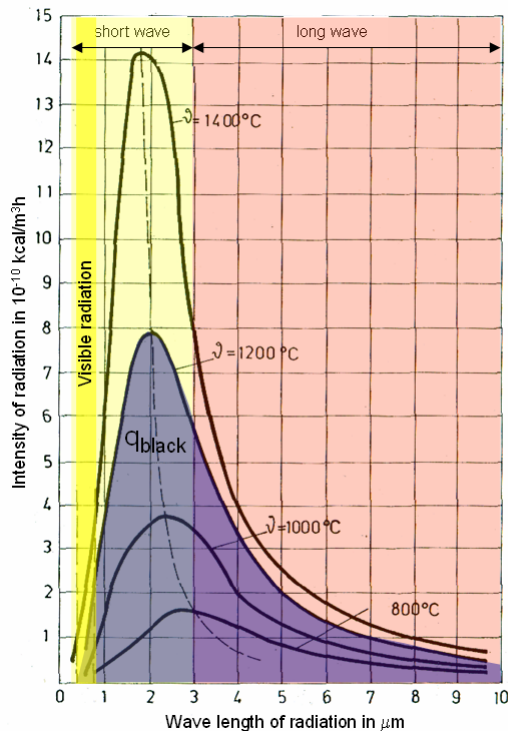
**PLANCK's radiation law of black bodies**

Old physical unit: $kcal/m^3h * 1.142 = W/m^3$

**STEFAN BOLTZMANN's law**    $q_{black} = \sigma * T^4$

| | | |
|---|---|---|
| $q_{black}$ | $W/m^2$ | Emission of a black body |
| $\sigma = 5.67*10^{-8}$ | $W/m^2K$ | BOLTZMANN's constant |
| T | K | Temperature |
| | | |
| $q = \varepsilon * C_{black} * (T/100)^4$ | | Emission of a gray body |
| $\varepsilon$ | --- | Degree of blackness |
| $C_{black} = 5.67$ | $W/m^2K$ | Radiation constant |

## General remarks about radiation components

Beside the air temperature, the short wave radiation and long wave radiation are two climate components having major influence on the thermal behavior of the building envelope and the building as whole. Both, the short wave radiation and long wave radiation consist of two components, the direct and diffuse sun radiation as well as the long wave emission and atmospheric counter (sky) radiation. For a proper modeling of the heat balance at building surfaces, all these radiation components have to be taken into account.

While the direct short wave radiation is a *directed* radiation, all other radiation components are *not directed*. The mechanisms of transformation of direct radiation into diffuse radiation are adsorption and dispersion (extinction) in the atmosphere (by ozone, carbon dioxide, water vapor, dust, pollution). The short wave radiation is a net influx (gain) of energy from the viewpoint of the building. The long wave radiation is always expressed as a balance, a difference between adsorption and emission.

The building establishes radiation equilibrium with the sky and the surrounding ground. While the sky can be modeled as an spheric half-infinite room, the ground can be seen as an flat plane perpendicular to vertical building surfaces. From the viewpoint of the building, there is a net gain of energy by direct and diffuse reflection of short wave radiation from the ground. At the same time, as the building itself, the ground acts as an long wave radiator.
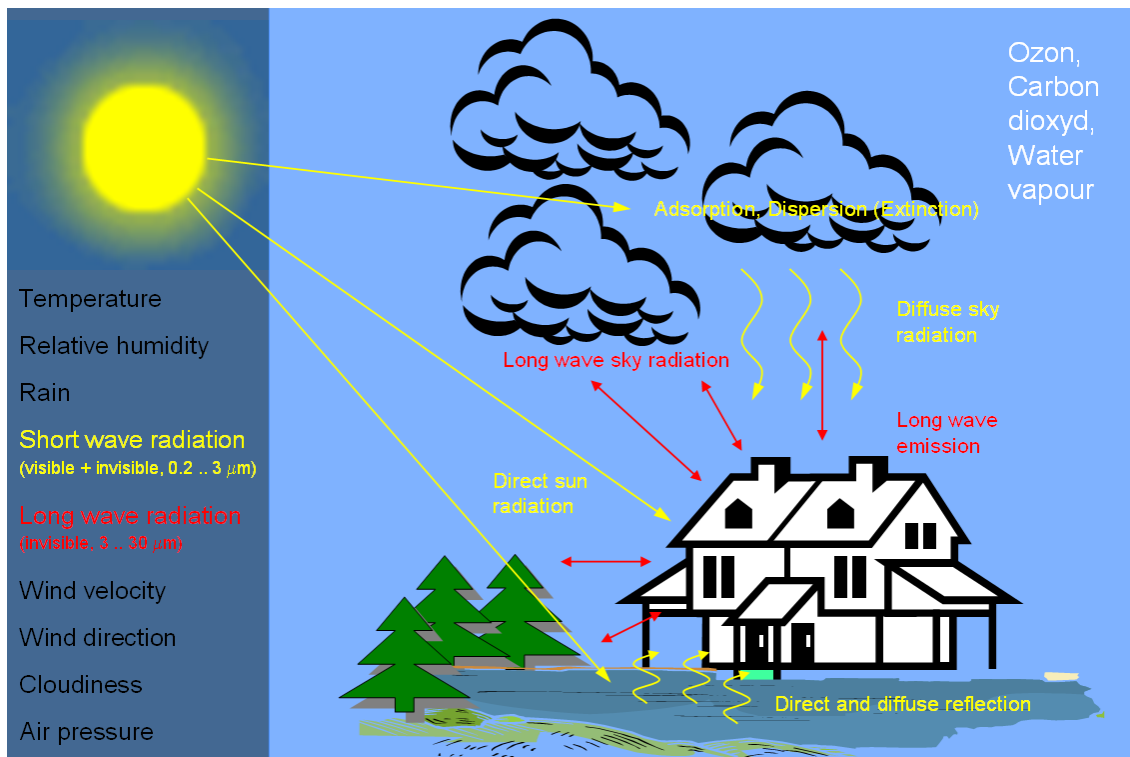
## *Radiation balances at the building surface*

Radiation balances can be established at the exterior surface of the building or the building component. According the absorption coefficient of the wall surface, the short wave sun radiation is divided into an adsorbed part and a reflected part. The absorbed radiation flux is taken up by the building element but only a part of it, the heat load contribution from the building element can be transferred to the indoor room. The heat load is reduced by convective heat transport (ventilation of the building element) and by higher long wave emission with higher surface temperature.

The long wave radiation balance is determined by the building component's emission coefficient and the surface temperature, the albedo and temperature of the surrounding ground and the cloudiness, relative humidity and temperature of the sky. For sake of simplification, the ground temperature and the sky temperature are usually set equal to the outdoor air temperature.

Climatic components

- Temperature
- Relative humidity
- Rain
- Short wave radiation*
- Long wave radiation*
- Wind velocity
- Wind direction
- Cloudiness
- Air pressure
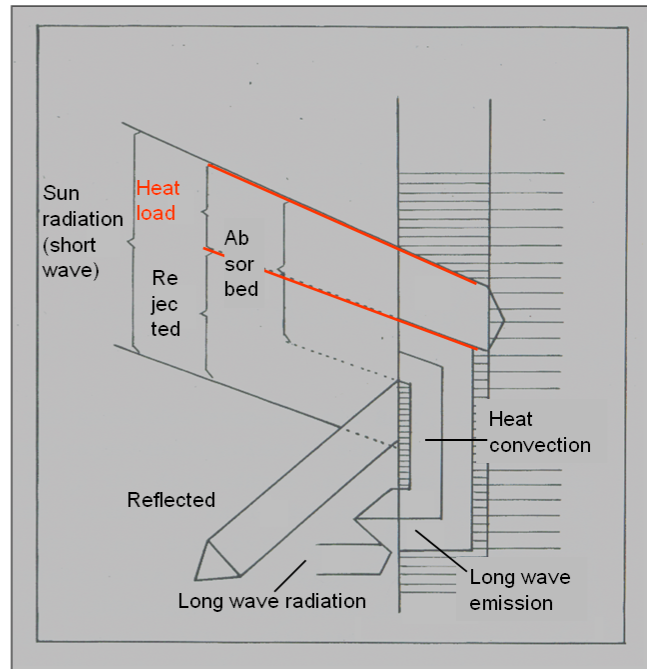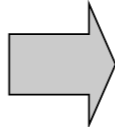
## 5.2.1   Experimental study

### *Effects of sun radiation on building elements*

The relevance of the effects of sun radiation on the thermal behavior of buildings can be demonstrated by an experimental study. In this study the short wave sun radiation is measured by a pyranometer. At the same time, the air temperature and the surface temperature of the building component have been measured as function of time. Knowing both temperatures, the overheating of the wall surface by radiation, the Surface-Overtemperature can be calculated.

Under adiabate conditions (no heat transfer to the interior) the Surface-Overtemperature would reach a maximum called Radiation-Overtemperature. The Surface/Radiation-Overtemperature ratio is a measure for the temperature amplitude attenuation that depends on the thermal storage capacity of the building element. Heavy buildings can store more heat than lightweight constructions; their temperature amplitude attenuation would be higher.

With knowledge of the Surface/Radiation-Overtemperatures, the long wave emission and the heat convection it is possible to calculate the heat load and the absorption coefficient of the building element. The study carried out in the frame of the Doctoral Thesis of Prof. Roloff, TU Dresden yielded absorption coefficients for a variety of surface colors and saturation degrees.

Measured:
Short Wave Sun Radiation
by pyranometer

Surface-Overtemperature = $(T_{surf} - T_{air})$
overheating of the wall surface

Calculated:
Attenuation coefficient
= SurfOvertemp / RadOvertemp

Radiation-Overtemperature
= Surface-Overtemperature but **without** heat load

Determined:
Heat load
= ExchCoeff * (RadOvertemp – SurfOvertemp)

Absorption coefficient
= absorbed heat flux / short wave radiation
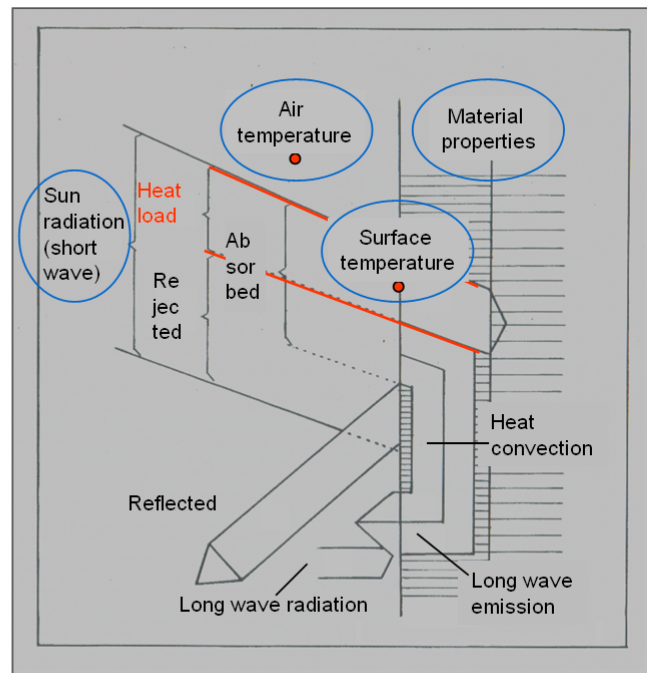


* PhD work of Prof. Dr.-Ing. Jürgen Roloff, 1970 (Institute of Building Climatology, TU Dresden)

## *Experimental results*

The first data plot below shows the influence of the surface color on the surface temperature of a heavy construction (concrete). The Surface-Overtemperature reaches from 7.5 K (white) to 30 K (black). The measurement was carried out at Jul 2-3, 1968 in Dresden, Germany.

The second data plot shows the influence of the heat storage capacity on the surface temperature. A difference of up to 9 K in Surface-Overtemperature between a heavy (concrete) and a lightweight construction (polystyrene) could be measured. A second effect could shown: the surface temperature falls below the air temperature in the night. This effect is called under-cooling by long wave emission in clear nights with little cloud covering. The measurement was carried out at Aug 14, 1968 in Dresden, Germany.

