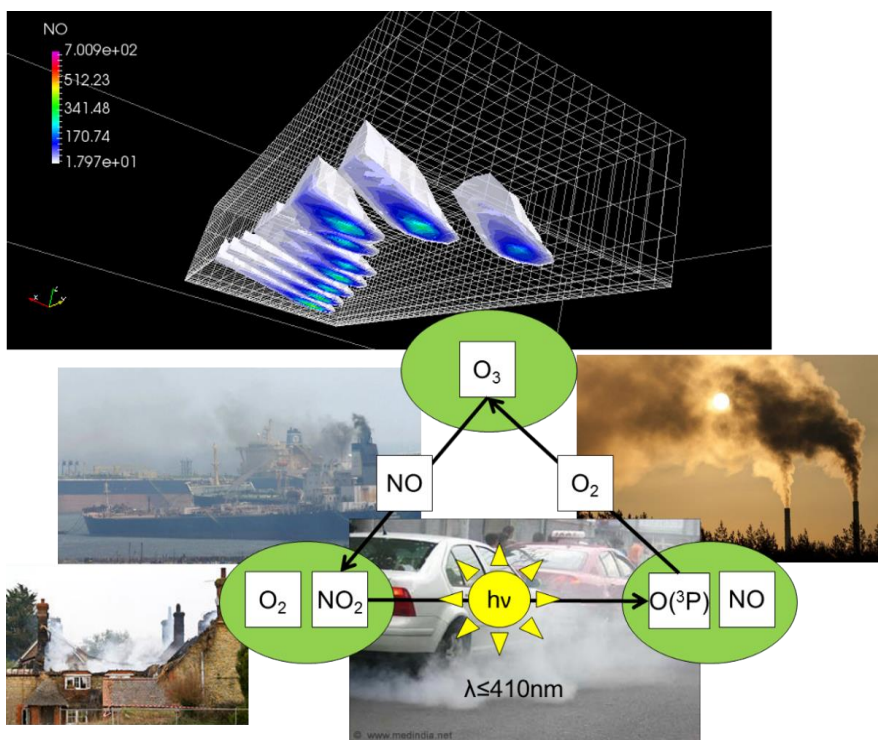


User's Guide

User's Guide EPISODE-CityChem version 1.2r City-scale Chemistry Transport Model

Matthias Karl¹ and Martin Ramacher¹

¹ Helmholtz-Zentrum Geesthacht, Department of Chemistry Transport Modelling, Geesthacht, Germany



Program manual

Preface

The CityChem extension to the EPISODE dispersion model has been developed by Dr. Matthias Karl (Helmholtz-Zentrum Geesthacht, HZG). EPISODE is an urban-scale air quality model developed at the Norwegian Institute for Air Research (NILU) appropriate for air quality studies at the local scale. Some parts of the CityChem extension include earlier developments at NILU, for example the WMPP modifications to plume rise from point sources and the basic EMEP45 chemistry mechanism and chemistry solver for computation of the photochemistry on the Eulerian grid.

The CityChem extension of the EPISODE model is distributed under the name EPISODE-CityChem under the terms of the RPL license version 1.5, <https://opensource.org/licenses/RPL-1.5>. By using the model you agree to the terms of usage given in the RPL license. Modifications to the distributed source code of EPISODE-CityChem and the usage of modified versions have to be reported to NILU and, for the parts that belong to the CityChem extension, in addition to HZG.

The User's Guide provides a detailed step-by-step description of how to set up and use the EPISODE-CityChem model. For a technical description of the physics and chemistry of the model it is referred to *Karl et al. (2019)*.

MSc Martin Ramacher contributed to this User's Guide by adding descriptions for creating the receptor grid, surface roughness, and land use input files and the section on installation of The Air Pollution Model (TAPM). Additionally, he has thoroughly tested the step-by-step manuals and the distributed software package. Josefine Feldner (Leuphana Univ., Lüneburg) contributed an R script for the additional evaluation of model results with openair.

The CityChem extension was designed for treating complex atmospheric chemistry in urban areas and for improved representation of the near-field dispersion in proximity of emission sources.

The developments within the CityChem extension include (1) new photochemistry schemes of multiple pollutants on the Eulerian grid, (2) optional reading of 3-dim. hourly boundary concentrations, (3) improved treatment of point source plumes, (4) a simplified model for the sub-grid treatment of street canyons, (5) several modifications to allow easier handling of the model output. Until now, EPISODE-CityChem has been applied for simulation of the air quality in Hamburg, Rostock, Gdansk and Riga.

Contents

	Page
Preface	1
Contents.....	3
1 Introduction	5
1.1 About EPISODE-CityChem.....	5
1.2 Model features.....	5
1.3 What this manual contains	11
2 Getting Started	12
2.1 Download	12
2.2 Requirements.....	14
2.3 Installation of TAPM	15
2.4 Directory structure.....	16
2.5 Installation of EPISODE-CityChem	17
2.6 Running EPISODE-CityChem.....	19
3 Model Input	21
3.1 Input Files.....	21
3.2 The CityChem model domain	22
3.3 User metadata file.....	24
3.4 Meteorology input	26
3.5 Emission input.....	32
3.5.1 Input files for point sources.....	33
3.5.2 Input files for line sources	34
3.5.3 Input files for area sources	35
3.6 Regional background input	36
3.7 Auxiliary input	37
3.7.1 Receptor grid	38
3.7.2 Auxiliary 2-D fields.....	42
4 Model Configuration.....	51
4.1 Main data.....	51
4.2 Site data	52
4.3 Time data.....	54
4.4 Meteorological data.....	55
4.5 Concentration data.....	59
4.6 Deposition data.....	62
4.7 Statistical data	63
4.8 Local photochemistry data	66
4.9 Grid model data	66
4.10 Area source data	69
4.11 Point source data	70
4.12 Line source data	72

5	Model Output	74
5.1	Spatial lon-lat maps (NCL)	77
5.2	Vertical profile sequence (NCL)	78
5.3	Concentration time series plots (NCL)	80
5.4	Statistical regression plots (NCL)	82
5.5	OpenAir script for additional analysis	84
5.6	Importing to ArcGIS	88
6	Utilities	89
6.1	MCWIND.....	89
6.2	UECT	95
6.3	BCONCC	104
7	Example data	111
8	References	115
	Appendix A Meteorological input files.....	117
	Appendix B Model output appendix.....	118
	Appendix C Installation of NetCDF	120

User's Guide for EPISODE-CityChem version 1.2r

1 Introduction

1.1 About EPISODE-CityChem

The EPISODE model with the CityChem extension is a chemistry-transport model (CTM) designed for the city-scale dispersion of pollutants. The model is applicable from local scale up to regional scale covering geographic extents of less than 10 km up to 100 km. Its purpose is to fill the gap between regional CTMs which are typically applied for mesoscale dispersion on one side and eddy-resolving models (e.g. Large Eddy Simulation (LES) models) typically applied for the micro-scale within urban areas, on the other side. EPISODE combines a 3-dimensional (3-D) Eulerian grid model with sub-grid Gaussian dispersion in a systematic manner. In EPISODE-CityChem, the chemical transformation of pollutants is considered in great detail within the Eulerian grid and in a simplified manner for the sub-grid dispersion. The model resolves the structure of urban emissions source objects such as streets and industry stacks. Typical applications are urban air quality studies, environmental impact assessment and sensitivity analysis of sector-specific emissions. CityChem is an extension of the EPISODE dispersion model developed by the Norwegian Institute for Air Research (NILU). A technical documentation of EPISODE can be found in *Slørdal et al.* (2003 and 2008). The original EPISODE model is an integral part of the Air Quality Information System AirQUIS 2003 (*Bøhler and Sivertsen, 1998*). Several distinct features of the EPISODE-CityChem model, presented in the next section, make this model in particular useful for studies on formation of ozone and aerosols in urban areas.

1.2 Model features

The EPISODE-CityChem model is embedding a series of features which are distinct from other dispersion models typically used for urban air quality studies:

1. Reading time-varying 3-D concentrations of compounds at the lateral and vertical boundaries from the Community Multiscale Air Quality (CMAQ) model as initial and boundary concentrations.
2. Reading prognostic meteorological fields from the Australian model TAPM (The Air Pollution Model). If the TAPM model is not available, utilities are provided to produce a diagnostic wind field based on meteorological observations.
3. All physical processes of the model can be switched on/off for debugging or sensitivity testing.
4. The model calculates time-dependent concentrations of air pollutants on a 3-D main grid with coarse resolution ($\leq 1 \times 1 \text{ km}^2$) and simultaneously local ground air concentrations with high resolution ($100 \times 100 \text{ m}^2$) as well as in selected locations (monitoring stations).
5. Providing concentration output and other diagnostic output in form netCDF files, enabling quick conversion into ArcGIS shapes.

6. EPISODE and the CityChem extension are distributed as open source code under the Open Reciprocal License. Modifications to the distributed source code of EPISODE-CityChem and the usage of modified versions have to be reported to NILU and, for the parts that belong to the CityChem extension, in addition to HZG. EPISODE-CityChem runs on Linux computers in single processor mode.

The City-scale Chemistry transport model CityChem extension uses the functionality of the EPISODE urban dispersion model to run simulations on an Eulerian grid as main grid with an embedded regular receptor grid which samples concentrations from Gaussian sub-grid dispersion calculations. The model solves the time-dependent advection/-diffusion equation on the Eulerian 3-D grid including complex chemical transformation and deposition processes. Finite difference method numerical are applied to integrate the solution forward in time. The user can choose from numerical scheme with varying degree of accuracy for solving the advection/-diffusion processes. Depending on the chosen chemistry options, EPISODE-CityChem can be run either as a tracer model or as a chemistry model. Running as tracer model implies that all air pollutants are only dispersed as (chemically inert) passive tracers, whereas running as chemistry model implies chemical transformation driven by photo-oxidation using either a detailed atmospheric chemistry mechanism or simple photo-stationary steady-state between ozone (O₃), nitrogen oxide (NO) and nitrogen dioxide (NO₂). In the current version, aerosol particles can be treated in terms of primary emitted particulate matter (PM) and/or of total particle number (PN). Particulate matter is separated in PM_{2.5} (PM with aerodynamic diameter ≤ 2.5 μm) and PM₁₀ (PM with aerodynamic diameter ≤ 10 μm). Secondary particle formation is not implement in the model yet; this is planned for one of the next versions of the CityChem extension. The model considers three categories of emissions: area source emissions, line source emissions and point source emissions.

Embedded sub-grid models for near-source dispersion

The sub-grid model for line sources (i.e. road traffic) is the Gaussian model HIWAY-2 (*Petersen, 1980*) with modifications. The sub-grid model for point sources (e.g. industry stacks) is the Gaussian segmented plume model SEGPLU (*Walker and Grønskei, 1992*). HIWAY-2 and SEGPLU are part of the EPISODE model.

HIWAY-2 is used to calculate concentration levels of pollutants from road traffic at distances tens to hundreds of meters downwind of the road, given that the surrounding terrain is relatively uncomplicated. Each street lane (or lane segment) with vehicle traffic is simulated as a straight, continuous, finite length, line source with a uniform emission rate. Pollutant concentrations caused by vehicle traffic are found by interpretation of the line source (line or line segment) as a finite sum of simple Gaussian point-source plumes, and the total line source contribution (of the line or line segment) is then derived by numerical integration (i.e. summation) over the length of the line source, thinking of the line source as a line-of-points.

In CityChem, the option to use a simplified street canyon model (SSCM) to compute concentrations for the receptor points in street canyons is introduced. The implemented street canyon model follows in most aspects the Operational Street Pollution Model (OSPM; *Berkowicz et al.*, 1997). The complex and diverse geometry of street canyons is approximated by three generic types. If the street canyon option is selected, each line source for which the geometric mid-point is located in a grid cell with urban land use (land use classes 32 - 35 defined by TAPM) is identified as street canyon.

A fundamental assumption of the street canyon model (OSPM and SSCM) is that when the wind blows over a rooftop in a street canyon, an hourly averaged recirculation vortex is always formed inside the canyon (*Hertel and Berkowicz*, 1989), as illustrated in Figure 1. The contribution of a line source l to the receptor concentration located within the cut section, i.e. the urban street canyon is the sum of the direct contribution ($C_{sdir,s}$) from the traffic plume plus a contribution from the recirculation of the traffic plume ($C_{srec,s}$) due to the vortex inside the canyon:

$$C_{line,s} = C_{sdir,s} + C_{srec,s} \quad (1)$$

The leeward receptor inside a street canyon is exposed to the direct contributions from the emissions inside the recirculation zone (unless the wind direction is close to parallel) and a recirculation contribution. For the receptor on the windward side, only the emissions outside the recirculation zone are taken into account for the direct contribution. If the recirculation zone extends through the whole canyon, no direct contribution is given to the windward receptor.

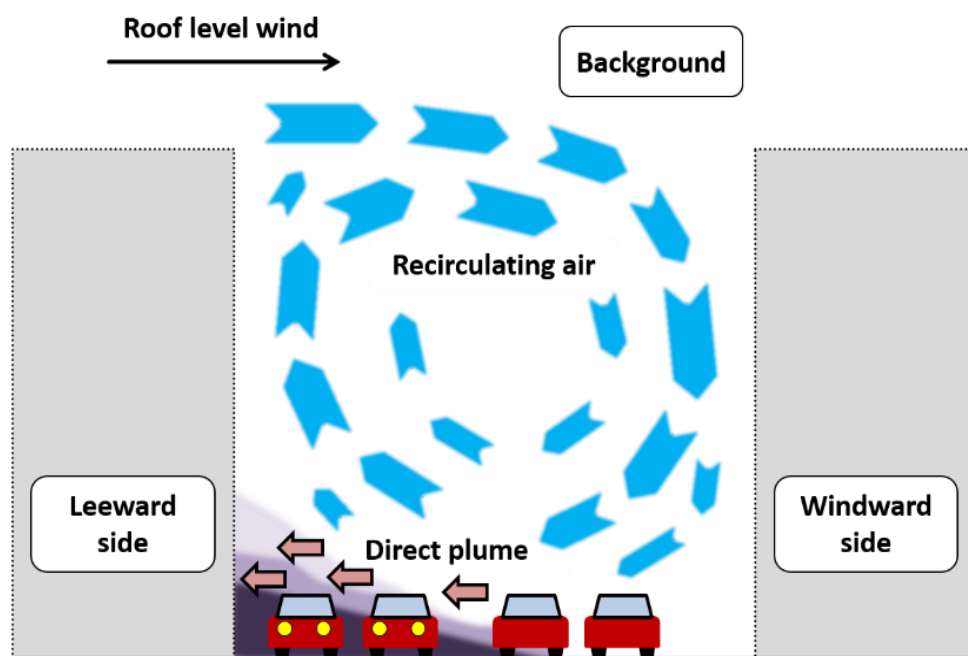


Figure 1: Illustration of the street canyon box model. In the figure, the recirculation zone extends through the whole street canyon, albeit the model can treat situations where this is not the case.

In CityChem, the street canyon model is coupled to the HIWAY-2 line source dispersion model. HIWAY-2 does not calculate line source concentration contributions to receptors that are upwind of a line source and for receptor points that are too close to the line source (in the latter case, the receptor points are moved away from the street in the EPISODE model). In SSCM, the wind direction at street level in the recirculation zone is mirrored compared to the roof level wind direction, which means that the leeward receptor (which is upwind from the line source when the roof top wind direction is used) receives both emitted pollutants from the traffic and the recirculated pollution. SSCM calculates the line source concentration contribution to a leeward receptor in a street canyon as sum of the direct contributions from the emissions inside the recirculation zone (unless the wind direction is close to parallel) and a recirculation contribution.

For a given street canyon, the concentration contribution from a line source to the receptor point is calculated either by HIWAY-2 or by SSCM. For all windward/leeward receptor points, which are (1) located within a model grid cell defined as street canyon cell, and (2) located close enough to a line source (i.e. within the actual street canyon), and (3) located at a road link with length > 8 m, the concentration contribution from the street is calculated by SSCM, whereas for all others HIWAY-2 is used.

The segmented plume model for point sources, SEGPLU, treats the emission from individual point sources as a temporal sequence of instantaneous releases of a specified pollutant mass. The subsequent position and pollutant concentration within each of the plume segments is then estimated. When the plume segment reaches a predefined horizontal or vertical extent, it is inserted into the (Eulerian) main grid cell containing its centre of mass. This size is optimally set as $\sigma_y/\Delta y = 4$ or $\sigma_z/\Delta z = 4$ (corresponds to user-defined input value 0.25), where σ_y and σ_z are the horizontal and vertical length scales of the plume segment, and Δy and Δz are the grid spacing in the horizontal and vertical direction, respectively. This is also done if the segmented plume experiences a large change in wind direction. If the plume segment is transported outside the model domain, its mass is lost.

In CityChem, the Worm Meteorological Pre-processor (WMPP) created by Sam-Erik Walker (NILU) is implemented for improved treatment of the plume rise from point sources (as briefly described in: *Karl et al.*, 2014). WMPP calculates the wind speed profile and the wind speed at the height of the plume using similarity theory.

Using SEGPLU for a large number of point sources might demand long computational times because EPISODE keeps track of each plume segment. The user can choose to use a simple elevated injection from point sources instead of the sub-grid plume treatment. In that case, the emission amount from a point source will be injected at a certain height (height of the final plume rise) into the grid cell area.

Interactions between the main grid and the receptor grid

On the main (Eulerian) grid, time-dependent 3-D concentration fields of the pollutants are calculated by solving the advection/-diffusion equation with terms for chemical reactions, dry deposition and wet deposition, and area emissions. The 2-D and 3-D fields of meteorological variables and the 2-D fields of area emissions are given as input to the model with the resolution of the main grid. Therefore the resolution of meteorology in EPISODE-CityChem corresponds to the grid resolution of the main grid. Emissions from point sources are added to the main grid concentration each time step. Emissions from line sources are added to the main grid concentrations following a procedure described in more detail below. As the model steps forward in time, an accurate account of the total pollutant mass from area- and line-sources is kept within the Eulerian grid model.