The third data plot shows again the influence of the heat storage capacity on the surface temperature. The under-cooling effect is relevant for lightweight constructions. All constructions on Polystyrene show under-cooling effects, the (blue marked) curve of the concrete construction stays clearly above the air temperature. The measurement was carried out at Aug 10-11, 1968 in Dresden, Germany.
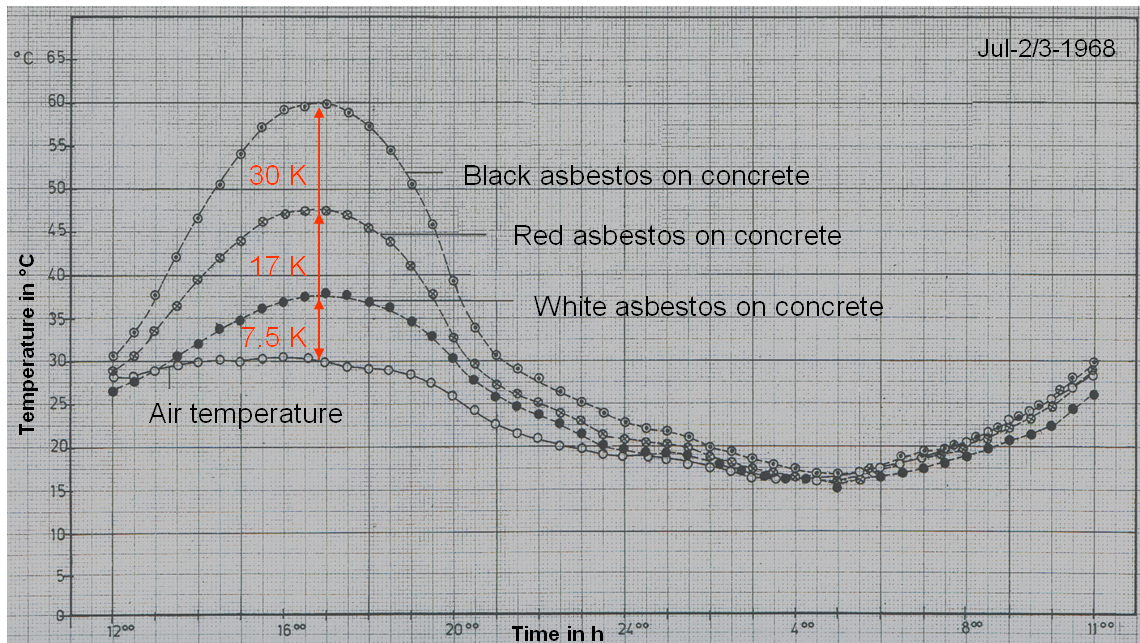
Propagation of temperature with time as function of surface color (absorption factor)
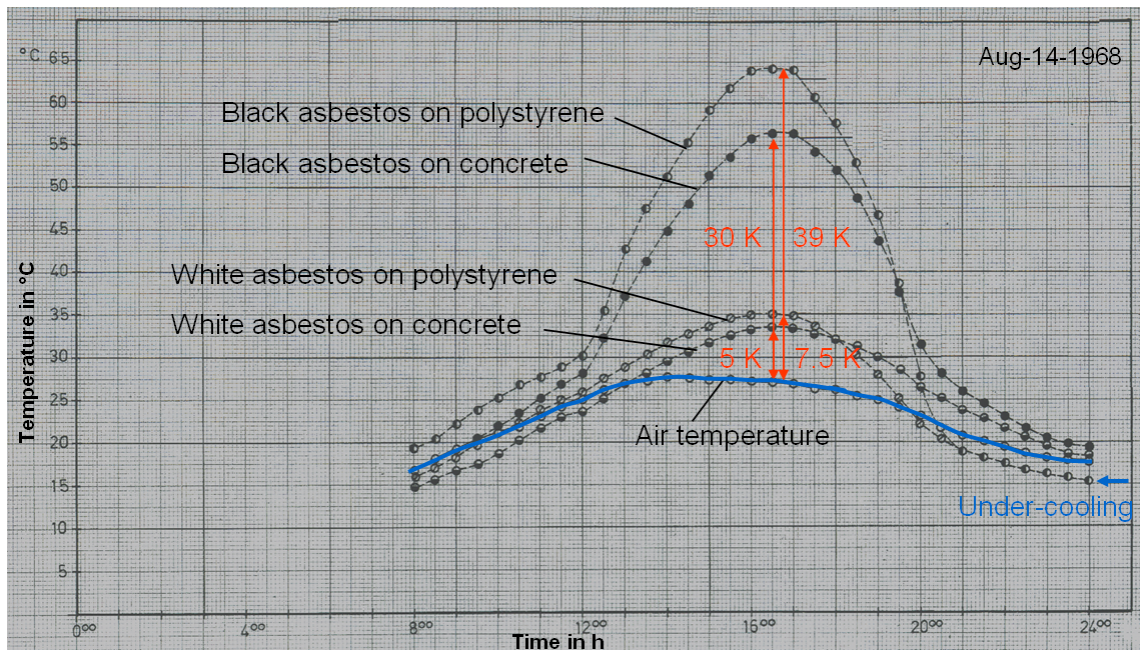


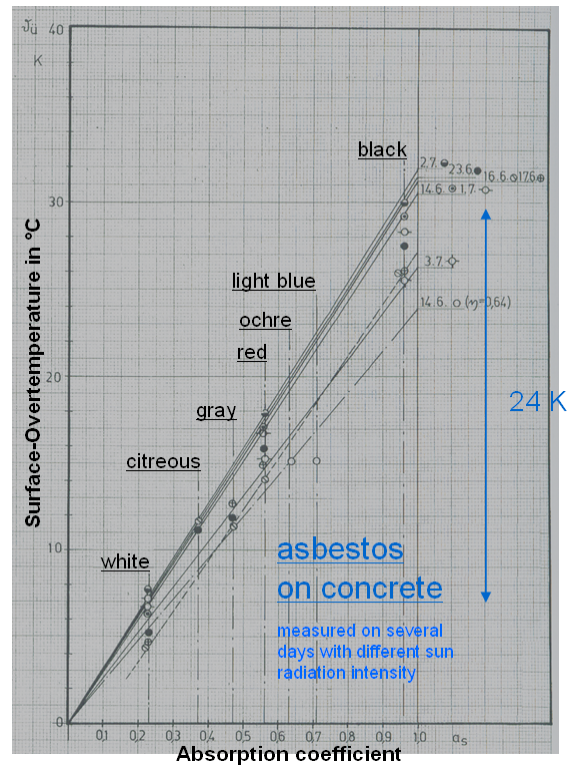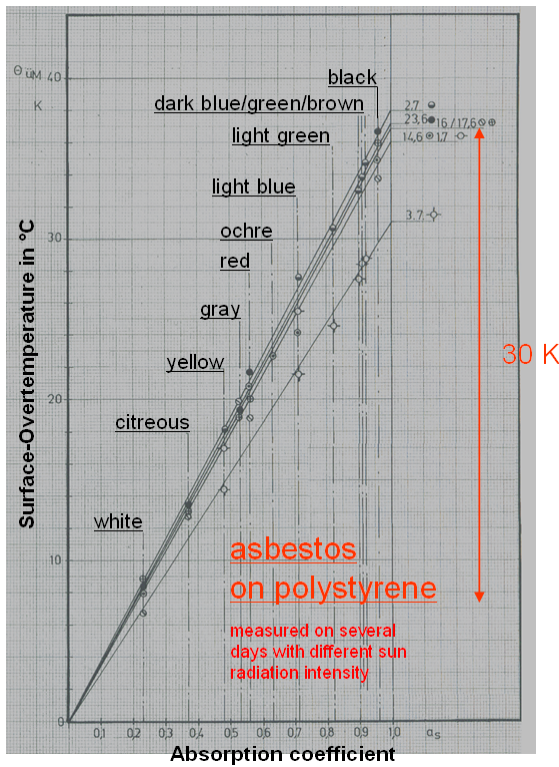Propagation of temperature with time as function of background material (attenuation factor)
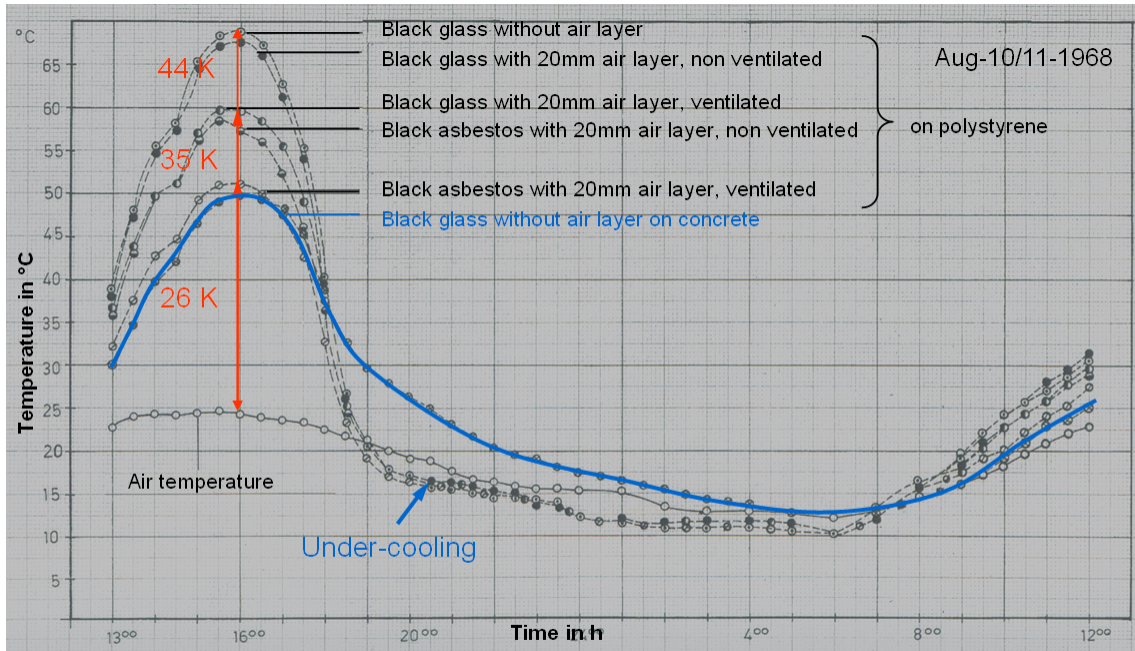
Propagation of temperature with time as function of constructional measures (air layer)

## Evaluation of results

As one result of the experimental study a relation between the Surface-Overtemperature and Absorption coefficients could be determined. This relation is basically linear but depends on the sun radiation intensity. The colors were assigned to the absorption coefficients, ranging from white to black.

Evaluation of the experimental results delivers absorption coefficients a function of the surface color, or better as function of the saturation degree of the surface color. The variation of the absorption coefficient within a color can be higher than within a specific saturation degree of different colors.

The table below is just an excerpt of the data gained from the experimental study and literature reviews.

Left chart: Surface-Overtemperature in °C vs Absorption coefficient — "asbestos on polystyrene, measured on several days with different sun radiation intensity" (30 K). Labels: black, dark blue/green/brown, light green, light blue, ochre, red, gray, yellow, citreous, white.

Right chart: Surface-Overtemperature in °C vs Absorption coefficient — "asbestos on concrete, measured on several days with different sun radiation intensity" (24 K). Labels: black, light blue, ochre, red, gray, citreous, white.

| Painting | Absorption coefficient |
|---|---|
| Oil white | 0.15 .. 0.24 |
| Lime white | 0.15 .. 0.30 |
| Beige | 0.30 .. 0.40 |
| Creamy | 0.28 .. 0.44 |
| Straw colored | 0.33 .. 0.45 |
| Chrome yellow | 0.33 .. 0.45 |
| Gold | 0.41 .. 0.56 |
| Light green | 0.33 .. 0.60 |
| Dark green | 0.78 .. 0.90 |
| Grass green | 0.80 .. 0.88 |
| Middle brown | 0.59 .. 0.73 |
| Light blue | 0.45 .. 0.69 |
| Turquoise | 0.85 .. 0.90 |
| Ultramarine | 0.90 .. 0.93 |
| Light red | 0.45 .. 0.68 |
| Dark red | 0.73 .. 0.90 |
| Vermilion | 0.80 |
| Brick red | 0.90 |

- Adsorption data available for approximately 400 materials

- Emission data available for approximately 130 materials

- Reflection coefficients for 24 kinds of surrounding area (albedo) reported

Color is less important but degree of saturation has more influence on absorption

## 5.2.2 Short wave radiation

### *The sun as source of short wave radiation*

The source of the short wave radiation arriving at the top of our atmosphere is the nuclear fusion process taking place in the sun. In the center of the sun and under high pressure and high temperature hydrogen is transmuted to helium. The exogenous reaction is a permanent source of energy that will last for the next five billion years; the half of the estimated live cycle of the sun.