The establishment of a regular receptor grid is an integral part of the CityChem extension; it enables higher resolution output required for comparison with monitor data acquired near line sources. The regular receptor grid in the EPISODE-CityChem model differs from the downscaling approach by *Denby et al.* (2014) which allocates sampling points at high density along roads and other line sources but much fewer further away from the line sources. While *Denby et al.* (2014) interpolate the model-computed high-density set of receptor concentrations to the desired output resolution using ordinary kriging, CityChem gives as output the receptor point concentrations with high resolution, usually $100 \times 100 \text{ m}^2$, covering the entire model domain. Figure 2 illustrates the combination of an Eulerian main grid (black dotted grid lines) and the regular receptor grid (blue dots) for the surface layer (ground air).

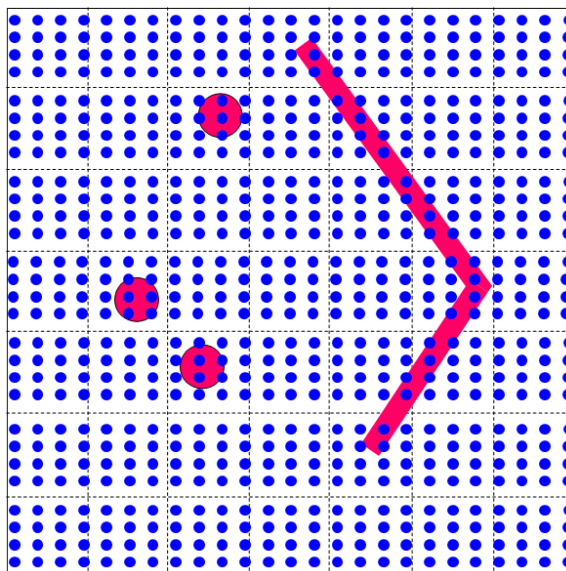


Figure 2: Illustration of the main grid and the regular receptor grid in EPISODE-CityChem. The black dotted lines represent the gridlines of the main grid while the blue dots represent the regular receptor grid. The objects in red are point source and line source objects, for which sub-grid dispersion is applied. Note that the size of the source objects is not to scale.

On the regular receptor grid, time-dependent surface concentration fields of the pollutants are calculated by summation of the main grid concentration of the corresponding grid cell, and the concentration contributions from the Gaussian sub-grid models due to line source emissions and point source emissions.

The hourly concentration C_r in an individual receptor point is thus defined as:

$$C_r = C_m(i, j, 1) + \sum_{l=1}^L C_{line, l} + \sum_{p=1}^P C_{point, p} \quad (2)$$

Where $C_m(i, j, 1)$ is the hourly main grid concentration of the grid cell $(i, j, 1)$ where the receptor point is located in. The grid (background) concentration C_m used in Equation (2) corresponds to a modified Eulerian 3-D main grid concentration (i.e. $C(i, j, k)$) to avoid that emissions of point and lines sources are counted twice. $C_{line, l}$ is the hourly concentration contribution of line source l calculated by the line source sub-grid model and $C_{point, p}$ is the hourly concentration contribution of point source p calculated by the point source sub-grid model.

For an individual receptor point, all line sources within a certain distance from the line, the so-called *influence distance*, will have a concentration contribution. The sub-grid concentration contributions from all L lines lying within the influence distance will be summed up and added to the receptor point concentration. The influence distance can be defined by the user for each line source, however the maximum influence distance is defined as half of the grid width in x-direction ($\Delta X/2$) (Slørdal et al., 2003). All P concentration contributions from point source plume segments calculated by the point source model are also added to C_r .

The receptor point concentration C_r is rather a diagnostic concentration and is calculated each time step based on current main grid concentrations and sub-grid concentration contributions. C_r is not added to the main grid concentration. The regular receptor grid is a sampling grid of concentrations in the receptor points. Receptor point concentrations are thought to represent the high-resolution ground concentration of a cell with grid cell area of the sampling grid.

The description of the mass redistribution from line sources to the Eulerian main grid that is implemented in the EPISODE model is given in Appendices A2 and A.3 of the report by Slørdal et al. (2003). Normally, the prescribed influence distance of a line source is less than (or equal to) $\Delta X/2$, as stated above. In this situation the line source will contribute to the gridded emissions of the Eulerian main grid during each model time step of the first part of the hour, (defined as: $t_{line} = 3600 - t_{inf}$). During the remaining part of the hour (i.e. during the influence time, t_{inf}) the line source emissions are not contributing to the Eulerian model grid. In CityChem, t_{inf} is set to zero, i.e. the line source emissions are contributing to the main grid concentrations during the entire simulation hour.

1.3 What this manual contains

This User’s Guide is a manual for the practical usage of the CityChem extension to the EPISODE model. A technical documentation of the model is provided elsewhere (Karl *et al.*, 2019). This User’s Guide describes the following items:

1. Building and running the EPISODE-CityChem model.
2. Creating meteorological input files.
3. Producing 3-D boundary conditions from CMAQ model output.
4. Creating auxiliary input files.
5. Configuring the run script of EPISODE-CityChem.
6. Graphical display of model output.

The following interfaces and utilities that come along with the CityChem-EPISODE model are described in this manual:

AERMAP	Topography pre-processor
MCWIND	Utility to generate a diagnostic wind field
tapm4cc	Interface to convert TAPM meteorology output
z0top4cc	Utility to generate auxiliary input
static4cc	Utility to generate auxiliary input
BCONCC	Interface to convert CMAQ concentration output
UECT	Interface for line source emissions
	Interface for point source emissions
	Interface for area source emissions
postcchem	Graphical display routines in NCL

Figure 3 provides an overview of the components of the model system:

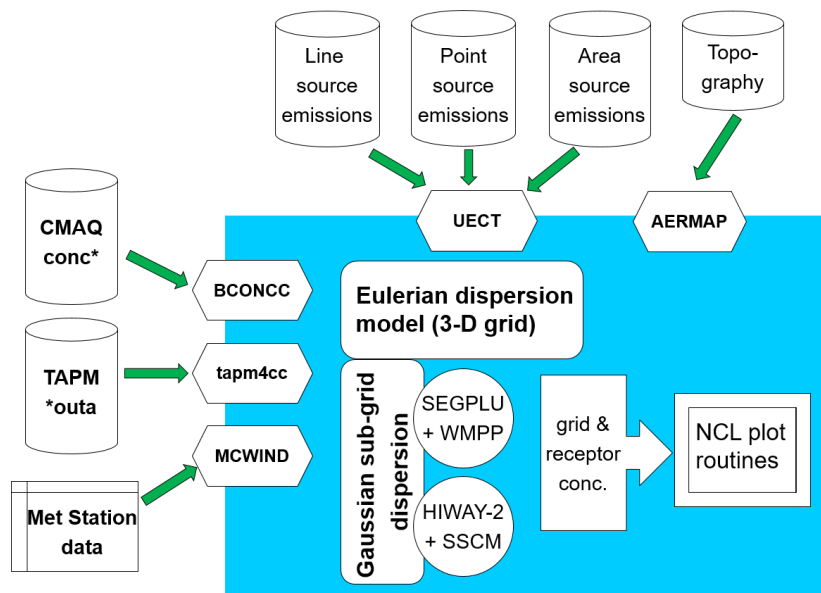


Figure 3: Overview of EPISODE-CityChem model components: utilities and interfaces.

2 Getting Started

2.1 Download

Distribution packages.

The distribution includes: 1) EPISODE-CityChem model source code, pre-processing utilities (BCONCC, MCWIND, etc.) and post-processing scripts and 2) a test data set for a one-month air quality simulation for Hamburg. The example data package contains the TAPM *.outa file and the boundary conditions files for use with EPISODE-CityChem.

Download location:

<ftp://ftp.hzg.de/outgoing/Karl/CITYCHEM>

Available on request by email to Matthias.Karl at hzg.de.

citychem-1.2r.tar.gz

Contains model source code and pre-processing utilities and graphic scripts [41.2 MB].

cc_example_data_new_201307.tar.gz

Contains example data for simulation of Hamburg model domain [352 MB].

Distribution license.

The CityChem extension package is distributed under the Open Reciprocal License, in sub-license of HZG in accordance with the license of NILU for EPISODE.

The following statement has to be added in all publications resulting from the use of the EPISODE-CityChem model:

“The CityChem extension, developed at Helmholtz-Zentrum Geesthacht (HZG) is designed for treating complex atmospheric chemistry in urban areas and improved representation of the near-field dispersion (Karl, 2018; Karl et al., 2019). The model is an extension of the EPISODE dispersion model to enable chemistry/transport simulations of reactive pollutants on city scale. EPISODE is an Eulerian dispersion model developed at the Norwegian Institute for Air Research (NILU) appropriate for air quality studies at the local scale. The EPISODE model and the CityChem extension are open source code subject to the Reciprocal Public License ("RPL") Version 1.5, <https://opensource.org/licenses/RPL-1.5>.”

Giving the following citations:

Karl, M. (2018): Development of the city-scale chemistry transport model CityChem-EPISODE and its application to the city of Hamburg, Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2018-8>, 2018.

Karl, M. Walker, S.-E., Solberg, S., and Ramacher, M. O. P. (2019): The Eulerian urban dispersion model EPISODE. Part II: Extensions to the source dispersion and photochemistry for EPISODE-CityChem and its application to the city of Hamburg, Geosci. Model Dev., <https://doi.org/10.5194/gmd-2018-325>, in review, 2019.

Terms of usage.

These are the RPL License terms of usage of the EPISODE-CityChem model software:

2.0 Acceptance Of License. You are not required to accept this License since you have not signed it, however nothing else **grants you permission to use, copy, distribute, modify, or create derivatives** of either the Software or any Extensions created by a Contributor. These actions are prohibited by law if you do not accept this License. **Therefore, by performing any of these actions You indicate Your acceptance of this License and Your agreement to be bound by all its terms and conditions.** IF YOU DO NOT AGREE WITH ALL THE TERMS AND CONDITIONS OF THIS LICENSE DO NOT USE, MODIFY, CREATE DERIVATIVES, OR DISTRIBUTE THE SOFTWARE. IF IT IS IMPOSSIBLE FOR YOU TO COMPLY WITH ALL THE TERMS AND CONDITIONS OF THIS LICENSE THEN YOU CAN NOT USE, MODIFY, CREATE DERIVATIVES, OR DISTRIBUTE THE SOFTWARE.

4.0 Grant of License From Contributor. By application of the provisions in Section 6 below, each Contributor hereby **grants You a world-wide, royalty-free, non-exclusive license, subject to said Contributor's intellectual property rights, and any third party intellectual property claims derived from the Licensed Software under this License, to do the following:**

4.1 **Use, reproduce, modify, display, perform, sublicense and distribute any Extensions Deployed by such Contributor or portions thereof, in both Source Code form or as an executable program,** either on an unmodified basis or as part of Derivative Works.

4.2 Under claims of patents now or hereafter owned or controlled by Contributor, to make, use, have made, and/or otherwise dispose of Extensions or portions thereof, but solely to the extent that any such claim is necessary to enable You to make, use, have made, and/or otherwise dispose of Licensed Software or portions thereof.

The complete terms of the RPL license should be consulted at <https://opensource.org/licenses/RPL-1.5>.

2.2 Requirements

Hardware requirements:

- Linux computer
- Intel or AMD processor, 64-bit. 1600-megahertz (MHz) or higher
- 4 gigabyte (GB) or more memory
- 18 gigabyte or more available disk space
- 1024 x 768 minimum display resolution

The example run takes about 23 hours on a Linux desktop PC with AMD A4-5300 APU processor (1.6 GHz) for the 1-month simulation of air quality in Hamburg.

Software requirements:

- gcc/gfortran Fortran90 compiler (version 4.4 or later).
It is also possible to use the Intel Fortran compiler (version 11 or later, and Intel Fortran Composer).
- netCDF library (version 3.6.0 or later). Detailed installation steps for Linux Ubuntu are given in Appendix C.
- Two options exist for the preparation of meteorological input to the model. The first option requires the installation of the air pollution model TAPM on a Windows PC (see section 2.3). The second option is the installation of the diagnostic wind field model MCWIND, which is included in the downloaded CityChem program package. MCWIND requires measurements at two heights from a meteorological mast at one site and additional wind measurements from a second site. More details on installation of MCWIND are given in section 6.1.
- Installation of the community multi-pollutant air quality model CMAQ. CMAQ is required if it is intended to use 3-D boundary conditions (BCON) in the CityChem simulation. CMAQ is open source and can be obtained here: <https://www.epa.gov/cmaq/download-cmaq/>. For a description of performing simulations with CMAQ it is referred to the CMAQ manuals at <https://www.epa.gov/cmaq/>. The installation of CMAQ is optional because constant boundary concentration values can be used instead of 3-D BCON concentration fields. The utility BCONCC, which is included in the downloaded CityChem program package, allows conversion of CMAQ concentration output into CityChem BCON files. More details on BCONCC are given in section 6.3.
- Optional: NCL (NCAR Command Language) for graphical visualization and for some input preparation steps (version 6.3.0 or later).
- Optional: NCO (netCDF Operators) for manipulation and analysis of netcdf output files (version 4.4.4 or later).
- Optional: Openair for additional graphical visualization (version 2.1-5 or later). Openair is an R package developed for the purpose of analysing air quality data (<http://davidcarslaw.github.io/openair/>). The use of openair requires the installation of R.

2.3 Installation of TAPM

Installation of the air pollution model TAPM on a Windows PC is necessary for producing the meteorological input files for EPISODE-CityChem when no meteorological observations are available.

It is no requirement to install TAPM for using EPISODE-CityChem. Note, if TAPM is not installed, then meteorological input files have to be created with MCWIND (Sect. 6.1) and auxiliary input data with other tools (Sect. 3.7.2).

TAPM consists of coupled prognostic meteorological and air pollution concentration components, eliminating the need to have site-specific meteorological observations. Instead, the model predicts the flows important to local-scale air pollution, such as sea breezes and terrain-induced flows, against a background of larger-scale meteorology provided by synoptic analyses (*Hurley, 2008*).

TAPM version 4 can be obtained by using the contact of the Australian research institute CSIRO form at:

<http://www.csiro.au/en/Research/OandA/Areas/Assessing-our-climate/Air-pollution/TAPM>

The approximate cost of a TAPM group license is 5000 Australian Dollars (status 12.12.2015).

To install TAPM V4 software, the user needs to be logged-on as a user with Administrative privileges in some versions of the Windows operating system (Windows XP, Windows 7 and later).

The TAPM software is in a directory called TAPM_V4.0_setup that contains a setup.exe installation program. TAPM_GUI can be installed by running setup.exe either manually or by using the Start-Settings-Control Panel-Add/Remove Programs utility in Windows. It is recommended that the directory name should be something like C:\TAPM. Note that the directory name must not contain spaces (e.g., 'Program Files'), as spaces in directory or file names can cause problems in some of the command-line batch files used by TAPM_GUI, and the name is restricted to be less than or equal to fifteen characters.

After installing the software, the GUI can be started by running the TAPM_GUI.EXE program from the installed directory.

For information on how to run the software see the tapm_v4_user_manual.pdf which can be found in the installation directory.

2.4 Directory structure

First create the directory `citychem-1.2r` in your home directory (i.e. `/home/user/`), `{your-home}`, by typing in a terminal:

```
{your-home}$ mkdir citychem-1.2r
```

Change to the new directory (i.e. `/home/user/citychem-1.1/`) and unpack the delivered tar-files into the new directory

```
{your-home}$ cd citychem-1.2r
{your-home}/citychem-1.2r$ tar -zxvf citychem-1.2r.tar.gz
{your-home}/citychem-1.2r$ tar -zxvf cc_example_data_new_201307.tar.gz
```

From now on the directory where CityChem is installed (i.e. `/home/user/citychem-1.2r/`) will be referred to as `{your-citychem}`. Table 1 shows the directory structure that will be created in `{your-citychem}`.

Table 1: Subdirectories in {your-citychem} created by unpacking the CityChem download.

Name of subdirectory	Content
<code>bin/</code>	Location of the executable program
<code>docu/</code>	User's Guide document
<code>GRAFICS/</code>	Graphical plots of the output
<code>testdata</code>	Input files of background concentrations at the domain boundaries, produced from CMAQ model. TAPM *.outa file. From unpacking the Example Data tar package.
<code>INPUT/emis</code>	Input files of emission data
<code>INPUT/tapm</code>	Input files of meteorological variables produced from TAPM model
<code>INPUT/other</code>	Auxiliary static input files
<code>INPUT/mcwind</code>	Input files of meteorological variables produced from MCWIND (empty directory)
<code>license/</code>	License files
<code>OUTPUT/</code>	Output files generated by CityChem
<code>postcchem/</code>	Scripts for the visualization of CityChem output
<code>preproc/</code>	CityChem user input script (meta data) and utilities for preparation of input files
<code>preproc/bconcc2.0</code>	Scripts for preparation of chemical boundary conditions based on CMAQ output
<code>preproc/uct2.0</code>	UCT program (emission conversion tool)
<code>preproc/mcwind</code>	MCWIND program (diagnostic wind field model)
<code>preproc/tapm4cc2.0</code>	Scripts for preparation of meteorological input based on TAPM output
<code>preproc/auxiliary</code>	Scripts for preparation of auxiliary input data
<code>SIMU/</code>	CityChem run scripts
<code>src/</code>	Source code of CityChem

2.5 Installation of EPISODE-CityChem

The installation is now done automatically. First you need to edit the config.user file in the citychem directory. The config.user file included in the distribution is consistent with the notations in this Users' Guide and should fit for most installations on a personal computer with Linux. Here is the explanation for the keywords that have to be set in the config.user file:

F90	Fortran compiler (gfortran or ifort)
F90FLAGS	Fortran compiler flags for normal compilation (debugging enabled)
F90OPTFLAGS	Fortran compiler flags for optimized run (fast but still high accuracy)
NCLIB	Netcdf Library linker statement (for Makefiles)
NCINC	Netcdf Include statement (for Makefiles)
HOMEPATH	Full directory path of your citychem directory (i.e. {your-citychem})
IOAPI	IOAPI 3.1 binary (default: Linux2_x86_64gfort)
IOAPIDIR	Full path of IOAPI 3.1 installation (IOAPI is included in the distribution)

The IOAPIDIR path is normally the subdirectory preproc/bconcc2.0/ioapi3.1/ included in the distribution, unless you want to use a different or previously built IOAPI3.1 installation. After editing the config.user file, the installation routine can be started.

For the installation, the script is called with two arguments. The first argument defines the installation process and the second argument defines, whether the installation is for the User's Guide example ("example") or for self-made input data of boundary conditions ("bcon").

For the example use:

<code>./installcc.csh city example</code>	Install CityChem (normal executable)
<code>./installcc.csh cityopt example</code>	Install CityChem with high speed
<code>./installcc.csh util example</code>	Install all pre-processing utilities

For the self-made "bcon" files use:

<code>./installcc.csh city bcon</code>	Install CityChem (normal executable)
<code>./installcc.csh cityopt bcon</code>	Install CityChem with high speed
<code>./installcc.csh util bcon</code>	Install all pre-processing utilities

For the explanation of boundary conditions see chapter 3.6.

For removing the installation, the script can be used as follows.

<code>./installcc.csh uninstall</code>	Remove all installations (citychem and utilities)
<code>./installcc.csh cleandata</code>	Remove all user-generated output data

To start the installation of EPISODE-CityChem, change to your citychem directory and open the C-Shell by typing ({your-citychem} is the directory where you have installed CityChem, for example, /home/user/citychem-1.2/):

```
{your-citychem}/$ csh
```

And run the installation routine for CityChem as normal executable:

```
{your-citychem}/$ ./installcc.csh city example
```

And after that you can install all pre-processing utilities:

```
{your-citychem}/$ ./installcc.csh util example
```

If you do not want to use the installation routine or experience problems with it, you can follow the instructions below to build the model. Otherwise you may ignore the remainder of this section.

Change to the subdirectory src/ and edit in Makefile.mk the variables INCLUDES and LIBS. Here the correct path of the netCDF library (see Appendix C) on your computer has to be given. By default the lines are:

```
INCLUDES = -I/usr/local/netcdf4/include
LIBS     = -L/usr/local/netcdf4/lib -lnetcdf -lnetcdff
```

There are two compilation options in Makefile.mk, the default option for testing with stricter control and the advanced option for speed optimization. The default is:

```
#flags for debugging (use for the test examples). Optimization (-O) flags cannot be used
together with -ggdb.
F90FLAGS = -ggdb -u -C -cpp -pedantic -ffpe-trap=invalid,underflow,zero -ftrapv
-fbounds-check -finit-real=nan -fbacktrace -fimplicit-none -finit-integer=n -ffree-form
```

For the first run of EPISODE-CityChem it is recommended to use the default. Later runs could be done with the advanced option. To use it, comments the above default option with a # in the first column, and use the F90FLAGS definition that follows two lines later in Makefile.mk:

```
#Optimized run -O2
F90FLAGS = -O2 -cpp -fbacktrace -ffree-form -fimplicit-none -finit-integer=n
-ffast-math -funroll-loops -m64 -dynamic
```

If the Intel® Fortran compiler is used for compilation and linking, then the F90 definition has to be changed:

```
#Compilation with gfortran
#F90 = gfortran
#Compilation with intel fortran
F90 = ifort
```

The two previous definitions of F90FLAGS have to be commented and the following compilation flag has to be activated:

```
#Compilation flags with intel fortran
F90FLAGS = -g -cpp -FR -traceback
```

For an optimized run with Intel® Fortran one might replace ‘-g’ by ‘-O2’ in this definition, once the first compilation and run has been successful.

Now build the program. In the terminal, change to subdirectory `src/` for compilation. After building, a symbolic link is added in the subdirectory `SIMU/` to start the program from there (`{your-citychem}` is the directory where you have installed CityChem, for example, `/home/user/citychem-1.2/`):

```
{your-citychem}/$ cd src
{your-citychem}/src$ make -f Makefile.mk
{your-citychem}/src$ cd ../SIMU
{your-citychem}/SIMU$ ln -s ../bin/citychem.exe
```

If you want to refresh the built at a later time, proceed as follows:

```
{your-citychem}/src$ make -f Makefile.mk cleanall
{your-citychem}/src$ make -f Makefile.mk
```

2.6 Running EPISODE-CityChem

All required information about physical and chemical options, input and output for a EPISODE-CityChem simulation is handed to the program in one run script, located in subdirectory `SIMU/`. The detailed description of the run script is in section 4. To start a run, open a terminal window and enter:

```
{your-citychem}/SIMU$ ./citychem.exe
```

This starts the model run. Next the program asks for the run script. Enter the filename (for example: `citychem_yourscript.txt`) manually in the terminal window. To continue the run the session on your computer has to remain open and the terminal window must not be closed.

Alternatively, the program can be started with one line:

```
{your-citychem}/SIMU$ ./citychem.exe citychem_yourscript.txt
```

Main debugging information is written during runtime to the log file (in the example: `CITYCHEM_Yourmachine_log.txt`) in the subdirectory `OUTPUT/`.

In case input files are missing or erroneous input values are provided, CityChem-EPISODE will most probably stop and issue the following warning:

'CITYCHEM stopped! Please read the log-file!'

Read the end of the log file (in the example: `CITYCHEM_Yourmachine_log.txt`) located in the subdirectory `OUTPUT/` carefully. It will help to identify the wrong input file.

A simulation run is usually carried out for the period of one month. The restart option allows performing runs for multiple months or one year. At the end of a simulation month, the 3-D concentration field of the main grid of the study domain of the last hour is written to a netCDF file, which can be used as initial

concentration (ICON) file for the next simulation month. In the run of the next simulation month, the ICON file is opened and the 3-D initial concentrations are used as starting concentrations for the main grid. The ICON file for the restart must have the same horizontal and vertical grid dimensions and the same chemical compounds. See section 4.9 for details on the restart option.

Example data is provided for performing a one-month test simulation with the EPISODE-CityChem model for the city of Hamburg in July 2013.

A step-by-step guidance for the test simulation of the example data is given in section 7. Please do not use the example data without the instructions from section 7.

It is recommended to read section 3 and section 4 before starting the test run. After reading the two chapters, you may return to section 7 for performing your first example run.

3 Model Input

3.1 Input Files

Four categories of input files have to be provided:

1. Meteorology input files;
2. Background chemistry (boundary condition) files;
3. Emission files; and
4. Auxiliary input files.

Several meteorological variables are required for a EPISODE-CityChem simulation. Table A1 in Appendix A shows which of the meteorological input files are mandatory and which are optional for simulations with the model. It also indicates which of the input files can be produced by TAPM and by the MCWIND program. A few input files are not produced by TAPM. The corresponding variables are calculated in the (internal) meteorological pre-processor of EPISODE.

For all chemical compound for which CityChem produces output (by default these are 18 compounds in total), chemical concentrations at the boundaries of the domain (lateral and vertical) can be provided as a 3-D field in an individual BCON input file. Alternatively a constant value for each of the compounds can be provided in the run-script.

For all chemical compound for which CityChem produces output, emissions from point sources, line sources (road traffic, ferry lines, airport lanes) and area sources can be provided. Emissions from line sources and area sources are in separate files for each individual compound. Currently four categories of area sources can be used: domestic heating, solvent use and agriculture, commercial and industrial sources, and shipping. For the four categories vertical profiles are pre-defined in the Fortran source code (src/area/casrc.f90). For point sources, one file contains all point sources and emitted compounds. A simulation without emissions can be done by omitting all area and line source emission files and by filling the point source file with zeros for all point sources.

A few auxiliary files are required for a model run. These include the definition of the regular receptor grid, static 2-D fields of topography and surface roughness, and hourly fractional cloud cover.

3.2 The CityChem model domain

The EPISODE-CityChem model domain relies on the Universal Transverse Mercator (UTM) coordinate system with WGS 84 as reference geoid. The model domain is characterized by a main grid with a usual grid cell width of $dx = 1000$ m and $dy = 1000$ m (cell widths can be chosen by the user) and a regular raster grid of receptor points (in distances of 100 m for example).

The dimensions of the main grid are specified in the run script of EPISODE-CityChem. The parameters nx and ny are the maximum number of cells in x-coordinate direction and y-coordinate direction respectively, and nz is the maximum number of vertical model layers.

The vertical coordinate of the model is derived from the vertical coordinate in the TAPM model. Table 2 displays the vertical model layers (model levels) and the corresponding layer heights for a typical set up. It is not recommended to change the vertical coordinate and the user's guide does not provide information how to do it.

The input and output time step of the model is one hour ("simulation hour"); n is the index for simulation hour, and $tend$ is the maximum number of simulation hours.

The EPISODE-CityChem model provides output for a certain number of chemical compounds (see Table 3), from here on called CityChem compounds. Parameter nc is the total number of CityChem compounds (by default $nc = 20$). Only for CityChem compounds, emission input can be provided. It is further possible to specify deposition velocities and wet scavenging coefficients for these compounds.

Table 2: Definition of the vertical model layers of CityChem.

Model level	Layer height (m)	Height above ground (m) at layer top
1	17.50	17.50
2	20.00	37.50
3	25.00	62.50
4	25.00	87.50
5	37.50	125.00
6	50.00	175.00
7	50.00	225.00
8	50.00	275.00
9	50.00	325.00
10	50.00	375.00
11	50.00	425.00
12	50.00	475.00
13	75.00	550.00
14	125.00	675.00
15	200.00	875.00
16	250.00	1125.00
17	250.00	1375.00
18	250.00	1625.00
19	250.00	1875.00
20	250.00	2125.00
21	250.00	2375.00
22	375.00	2750.00
23	500.00	3250.00
24	500.00	3750.00

Table 3: Chemical compounds with input and output in CityChem.

Index ic	Chemical compound	Full name
1	O3	Ozone
2	NO	Nitrogen oxide
3	NO2	Nitrogen dioxide
4	H2O2	Hydrogen peroxide
5	N2O5	Dinitrogen pentoxide
6	HNO3	Nitric acid
7	SO2	Sulphur dioxide
8	Sulphate	Sulphuric acid
9	CO	Carbon monoxide
10	C2H6	Ethane
11	HCHO	Formaldehyde
12	CH3CHO	Acetaldehyde (and higher aldehydes)
13	C2H4	Ethene
14	C3H6	Propene (and other n>4 alkenes)
15	PAN	Peroxy acetyl nitrate
16	nC4H10	n-Butane (and other alkanes)
17	CH3COC2H5	Methyl ethyl ketone (and other ketones)
18	o-Xylene	o-Xylene (and other aromatic HCs)
19	PM2.5	Fine Particulate Matter, Dp ≤ 2.5 µm
20	PM10	Particulate Matter; Dp ≤ 10 µm

3.3 User metadata file

Before beginning with the preparation of input files for a model simulation run, the user metadata file for the simulation run has to be configured. The user metadata file includes all meta information required by the input preparation tools for meteorology (TAPM), boundary/background concentrations (CMAQ), emissions and the auxiliary data. The user metadata file is read by the interfaces for preparation of meteorology, boundary concentrations, emissions and auxiliary data. It is one file that controls all pre-processing utilities. The user metadata file has to be named “cctapm_meta.inp” and located in the same directory as the respective pre-processing tool.

A date stamp (start and end date) is added as suffix to all files, produced by the pre-processing utilities in order to allocate input files to a certain simulation period.

A template of the user input file “cctapm_meta.inp” is located in {your-citychem}/preproc/. Figure 4 shows a snapshot of the user metadata file. A list is provided below which describes the content of the user metadata file line-by-line.

```

1  '{SHEBA-Riga---City--testwithtapm-0001}'      ! Simulation-ID, exact 38 characters long
2  '/home/karl/Meteo/tapm_1km_20130101_20130201.outa' ! Filename of TAPM tapm_YYYYMMDD_YYYYMMDD.outa file ('' if not available)
3  '/home/karl/CMAQ/'                             ! Input path of CMAQ CONC.scenario.YYYYJJJ and METCRO3D.scenario.YYYYJJJ files ('' if not available)
4  '/home/karl/Emissions/UECT/input/Riga_Pointerc.csv' ! Filename of csv-file containing point source yearly emission data (full path)
5  '/home/karl/Emissions/UECT/input/Riga_Linearc.csv' ! Filename of csv-file containing line source yearly emission data (full path)
6  '/home/karl/Emissions/UECT/input/Riga_Areasrc.csv' ! Filename of csv-file containing area source yearly emission data (full path)
7  '/home/karl/Emissions/UECT/output/Riga/'        ! Output files path name
8  'uect_log.txt'                                  ! Log file (in output path)
9  'CC'                                            ! City-scale model choice ['CC'=CityChem, 'TP'=TAPM]
10 'ASE'                                           ! Output type choice      ['FSE'=points, 'LSE'=lines, 'ASE'=areas, 'ALL'=all]
11 1                                                ! Output for CityChem as ASCII or Binary [1=ASCII, 2=Binary]
12 744                                             ! Number of hours for output, one month (e.g. 744)
13 '20120101'                                     ! Start date of input  FORMAT='YYYYMMDD'
14 '20120131'                                     ! End date of input    FORMAT='YYYYMMDD'
15 2012,01,01                                     ! Start date of input  FORMAT=YYYY,MM,DD
16 2012,01,31                                     ! End date of input    FORMAT=YYYY,MM,DD
17 50                                              ! Number grid cells in x-direction of the model domain
18 50                                              ! Number grid cells in y-direction of the model domain
19 100.0                                           ! Horizontal resolution (grid width dx) in meters of gridded area source input
20 400.0                                           ! Horizontal resolution (grid width dx) in meters of the model domain
21 318256,6310096                                 ! Grid Origo (south-west corner) in (truncated) UTM coordinates (x,y) of the model domain
22 '35N'                                           ! UTM zone, e.g. '35N'
23 725                                             ! 725 Number of sources in the point input
24 2806                                           ! 2806 Number of sources in the line input
25 225                                             ! 225 Number of sources in the area input
26 0                                               ! Output netCDF area sources choice [1=Yes, 0=No]
27

```

Figure 4: Screenshot of the “cctapm_meta.inp” user metadata input file.

A line-based list of simulation input parameters:

- simid* Simulation-ID must be a string starting with a left parenthesis and ending with a right parenthesis and an exact length of 38 characters.
- tapm_path* Full path and file name of the TAPM *.outa file. (see section 3.4). Format of the file should be tapm_YYYYMMDD_YYYYMMDD.outa. Only required if TAPM is used to produce the meteorological fields for the simulation. Otherwise enter ‘’.
- cmaq_path* Input path of the CMAQ concentration output files CONC* and the MCIP output files METCRO3D*. Only required if the utility BCONCC (section 6.3) is used to generate 3-D

	boundary concentration fields for the simulation. Otherwise enter ‘.’.
<i>point_path</i>	Full path and file name of the input file of point sources. The input file has to be in CSV format containing the yearly emission totals for each point source. If not available enter ‘.’.
<i>line_path</i>	Full path and file name of the input file of line sources. The input file has to be in CSV format containing the yearly emission rate for each line source. If not available enter ‘.’.
<i>area_path</i>	Full path and file name of the input file of area sources. The input file has to be in CSV format containing the yearly emission totals for each area source. If not available enter ‘.’.
<i>log_file</i>	File name of the log file of UECT. The log file will be written to the directory where UECT was executed.
<i>model</i>	City-scale model option used by UECT to produce emission input files. Enter ‘CC’ for CityChem or ‘TP’ for TAPM.
<i>source</i>	Emission type for output of emission files, used by UECT. Enter ‘PSE’ for point source input, ‘LSE’ for line source input, ‘ASE’ for area source input or ‘ALL’ for all source types.
<i>ep_fmt</i>	Output file format option. Select <i>ep_fmt</i> = 1 for ASCII output files or <i>ep_fmt</i> = 2 for binary output (input files for CityChem) or <i>ep_fmt</i> = 3 for netCDF output (only enabled for tapm4cc.exe).
<i>hmonth</i>	Numbers of simulation hours (for one month).
<i>start_string</i>	Start date as string. Format has to be ‘YYYYMMDD’.
<i>end_string</i>	End date as string. Format has to be ‘YYYYMMDD’.
<i>startdate</i>	Start date separated by comma. Format: YYYY,MM,DD.
<i>enddate</i>	End date separated by comma. Format: YYYY,MM,DD.
<i>nx</i>	Number of grid cells of the model domain (main grid) in x-direction.
<i>ny</i>	Number of grid cells of the model domain (main grid) in y-direction.
<i>dxarea</i>	Grid width dx of the cells in the area source emission file.
<i>dx</i>	Grid width dx of the cells of the model domain (main grid).
<i>utm_x, utm_y</i>	UTM coordinates (x, y) of the model grid origin (south-west corner of the rectangular domain). UTM x-coordinate <i>utm_x</i> has to be truncated, i.e. not containing the two leading digits which indicate the UTM zone.
<i>utm_zone</i>	UTM zone as string, for example: ‘35N’.
<i>nsopp</i>	Number of point sources in the point source input file to UECT.
<i>nsoll</i>	Number of line sources in the line source input file to UECT.
<i>nsoaa</i>	Number of area sources in the area source input file to UECT.
<i>ncout</i>	Option for getting area emission output from UECT in one extra netCDF file. Enter 1 if netCDF extra output is wanted or 0 if not.