A fractional amount of the sun energy is transmitted to the surface of the earth by radiation. The radiation flux density at top of the atmosphere is called solar constant. The solar constant is a function of time (season) that depends on solar-system and terrestrial influencing factors.

The radiation that passes through the atmosphere undergoes an extinction process (dispersion, reflection and adsorption) and only a certain percentage of it arrives at the surface of the earth. The normal radiation is the direct radiation flux density measured on a surface perpendicularly directed to the sun (one imagine a screen that moves with the sun). Dispersed and reflected radiation reappears as diffuse radiation which is still a (non-directed) kind of short wave radiation.

Both, direct and diffuse short wave radiation are balanced together at the building surface. The reflectivity of the surrounding is taken into account, too. Since the radiation is measured usually on a horizontal surface, an angle transformation from the horizontal component to the component normal to the wall surface is required for simulations.



Emission of radiative energy by the sun (nuclear fusion)
hydrogen > helium
*Surface of the sun*

„Solar constant"   ~ declination and ecliptical lenght of the sun (season)
~ time shift (longitude, time zone)
*Above the atmosphere*

Normal radiation   ~ sun height (latitute)
~ azimuth (day time)
~ absorption, dispersion, extinction (elevation)
~ opaqueness factor (air pollution)
*Surface of the earth*

Direct radiation   ~ inclination and orientation of the surface

Diffuse radiation   ~ inclination of the surface
~ albedo (diffuse reflection of surrounding)
*Surface of the building*

## Solar system-related and terrestrial influencing factors on solar constant

One of the influencing factors on the solar constant S is the **velocity of circulation** of the earth in the orbit. It changes with the season by about 1 hour. The graphics below show time shift against time for the location Tampa in Florida.
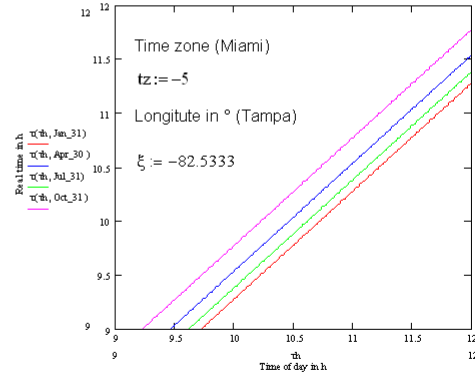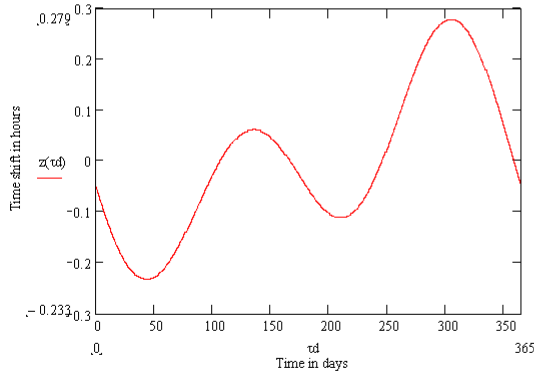
**Velocity of circulation changes with time**

Earth

Earth orbit

Periphel

Sun

77,94°

Equator

Vernal equinox
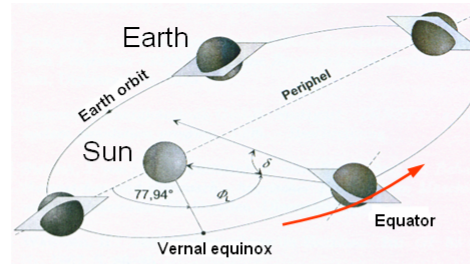
**Time zone and longitude determine real time**

$$\tau(\tau h, \tau d) := \tau h + z(\tau d) + \frac{1}{15} \cdot \xi - tz$$

**Hour angle** $\gamma h$ in $°$

$$\gamma h(\tau h, \tau d) := 15 \cdot \tau(\tau h, \tau d)$$



Time zone (Miami)

$$tz := -5$$

Longitude in ° (Tampa)

$$\xi := -82.5333$$

$$z(\tau d) := 1.1 \cdot 10^{-4} + 0.123 \cos G \left( 2 \cdot \tau d \cdot \frac{180}{365} + 85.88 \right) + 0.166 \cos G \left( 4 \cdot \tau d \cdot \frac{180}{365} + 108.896 \right) + 0.005645 \cos G \left( 6 \cdot \tau d \cdot \frac{180}{365} + 195.195 \right)$$

A second influencing factor on the solar constant S is the **declination** of the sun. The declination of the sun is the angle between the equatorial plane of the earth and the earth orbit plane. It changes by more than 40° over the year.

Slope of the equatorial plane changes with time (relativly to the earth orbit plane)

= Declination of the sun
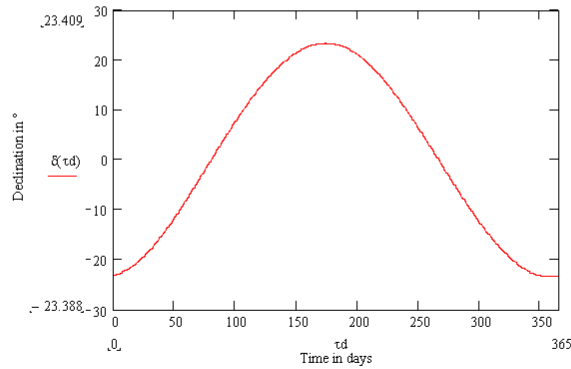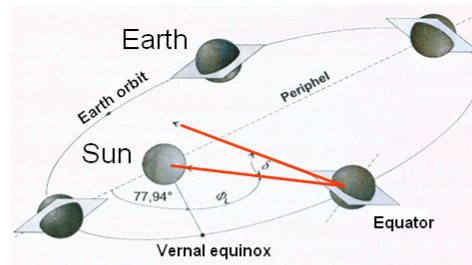


$$\delta(\tau d) := 0.395 - 23.256 \cos\!\left(2 \cdot \tau d \cdot \frac{180}{365} + 9.053\right) - 0.392 \cos\!\left(4 \cdot \tau d \cdot \frac{180}{365} + 5.329\right) - 0.176 \cos\!\left(6 \cdot \tau d \cdot \frac{180}{365} - 10.084\right)$$

The third influencing factor on the solar constant S is the **ecliptical length**, the angle between current position of the earth and that at spring begin. With ecliptical length the distance between sun and earth changes. The three influencing factors together cause an annual variation of the solar constant by about 7% or 100 W/m$^2$. The solar constant S reaches a maximum in summer time (mid of year) and a minimum in winter time (end of year).
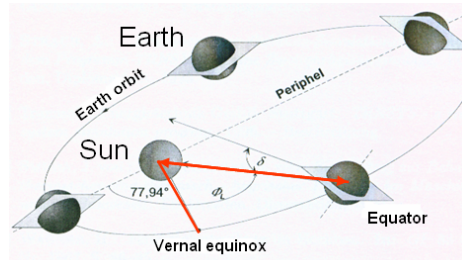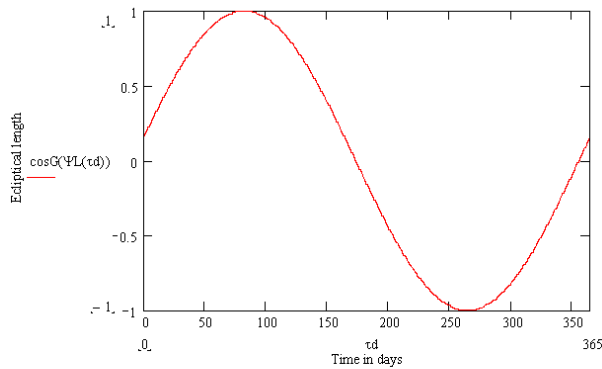
$$\Psi L(\tau d) := 0.986(\tau d - 2.875) + 1.914 \sin G\lfloor 0.017(\tau d - 2.875)\rfloor - 77.94$$
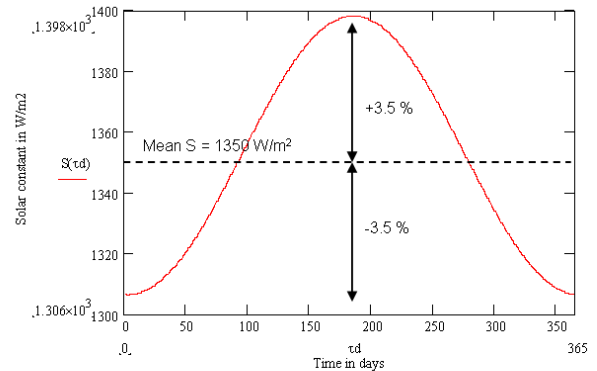
$$S(\tau d) := 1352\left(1 - 0.017\cos G(\Psi L(\tau d) + 77.94)\right)^2$$

## *Solar and local reference systems*

For calculation of the sun radiation a s function of position (on the earth) and time the introduction of a reference system is necessary. There are two reference systems: the **equatorial** system that introduces the solar angles **hour angle** and **declination** and the **horizontal** system that introduces the terrestrial angles **azimuth** and **sun height**. The hour angle and the azimuth describe the angle related to the rotation of the earth. The declination and the sun height are a measure for the current vertical position of the sun.

The equatorial system uses a plane parallel to the equator. The horizontal system relates to a horizontal plane from the viewpoint of the respective location on the earth. Therefore, the relation between the angles of both reference systems depend on the respective location.
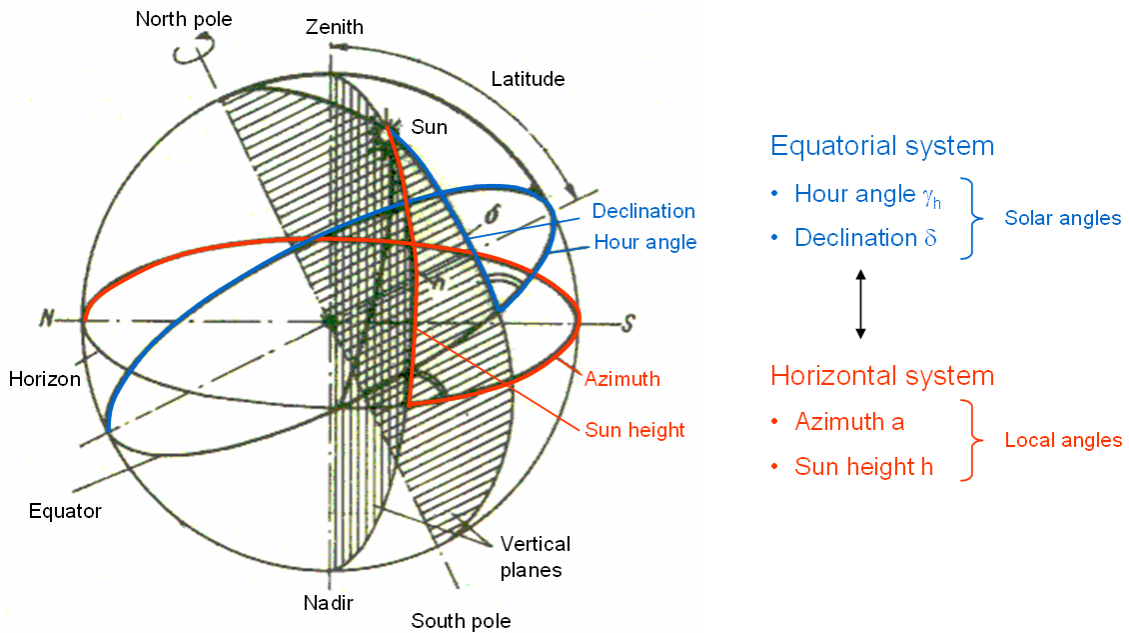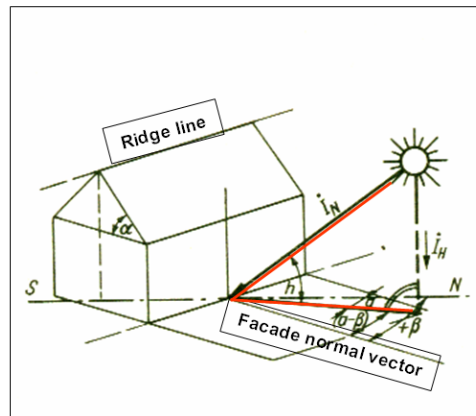
For radiation calculations the angles of the horizontal system are relevant. The definition of the local angles azimuth and sun height at the building surface coincides with that of the horizontal reference system (see sketches below). Knowing the influencing factors one can develop a set of formulas to calculate the direct sun radiation as function of location and time (still without influences of the atmosphere).
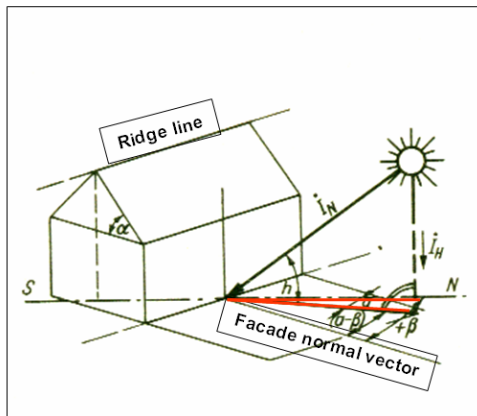
<span style="color:red">Azimuth a</span>　　　　　　　　　　　<span style="color:red">Sun height h</span>



$$\sin\_a(\tau h,\tau d):=\frac{\cos G(\delta(\tau d))\cdot\sin G(\gamma h(\tau h,\tau d))}{\cos G(h1(\tau h,\tau d))}$$

$$\sin\_h(\tau h,\tau d):=\sin G(\phi)\cdot\sin G(\delta(\tau d))-\cos G(\phi)\cdot\cos G(\delta(\tau d))\cdot\cos G(\gamma h(\tau h,\tau d))$$

### *Atmospheric influencing factors*

Another influencing factor on the sun radiation is the atmosphere itself or better, its

thickness, humidity and degree of pollution.