3.4 Meteorology input

Table A1 in Appendix A gives an overview of the mandatory and optional meteorological input files for simulations with EPISODE-CityChem. It is possible to use 2-D and 3-D fields of meteorological variables computed by the TAPM model using an inner model grid for TAPM which has the same model domain extent and horizontal resolution.

If TAPM is not available, it is possible to use a minimum set of meteorological variables (i.e. the mandatory meteorological input files) produced with the MCWIND model based on observational meteorological data (see section 6.1).

TAPM produces an output file with the suffix *.outa, from here on referred to as TAPM outa-file. The TAPM built-in tool TAPM2OUTA.exe converts selected *.out and *.rfl output files of TAPM for a particular date range to an *.outa file, in ASCII format. In the **TAPM Graphical User Interface (GUI)** chose from the menu “Utilities” – “Convert *.out,*.rfl files to ASCII *.outa file” (Figure 5). The TAPM outa-file has to include 30 vertical model layers. For a simulation period of one month it is recommended to create the *.outa file with start date 1st of the month and end date 1st of the next month.

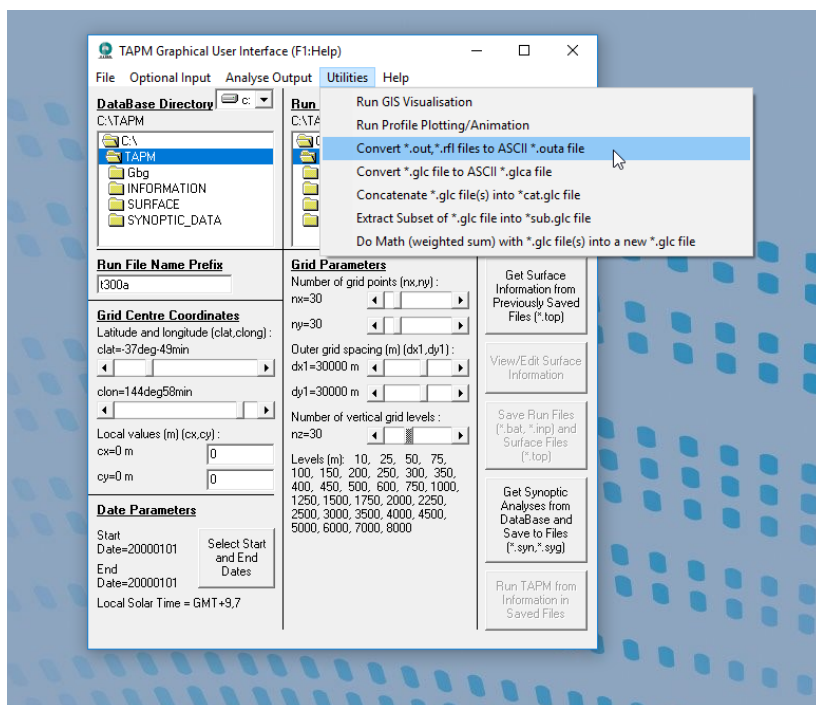


Figure 5: TAPM Graphical User Interface, selection of outa-file creation in the Utilities menu.

Fortran program **tapm4cc.exe**, included in the EPISODE-CityChem distribution, creates 2-D and 3-D input files for meteorology by reading the TAPM outa-file and writing files for all meteorological variables as input to CityChem.

It is recommended to select the binary output option ($ep_fmt = 2$) in the user metadata file (section 3.3), since binary files need less space on the computer disk. The option to write netCDF output ($ep_fmt = 3$) has to be used when it is planned to use temperature to calculate area emissions for residential heating with UECT. If the netCDF option is used, all meteorological variables from the TAPM out-file will be written to files in netCDF format. However, only the netCDF files containing air temperature at ground and vertical temperature gradient need to be stored for preparing temperature-dependent residential heating emissions.

Furthermore, it is recommended to use 2 extra hours for a one month simulation (hmonth), so instead of $24 \times 31 = 744$ hours, better use 746 hours.

Before using the `tapm4cc` tool, change the output file path in the user metadata file (“`cctapm_meta.inp`”) to ‘`./INPUT/tapm`’.

The program `tapm4cc.exe` is built by the automated installation routine:

```
{your-citychem}/$ ./installcc.csh util example
or (for self-made bcon files),
{your-citychem}/$ ./installcc.csh bcon example
```

Alternatively, the program **tapm4cc.exe** can be built as follows: Change to the subdirectory `preproc/tapm4cc2.0/` and edit in `Makefile.tapm` the variables `INCLUDES` and `LIBS`. Here the correct path of the netCDF library (see Appendix C) on your computer has to be given. By default the lines are:

```
INCLUDES = -I/usr/local/netcdf4/include
LIBS     = -L/usr/local/netcdf4/lib -lnetcdf -lnetcdff
```

Now build the program.

```
{your-citychem}/preproc/tapm4cc2.0/$ make -f Makefile.tapm
```

This creates **tapm4cc.exe**. With a symbolic link it becomes executable from subdirectory `preproc/`:

```
{your-citychem}/preproc/$ ln -s ./tapm4cc2.0/bin/tapm4cc.exe
```

Now `tapm4cc.exe` can be run. The meteorological input files are written to the output directory given in the user metadata file:

```
{your-citychem}/preproc/$ ./tapm4cc.exe
```

On the following pages the format of all input files containing meteorological fields are described.

File format is given in Fortran notation, where `ii` and `jj` are grid cell indices; `ix`, `iy`, and `iz` are the indices for cells along the x-coordinate direction, y-coordinate direction, and z-coordinate (vertical) direction of the domain, respectively. All numerical variables should be in float format `e16.8` in the input files.

- Air temperature at ground and vertical temperature gradient:
First air temperature field at ground (tair in degC) and second vertical temperature gradient field (dtdz in degC/m) for every simulation hour.
Read format (ascii):

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((i,j,tair(ix,iy),ix=1,nx),iy=1,ny)
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((i,j,dtdz(ix,iy),ix=1,nx),iy=1,ny)
```
- Wind u-, v-, w-components:
For each vertical layer, starting with lowest layer, first field of wind u-component (u in m/s), second field of wind v-component (v in m/s), third field of wind w-component (w in m/s) for every simulation hour.
Read format (ascii):

```
do iz=1,nz
  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((i,j,u(ix,iy,iz),ix=1,nx),iy=1,ny)

  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((i,j,v(ix,iy,iz),ix=1,nx),iy=1,ny)

  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((i,j,w(ix,iy,iz),ix=1,nx),iy=1,ny)
enddo
```
- Turbulence sigma-vw:
For each vertical layer, starting with lowest layer, first field of turbulence sigma-v (sigv in m/s) and second field of turbulence sigma-w (sigw in m/s) for every simulation hour.
This field is not produced by TAPM.
Read format (ascii):

```
do iz=1,nz
  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((i,j,sigv(ix,iy,iz),ix=1,nx),iy=1,ny)

  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((i,j,sigw(ix,iy,iz),ix=1,nx),iy=1,ny)
enddo
```
- Aerodynamic resistance:
Aerodynamic resistance field (aero in in s/m) for every simulation hour.
This field is not produced by TAPM.
Read format (ascii):

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((i,j,aero(ix,iy),ix=1,nx),iy=1,ny)
```

- **Mixing height:**
Mixing height field (hmix in m) for every simulation hour.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,hmix(ix,iy),ix=1,nx),iy=1,ny)`
- **Surface friction velocity scale:**
Friction velocity scale field at ground (ustr in m) for every simulation hour.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,ustr(ix,iy),ix=1,nx),iy=1,ny)`
- **Sensible heat flux**
Sensible heat flux field (shfl in W/m²) for every simulation hour.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,shfl(ix,iy),ix=1,nx),iy=1,ny)`
- **Latent (evaporative) heat flux:**
Latent heat flux field (lhfl in W/m²) for every simulation hour.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,lhfl(ix,iy),ix=1,nx),iy=1,ny)`
- **Land use category:**
Land use category is not used in the current version.
[For creation of the land use file, please refer to section 3.7.2](#)
Land use category (lanu corresponding to TAPM land use index) as one 2-D-field.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,lanu(ix,iy),ix=1,nx),iy=1,ny)`
- **3-D air temperature:**
For each vertical layer, starting with lowest layer, field of in-situ temperature (ins_t in K) for every simulation hour.
Read format (ascii):
`do iz=1,nz`
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,ins_t(ix,iy,iz),ix=1,nx),iy=1,ny)`
`enddo`
- **3-D Potential temperature:**
For each vertical layer, starting with lowest layer, field of potential temperature (pot_t in K) for every simulation hour.
Read format (ascii):
`do iz=1,nz`
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,pot_t(ix,iy,iz),ix=1,nx),iy=1,ny)`
`enddo`

- TAPM 3-D mid-layer geopotential height equivalent:
 For each vertical layer, starting with lowest layer, model layer level midpoint (gpot in m) derived from TAPM model.
 Read format (ascii):

```
do iz=1,nz
  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((ii,jj,gpot(ix,iy,iz),ix=1,nx),iy=1,ny)
enddo
```
- Surface water vapour flux:
 Surface water vapour flux field (mflx in kg/(m²s)) for every simulation hour. This field is not produced by TAPM.
 Read format (ascii):

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,mflx(ix,iy),ix=1,nx),iy=1,ny)
```
- Surface momentum stress:
 First field of surface momentum stress in x-direction (taus_x in N/m²) and second field of surface momentum stress in y-direction (taus_y in N/m²) for every simulation hour.
 This field is not produced by TAPM.

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,taus_x(ix,iy),ix=1,nx),iy=1,ny)
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,taus_y(ix,iy),ix=1,nx),iy=1,ny)
```
- Potential temperature scale:
 Potential temperature scale field (ptstr in K) for every simulation hour.
 Read format (ascii):

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,ptstr(ix,iy),ix=1,nx),iy=1,ny)
```
- Potential virtual temperature scale:
 Potential virtual temperature scale field (pvstr in K) for every simulation hour.
 Read format (ascii):

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,pvstr(ix,iy),ix=1,nx),iy=1,ny)
```
- Convective velocity scale:
 Convective velocity scale field (wstr in m/s) for every simulation hour.
 Read format (ascii):

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,wstr(ix,iy),ix=1,nx),iy=1,ny)
```
- Total solar radiation (global radiation):
 Total solar radiation field (tsrad in W/m²) for every simulation hour.
 Read format (ascii):

```
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,tsrad(ix,iy),ix=1,nx),iy=1,ny)
```


- 2D ground temperature:
Ground (screen-level) temperature field (gtmp in K) for every simulation hour.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,gtmp(ix,iy),ix=1,nx),iy=1,ny)`
- Precipitation:
Precipitation (rainfall) field (prec in mm/h) for every simulation hour.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,prec(ix,iy),ix=1,nx),iy=1,ny)`
- Relative humidity:
Ground (screen-level) relative humidity field (rhum as fraction; values: 0, ..., 1) for every simulation hour.
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,rhum(ix,iy),ix=1,nx),iy=1,ny)`
- Cloud cover:
Cloud fraction field (clou as fraction; values: 0, ..., 1) for every simulation hour.
[For creation of the cloud fraction file, please refer to section 3.7.2](#)
Read format (ascii):
`read("(2a10.2i3)") TEXT1, TEXT2, nx, ny`
`read(*) ((ii,jj,clou(ix,iy),ix=1,nx),iy=1,ny)`

3.5 Emission input

For all chemical compound for which EPISODE-CityChem produces output, emissions from point sources, line sources (road traffic, ferry lines, airport lanes) and area sources can be provided. Emissions from line sources and area sources are in separate files for each individual compound.

File format is given in Fortran notation, where ii and jj are grid cell indices; ix and iy the indices for cells along the x -coordinate direction and y -coordinate direction of the domain, respectively. All emission data values should be in float format $e16.8$ in the input files.

Section 6.2 describes the use of the utility UECT (Urban Emission Conversion Tool) to prepare the various emission input files for point sources, line sources and area source categories. UECT prepares emission input files for CityChem containing hourly varying emission data based on emission data of geo-referenced yearly emission totals for nitrogen oxides (NO_x), non-methane volatile organic compounds (NMVOC), carbon monoxide (CO), sulphur dioxide (SO_2), ammonia (NH_3), as well as particulate matter $\text{PM}_{2.5}$ and PM_{10} , defined for each source. Missing emission totals are indicated by -999 in the input to UECT.

The geo-reference for point sources is the (x,y) -coordinate of the point, the geo-reference for lines sources is the start (x,y) -coordinate together with the end (x,y) -coordinate of the line, and the geo-reference for area sources is the (x,y) -coordinate of the lower left (southwest) corner together with the (x,y) -coordinate of the upper right (northeast) corner of the quadratic area cell. Area sources have to be located within a regular Cartesian grid, i.e. when area sources from an ArcGIS polygon shape are used, these have to be intersected first with a raster grid. Figure 6 illustrates the three emission source objects.

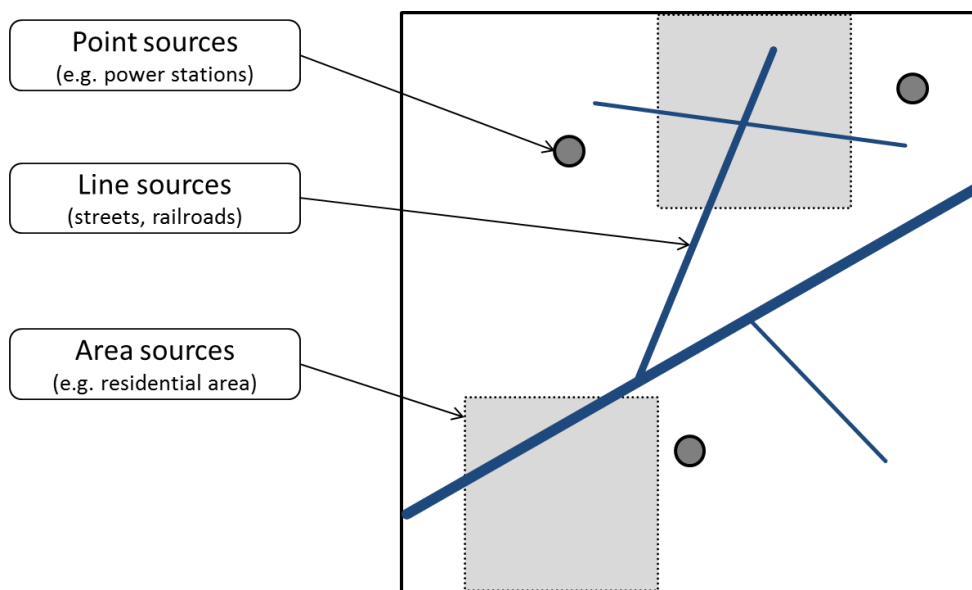


Figure 6: Illustration of the three emissions source objects.

3.5.1 Input files for point sources

In this section the format of input files containing point source emissions are described. All information about the point sources (exhaust stack properties) and emissions of all compounds (of the CityChem output) for all point sources in the model domain are included in one point emission file. The point source file contains first 4 lines with information about chemical compounds and number of sources and second a field of coordinates, exhaust stack information and compound emission data (qemvec in g/s) for each point source followed by one line of the date of emissions. The second part is read every simulation hour.

Read format (ascii):

```
read(*) nqvec,(cmpvec(j),jvec(j),j=1,nqvec)
```

Where nqvec is the number of compounds (has to be the same as the total number of chemical compounds which are included in the CityChem output), j is an index of compounds, cmpvec is a string (a10) with the name of the compound, and jvec is the running number of the compound.

```
read(*) iqu,iqt
```

Reads the indicator of emission data units (iqu) and temperature units (iqt). For units g/s and degC both indicators are 1.

```
read(*) (chtime(i),i=1,6)
```

Reads the start date and time of the emission data in format

yy m d h min sec

```
read(*) np
```

Reads the total number of point sources (np) for which exhaust stack and emission data is provided.

Then read for every simulation hour:

```
do ip = 1, np
```

```
  read(*) qidv(ip),qxv(ip),qyv(ip),qzv(ip),qhsv(ip),qdiv(ip), &
    qhbv(ip),qwbv(ip),qtev(ip),qtgv(ip),qvgv(ip), &
    (qemvec(j,ip),j=1,nqvec)
```

```
enddo
```

```
read(*) (chtime(i),i=1,6)
```

Reads the following list of variables for each point source ip:

qidv	point source ID
qxv	x-coordinate of point source
qyv	y-coordinate of point source
qzv	z-coordinate of point source
qhsv	Exhaust stack height (m)
qdiv	Exhaust stack diameter (m)
qhbv	Stack building height (m) (optional)
qwbv	Stack building width (m) (optional)
qiev	Point source thermal energy (MW) (optional, not used)
qtgv	Exhaust temperature (degC)
qvgv	Exhaust flux velocity (m/s)

Finally reads a line with the date and time of the simulation hour. Default values for building height is 10 (m), for building width is 20 (m) and for thermal energy is -9900 (missing value).

3.5.2 *Input files for line sources*

The input data set for line sources (linear mobile sources from traffic) consists of one line source meta information file and line emission data files for each chemical compound of the CityChem output. A missing line emission input file for a specific compound is ignored by CityChem and means zero emission for this compound.

The line source meta information file (linesource_metadata.txt) contains the data of all line sources (i.e. road links) for which emissions are allocated. The total number of line sources is nql. It starts with 8 lines of comments, followed by a list of the line source data (x- and y-coordinates of start and end point, width of the lane, etc.):

Read format (ascii):

```
read("(a50)") COMMENT1
read("(a50)") COMMENT2
read("(a50)") COMMENT3
read("(a50)") COMMENT4
read("(a50)") COMMENT5
read("(a50)") COMMENT6
read("(a50)") COMMENT7
read("(a50)") COMMENT8
```

```
do L = 1, nql
```

```
  read(*) iqlv,x1v,x2v,y1v,y2v,z1v,z2v,wv,rmaxv,qlriv
enddo
```

Reads the following list of variables for each line source L:

```
iqlv    line source ID
x1v     Line source start point x-coordinate
x2v     Line source end point x-coordinate
y1v     Line source start point y-coordinate
y2v     Line source end point y-coordinate
z1v     Line source start point z-coordinate (model layer level;
        0 for ground level)
z2v     Line source end point z-coordinate (model layer level;
        0 for ground level)
wv      Line source width of lane (m)
rmaxv   Maximum influence distance for receptor points;
        highest value is 500 m (m)
qlriv   Line source index (0 if no line receptor point output,
        1 if line receptor point output wanted)
```

For each chemical compound of the CityChem output a line source emission data file can be provided. A line source file contains the emission data (ql1v in g/s) for an individual chemical compound for all line sources for every simulation hour. Please note that the emission data unit is g/s and not g/(s m). Therefore emission data given per line source length has to be multiplied by the line source length.