First, the elevation or altitude of the respective location matters. Since locations can vary by 8 km in altitude and a simple estimation of the thickness of the atmosphere (by knowledge of the total air mass and application of the barometric formula) gives a value of about 10 km, the conclusion is justified that it may have a major influence. The graphics below shows the percentage of radiation arriving at the surface of the earth. In the morning (sun rise) and in the evening (sun set) the traveling distance of the sun beam elongates due to low values of sun height. Elongated traveling distance means higher extinction.

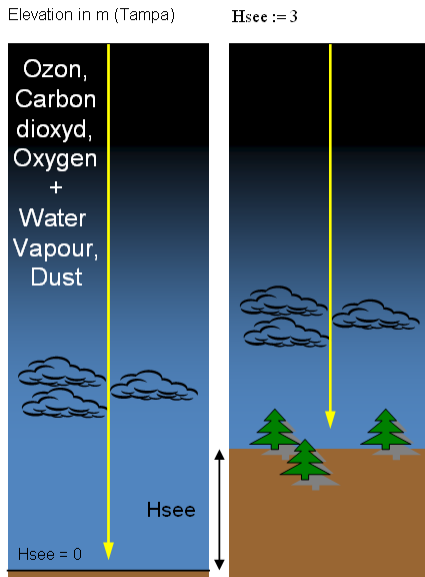Extinction of sun radiation by clean/dry atmosphere
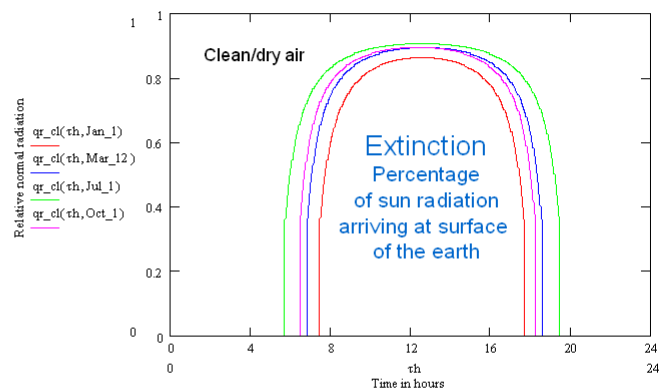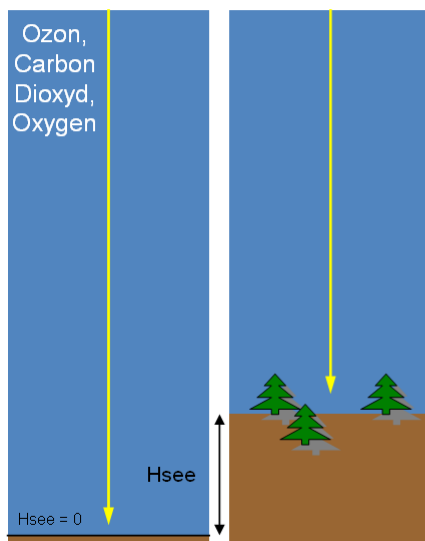


$$Q(\tau h, \tau d) := \frac{9.381 \left( \sin\_h(\tau h, \tau d) + \sqrt{1.04 \cdot 10^{-4} + \sin\_h(\tau h, \tau d)^2} \right)}{2.002 \left( 1 - \text{Hsee} \cdot 10^{-4} \right)} + 0.912$$

$$qr\_cl(\tau h, \tau d) := \text{wenn}\left( \sin\_h(\tau h, \tau d) \geq 0, \exp\left( -\frac{1}{Q(\tau h, \tau d)} \right), 0 \right)$$

The previous consideration relates to a dry and clean atmosphere. To account for water vapor and dust one introduced the so-called opaqueness factor. Its definition is given by the quotient of logarithms of the radiation of a real and a clear/dry atmosphere. It can be expressed as the number of clean/dry atmospheres needed to have the same extinction effect as the real atmosphere.

<span style="color:#cc3300">Opaqueness of the atmosphere</span>

Elevation in m (Tampa)     $Hsee := 3$

Ozon,
Carbon
dioxyd,
Oxygen
+
Water
Vapour,
Dust

Hsee

Hsee = 0

<span style="text-decoration:underline">Definition of opaqueness factor</span>

$$q_{rel,nor} = q_{rel,clear}{}^{T}$$

$T = 3$

Real atmosphere

=
same
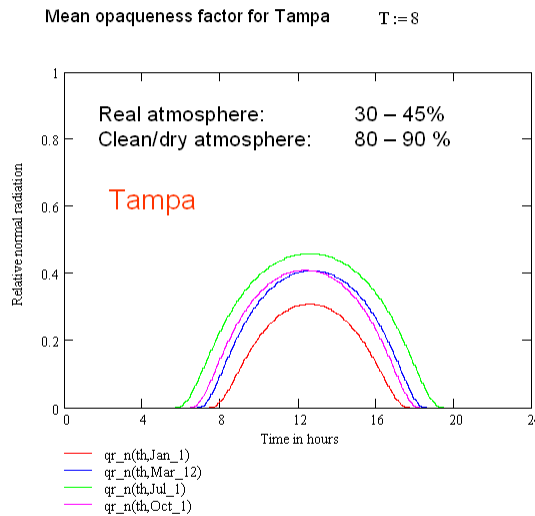extinction

Clean/dry atmosphere
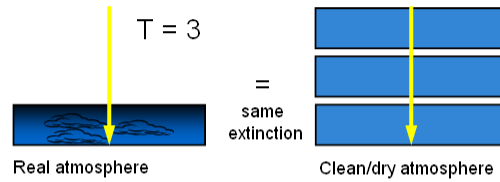
## *Short wave radiation modeling versus measurements*

Calculation of the short wave radiation from the considerations above requires estimation of the opaqueness factor. The graphics below show a comparison of the measured and calculated radiation for Tampa, FL with an estimated opaqueness factor of T=8. The estimation is based on the equality of the annual energy amount transmitted by short wave radiation.

The calculated radiation can reproduce the mean course of measured radiation quite well but it doesn't account for short term influences and local fluctuations.

Comparison of measured and calculated horizontal radiation

Mean opaqueness factor for Tampa   $T := 8$

Real atmosphere: $30 - 45\%$
Clean/dry atmosphere: $80 - 90\%$

Tampa
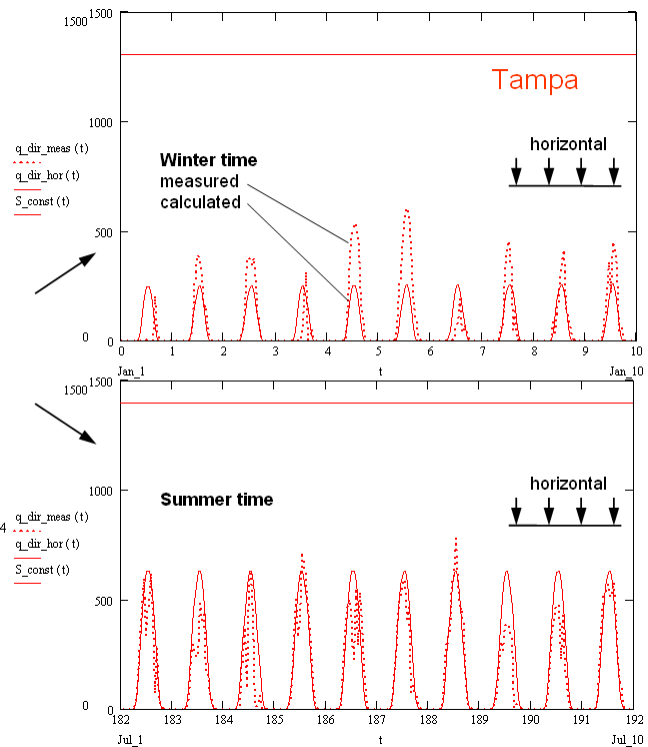
## *Direct sun radiation on building surfaces*

The horizontal component $q_{dir,hor}$ of the normal radiation is measured on a horizontal surface. Its vertical component $q_{dir,ver}$ can be calculated either from the horizontal component (in case of measured radiation) or both components are calculated from the normal radiation $q_{dir,nor}$ (in case of calculated radiation).The direct sun radiation on building surfaces is calculated from both, the horizontal and the vertical components of the normal radiation. Parameters are the slope of the building element and its orientation. The angle between azimuth and facade normal vector is called relative azimuth.

Knowing these parameters the radiation flux to the building surface is the sum of the contributions perpendicular to the building element from the the horizontal and the vertical components of the normal radiation. The radiation flux normal to a vertical wall (East, South and West) and to a flat roof are shown in the graphics below for the location Tampa, FL.

Angle relations of direct sun radiation
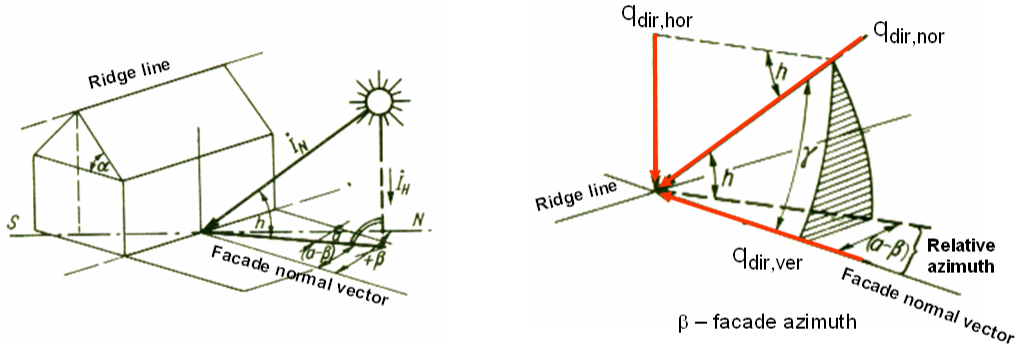


$$q_{dir,ver} = q_{dir,hor} * \cos(a-\beta)/\tan(h)$$

$$q_{dir,hor} = q_{dir,nor} * \sin(h)$$
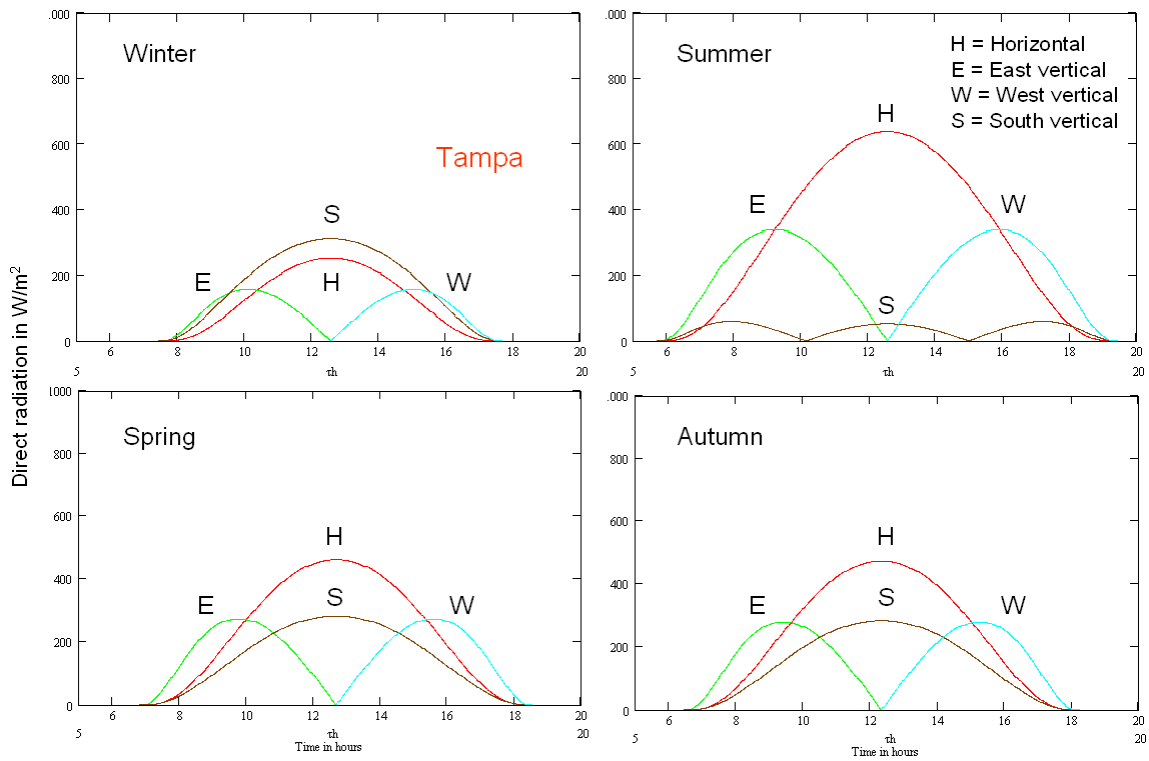


## *Diffuse sun radiation on building surfaces*

On crossing the atmosphere the direct sun radiation is reflected, adsorbed and dispersed. This process is called extinction. The direct sun radiation is partially transmuted into diffuse radiation; another part is reflected and doesn't reach the surface of the earth.

The "missing" radiation can be calculated from the difference between the solar constant and the direct radiation arriving at the surface of the earth. About one third of that "missing" radiation is transmuted into diffuse radiation. The rest is reflected back in the space.

A comparison of calculated and measured diffuse radiation is shown below. The calculated radiation can reproduce the mean course of measured radiation quite well but it doesn't account for short term influences and local fluctuations.
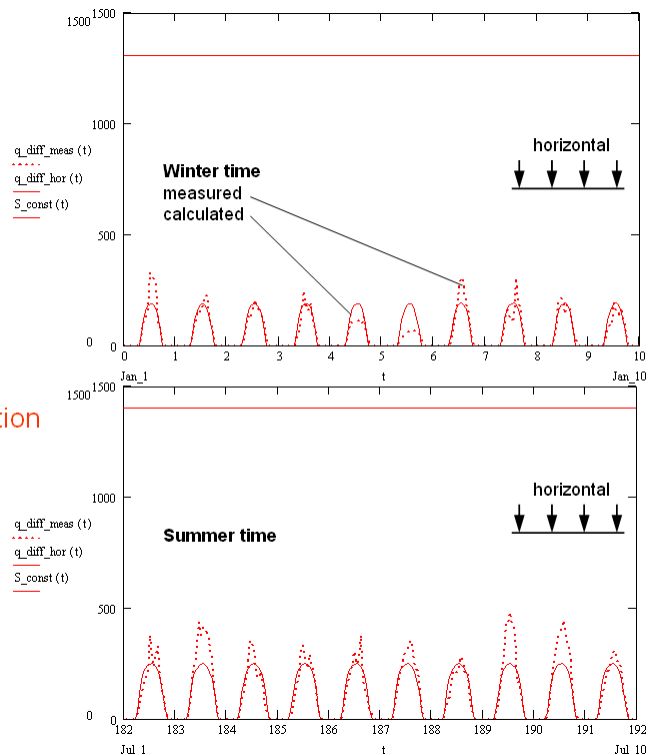
Diffuse radiation

Calculation

$$q_{diff,hor} = X*\underbrace{(S - q_{dir,norm})}_{Extinction + Opaquness}*sin(h)$$

$X = 33 \%$

About 33% of the absorbed/dispersed direct radiation arrives as diffuse radiation at the surface of the earth.



## 5.2.3  Long wave radiation

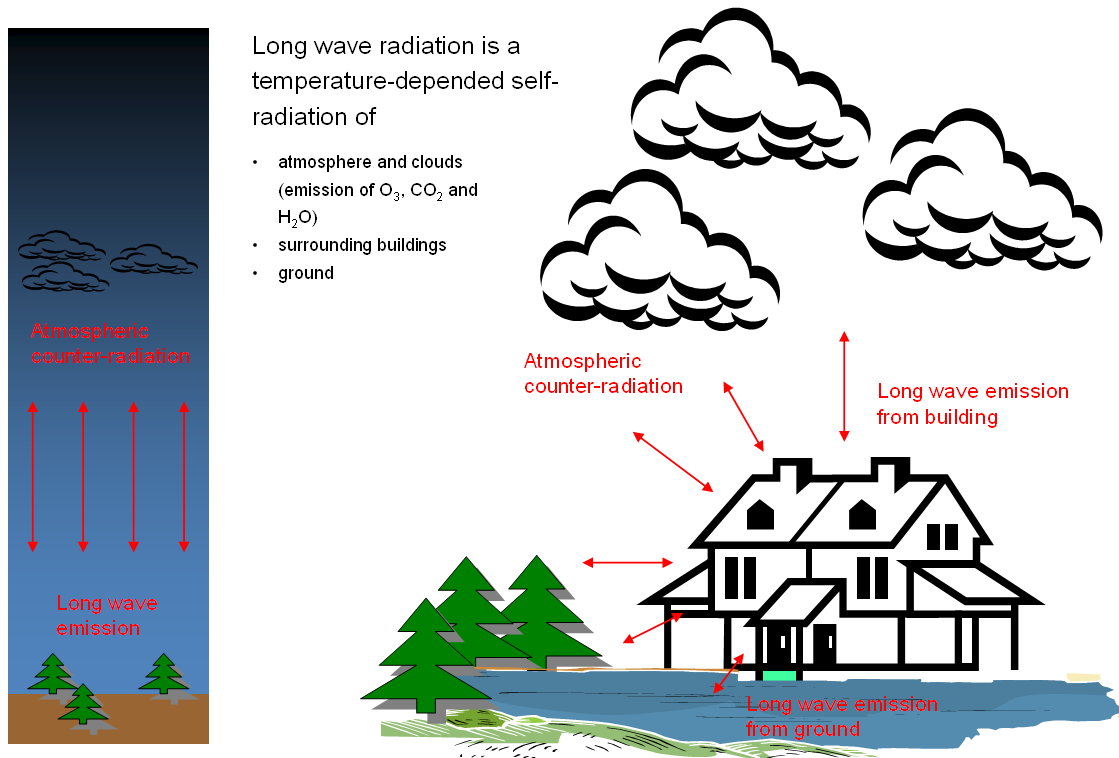### *Long wave radiation equilibrium between building, sky and ground*

The building establishes long wave radiation equilibrium with the sky and the surrounding ground. The building itself is seen as a long wave radiator but the sky and the ground as well. The radiation emitted by the building and the ground is called long wave emission, the radiation emitted by the sky is named atmospheric counter radiation or sky radiation.

The geometry of the building is known. It usually has plane, sloped and oriented surfaces. The sky can be modeled as an half-infinite spheric room or as a flat plane and the ground is regarded as an flat plane perpendicular to vertical building surfaces.

The long wave radiation balance is determined by the radiation equilibrium between

building, sky and ground. The influencing parameters are the emission coefficients of the building component and of the surrounding ground (albedo). The emission coefficient of the sky plays no role. The surface of the sky is much bigger compared to the building surface.

For radiation calculations after Boltzmann's law the surface temperatures of all radiators must be known. This is easily done for the building; its surface temperature is part of the numerical solution, but this is more complicated for the ground and the sky. For sake of simplification, the ground temperature and the sky temperature are usually set equal to the outdoor air temperature. This in not problematic for the ground emission but for the sky emission a model has to be applied that corrects for air humidity and cloudiness.

Long wave radiation is a temperature-depended self-radiation of

- atmosphere and clouds (emission of $O_3$, $CO_2$ and $H_2O$)
- surrounding buildings
- ground

Atmospheric counter-radiation

Long wave emission

Atmospheric counter-radiation

Long wave emission from building

Long wave emission from ground

## *Ground emission modeling*

We assume that the building surface, the ground and the sky can be treated as planes in radiation equilibrium. The best way to approach the physics of radiation equilibrium is to start with a simple model: with two parallel planes. Considering the emission and the reflection (reflection of one plane depends on emission and reflection of the opposite one) of both planes, the single flux of each plane to the other one can be easily derived.

The total or resulting flux between the two planes is given by the sum of the single flux of each plane. Rearranging the terms leads to the equation below that contains an expression for the radiation exchange coefficient. The emission coefficients of both planes are parameters.

For parallel planes …
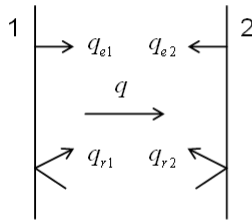


| | | |
|---|---|---|
| $q_{e1} = \varepsilon_1 \sigma T_1^4$ | Emission of plane 1 | W/m$^2$ |
| $q_{r1} = r_1 q_2$ | Reflection of plane 1 | W/m$^2$ |
| $q_1 = q_{e1} + q_{r1}$ | Flux of plane 1 to plane 2 | W/m$^2$ |
| $q_{e2} = \varepsilon_2 \sigma T_2^4$ | Emission of plane 2 | W/m$^2$ |
| $q_{r2} = r_2 q_1$ | Reflection of plane 2 | W/m$^2$ |
| $q_2 = q_{e2} + q_{r2}$ | Flux of plane 2 to plane 1 | W/m$^2$ |
| $q_\| = q_1 - q_2$ | Total flux | W/m$^2$ |

using

| | |
|---|---|
| $\varepsilon = a$ | Emission coefficient |
| $r = 1 - a$ | Reflection coefficient |
| $a$ | Absorption coefficient |
| $\sigma$ | Boltzmann constant    W/m$^2$K$^4$ |
| $T$ | Temperature    K |

with   $q_1 = \dfrac{q_{e2} + r_1 q_{e1}}{1 - r_1 r_2}$   and   $q_2 = \dfrac{q_{e1} + r_2 q_{e2}}{1 - r_1 r_2}$   follows

$$q_\| = \sigma \frac{T_1^4 - T_2^4}{\dfrac{1}{\varepsilon_1} + \dfrac{1}{\varepsilon_2} - 1}$$

Resulting long wave radiation flux between two parallel planes.

Using the formula above for flux between parallel planes we can describe the radiation balance between planes that enclose an arbitrary angle. Usually the wall inclination is 90° for vertical building elements and a horizontal ground. It is evident that the transformation factor from parallel planes to non-parallel planes is given by the sinus of the wall inclination divided by 2. Using that, the equation below in the box can be written.

One question remaining concerns the ground temperature. Since it is usually not part of the numerical solution it must be known or estimated. Using the air temperature, which is part of meteorological data files, the calculated ground emission can be compared with data from a meteorological station when available. For the location Germany-middle the calculated data fit very well indicating than the meteorological data were calculated in the same way but using an emission coefficient of 0.9. This, on the other hand, is important to know when using meteorological data for simulation.