```

Read format (ascii):
read("(a50)") COMMENT1
read("(a50)") COMMENT2
read("(a50)") COMMENT3
read("(a50)") COMMENT4
read("(a50)") COMMENT5
read("(a50)") COMMENT6
read("(a50)") COMMENT7
read("(a50)") COMMENT8

```

Then the line source emission data for all line sources for every simulation hour are following:

```

do n = 1, tend
  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  do L = 1, nql
    read(*) ql1v(L)
  enddo
enddo

```

3.5.3 *Input files for area sources*

In this section the format of input files containing area source emissions are described. For each chemical compound for which EPISODE-CityChem produces output an area emission file can be provided for each emission category (residential heating; solvent use and agriculture; industrial and commercial sources; shipping). A missing area emission input file for a specific compound in a specific category is ignored by CityChem and means zero emission for this compound and category.

An area emission file contains first 8 lines of comments and then for every simulation hour the field of area emissions (qaorig in g/s) for the specific compound and category.

```

Read format (ascii):
read("(a50)") COMMENT1
read("(a50)") COMMENT2
read("(a50)") COMMENT3
read("(a50)") COMMENT4
read("(a50)") COMMENT5
read("(a50)") COMMENT6
read("(a50)") COMMENT7
read("(a50)") COMMENT8

do n = 1, tend
  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((ii,jj,qaorig(ix,iy),ix=1,nx),iy=1,ny)
enddo

```

If TEXT1 or TEXT2 contain the keyword 'no index' then the file is read without the grid cell indices:

```

read(*) ((qaorig(ix,iy),ix=1,nx),iy=1,ny)

```

This is the usual format.

3.6 Regional background input

In EPISODE-CityChem it is possible to use regional background concentrations in the form of a 3-D boundary conditions (BCON) field generated from output of the community multi-pollutant air quality model CMAQ (*Byun and Ching, 1999; Byun and Schere., 2006; Appel et al., 2013*). The background concentrations are adopted for the grid cells directly adjacent to the grid cells of the model domain (with $n_x \times n_y$ cells per model layer) and also for the vertical model layer that is on top of the highest model layer.

There are two options for using background concentrations in the simulation run. The first option is to use one constant concentration value for each CityChem compound. The second option is to use the hourly 3-D BCON fields, one file for each compound. The decision is made in the run script with the background concentration option flag. It is set to 1 for constant values and to 3 for 3-D BCON files (section 4.9).

Constant background concentrations:

The first option, to use constant values. The option should be chosen if no CMAQ model concentration data is available. This option requires only one background concentration value per CityChem compound (in unit $\mu\text{g}/\text{m}^3$ for gas-phase species and particulate matter, in unit particles/ cm^3 for particle numbers), which will be constant throughout the simulation period. This option is very convenient for running model tests or for getting started quickly with the simulation run for a city. However, for research purposes and for air quality management studies it is recommended to use the hourly 3-D BCON fields from a CMAQ simulation to constrain the flow of pollutants into and out of the model domain.

CMAQ 3-D boundary conditions:

For obtaining the 3-D boundary conditions it is necessary to run the CMAQ model for a regional domain of several hundreds of kilometres, including the domain of the target city. The 3-D BCON file has to be provided for each CityChem compound. The program will stop if the 3-D BCON option was selected and the 3-D BCON file does not exist. The utility BCONCC has been developed to convert the CMAQ concentration output (CONC* files) to EPISODE-format (binary) files. The technical guide line to use the utility BCONCC is described in section 6.3. The meta information on 3-D BCON conversion has to be included in the user metadata file (“cctapm_meta.inp”) described in section 3.3.

Format of the file:

For each vertical layer, starting with lowest layer, field of background concentration for every simulation hour.

Read format (binary):

```
read(*) TEXT1,TEXT2,nx_D,ny_D, ( ( BCON(ic,ix,iy,iz),
                                ix=1,nx+2*nbcx),iy=1,ny+2*ncby)
```

Where $n_x_D = n_x + 2 \cdot n_{bcx}$ and $n_y_D = n_y + 2 \cdot n_{bcy}$, with n_{bcx} and n_{bcy} the number of boundary grid cells (=1) adjacent to each domain border.

3.7 Auxiliary input

There are some auxiliary input files which need to be located in the subdirectory INPUT/other/. Table 4 shows a list of the auxiliary input files.

Table 4: Auxiliary input files.

Filename	Content	Category
receptor_stations_raster.txt	x- and y-coordinates of monitoring station and raster grid of receptor points	geography
old_plume.dat	Plume segments from previous time step	point sources
landuse_episode.asc (ascii) or landuse_episode.fld (binary)	Static 2-D field of land use information	geography
topo.asc (ascii) or topo.fld (binary)	Static 2-D field of topography (terrain heights)	geography
surfrough_episode.asc (ascii) or surfrough_episode.fld (binary)	Static 2-D field of surface roughness	meteorology
clou_episode.asc (ascii) or clou_episode.fld (binary)	2-D field of cloud fraction for every simulation hour	meteorology

On the following pages a detailed description of the input files, their file format and how they can be created, is given.

Plume segments from previous time step:

The option to provide the remaining plume segments from the previous time step (one time step before the start time of the simulation) is a preliminary development. Therefore it is intended to provide only a dummy file with zero remaining plume segments.

A dummy file old_plume.dat is located in the subdirectory INPUT/other. It needs to be prepared for the simulation run.

The second line of the file contains the start date. It needs to be changed to the actual start date of the simulation:

<i>Start column:</i>	<i>property</i>	<i>default value</i>
11	number of plume segments	0
20	simulation start year	2013
35	simulation start month	1
47	simulation start day	1
59	simulation start minute	0

3.7.1 Receptor grid

The definition of a regular raster grid of receptor points (“receptor grid”) and geographic positions of air quality monitoring stations is essential to get output of concentration time series at high spatial resolution or in a specific point.

The file `receptor_stations_raster.txt` (an example of this file is in subdirectory `INPUT/other/`) first contains x- and y-coordinates of the monitoring stations and second contains the list of x-coordinates and y-coordinates of the points from the regular receptor raster grid. For the included monitoring stations, time series of hourly pollutant concentrations will be computed and written to the model output.

Format of the file:

List of x-coordinates and y-coordinates of the receptor points.

Read format (ascii):

```
read("(a256)") sim_id
read("(a256)") COMMENT1
read("(a256)") COMMENT2
```

Then the list of monitoring sites:

```
do m = 1, nmp
  read(*) sim_id, xcor(m),ycor(m),zcor(m),rcmax(m),name(m)
enddo
```

Then the list of points from the regular receptor raster grid:



```
do r = 1, nrp
  read(*) sim_id, xcor(r),ycor(r),zcor(r),rcmax(r),name(r)
enddo
```

Where `sim_id` is a text string containing a user-defined simulation id for the model run, `nmp` is total number of monitoring sites, `nrp` is the total number of receptor points in the raster grid, and `name` is a text string with the name or number of the receptor point / station. Further, `rcmax` is the maximum distance (in m) of a source to the receptor point, to be taken into account. The variable is not used and therefore missing value (-9900) should be entered. The location of the point is defined by `xcor`, the x-coordinate (UTM-coordinate in m), `ycor`, the y-coordinate (UTM-coordinate in m), and `zcor`, the height (in m) above ground.

Coordinates of the monitoring stations have to be added manually in the beginning of the input file.

The raster grid of receptor points can be created in Esri® ArcMap™. The result will be an ASCII file, which contains x- and y-coordinates of receptor points in a grid of predefined size. To create this grid:

- ArcMap™ has to be launched.
- Right-mouse click on the current “Layers” and go to Properties. Select Coordinate System, then click through Projected Coordinate Systems, UTM, WGS1984, Northern Hemisphere, until you find the UTM zone of the research area. This turns the layer and all added features into the right projection.

- A random Basemap should be added by using the ‘Add data’  and ‘Add basemap ...’  buttons.
- An overview of the research area should be displayed by navigating to it.

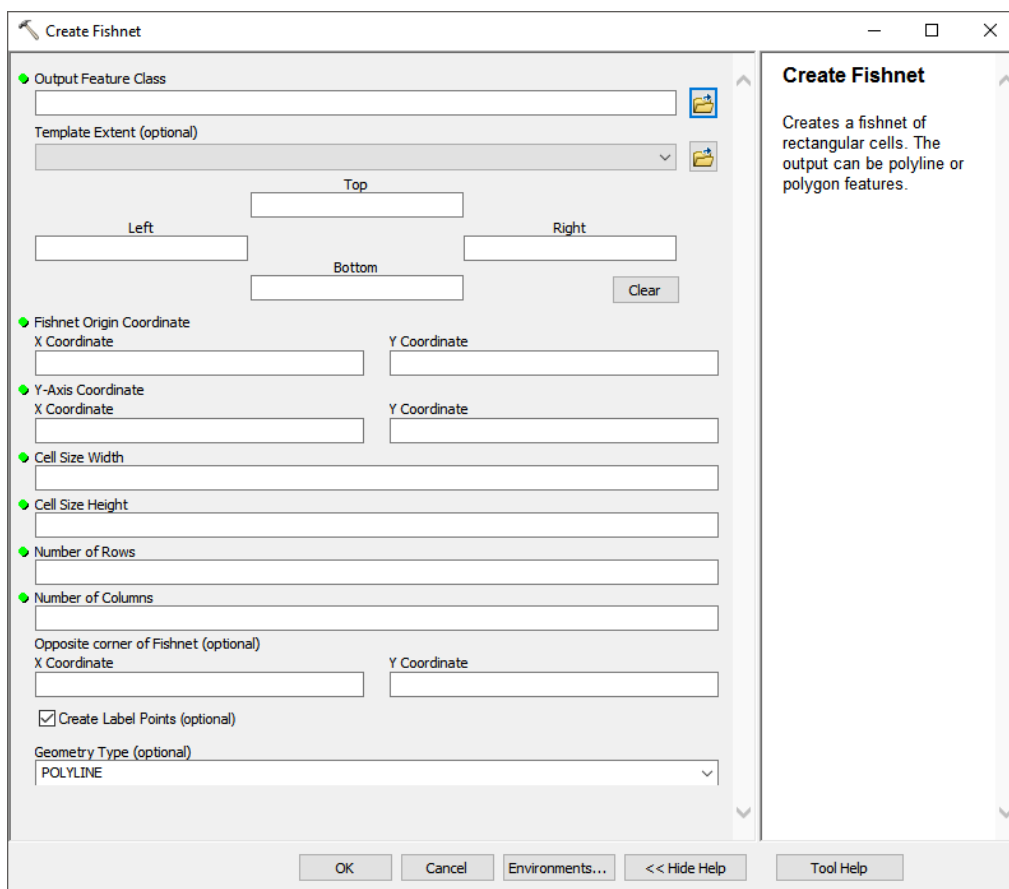



Figure 7: Screenshot of the Esri® ArcMap™ tool “Create Fishnet (Data Management)”.

- To create a grid of receptor points within ArcGIS, the ‘Fishnet’ tool will be used. This tool creates both, a grid of cells and a grid of points which are located in the centre of the cells:
 - Open the tool ‘Create Fishnet (Data Management)’ whether by navigating through the toolbox or by using the Search. Figure 7 shows a screenshot of the ‘Fishnet’ tool.
 - Output Feature Class: The output folder and name of the Fishnet has to be chosen.
 - Fishnet origin coordinate: It is necessary to know the originate X and Y coordinates of the raster, which are located in the SW corner of the raster grid of receptor cells – if there are only coordinates for the SW receptor points it is necessary to calculate the origin of the SW receptor cell in dependence on the cell size, by subtracting half of the cell size in X and Y direction from the receptor point.

- X coordinate: The X coordinate of the SW corner of the receptor grid in accordance to the chosen projection or coordinate system has to be entered.
 - Y coordinate: The Y coordinate of the SW corner of the receptor grid in accordance to the chosen projection or coordinate system has to be entered.
 - Y-Axis Coordinate: The Y-axis coordinate is used to orient the fishnet. The fishnet is rotated by the same angle as defined by the line connecting the origin and the y-axis coordinate.
 - X coordinate: The X coordinate of any point that will be connected to the origin for creating a line to orient the grid, in accordance to the chosen projection or coordinate system has to be entered.
 - Y coordinate: The Y coordinate of any point that will be connected to the origin for creating a line to orient the grid, in accordance to the chosen projection or coordinate system has to be entered.
 - Cell Size Width: The cell size width of each cell in meter has to be entered
 - Cell Size Height: The cell size height of each cell in meter has to be enter
 - Number of Rows: The Number of rows has to be entered
 - Number of Columns: The number of columns has to be entered
 - Create Label options (optional): This option has to be activated
 - Geometry Type (optional): The option POLYGON has to be chosen.
 - By clicking ‘OK’ the fishnet will be created.
- The fishnet will be created and both features, the cells and the points, will be displayed upon your research area.
 - To add the x- and y- coordinates to every point, the ‘Add XY Coordinates (Data Management)’ tool has to be opened and the created fishnet with the suffix ‘_label’ has to be selected. By clicking OK, every point will get its x- and y- coordinate related to the chosen coordinate system or projection.
 - To export the grid into an ASCII file, the created fishnet with the suffix “_label” in the table of contents has to be selected and within the context menu the ‘Open Attribute Table’ has to be selected.
 - A window ‘Table’ will open. In the table options  ‘Export...’ has to be chosen. Figure 8 shows a screenshot of the ‘Export Data’ pop-up window.
 - ‘Export: All records’ has to be selected and a place and output format for the table has to be defined.
 - By clicking ‘OK’ the file with all receptor points and their XY coordinates will be created at the chosen destination with the chosen file format.

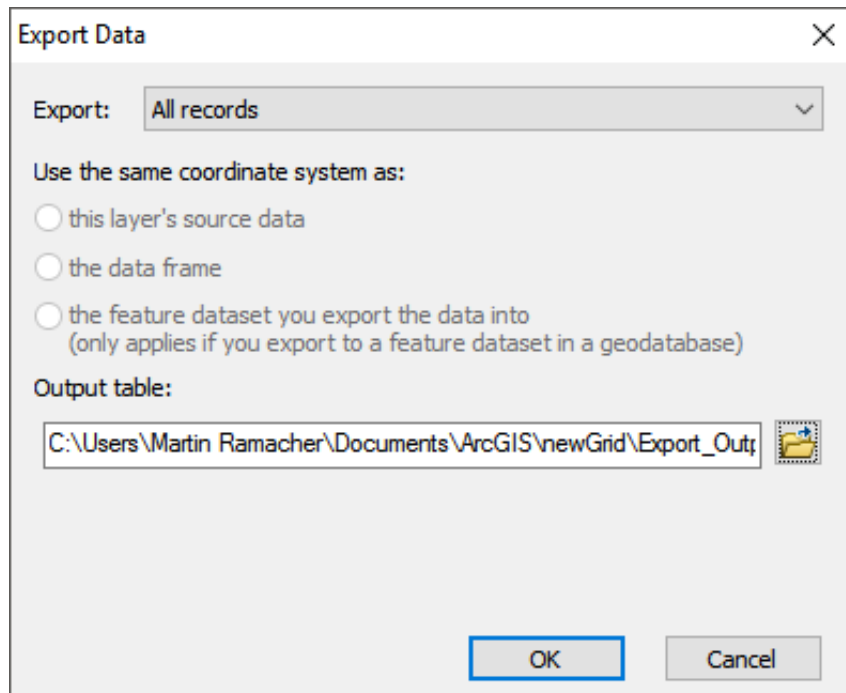


Figure 8: Screenshot of the Esri® ArcMap™ pop-up window “Export Data”.

Finally the output table is included by copy-paste in an Excel worksheet, the remaining columns (fields sim_id, zcor, rcmx and name) are added manually. The resulting Excel table is transferred to the file receptor_stations_raster.txt by copying the relevant part, and pasting it to a text file in a text editor of your choice.

3.7.2 Auxiliary 2-D fields

Mandatory 2-D fields as input to EPISODE-CityChem include a static field of topography (terrain heights), a static field of surface roughness length (homogeneous or heterogeneous), and a time-varying field of fractional cloud cover. Further, a static field of land use (homogeneous or heterogeneous) can be created, which is required by the utility MCWIND.

Auxiliary 2-D fields input files:

1. The first auxiliary input is the **static 2-D field of topography**, i.e. terrain elevation. The topography input file topo.asc (ASCII format) or topo.fld (binary format) is a static file that contains the terrain heights (in m) of the model grid.

The topography input file can be provided in either ASCII or binary format.

Format of the file

Topography (terrain heights) field (topo in m) as one 2-D field for the model domain.

Read format (ascii):

```
read("(2a10.2i3") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,topo(ix,iy),ix=1,nx),iy=1,ny)
```

2. The second auxiliary input is the **static 2-D field of land use**. A land use input file in CityChem format is required by MCWIND, but not used in the calculation. EPISODE-CityChem currently does not require land use information, however it is planned for future versions.

Format of the file:

Land use classes (lu) as one 2-D field for the model domain. Land use classification is adopted from TAPM. The value of lu is dimensionless integer between -1 and 38.

Read format (ascii):

```
read("(2a10.2i3") TEXT1, TEXT2, nx, ny
read(*) ((ii,jj,lu(ix,iy),ix=1,nx),iy=1,ny)
```

3. The third auxiliary input is the **static 2-D field of surface roughness length**. The surface roughness input file can be provided in either ASCII or binary format.

Format of the file:

Surface roughness length field (z0 in m) as one 2D field for the model domain.