Modeling of long wave radiation balance between building and surrounding ground



$\beta$    Wall inclination

$0° = horizontal (flat roof)$

$90° = vertical$
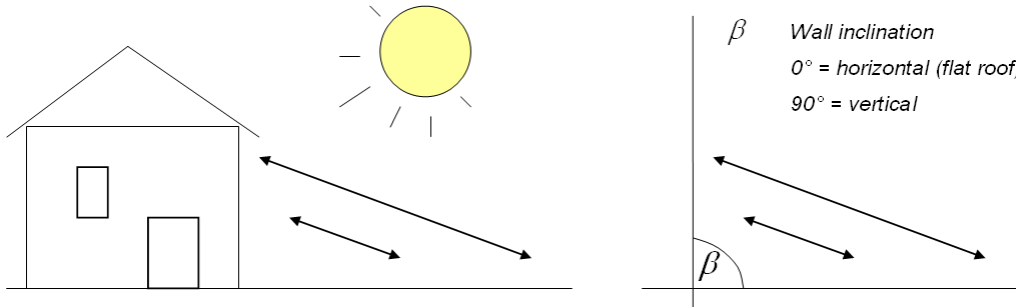
Flux between sloped building element and ground

$$q_{grd,\beta} = \cfrac{1}{\cfrac{1}{\varepsilon_{grd}} + \cfrac{1}{\varepsilon_{bld}} - 1} \sin\left(\frac{\beta}{2}\right) \cdot \sigma\left(T_{grd}^4 - T_{bld}^4\right)$$

$$q_\beta = \sin\left(\frac{\beta}{2}\right) \cdot q_\parallel$$

$q_\beta$    flux between two planes enclosing an angle $\beta$

$q_\parallel$    flux between two parallel planes

*Meteorological data are fitted well with* $T_{sky} = T_{air}$ *and* $\varepsilon_{ground} = 0.9$.
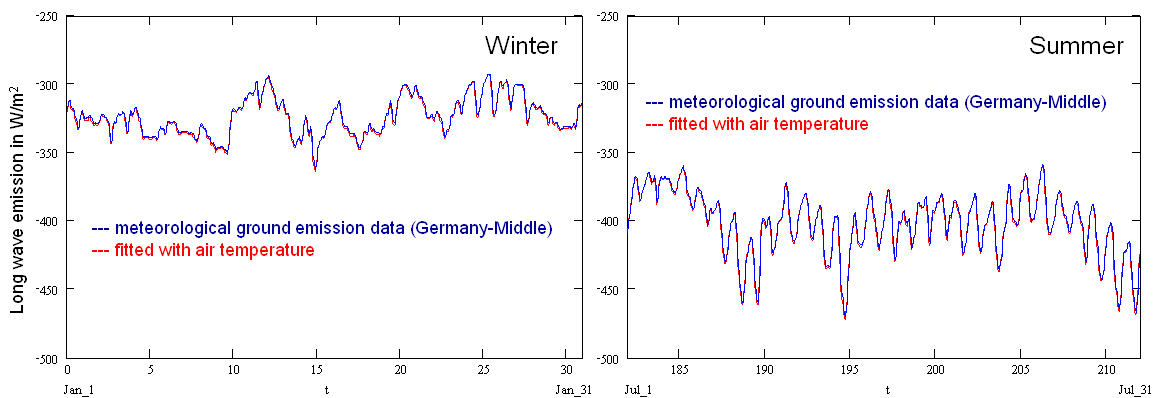
Fit of meteorological data with

$$q_{grd} = \varepsilon_{grd} \cdot \sigma \cdot T_{grd}^4$$

and

$$\varepsilon_{grd} = 0.9$$
$$T_{grd} = T_{air}$$

Comparison with data from a meteorological station ...



# Sky emission modeling

At first sight the sky emission modeling seems to be very similar to the ground emission modeling. The sky can be regarded as an plane opposite to the ground and just the sinus of the wall inclination angle changes to cosinus. Problems are caused by the unknown

parameters emission coefficient and temperature of the sky.

For simplified approaches, the sky can be seen as an half-infinite spheric room with an area very much larger than the building's surface area. An emission coefficient of $\varepsilon_{sky} = 1$ would follow for this model. The problem of the unknown sky temperature remains unsolved.

Modeling of long wave radiation balance between building and atmosphere
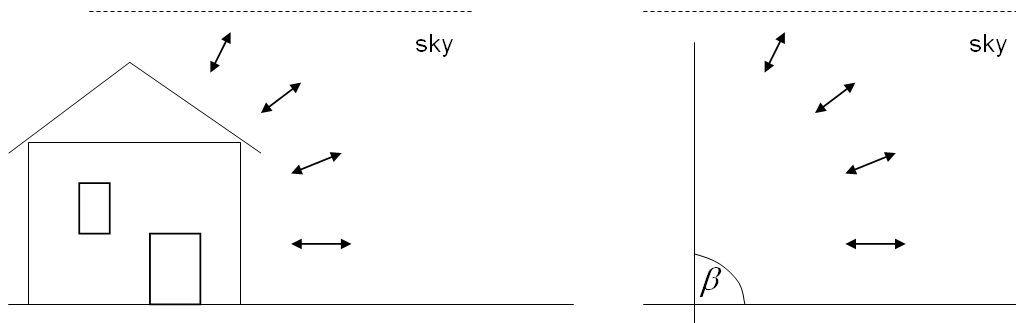


Flux between sloped building element and sky

$$q_{sky,\beta} = \frac{1}{\dfrac{1}{\varepsilon_{sky}} + \dfrac{1}{\varepsilon_{bld}} - 1} \cos\left(\frac{\beta}{2}\right) \cdot \sigma\left(T_{sky}^4 - T_{bld}^4\right)$$

Sometimes, $\varepsilon_{sky} = 1$ is set for a large sky area.

$$q_\beta = \cos\left(\frac{\beta}{2}\right) \cdot q_\parallel$$

$q_\beta$     *flux between two planes enclosing an opposite angle of $\beta$*

$q_\parallel$     *flux between two parallel planes*

### A new model for the long wave sky emission

Modeling of the long wave sky emission or atmospheric counter-radiation is part of meteorological research. Different models have been developed and applied to account for the influences of humidity and cloudiness. Since both components, humidity and cloudiness together with temperature are usually given in climatic data files we strive for a model that can reproduce atmospheric counter-radiation data from that input.

For the clear sky (no clouds) the model of *Angstroem* is applied. It describes the humidity-dependence of the sky emission coefficient by a power function as written below with water vapor pressure as exponent. This formula doesn't account for cloudiness.

For overcast sky, the formula is expanded by two further terms. The first term describes the increase of the emission coefficient with cloudiness by a linear relationship. The maximum alternates with the season, two maximum values for summer and winter have been determined. The transition from summer to winter is described by a sinus function of the year time. The second term is similar to the model of *Feussner* who considered a quadratic relationship that is here reduced to a linear one since the first term was introduced.

The arguments of the model are the cloudiness, the vapor pressure in the air and the year time. The parameters given below have been determined by comparison of the

calculated sky radiation data with data available from meteorological stations, here Germany-North, Germany-Middle and Germany-South. Despite an unique set of parameters has been used for different locations the agreement between meteorological data and the fitted curves is satisfying. This is an indication that the model could be generally applicable. For many meteorological stations no sky radiation is available. By applying the model below it should be possible to generate more accurate data.

<u>Clear sky</u>

The emission coefficient depends on vapor pressure (concentration) in the air

$$q_{sky} = \underbrace{\varepsilon_{clear} \cdot (a - b \cdot 10^{-c \cdot p_v^{air}})}_{\varepsilon_{sky,clear}(p_v^{air}) \approx\ 0.75\ (winter)\ ...\ 0.80\ (summer)} \cdot \sigma \cdot T_{sky}^4 \qquad atmospheric\ counter\text{-}radiation\ (no\ clouds)\quad W/m^2$$

<u>Overcast sky</u>

More clouds – more emission

Low clouds influence more than high clouds

Thick, dense clouds: $\varepsilon_{sky} \longrightarrow 1$

Mathematical model that fits meteorological data best using $T_{air}$ instead of $T_{sky}$ :

$$\varepsilon_{sky}(B, p_v^{air}, t_a) = \left(\varepsilon_{clear} + \Delta\varepsilon(t_a) \cdot B\right) \cdot \left(1 + k \cdot B\right) \cdot \left(a - b \cdot 10^{-c \cdot p_v^{air}}\right)$$

$$\Delta\varepsilon(t_a) = \varepsilon_{summer} \cdot w(t_a) + \varepsilon_{winter} \cdot \left(1 - w(t_a)\right) - \varepsilon_{clear}$$

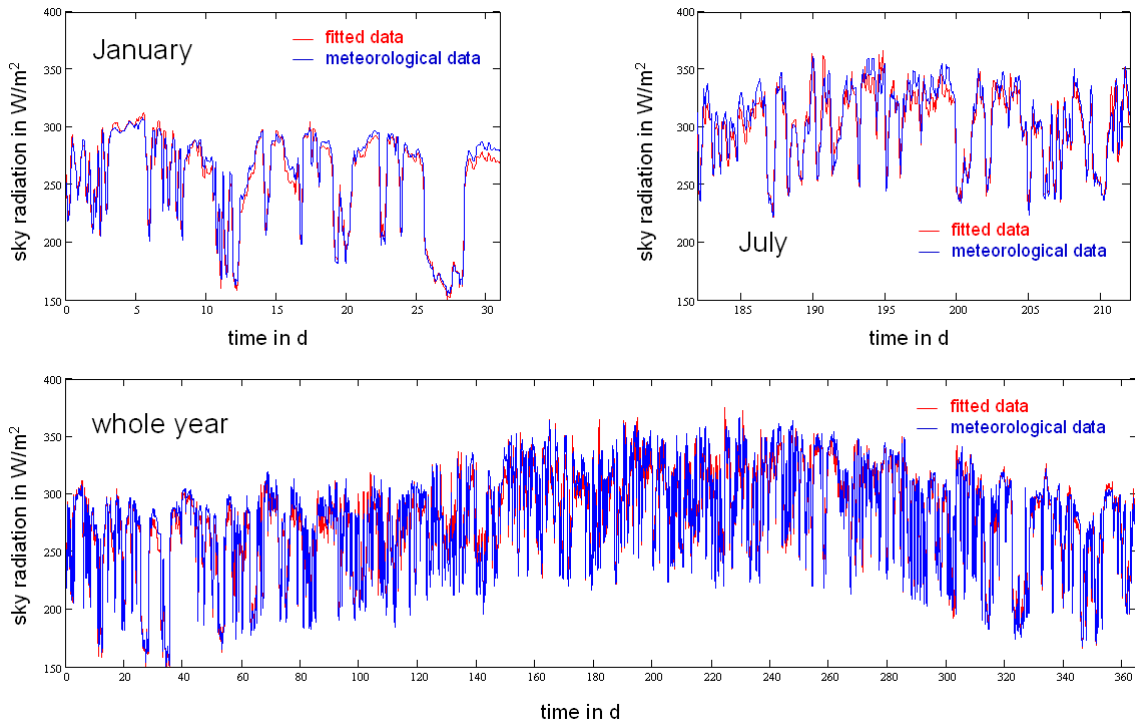$$w(t_a) = \sin\left(\frac{\pi \cdot t_a}{365\ d}\right)$$