```

Read format (ascii):
read("(2a10.2i3)") TEXT1, TEXT2, nx, ny3333
read(*) ((i,j,z0(ix,iy),ix=1,nx),iy=1,ny)

```

4. The fourth auxiliary input is the **dynamic 2-D field of fractional cloud cover**. The cloud fraction input file can be provided in either ASCII or binary format.

TAPM does not produce the cloud fraction file. Please note that in EPISODE-CityChem the total solar radiation produced by TAPM (if supplied) is used instead of cloud fraction to calculate the photolysis rate constants but still the cloud fraction input file has to be provided. MCWIND produces the cloud fraction file based on the meteorological observation of cloud fraction one met station.

Format of the file:

Cloud fraction field (clou as fraction; values: 0, ..., 1) for every simulation hour.

```

Read format (ascii):
do n = 1,tend
  read("(2a10.2i3)") TEXT1, TEXT2, nx, ny
  read(*) ((i,j,clou(ix,iy),ix=1,nx),iy=1,ny)
enddo

```

The next part of this section describes the procedure of creating input files of auxiliary 2-D fields from TAPM output, the last part of this section describes the procedure for the case that TAPM is not available.

Producing auxiliary 2-D fields from TAPM output

Topography data is included in the TAPM outa-file. The topography file is automatically created when using the tool **tapm4cc.exe** located in directory {your-citychem}/preproc/tapm4cc/. The file is written in ASCII format to subdirectory INPUT/other/ (when using the default user metadata file).

The vegetation (land use) types used in TAPM are based on a CSIRO Wildlife and Ecology Categorisation (Hurley, 2008). Table 5 lists the TAPM land use classes together with vegetation height (h_f in m) and surface roughness length (z_0 in m). In addition there are eight urban land use classes (index 31-38). A spatial heterogeneous surface roughness input file can be produced based on the TAPM land use information. The roughness length for different TAPM land use classes (“vegetation types”) is derived with the vegetation heights used for the TAPM land use classes. For vegetation land use, roughness length (z_0 in m) is obtained from the vegetation height by division with 10. For urban land use, z_0 is defined in Table 2 of the technical documentation of TAPM (Hurley, 2008). Table 5 lists the vegetation height and surface roughness length of all TAPM land use classes.

The **TAPM Graphical User Interface (GUI) Surface Window** (appears by pressing the View/Edit Surface Information button in the TAPM GUI), allows the user to save the land use classes information of the selected grid (“Pollution Grid”) in a *.top file, when the “Save Run Files (*.bat, *.inp) and Surface Files (*.top)” button is pressed. The saved *.top file should be opened with a text editor. The land use data in the *.top file starts after $1 + 2[(x \times y)/10]$ lines (with x and y as raster dimensions, given in the first line of the *.top file) and ends after of $x \times y$ values or $(x \times y)/10$ lines. Now, the land use type values in the *.top file can be copied & saved to a new file and saved as landuse.top to INPUT/other/. All land use class indices in landuse.top with a value below 10 have to be preceded by a space (e.g. replace ‘4’ by ‘ 4’). All values have to be separated by a single space ‘ ’ (replace ‘,’ or ‘;’ separator by ‘ ’). The tool **z0top4cc.exe** will read the land use class values in landuse.top and attribute the corresponding z_0 values based on Table 5. It also produces a homogenous cloud fraction clou field with a constant value for every simulation hour.

Tool **z0top4cc.exe** will write landuse_episode.asc, surfrough_episode.asc, and clou_episode.asc. Before using the tool change the output file path in the user metadata file to ‘./INPUT/other’. This is the directory for the input file landuse.top and for the output files. It is recommended to set the output format to ASCII in the user metadata file. Now build **z0top4cc.exe** as follows:

```
{your-citychem}/preproc/auxiliary/$ make -f Makfile.zo
```

This creates **z0top4cc.exe**. With a symbolic link it becomes executable from subdirectory preproc/:

```
{your-citychem}/preproc/$ ln -s ./auxiliary/bin/z0top4cc.exe
```

Now **z0top4cc.exe** can be run:

```
{your-citychem}/preproc/$ ./z0top4cc.exe
```

Table 5: TAPM land use classes, vegetation heights and surface roughness length.

Index	TAPM land-use class	Vegetation height (m)	z0 (m)
-1	Permanent snow/ice	–	0.000
0	Water	–	0.000
1	Forest – tall dense	42.00	4.200
2	Forest – tall –mid-dense	36.50	3.650
3	Forest – dense	25.00	2.500
4	Forest – mid-dense	17.00	1.700
5	Forest – sparse (woodland)	12.00	1.200
6	Forest – very sparse	10.00	1.000
7	Forest – low dense	9.00	0.900
8	Forest – low mid-dense	7.00	0.700
9	Forest – low sparse (woodland)	5.50	0.550
10	Scrub-land – tall mid-dense (scrub)	3.00	0.300
11	Scrub-land – tall sparse	2.50	0.250
12	Scrub-land – tall very sparse	2.00	0.200
13	Scrub-land – low mid-dense	1.00	0.100
14	Scrub-land – low sparse	0.60	0.060
15	Scrub-land – low very sparse	0.50	0.050
16	Grassland – sparse hummock	0.50	0.050
17	Grassland – very sparse hummock	0.45	0.045
18	Grassland – dense tussock	0.75	0.075
19	Grassland – mid-dense tussock	0.60	0.060
20	Grassland – sparse tussock	0.45	0.045
21	Grassland – very sparse tussock	0.40	0.040
22	Pasture/herb-field – dense (perennial)	0.60	0.060
23	Pasture/herb-field – dense (seasonal)	0.60	0.060
24	Pasture/herb-field – mid-dense (perennial)	0.45	0.045
25	Pasture/herb-field – mid-dense (seasonal)	0.45	0.045
26	Pasture/herb-field – sparse	0.35	0.035
27	Pasture/herb-field – very sparse	0.30	0.030
28	Littoral	2.50	0.250
29	Permanent lake	–	0.000
30	Ephemeral lake (salt)	–	0.000
31	Urban	10.00	1.000
32	Urban (low)	8.00	0.400
33	Urban (medium)	12.00	0.600
34	Urban (high)	16.00	0.800
35	Urban (cbd)	20.00	2.000
36	Industrial (low)	10.00	0.500
37	Industrial (medium)	10.00	1.000
38	Industrial (high)	10.00	1.500

Producing auxiliary 2-D fields without TAPM

The terrain pre-processor AERMAP (EPA-454/B-03-003, October 2004) of the U.S. EPA air dispersion model AERMOD (*U.S. EPA*, 2004) can be used to create the topography file. AERMAP is a terrain pre-processor which coordinates the allocation of terrain elevation data from several digitized data bases to a user-specified model grid.

The version of AERMAP downloaded for this guideline was version 11103. AERMAP is included in the distributed CityChem tar-file. The AERMAP is documented at: <https://www.epa.gov/scram/air-quality-dispersion-modeling-related-model-support-programs>.

The AERMAP documentation and source code (aermap_source.zip) is available from the U.S. EPA website:

https://www3.epa.gov/ttn/scram/models/aermod/aermap/aermap_source.zip

Installation of AERMAP: a separate installation of AERMAP is normally not required, since it becomes installed when you issue:

```
{your-citychem}/$ ./installcc.csh util example
```

If this is not working for any reason, copy aermap.zip to directory {your-citychem}/preproc/auxiliary/aermap/ and extract the content of the zip folder to this directory. Use Makefile.aermap in subdirectory preproc/auxiliary/ to build the executable program AERMAP (aermap.exe):

```
{your-citychem}/preproc/auxiliary/$ make -f Makfile.aermap
```

This creates aermap.exe. With a symbolic link it becomes executable from subdirectory preproc/auxiliary/:

```
{your-citychem}/preproc/auxiliary/$ ln -s ./bin/aermap.exe
```

Here follows a step-by-step description to create topo.asc (or topo.fld) using AERMAP:

- Beginning with the version 09040, AERMAP can process terrain data derived from NASA's Shuttle Radar Topography Mission (SRTM, *Rodriguez et al.*, 2005). SRTM digital elevation data is available at 3 arc-second resolution. SRTM3 has a spatial resolution of ca. 100 m and uses WGS 84 as reference geoid. Download Digital Elevation Data SRTM3 for the region of interest from the map of the world at:

http://www.viewfinderpanoramas.org/Coverage%20map%20viewfinderpanoramas_org3.htm

Unpacking the downloaded zip folder results several SRTM3 one by one degree latitude and longitude tiles as *.hgt.zip files. File names refer to the latitude and longitude of the lower left corner of the tile. For the latitude-longitude area of interest the correct *.hgt.zip file can be located according to “NlatElon.hgt.zip”. For example, when looking for the area at latitude 53°-54° N and at longitude 9°-10° E, the file to be used is N53E009.hgt.zip.

- Extract the selected *.hgt.zip file to subdirectory preproc/auxiliary/srtm3. Each SRTM3 zip folder contains one HGT file which is height data in a binary format. It needs to be converted to GeoTIFF format before it can be used with AERMAP.
- First, download the shell script “srtm_generate_hdr.sh” at grassbock.org:

<http://www.grassbook.org/wp-content/uploads/neteler/rs/srtm/>

Save to subdirectory preproc/auxiliary/.

- Second, install GDAL on your Linux computer. On Ubuntu Linux, the installation starts by typing:

```
{your-citychem}/preproc/auxiliary/$ sudo apt-get install gdal-bin
```

The installer command will be different for other Linux systems.

- After successful installation of GDAL the shell script can be used. The shell scripts looks for *.hgt.zip files. In case the height file is not zipped, do the following:

```
{your-citychem}/preproc/auxiliary/srtm3$ zip N53E009.hgt.zip N53E009.hgt
```

You can then start the conversion by typing:

```
{your-citychem}/preproc/auxiliary/$ ./srtm_generate_hdr.sh
./srtm3/N53E009.hgt.zip
```

```
{your-citychem}/preproc/auxiliary/$ mv N53E009* ./srtm3/
```

This will create N53E009.tif and other files in subdirectory preproc/auxiliary/srtm3. The *.tif is the GeoTIFF file that can be used as input for AERMAP.

- Edit the AERMAP user control file “aermap.inp” provided in subdirectory preproc/auxiliary/. ”aermap.inp” is an input file in ASCII format that specifies the filenames and type of databases being processed and the parameters related to the model study domain. The following lines need to be edited:

DATAFILE Provide name of the input *.tif file

DOMAINXY

Lower left corner and upper right corner of the model domain in format UTME UTMN ZN; where UTME is the east-west coordinate in m, UTMN is the south-north coordinate in m and ZN is the UTM zone. Subtract 1000 m from UTME and from UTMN of the lower left corner and add 1000 m to the UTME and to the UTMN of the upper right corner to allow finding of the corresponding receptor points by AERMAP.

ANCHORXY

Anchor point, i.e. mid-point of the model domain in format X-point Y-point UTME UTMN ZN NAD; where X-Point and Y-point are 0.0, UTME and UTMN of the mid-point are obtained for a 20 km x 20 km domain by adding 10 000 m to UTME and to UTMN of the lower left corner, ZN is the UTM zone of the mid-point, and NAD is set to 3 for WGS84 (see Table 3-1 of the AERMAP User's Guide).

XYINC

For output on Cartesian grid. Between GRIDCART CART1 STA and GRIDCART CART1 END in format xinit xnum dx yinit ynum dy; where xinit and yinit are the start points in m on the x-coordinate and y-coordinate, xnum and ynum are the number of receptor points (i.e. grid cells) in x- and y-direction, dx and dy are the grid cell width in m in x- and y-direction. For the example of a 20 km x 20 km domain with 1-km grid width, xinit and yinit are -10000.0, xnum and ynum are 20, and dx and dy are 1000.0.

- Figure 9 shows a screenshot of “aermap.inp” input file of AERMAP.

```

1 CO STARTING
2 TITLEONE Using 3 arc-second SRTM File for N60E024
3 TITLETWO With NAD83-Equivalent Anchor Point
4 DATATYPE NED
5
6 ** Use 3 arc-second SRTM File for Helsinki area (WGS84)
7 ** N60E024 and N60E025 needed to cover Helsinki area
8 DATAFILE "../geotiff/N60E024.tif" tiffdebug
9 DATAFILE "../geotiff/N60E025.tif" tiffdebug
10
11 ** Request ALL Debug Output Files
12 DEBUGOPT ALL
13
14 ** x-500 m x+500m
15 ** Lower Left corner Upper right corner
16 ** LL UTME LL UTMN Zn UR UTME UR UTMN Zn
17 DOMAINXY 378455.0 6661198.0 35 394455.0 6676198.0 35
18
19 ** NAD83 anchor point: (use NAD84 for WGS84)
20 ** NAD code in TABLE 3-1 of AERMAP User Guide
21 ** WGS84 = 3
22 ** Mid-point = x1+7.5km, y1+7.5km
23 ** X-point Y-point UTME UTMN Zn NAD
24 ANCHORXY 0.0 0.0 386455.0 6668698.0 35 3
25
26 NADGRIDS ..\
27 RUNORNOT RUN
28 CO FINISHED
29
30 ** Output elevation on a Cartesian grid
31 ** xinit xnum dx yinit ynum dy
32 RE STARTING
33 GRIDCART CART1 STA
34 | XYINC -7500.0 150 100.0 -7500.0 150 100.0
35 GRIDCART CART1 END
36 RE FINISHED
37
Normal text file length:1370 lines:43 Ln:2 Col:54 Sel:0|0 Dos/Windows UTF-8 INS

```

Figure 9: Screenshot of the “aermap.inp” configuration file for AERMAP.

- Now AERMAP can be run. AERMAP produces a terrain elevation output file (AERMAP.REC) in ASCII format. Other output files from AERMAP can be ignored (list file, plot file, save file):

```
{your-citychem}/preproc/auxiliary/$ ./aermap.exe
```

```
{your-citychem}/preproc/auxiliary/$ mv *.OUT *.out *.REC TiffDebug* ./output/
```

- Fortran tool **static4cc.exe** reads the file AERMAP.REC and converts it into topo.asc or topo.fld.

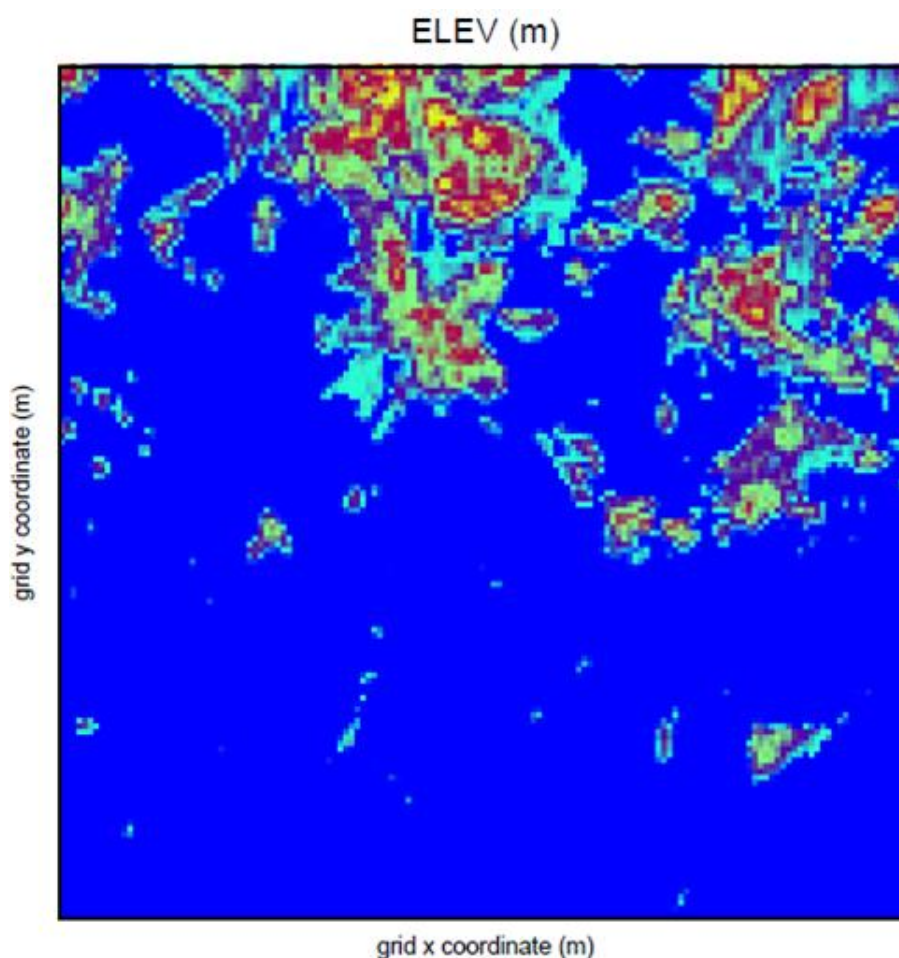


Figure 10: Terrain elevation map (heights in m) of the Helsinki bay with a grid resolution of 100 m, produced from SRTM3 data using the AERMAP terrain pre-processor. Blue colour is sea surface, yellow to red colours show elevations up to 50.9 m.

An example of the elevation map produced from SRTM3 data using the AERMAP terrain pre-processor is shown in Figure 10.

To use the tool **static4cc.exe**, the following changes need to be made in the Fortran source code `aermap_reader.for` in subdirectory `preproc/auxiliary/src/`: change the number of output lines for one elevation receptor point in the `AERMAP.REC` file `nrec`, and the filename of the `AERMAP.REC` file `aermap_name`. It is also possible to change the value of the surface roughness length for urban land use, `z0` that can be found approximately in line 224 of `aermap_reader.for` (the default value is 0.4 m).

Before using the tool change the output file path in the user metadata file to `'./INPUT/other'`. This is the directory for the output files. It is recommended to set the output format to ASCII in the user metadata file. The `AERMAP.REC` file obtained from the `AERMAP` run has to be copied to the subdirectory `INPUT/other`.

A separate installation of the tool **static4cc.exe** is normally not required, since it becomes installed when you issue:

```
{your-citychem}/$ ./installcc.csh util example
```

If this is not working for any reason, **static4cc.exe** can be built as follows:

```
{your-citychem}/preproc/auxiliary/$ make -f Makfile.aux
```

This creates **static4cc.exe**. With a symbolic link it becomes executable from subdirectory `preproc/`:

```
{your-citychem}/preproc/$ ln -s ./auxiliary/bin/static4cc.exe
```

Now **static4cc.exe** can be run. The static input files are written to subdirectory `INPUT/other` as specified in the user metadata file:

```
{your-citychem}/preproc/$ ./static4cc.exe
```

The tool **static4cc.exe** also produces the land use file `landuse_episode.asc`, needed by `MCWIND`, and the surface roughness file, `surfrough_episode.asc`. Running the tool **static4cc.exe** allocates a homogenous 2-D field with a constant value of `lu` and `z0` representing either urban land use or water surfaces. Both files are written to subdirectory `INPUT/other`.

`MCWIND` produces the dynamic fractional cloud cover file `clou_episode.asc` based on the meteorological observation of cloud fraction one met station. Running `MCWIND` is described in section 6.1. `MCWIND` requires the files created with tool **static4cc.exe** as input. The file `clou_episode.asc` containing the 2-D field of fractional cloud cover for every simulation hour is written to subdirectory `INPUT/other/`.

4 Model Configuration

All required information about physical and chemical options, input and output for a EPISODE-CityChem simulation is handed to the program in one run script, located in subdirectory SIMU/. This chapter gives a detailed explanation of each line in the run script, and a step by step explanation how to configure the simulation run. Comments in this run script are indicated by '*' in first column. Leaving an empty space in the first column causes the file in the same line not to be used by the model.

4.1 Main data

The run script starts with a “MAIN DATA” section.

```
* MAIN DATA
*
* Unique ID for Simulation, Blank indicate no ID given.
* Must be 38 characters long including left and right parenthesis.
{SIMUL-HamburgCity--yougiveaname-0001}
```

In this line the simulation id has to be provided. It is the same simulation id that is used in emission files and other input files that belong to the specific simulation. The simulation id must be 38 characters long including the left and right parenthesis. The simulation id is mandatory.

```
* Testbench log file
CITYCHEM_Yourmachine_log.txt
```

This is the filename of the log file for the simulation. The log file will be created in subdirectory SIMU/. Debugging information will be written to it during the simulation.

```
* Run log file (written for airquis)
* this log-file is not created by CityChem
EPISODE_internal_log_201307.txt
*
* LHS example of new INFO-file
out_xxxnn/EPISODE_another_dummyfile.txt
```

The next two input lines are not used by the program, but have to be filled. The first is an internal log-file, which is created when Episode is integrated in an AirQUIS simulation. There is no need to create the two files.

4.2 Site data

The next section of the run script is the “SITE DATA” section.

```
* SITE DATA
*
* pisiteexternaldata = 0 to 3. If 0 then read from external file inside of EPISODE.
*           If 1 then read from Integrated AirQUIS/MCWIND.
*           If 2 then linked with UM Met-input.
*           If 3 then linked with TAPM Met-input.
* Only options 1, 2 and 3 can be used with CityChem
3
```

In this line, either ‘1’, ‘2’ or ‘3’ can be entered. If meteorological input from TAPM should be used, then ‘3’ has to be entered. If TAPM is not available, then meteorological input generated by the MCWIND model can be used, and ‘1’ has to be entered. Option ‘2’ is for use of other meteorological input fields, e.g. from UM (Unified Model), MM5 or WRF. Option ‘2’ has not been tested yet, and the possibility to read meteorology from WRF is under development.

```
* Name of site
"Hamburg"
```

Enter the name of the city as text string.

```
* Site latitude and longitude (deg) of the meteo observation station
53.60833,9.925000
```

Enter latitude and longitude coordinates (in decimal degrees) of the observation site for meteorology.

```
* Number of observation sites (listed in receptor station file)
20
```

Enter the total number of monitoring stations that are included in the receptor file receptor_stations_raster.txt, located in subdirectory INPUT/other/.

```
* Grid origo in (truncated) UTM coordinates (m) for the MAIN DOMAIN GRID
551750,5918656
```

Enter the UTM coordinates (in m) of the origin (SW-corner) of the model domain. The UTM x-coordinate has to be truncated (i.e. without the two leading digits that represent the UTM zone).

```
* UTM zone of the site (e.g. 32N)
"32N"
```

Enter the UTM zone where the city is located as text string.

```
* EPSG number, spatial reference epsg projection (http://spatialreference.org/ref/epsg/)
"32632"
```

Enter the UTM projection zone as EPSG number.

```
* Number of Main and Boundary grid cells in the MAIN DOMAIN GRID: NX,NY,NZ,NBCX,NBCY,NBCZ
* NBCX,NBCY,NBCZ are set to 1
30,30,24,1,1,1
```

Enter the maximum number of cells in x-coordinate direction and y-coordinate direction, the maximum number of vertical model layers and the number of

additional grid cells for boundary concentration input in x-,y-,and z-direction. Values are separated by comma.

*** Angle between grid x-axis and UTM x-axis (deg) in the MAIN DOMAIN GRID**
0.0

Enter the angle (in degrees) between the x-coordinate axis of the study grid and the UTM x-coordinate axis. CityChem simulations are expected to be in UTM-projection with WGS84 as reference geoid. It is recommend to use the value 0.0 here.

*** Length of MAIN DOMAIN GRID cells DX,DY,DZ(1:NZ) (m) in the MAIN DOMAIN GRID**
1000.0,1000.0,17.50,20.00,25.00,25.00,37.50,50.00,50.00,50.00,50.00,50.00,75.00,125.00,200.00,250.00,250.00,250.00,250.00,250.00,375.00,500.00,500.00

Enter grid cell width in x-direction (in m), grid cell width in y-direction (in m), and the layer heights (in m) of the vertical model layers (with increasing height) as comma-separated values in one line.

*** Main grid topography file/value**
../INPUT/other/topo.asc

In this line the path of the location of the topography file is entered. See section 3.7.2 how to produce this file.

*** Surface roughness file/value (constant value)**
../INPUT/other/surfrough_episode.asc

In this line the path of the location of the surface roughness file is entered. See section 3.7.2 how to produce this file.

*** Surface Albedo file/value (1 = Total reflection; 0 = Black body) Default 0.3?**
inp_XXXXN/albedo.fld

In this line the path of the albedo file can be entered. Currently, the albedo file is not used by CityChem. Sufficient to enter a dummy path.

*** Receptor points file/value**
*** receptor stations and receptor raster grid 100x100 m2 (in total 20+90000 points)**
*** format: 1) list of observation sites, 2) raster points of 100 m grid**
../INPUT/other/receptor_stations_raster.txt

In this line the path of the location of the receptor point file is entered. See section 3.7.1 how to produce this file.

*** Exporting both receptor and building conc. (1) or just receptor conc. (0) to the receptor-files**
*** If 1 the receptor-concentrations are written vertically.**
*** option is not used**
2

The option to export receptor point and building points (i.e. receptor point at the location of a building) is not supported by the model. Enter '2' in this line.

*** Gridded values of total-population (the building population is subtracted from this)**
*** dummy file is not used**
inp_XXXXN/population.txt

In this line the location of the population data files is entered. Currently, the population data file is not used by the model. Sufficient to enter a dummy path.

4.3 Time data

The next section of the run script is the “TIME DATA” section.

```
* TIME DATA
*
* LTOGMT is the number of hours between LOCAL TIME and GMT.
* Examples: LTOGMT = -1 for Germany/Norway in wintertime (CET).
*           LTOGMT = -2 for Germany/Norway in summertime (CEST).
*           LTOGMT = -4 for ABU DHABI.
-2
```

In the line the time difference in hours between local time (LT) and Coordinated Universal Time (UTC) is entered. For Germany, the value is ‘-1’ in the wintertime period (i.e. $LT-1 = UTC$) and ‘-2’ in the summertime period (i.e. $LT-2 = UTC$).

```
* Begin year,month,day,hour(0-23)
2013,07,01,00
```

Enter start date of the simulation in the format *yyyy,mm,dd,hh*.

```
* End year,month,day,hour(0-23)
2013,08,01,00
```

Enter end date of the simulation in the format *yyyy,mm,dd,hh*.

```
* Timestep factor
1.0 NB! Should not be changed! 0.5 will apply half allowed DT-value.
```

In this line the model’s internal time step can be changed. It is strongly recommended not to change it. The value should be ‘1.0’.

```
* Simulation time between results output
* do not change
3600.0
```

Enter the time interval for the model output (in seconds). The value should be ‘3600.0’ (hourly output). Other time intervals have not been tested, therefore it is not recommended to change the output time interval.

4.4 Meteorological data

The next section of the run script is the “METEOROLOGICAL DATA” section.

```
* METEOROLOGICAL DATA
*
* pimeteexternaldata = 0 to 3. If 0 then read from external file inside of EPISODE.
*           If 1 then read from MCWIND wind field.
*           If 2 then read from UM Met-input.
*           If 3 then read from TAPM Met-input.
3
```

In this line the option for reading meteorological input data can be selected. The same value as for ‘pimeteexternaldata’ in the “SITE DATA” section has to be used here. Valid values are ‘1’, ‘2’ or ‘3’. If meteorological input from TAPM is to be used, then ‘3’ must be entered.

If TAPM is used, then several meteorological input files have to be provided which are not needed for the option ‘1’.

If TAPM is not available, then meteorological input generated by the MCWIND model can be used, and ‘1’ has to be entered. Option ‘2’ is for use of other meteorological input fields, e.g. from UM (Unified Model), MM5 or WRF. Option ‘2’ has not been tested yet, and the possibility to read meteorology from WRF is under development.

```
* ITURB (= 1 If Delta-T, = 2 Global-radiation data) Applied in case of Integrated AirQUIS/MCWIND
simulations. (Default = 1)
1
```

In this line the option for the calculation of turbulence parameters by the internal meteorological pre-processor has to be given. The default option is ‘1’, which selects the profile method. The alternative option is ‘2’, which selects the radiation balance method. The second option has however not been tested until now and it is recommended to use the profile method.

```
* IVERTDIFF vertical eddy diffusivity (1 = NILU Standard (NILU_METHOD 4), 2 = New Urban
(NILU_METHOD 6))
1
```

In this line the option for the calculation of the vertical Eddy diffusivity by the internal meteorological pre-processor has to be given. The default option is ‘1’, which selects the standard method used by NILU. The alternative option is ‘2’, which selects the new urban method taking into account a background diffusivity due to urban heat island effect and a new parameterization for stable conditions.

```
* ZT_LOWER ZT_UPPER (NB: ZT_LOWER can not be equal to 0) Default values should be given.
8.0 25.0
```

In this line the values of the lower and upper measurement height (in m) for the profile method should be given. If MCWIND was used to produce meteorological data, the same heights as in MCWIND should be applied.

*** MCWIND, TAPM or UM Temperature (degC) and vertical temperature gradient (homogeneous and constant data) (or 2-values)**
 ../INPUT/tapm/T_and_dtdz.fld

Enter the path of the location of the file containing air temperature and vertical temperature gradient. The file can be produced by TAPM or MCWIND and is mandatory. If the option for reading meteorological input is '3', then the 3-D air temperature field will be used in the simulation instead of T_and_dtdz.

*** iant_comp = 2 if u and v, = 3 if u, v and w**
 3

Option 'iant_comp' is not used in the model. A value '2' or '3' should be entered.

*** MCWIND, TAPM or UM Wind u- and v- (and possibly w-) components in NZ layers (or 2*NZ-values)**
 ../INPUT/tapm/wind.fld

Enter the path of the location of the file containing the u- and v-components of the wind field. The file can be produced by TAPM or MCWIND and is mandatory.

*** TAPM turbulence sigma-vw file**
*** missing data is tolerated**
 inp_xxxxnn/sigmavw.fld

Optionally, in this line the turbulence sigma-vw file path is entered.

*** TAPM aerodynamic resistance file**
*** missing data is tolerated**
 inp_xxxxnn/aeroresist.fld

Optionally, in this line the aerodynamic resistance file path is entered.

*** TAPM or UM Mixing height (m) (or 1 value)**
*** missing data is tolerated**
 ../INPUT/tapm/hmix.fld

Enter the path of the location of the file containing mixing height. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** TAPM 2D Surface (Scalar) Friction velocity field (m/s) (or 1 value)**
*** missing data is tolerated**
 ../INPUT/tapm/ustar.fld

Enter the path of the location of the file containing friction velocity. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** TAPM or UM 2D Surface Sensible Heat Flux field (W/m2) (or 1 value)**
*** missing data is tolerated**
 ../INPUT/tapm/sens_heatfl.fld

Enter the path of the location of the file containing sensible heat flux. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** TAPM or UM 2D Surface Latent (Evaporative) Heat Flux field (W/m²) (or 1 value)**
*** missing data is tolerated**
 ../INPUT/tapm/evap_heatfl.fld

Enter the path of the location of the file containing latent heat flux. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** TAPM or UM Land use data [Variable: LANU(nx,ny)] (or 1 value)**
*** missing data is tolerated**
 ../INPUT/other/landuse_episode.asc

Enter the path of the location of the file containing land use data. The file is produced according to the description in section 3.7.2. The land use file must be provided if the street canyon option is activated.

*** TAPM or UM 3D-temperature field (K) (or NZ-values)**
*** missing data is tolerated**
 ../INPUT/tapm/temp.fld

Enter the path of the location of the file containing the 3-D air temperature field. The file can be produced by TAPM and is optional. However, if the option for reading meteorological input is '3', then the file must be provided and in that case the 3-D air temperature field will be used in the simulation instead of T_and_dtdz.

*** TAPM 3D Potential temperature field (K) (or NZ-values)**
*** missing data is tolerated**
 ../INPUT/tapm/pot_T.fld

Enter the path of the location of the file containing the 3-D potential temperature field. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** TAPM 3D (layer mid)? "geopot height"-equivalent (m) (or NZ-values)**
*** missing data is tolerated**
 ../INPUT/tapm/3D_EPISODE_z_abg.fld

Enter the path of the location of the file containing 3-D mid-layer geopotential height equivalents. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** UM 2D Surface Moisture Flux field (kg m⁻² s⁻¹) (or 1 value)**
*** missing data is tolerated**
 inp_XXXXN/mflx_episode.fld

Enter the path of the location of the file containing the surface moisture flux. The file is not produced by TAPM and is optional.

*** UM 2D-Vector surface momentum stress "taus_x" and "taus_y" (N/m²) (or 2-values)**
*** missing data is tolerated**
 inp_XXXXN/sfws_episode.fld

Enter the path of the location of the file containing surface momentum stress. The file is not produced by TAPM and is optional.

*** TAPM 2D Surface (Scalar) Potential temperature scale field (m/s) (or 1 value)**

*** missing data is tolerated**

../INPUT/tapm/ptstar.fld

Enter the path of the location of the file containing potential temperature scale. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** TAPM 2D Surface (Scalar) Potential Virtual temperature scale field (m/s) (or 1 value)**

*** missing data is tolerated**

../INPUT/tapm/pvstar.fld

Enter the path of the location of the file containing potential virtual temperature scale. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** TAPM 2D Surface (Scalar) convective velocity scale field (m/s) (or 1 value)**

*** missing data is tolerated**

../INPUT/tapm/wstar.fld

Enter the path of the location of the file containing convective velocity scale. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** MCWIND or TAPM 2D Surface (Scalar) Total solar radiation field {Global radiation??} (W/m2) (or 1 value)**

../INPUT/tapm/tot_solar_rad.fld

Enter the path of the location of the file containing total solar radiation. The file can be produced by TAPM and by MCWIND and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** UM 2D Ground temperature field (Deg. C) (or 1 value)**

*** missing data is tolerated**

../INPUT/tapm/T_surf.fld

Enter the path of the location of the file containing ground temperature. The file can be produced by TAPM and is optional. If the option for reading meteorological input is '3', then the file must be provided.

*** MCWIND, TAPM or UM Precipitation (rainfall) (mm/h) (or 1 value)**

../INPUT/tapm/prec.fld

Enter the path of the location of the file containing rainfall. The file can be produced by TAPM and by MCWIND and is mandatory.

*** MCWIND, TAPM or UM Relative humidity (0-1) (or 1 value)**

../INPUT/tapm/RH_screen.fld

Enter the path of the location of the file containing relative humidity. The file can be produced by TAPM and by MCWIND and is mandatory.

* MCWIND, TAPM or UM Cloud cover (0=Clear sky, 1=Overcast) (or 1 value)
 ../INPUT/other/clou_episode.asc

Enter the path of the location of the file containing cloud cover. The file can be produced by TAPM and by MCWIND and is mandatory.

4.5 Concentration data

The next section of the run script is the “CONCENTRATION DATA” section.

* CONCENTRATION DATA

*

* Number of compounds

20

Enter the total number of CityChem compounds (nc), i.e. the chemical species for which output is generated, and for which background concentrations and emissions can be provided.

* Number of above compounds which is not included in the chemical scheme (n_nochem)

* NSPEC in grid box chemistry is NC - n_nochem

2

Enter the number of the CityChem compounds that are not included in the EMEP-45 chemistry scheme (n_nochem), i.e. are not reacting chemically.

* Number of additional compounds that are transported (n_advect)

* should be the same as n_nochem, so that unreactive species are transported

2

Enter the number of non-reactive compounds that should be transported (n_advect). It is recommended to use the same number as for the non-reactive species.

* Compound name, unit, indicators for output of concentration, dry and wet deposition,

* molecular weight, dry deposition velocity, wet deposition scavenging ratio and

* compound half-value time (half-life) (-9900 means no radioactive decay).

* WdepSR

* wet scavenging ratio $WdepSR = WdepCoeff * (P(mm/h)/(1 mm/h))$

* WdepCoeff (rain) SO2: 3.e-5, SO4: 1.e-4, HNO3: 1.e-4

* typical P: 796 mm/yr / 3192 h/yr = 0.25 mm/h (Cologne, Germany, 133 rain days)

* WdepSR SO2: 7.5e-6 (EMEP: 0.5e-6), SO4: 2.5e-5, HNO3: 2.5e-5

* But here WdepSR looks more like fraction scavenged (1.0 for SO4 and HNO3)

* WdepSR SO2: 7.5/25 = 0.3, PM2.5 = SO4 = 1.0, H2O2 = HNO3 = 1.0

*

* DDepV in cm/s

*

*PM2.5 MW and DDepV are pure guesses (not included in chemistry)

*

*Id-nr	Compound	Unit	ConI	DDepI	WDepI	MolW	DDepV	WDepSR	THalf/Kcoag
4	'O3	'ug/m3	1	1	1	48	0.2E+00	0.0E+00	0.0E+00
7	'NO	'ug/m3	1	0	0	30	0.0E+00	0.0E+00	0.0E+00
8	'NO2	'ug/m3	1	1	1	46	0.2E+00	0.0E+00	0.0E+00
6	'H2O2	'ug/m3	0	1	0	34	0.5E+00	1.0E+00	0.0E+00
10	'N2O5	'ug/m3	0	0	0	108	0.0E+00	0.0E+00	0.0E+00
11	'HNO3	'ug/m3	1	1	1	63	1.0E+00	1.0E+00	0.0E+00
12	'SO2	'ug/m3	0	1	1	64	0.0E+00	0.3E+00	0.0E+00
13	'Sulphate	'ug/m3	1	1	1	96	0.1E+00	1.0E+00	0.0E+00
20	'CO	'ug/m3	1	0	0	28	0.0E+00	0.0E+00	0.0E+00
21	'C2H6	'ug/m3	0	0	0	30	0.0E+00	0.0E+00	0.0E+00
17	'HCHO	'ug/m3	1	0	0	30	0.0E+00	0.0E+00	0.0E+00
23	'CH3CHO	'ug/m3	0	0	0	44	0.0E+00	0.0E+00	0.0E+00
32	'C2H4	'ug/m3	0	0	0	28	0.0E+00	0.0E+00	0.0E+00
25	'PAN	'ug/m3	1	0	1	121	0.0E+00	0.0E+00	0.0E+00
27	'nC4H10	'ug/m3	1	0	0	58	0.0E+00	0.0E+00	0.0E+00
29	'CH3COC2H5	'ug/m3	0	0	0	72	0.0E+00	0.0E+00	0.0E+00
34	'C3H6	'ug/m3	0	0	0	42	0.0E+00	0.0E+00	0.0E+00
36	'oXylene	'ug/m3	0	0	0	106	0.0E+00	0.0E+00	0.0E+00
69	'PM2.5	'ug/m3	1	1	1	90	0.2E+00	1.0E+00	0.0E+00
70	'PM10	'ug/m3	1	1	1	90	0.2E+00	1.0E+00	0.0E+00

A list of the CityChem compounds.

The non-reactive species have to be listed at the end of the list of CityChem compounds. Optionally, isoprene can be added in the list (before PM2.5) but then the number of compounds needs to be increased to 21 and emission/boundary condition input files have to be provided.

Column 1 is Id-nr, an arbitrary ID of the compound, not used by the program. Column 2 is a text string (10 characters long) with the specific compound name in the EMEP-45 chemistry scheme. Column 3 is the concentration unit of the output, typically $\mu\text{g}/\text{m}^3$, as text string (10 characters long). The next three columns are indicators for concentration, dry deposition and wet deposition output, not used by the program. Column 7 is molecular weight in g/mol. Column 8 is dry deposition velocity in cm/s. Column 9 is wet scavenging ratio. Column 10 is half-life time either towards radioactive decay or towards coagulation of particles, in seconds. It is possible include more non-reactive species at the end of the list. Addition of a new compound requires the following steps 1) change the number of compounds, nc, n_nochem, and n_advect above, 2) provide background concentration for that compound, 3) provide emission files for that compound or leave an empty space as placeholder for the emission file.

*** Particle number concentrations, three size categories**