<u>Arguments:</u>

$B$ — Cloudiness ($B = 0...1$) -

$p_v^{air}$ — Water vapor pressure — Pa

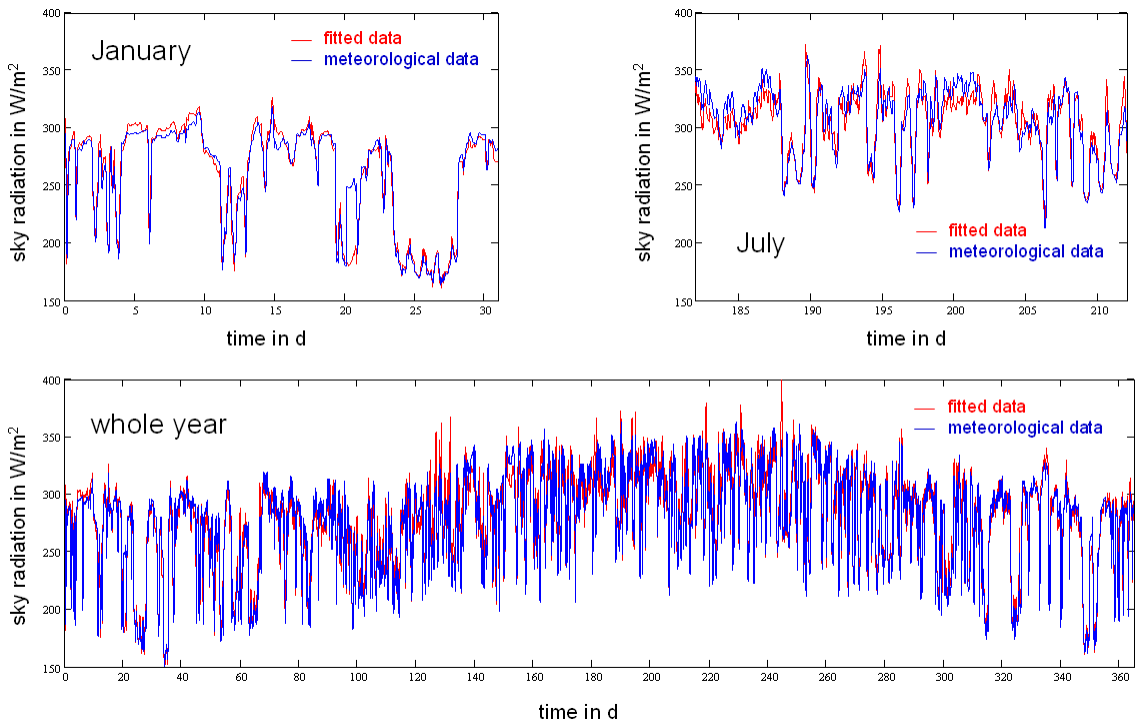$t_a$ — Time of the year (0...365) d

<u>Parameters:</u>

$k = 0.2$
$a = 0.82$
$b = 0.25$
$c = 1.333224 \cdot 10^{-3}$
$\varepsilon_{clear} = 0.75$
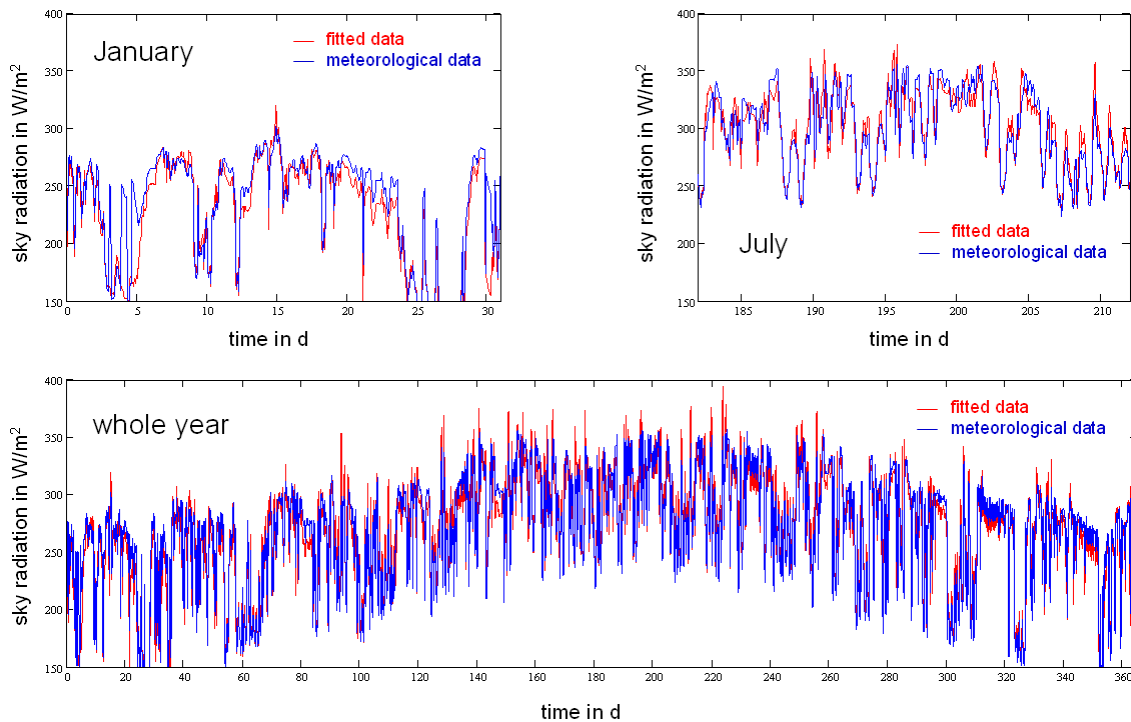$\varepsilon_{summer} = 0.86$
$\varepsilon_{winter} = 0.94$

**Fitted atmospheric counter-radiation data for Germany-North**

**Fitted atmospheric counter-radiation data for Germany-Middle**



**Fitted atmospheric counter-radiation data for Germany-South**

**Literature**

Angstroem, A. K. (1924). Solar and terrestrial radiation. Quarterly Journal of the Royal Meteorological
Society 50, 121–125.
Feussner, K. and P. Dubois (1930). Truebungsfaktor, precipitable water, Staub. Gerlands Beitraege zur
Geophysik 27, 132–175.

# 5.3   Climate data file formats

For CHAMPS-BES simulations two different climate data file formats can be used. The CLI format (short for CLImate data file) is the format used in DELPHIN 4. The CCD format (Common Climata Data format) is more general and simpler and therefore recommended if you want to create or convert your own climate data files.

## CCD Files

### General structure

CCD files contains climate data that can be used to alter boundary conditions in time. Since boundary conditions only describe, how a certain climatic component affects the structure, usually changing the climatic component is enough. CCD files have a very simple format. Take a look a the following example:

```
# Common Climate Data file format
# Air temperatures (including radiation effects)
```

```
TEMPER C
0        0:00:00      -2
0        6:00:00      -4
0       10:00:00       4
0       14:00:00      10
0       20:00:00       8
0       23:00:00       1
1       00:00:00      -1
1        6:00:00      -4
.
.
.
```

Let's take a look at the different sections. All lines beginning with a # character (that means if the # is the first character in the line) are comments and ignored. So are empty lines.

The first required line is the line with the keyword indicating the quantity and the unit of the values. Quantities are:

```
Keyword          Description               Base unit

TEMPER           'Temperature'             C
RELHUM           'Relative humidity'       -
VAPPRES          'Water vapour pressure'   Pa
WATCONT          'Water content'           m3/m3
CAPPRESS         'Capillary pressure'      Pa
PREHEAD          'Pressure head'           m
WATFLUX          'Imposed moisture flux'   kg/m2s
HEATFLUX         'Imposed heat flux'       W/m2
GASPRESS         'Gas (air) pressure'      Pa
```

The units used in the climate data file must be convertible into the given base units. Note, for relative humidity the unit % will be mostly used.

After the line with the keyword and unit the actual climate data follows. Each line corresponds to a specific time point and gives the value at this time point. The format is as follows

```
ddd    hh:mm:ss    v
```

The number of spaces between the values is arbitrary (but at least one space or tab character is required between the numbers). The date and time given by ddd and hh:mm:ss indicate the date/time of the year based on **midnight January 1st**. The number of days is zero based, so for example:

```
5  6:00:00  15
```

defines the climate data value 15 (in some unit given above) on January *6th* at 6am (five days after the January 1st). The days are numbered throughout the year, 365 days correspond to one year. In the simulation time leap years are not considered. If you have measured data from a leap year, remove the 29th of February !

If you have data from multiple years, simple continue numbering:

```
367  10:00:00  15
```

Means (367 - 365 = 2) that this is data for January 3rd of the second year.

Climate data does not need to start at midnight January 1st. If you have data starting 50 days into the year, simple start the climate data file with day 50 and set the simulation start time to 50 days into the year (see Modeling/Simulation Setup Dialog).
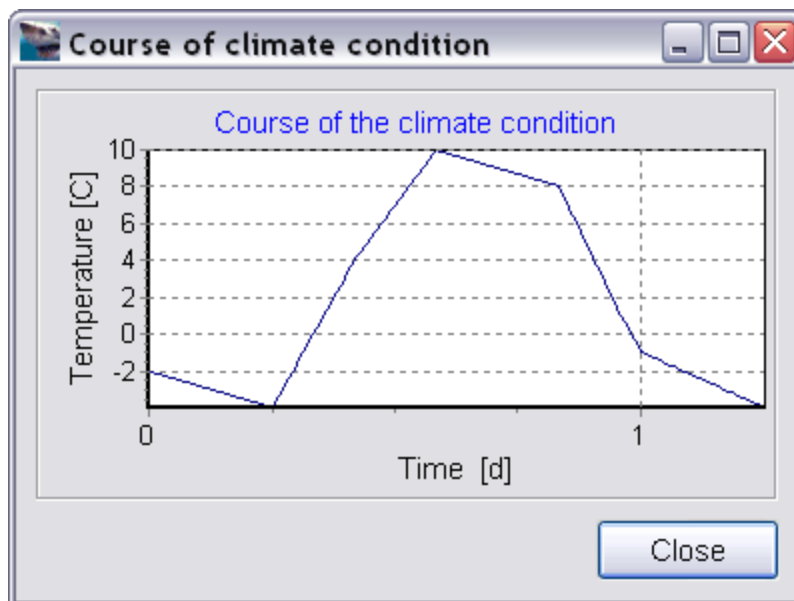
**Relating climate data to simulation time**

Normally the simulation starts at midnight January 1st. If you want to define data relative to the simulation time (typically this is the option for lab experiments or benchmarks) simply use relative date/time points in your climate data file starting with 0 00:00:00.

If you have realistic climatic data (reference year) and you want to run a simulation that starts sometime during the year, simply set the simulation start data and time. The climate data should be entered in the climate data file starting with January 1st all through December 31st.

**Some examples of what might be needed**

Look again at the example above. In realistic climatic conditions you would assume a transient change of temperature between these data points. However, for simulation of benchmarks or laboratory experiments, you may want to specify "jumping" climatic data. Look at the plot of the example above to see the difference:



Climate data in interpolation mode
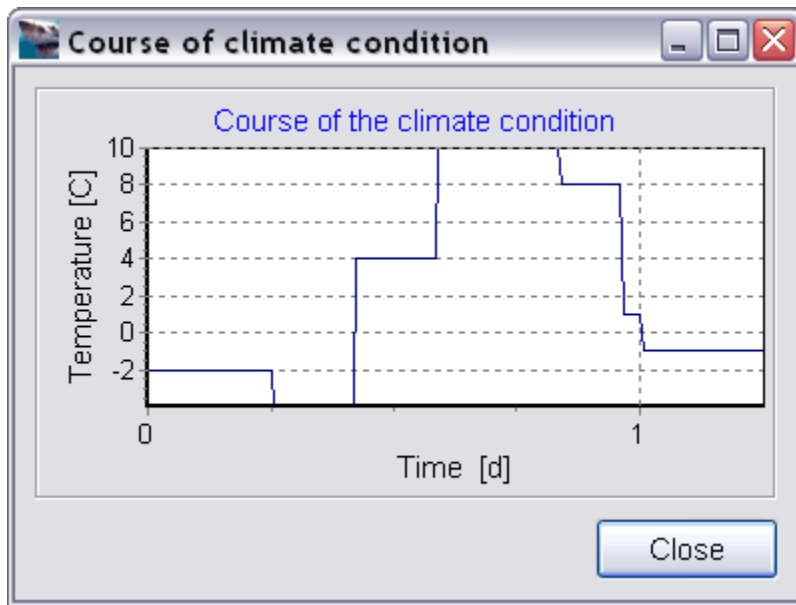
Climate data in "jumpy" mode.

You can change whether to use interpolation or not in the Climate Condition Edit Dialog.

**Problems, pitfalls and things to know**

If the simulation time lies outside the data given in the climate data file, the first or last value are used respectively. For example: You climate data file starts at day 10 and runs until day 20 but your simulation starts January 1st and runs until the end of May, for the first 10 days the first value giving in the climate data file is used. Then, after day 20, the value of day 20 is kept until the simulation is finished.

Sometimes if you have "jumpy" weather data the instantaneous jump in a boundary condition may cause numerical problems. In this case it helps if you define a data points with the previous value shortly before the actual "jump" and use interpolation as in the following example:

```
# Common Climate Data file format
# Air temperatures (including radiation effects)

TEMPER C
0     0:00:00    -2
0     5:59:59    -2
0     6:00:00    -4
0     9:59:59    -4
0    10:00:00     4
.
.
.
```

Cyclic weather data is not supported in the CCD format. Simply create a longer climate data file covering several years (cut'n paste the data to the end again, but ensure that each year has exactly 365 days!). There can be as many data entries as needed and the data lookup time is negligible compared to other computational costs.

### *CLI Files*

CLI files are the default files from DELPHIN 4 and are also supported in CHAMPS-BES. However, since the CCD files are easier to create and typically smaller, it is recommended to use them for your own climate data files.

(detailed documentation of the CLI format can be found in the DIM3/DELPHIN4 manual).

## 5.4    Output file reference

In this chapter the format of the CHAMPS-BES Output files is described so that user defined import filters may be written. A C++ implementation of an import filter for CHAMPS-BES Output files is part of the ***CHAMPS-BES Output File Converter*** installation.

The output of a CHAMPS-BES simulation is stored in a single output directory, which may be specified in the project file (section OUTPUTS). All CHAMPS-BES output files have a header and data section. All header information is stored in the first lines of the file whereas each header line starts with a # (hash character). They can also be treated as comments and be ignored.

Note that every output quantity is stored in a separate file.

**Output file header**
Two different types of outputs are created by the CHAMPS-BES Solver: Field and Flux outputs. Field outputs are created when a quantity in an element is monitored, such as temperature or moisture content. Flux outputs are created when a flux over an element/cell boundary is monitored. The files type can be distinguished by the type information in the header.

A typical field output file header looks like:

```
# CHAMPS-BES Output file
# TYPE          = FIELD
# PROJECT_FILE  = D:\examples\hamstad_benchmark_4.dpj
# CREATED       = Fri Nov 11 15:55:53 2005
# QUANTITY      = Total moisture mass
# SPACE_TYPE    = SINGLE
# TIME_TYPE     = NONE
# VALUE_UNIT    = kg/m3
# TIME_UNIT     = h
```

A typical flux output file header looks like:

```
# CHAMPS-BES Output file
# TYPE          = FLUX
# PROJECT_FILE  = D:\examples\hamstad_benchmark_4.dpj
# CREATED       = Fri Nov 11 15:55:53 2005
# QUANTITY      = Heat flux
# SPACE_TYPE    = SINGLE
# TIME_TYPE     = NONE
# VALUE_UNIT    = kg/m3
# TIME_UNIT     = h
```

The value of the parameter QUANTITY is a string describing the quantity that is monitored.

---

The parameter SPACE_TYPE can have one of the values:
- SINGLE - one value for each element/side
- MEAN - the values of all selected elements/sides are averaged (using weighted average)
- INTEGRAL - the values of all selected elements/sides are integrated in space (by multiplying the values with the volumes/areas)

The parameter TIME_TYPE can have one of the values:
- NONE - the values are written as they are obtained at the output time point
- MEAN - the values calculated between output time points are averaged in time
- INTEGRAL - the values calculated between output time points are integrated in time

After the header section the geometry section follows:

**Geometry section**
The geometry section contains a number of tables. A table starts with a table definition, for instance:

```
TABLE <table ID keyword>
```

The first table is the materials table, listing the materials used in the structure. The keyword for the material table is MATERIALS. The table contains as many lines as materials. Each line has the format:

```
<material number>  <material color id> <material name>
```

The color ID is an integer value corresponding to an RGB encoded color. An example for such a material table is:

```
TABLE MATERIALS
1    1231665    Ceramic brick
2    3312756    Mineral wool
3    4322234    Gypsum board
```

The next table currently used in the geometry file is the grid table, containing the widths and heights of the columns and rows of the grid. The keyword for this table is GRID. The table contains 3 lines. The first line contains the widths of the columns of the grid (x-dimensions). The second line contains the heights of the rows (y-dimensions). The last line contains the thickness (z-dimension) or in the case of 3D rotation symmetric geometry a single 0. The dimensions are all in m (meter).
For a simple 1D structure, the table could look as follows:

```
TABLE GRID 0.1 0.1 0.1 0.02 0.02 0.01 0.01 1 1
```

The next table in the geometry file is the element geometry table, with the keyword ELEMENT_GEOMETRY. It contains as many lines as elements for which quantaties are calculated. Each line has the following format:

```
<element number> <x-coordinate> <y-coordinate> <column> <row> <material number>
```

The x and y coordinates are hereby the coordinates of the element center points and given in m. The origin of the coordinates is the bottom left (as in a normal Cartesian coordinate system). The origin of the columns and rows is the top left, furthermore the column and row indices start with 0.

An example element geometry table for a 1D structure is given below:

```
TABLE ELEMENT_GEOMETRY
1    0.05   0.5   0   0   1
2    0.15   0.5   1   0   1
3    0.25   0.5   2   0   1
4    0.31   0.5   3   0   2
5    0.33   0.5   4   0   2
6    0.345  0.5   5   0   3
7    0.355  0.5   6   0   3
```

Also a table containing geometry information for the sides is included in the geometry file. The keyword for this table is SIDES_GEOMETRY. Each line in this table has the following format:

```
<side number> <x-coordinate> <y-coordinate> [horizontal=0|vertical=1]
```

The x and y coordinates are hereby the coordinates of the middle point of the side given in m. The last value is either 0 or 1, corresponding to a horizontal orientation of the side or vertical respectively. Horizontal means that the fluxes are calculated from top to bottom or bottom to top.

An example sides geometry table for a 1D structure (all vertical sides) is given below:

```
TABLE SIDES_GEOMETRY
1    0.0    0.5   1
2    0.1    0.5   1
3    0.2    0.5   1
4    0.3    0.5   1
5    0.4    0.5   1
6    0.5    0.5   1
```

## Data section
After the geometry section the actual output data follows. Depending on the type of the output file (field or flux) the data section starts with a keyword ELEMENTS or SIDES.

The data section of field fils starts with an ELEMENTS entry, which contains the element numbers in the order the values are printed later. If the values for the elements in the range 1..8 are printed, the following definition is used:

```
ELEMENTS = 1 2 3 4 5 6 7 8
```

If only certain values of a few elements will be printed, the following definition would represent a subset of all elements:

```
ELEMENTS = 5 6 7 8 12 13 14 15 17 18 19 20
```

The data section of flux files starts with an SIDES entry, since fluxes are always calculated over the sides of elements. The SIDES entry contains, similar to the ELEMENTS entry in field data files, the numbers of the sides used for output in this output file.

Finally the data block follows. In CHAMPS-BES Output Files the data has the same format regardless of grid/construction type or selected output range and format. Generally data is stored as

```
<time point in some unit>    <first value> <second value> ... <nth value>
```

whereas the number of values may vary. This grid/output format independent data format allows to use the same import filter for all kinds of output files.

The values are corresponding to the elements indicated by the numbers in the elements definition. There are as many values as element numbers, unless MEAN or INTEGRAL is specified as space type. In that case only a single value is written.

For flux outputs also a single value is written per side (unless MEAN or INTEGRAL format is used). The orientation of the fluxes, whether they are horizontal or vertical fluxes (x or y direction) has to be obtained from the SIDES geometry table in the geometry file.

## 5.5 Project file reference

This section contains a brief description of the CHAMPS-BES project file format and the data included therein.

### *General project file structure*

CHAMPS-BES project files are human readable (and editable) ASCII files with a hierarchical structure. Sections and subsections are enclosed in brackets as in the following example.

```
[SECTION]

  [SUBSECTION]
    ...
    ...

  [SUBSECTION]

    [SUBSUBSECTION]
      ...
      ...

    [SUBSUBSECTION]
      ...
      ...

  [SUBSECTION]
    ...
    ...

[SECTION]
    ...
```

### Comments
Lines starting with a ; (semi-colon) are comments and ignored.

### Place holders
Special placeholders can be used when external files (e.g. climate data) is referenced:

| | |
|---|---|
| `$(PROJECT_DIR)` | references the directory where the project file is saved into |
| `$(INSTALL_DIR)` | references the directory where the solver is installed |

### Main project file sections

The main project sections in a CHAMPS-BES project file are:

| | |
|---|---|
| PROJECT_INFO | contains information about the simulation and project file |
| INIT | contains general simulation and solver settings |
| MATERIALS | contains material parameters |
| CONDITIONS | contains definitions of climatic conditions, boundary conditions, initial conditions as well as contact and field conditions |
| DISCRETISATION | contains the geometrical information and discretisation data |
| OUTPUTS | contains output definitions |
| ASSIGNMENTS | contains assignments (links geometry with materials, conditions and outputs) |

### *Typical data types stored in the project file*

#### Parameters
Parameters are used for typical physical properties and are defined by a value (typically double precision) and a unit.

#### Strings
Used for identification and description, but also for initialization strings or flags. Strings do not need to be enclosed in quotation marks.

#### Flags
Used to enable/disable features, use 'yes' or 'no' as values.

#### Functions
Functions are actually linear splines/data tables and stored in 3 consecutive lines. The first line contains the 'Function' keyword followed by a string identification of the function type. The second line contains all x values and the third line contains all y values. The values must be separated by a whitespace (space or tab character) and the number of x values must match the number of y values.

## 5.6    Material file reference

The following material file is a prototype for CHAMPS-BES material files. Lines starting with a ; (semi-colon) are comments and ignored (comments are written in blue in the following text).

```
; This is a typical material data file for input into the CHAMPS-BES solver
; For all parameters where you can specify a unit, you may also use a different
; unit that can be converted into the base units given below in the examples

; Lines with a ';' at the begin are comments and ignored.

; The keyword [MATERIAL] starts a material definition
[MATERIAL]

; The general data section contains basic parameters
;    NAME    - A unique material identification string
;    CET     - Heat capacity of dry matrix
;    RHO     - Bulk density
```

```
;     OPOR    - Theta_por : Porosity (vacuum saturation value)
;     OEFF    - Theta_eff : Maximum liquid moisture content (defines upper limit of
;               calculatable moisture content range)
;     LAMBDA  - Thermal conductivity for dry material
;     FLAGS   - Add several flags to define particular properties of a material
;
;               Available flags:
;                  AIR - defines an air material and special behavior for
;                        sorption characteristics
;                  AIRTIGHT, VAPOURTIGHT or WATERTIGHT
;                       - make material tight for transports
;

  [GENERAL]
    NAME               = Prototype material
    CET                = 1000 J/kgK
    RHO                = 2000 kg/m3
    OPOR               = 0.2 m3/m3
    OEFF               = 0.19 m3/m3
    LAMBDA             = 0.8 W/mK


; The identification section contains additional information for the
; classification of the material
;     SOURCE     - A source identification string
;     CATEGORY   - One or more category identification strings, separate different
;                  categories with ':' characters
;     COLOR      - RGB encoded color, stored as BGR triple (so pure red is #0000ff)
;                  and as decimal number

  [IDENTIFICATION]
    SOURCE             = IBK, TU Dresden, Germany
    CATEGORY           = Brick
    COLOR              = 10000


; The MRC section contains moisture storage function parameters.
; Required for the simulation is at least one moisture storage function and
; its reversed function.
;
; Available TYPE keywords are: GAUSS, VG, OL_PC_SPLINE and OL_RH_SPLINE
;
; VG_N, VG_P are needed for type VG
; GAUSS_N and GAUSS_P are needed for type GAUSS
;
; Available FUNCTION keywords are (case sensitive) :
;     Ol(pC)  pC(Ol)  Ol(RH)  RH(Ol)
;
; Recommended moisture storage function, if calculating in oversaturated range:
;   TYPE = OL_PC_SPLINE   and functions Ol(pC) and pC(Ol)
;
; If calculating in hygroscopic range, using the sorption isotherm is possible, too:
;   TYPE = OL_RH_SPLINE   and functions Ol(RH) and RH(Ol)
;

  [MRC]
    TYPE               = VG
    VG_N               = 2
    VG_P               = 0.3 1.25e-05 1.65 0.7 1.8e-05 6
    FUNCTION           = pC(Ol)
    0                  8.81299128024e-07  9.11244499073e-07  9.26217184598e-07  ...
    12.2               12.1767711639      12.1544551849      12.1435546875       ...

; This section contains moisture transport functions
  [MOISTTRANS]
    FUNCTION           = lgKl(Ol)
    0                  8.03205886396e-05  9.61229018679e-05  0.000115034168771  ...
    −20.50314295       −20.50314295       −20.18314185       −19.86314206        ...
    FUNCTION           = lgKv(Ol)
    0                  0.00125            0.0025             0.00375             ...
    −11.192213         −11.192206         −11.192227         −11.192277          ...

; This section contains thermal conductivity
  [HEATTRANS]
    TYPE               = LINEAR
```

```
LINEAR_FACTOR      = 4.5
```

Back Cover