```
*43 'pnc1 ' 'num/cm3 ' 1 1 1-9900 0.904 0.4 6.31E-9
*44 'pnc2 ' 'num/cm3 ' 1 1 1-9900 0.202 0.4 5.58E-9
*45 'pnc3 ' 'num/cm3 ' 1 1 1-9900 0.032 0.4 8.82E-10
```

Simulation of particle number concentration in three size categories following the concept described by *Karl et al.* (2016) is possible by including the three lines (by deleting "*" in the first column). Addition of a new compound requires the following steps 1) change the number of compounds, nc, n_nochem, and n_advect above, 2) provide background concentration for that compound, 3) provide emission files for that compound or leave an empty space as placeholder for the emission file.

*** Instantaneous (0) or NTS-Averaged (1) output of Main grid concentrations**

0

Enter 0 to compute instantaneous model grid concentrations (i.e. concentration at the end of one simulation hour) or enter 1 to compute NTS-averaged model grid concentrations (i.e. concentration as average of intermediate concentrations during one simulation hour).

*** NETCDF FILE**

*** Main grid hourly concentrations file**

*** Total concentration 3D-field C() instantaneous**

../OUTPUT/concmhour.nc

Location of the netCDF file containing instantaneous hourly 3-D model grid concentration output for all CityChem compounds.

*** Receptors concentrations file**

res_XXXXN/concr_no2.asc

Location of the ASCII file containing hourly NO₂ concentrations at all receptor points. Output file is optional.

*** Line source receptors concentrations file**

res_XXXXN/concl_no2.asc

Location of the ASCII file containing hourly NO₂ concentrations at the receptor points associated with line sources. The line source receptor points are generated internally. Concentrations are only written for lines sources with the flag qlriv set to 1 (see section 3.5.2). Output file is optional.

4.6 Deposition data

The next section of the run script is the “DEPOSITION DATA” section.

```
* DEPOSITION DATA
*
* Main grid dry deposition file
* res_XXXXXX/ddepn_o3.prn
*
* Main grid wet deposition file
* res_XXXXXX/wdepn_o3.prn
```

The output of dry deposition and wet deposition fields is deactivated in the current version.

```
* Receptor point dry deposition file
res_XXXXXX/ddepr_no2.prn
```

Location of the ASCII file containing hourly dry deposition flux of NO₂ at all receptor points. Output file is optional.

```
* Receptor point wet deposition file
res_XXXXXX/wdepr_no2.prn
```

Location of the ASCII file containing hourly wet deposition flux of NO₂ at all receptor points. Output file is optional.

```
* Line sources dry deposition file
res_XXXXXX/ddepl_no2.prn
```

Location of the ASCII file containing hourly dry deposition flux of NO₂ at the receptor points associated with line sources. The line source receptor points are generated internally. Concentrations are only written for lines sources with the flag qlriv set to 1 (see section 3.5.2). Output file is optional.

```
* Line sources wet deposition file
res_XXXXXX/wdepl_no2.prn
```

Location of the ASCII file containing hourly wet deposition flux of NO₂ at the receptor points associated with line sources. The line source receptor points are generated internally. Concentrations are only written for lines sources with the flag qlriv set to 1 (see section 3.5.2). Output file is optional.

4.7 Statistical data

The next section of the run script is the “STAT MODEL DATA AND EXPOSURE USER INPUT” section.

```
* STAT MODEL DATA AND EXPOSURE USER INPUT
*
* "pxlimitvh": Hourly limit value in (ug/m3)
* 50.0
*
* "pinallowh": Number of hours allowed above the hourly limit value
* 3
```

Exposure calculation is not implemented in the current version of CityChem. The corresponding lines in the run script have to be commented.

```
* "nhighh": Number of highest hourly concentration values stored on STAT_C... files
0
```

Enter the number of the highest hourly concentration values which shall be stored in the corresponding output file. Not tested yet.

```
* "nhighd": Number of highest daily concentration values
0
```

Enter the number of the highest daily concentration values which shall be stored in the corresponding output file. Not tested yet.

```
* Main grid nhighh-concentrations file
* res_XXXXXX/statcmhighh_no2.txt
*
* Receptors nhighh-concentrations file
* res_XXXXXX/statcrhighh_no2.txt
*
* Line source receptors nhighh-concentrations file
* res_XXXXXX/statclhighh_no2.txt
```

The output of highest hourly concentration values (for exposure calculation) is not activated. The corresponding lines in the run script have to be commented.

```
* Calculate hourly and daily mean indicator (1=Yes,0=No)
1
```

Enter ‘1’ to activate the output of hourly concentrations for the model grid, for the receptor grid and for the stations as well as daily average concentration at the line source receptor points. Enter ‘0’ to switch off writing this output.

```
* Calculate overall mean indicator (1=Yes,0=No)
1
```

Enter ‘1’ to activate the output of concentrations, averaged over the simulation period (typically one month to obtain monthly average concentrations) for the model grid, for the receptor grid and for the stations. Enter ‘0’ to switch off writing this output.

* Main grid nhighd-concentrations file (If "nhighd"=0, then blank in first column)
 * res_XXXXXX/statcmhighd_no2.txt
 *
 * Receptors nhighd-concentrations file
 * res_XXXXXX/statcrhighd_no2.txt
 *
 * Line source receptors nhighd-concentrations file
 * res_XXXXXX/statclhighd_no2.txt

The output of highest daily concentration values (for exposure calculation) is not activated. The corresponding lines in the run script have to be commented.

* NETCDF FILES
 * hourly and daily average netCDF files
 *
 * Main grid hourly average concentrations file
 ../OUTPUT/statmainhour.nc

Location of the netCDF file containing hourly 2-D model grid ground concentration output for all CityChem compounds.

* Receptors hourly average concentrations file
 ../OUTPUT/statrecphour.nc
 ../OUTPUT/statmonihour.nc

First line: Location of the netCDF file containing hourly 2-D concentration output from the regular receptor grid for all CityChem compounds. Second line: Location of the netCDF file containing hourly concentration output at monitoring stations (included in the receptor file receptor_stations_raster.txt, located in subdirectory INPUT/other/) for all CityChem compounds.

* Line source receptors daily average concentrations file
 ../OUTPUT/statclaved_no2.txt

Location of the ASCII file containing daily average NO2 concentration output for all line source receptor points. The line source receptor points are generated internally. Concentrations are only written for lines sources with the flag qlriv set to 1 (see section 3.5.2).

* Calculate daily statistics indicator (1=Yes,0=No)
 0

Option to compute hourly and daily highest concentration values. Enter '0'.
 Option included for future statistical calculation of highest daily concentrations.

* NETCDF FILES
 * Overall average netCDF files
 * Main grid overall average concentrations file
 ../OUTPUT/statmain.nc

Location of the netCDF file containing 2-D model grid concentration averaged over the simulation period (typically one month to obtain monthly average concentrations) for all CityChem compounds.

*** Receptors overall average concentrations file**
 ../OUTPUT/statrecp.nc

Location of the netCDF file containing concentration averaged over the simulation period (typically one month to obtain monthly average concentrations) from the regular receptor grid for all CityChem compounds.

*** Stations overall average concentrations file**
 ../OUTPUT/statmoni.nc

Location of the netCDF file containing concentration averaged over the simulation period (typically one month to obtain monthly average concentrations) at monitoring stations (included in the receptor file receptor_stations_raster.txt, located in subdirectory INPUT/other/) for all CityChem compounds.

*** Line source receptors overall average concentrations file**
 * res_XXXNN/statclavea_no2.txt

Location of the simulation-averaged NO₂ concentration at line source receptor points. The line in the run script has to be commented.

*** Main grid HIGHH population exposure file**
 * res_XXXNN/statmhighh_no2.txt

*** Building HIGHH population exposure file**
 * res_XXXNN/statnrhighh_no2.txt

*** Main grid HIGHD population exposure file**
 * res_XXXNN/statmhighd_no2.txt

*** Building HIGHD population exposure file**
 * res_XXXNN/statnrhighd_no2.txt

The output of highest daily concentration values for exposure calculation is not activated. The corresponding lines in the run script have to be commented.

*** Main grid yearly average population exposure file**
 * res_XXXNN/statmavey_no2.txt

*** Building yearly average population exposure file**
 * res_XXXNN/statravey_no2.txt

The output of yearly average exposure files is not activated. The corresponding lines in the run script have to be commented.

*** INPUT Background concentrations file/value**

```

../INPUT/cmaq/cbg_o3.fld
../INPUT/cmaq/cbg_no.fld
../INPUT/cmaq/cbg_no2.fld
../INPUT/cmaq/cbg_h2o2.fld
../INPUT/cmaq/cbg_n2o5.fld
../INPUT/cmaq/cbg_hno3.fld
../INPUT/cmaq/cbg_so2.fld
../INPUT/cmaq/cbg_sulphate.fld
../INPUT/cmaq/cbg_co.fld
../INPUT/cmaq/cbg_c2h6.fld
../INPUT/cmaq/cbg_hcho.fld
../INPUT/cmaq/cbg_ch3cho.fld
../INPUT/cmaq/cbg_c2h4.fld
../INPUT/cmaq/cbg_pan.fld
../INPUT/cmaq/cbg_nc4h10.fld
../INPUT/cmaq/cbg_ch3coc2h5.fld
../INPUT/cmaq/cbg_c3h6.fld
../INPUT/cmaq/cbg_oxylen.fld
../INPUT/cmaq/cbg_pm25.fld
../INPUT/cmaq/cbg_pm10.fld

```

If constant value option was chosen in the previous line, enter one background concentration value for each CityChem compound per line. If 3-D input files option was chosen in the previous line, enter one BCON file path for each CityChem compound per line.

*** OFFSET Ozone (O3) in BCON file in ug/m³**

0.0

*** OFFSET PM (PM2.5 & PM10) in BCON file in ug/m³**

0.0

Adding a fix positive offset to the boundary concentration of O₃ (in µg/m³) and PM (in µg/m³) given in the BCON file. This option can be used to calibrate the model.

*** RESTART option for multi-month runs**

*** 0: No Restart (use the BCON (background) values)**

*** 1: Restart (use 3-D ICON files listed below)**

0

The restart option allows performing multi-month runs. If the restart option is set to '1', the model will use the file with 3-D initial concentrations as start concentrations for the main grid. If the restart option is set to '0', the model will use the background concentrations (3-D field or constant value) as start concentrations for the main grid.

*** RESTART initial conditions**

*** INPUT Old 3D concentrations files**

*** Must be provided if Restart option is 1**

*** If Restart option is 0 or files are missing, background values are used**

```
../INPUT/other/icmhour_20130630.nc
```

Enter the file with the 3-D field of initial concentrations if the restart option is set to '1'. The file contains the main grid 3-D concentrations of the last hour before the simulation starting hour. The file must have the same grid dimensions and the same chemical compounds. If the run is with no restart leave an empty line or a space in first column.

* Last simulation hour instantaneous concentration 3D-field C0
 * THIS WILL BE THE IC FILE FOR THE RESTART
 ../OUTPUT/icmhour_20130731.nc

This netCDF output file will contain the 3-D concentration field of the main grid from the last hour of the simulation. The file can be used as initial concentration file for the next simulation period (month).

* DIAGNOSTIC Output of photochemistry on main grid
 ../OUTPUT/photmhour.nc

Location of the netCDF output file containing diagnostic information on the model grid photochemistry: hourly 2-D surface fields of concentrations of some radicals (OH, HO₂, CH₃O₂, NO₃), selected reaction rate coefficients, photolysis frequencies, and meteorological variables.

4.10 Area source data

The next section of the run script is the “AERA SOURCE MODEL DATA” section.

* AREA SOURCES MODEL DATA
 *
 * Add area sources subgrid model to results
 *0 NB! Should not be changed!
 0

Option for adding area source emissions to sub-grid models. Enter ‘0’ (do not change).

* Number of area sources
 4

Enter the number of area source emission categories (e.g. residential heating; solvent use and agriculture; industrial and commercial sources; shipping). The number of emission categories is 4 and should not be changed. Note that for each emission category, emission files have to be provided.

* Area sources upper level (domestic heating) emissions files/values
 emis_XXXNN/asrc_domestic_uple_20130701_20130731_o3.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_no.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_no2.txt
 emis_XXXNN/asrc_domestic_uple_20130701_20130731_h2o2.txt
 emis_XXXNN/asrc_domestic_uple_20130701_20130731_n2o5.txt
 emis_XXXNN/asrc_domestic_uple_20130701_20130731_hno3.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_so2.txt
 emis_XXXNN/asrc_domestic_uple_20130701_20130731_sulphate.txt
 emis_XXXNN/asrc_domestic_uple_20130701_20130731_co.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_c2h6.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_hcho.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_ch3cho.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_c2h4.txt
 emis_XXXNN/asrc_domestic_uple_20130701_20131231_pan.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_nc4h10.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_ch3coc2h5.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_c3h6.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_oxylen.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_pm25.txt
 ../INPUT/emis/asrc_domestic_uple_20130701_20130731_pm10.txt

Enter location of area source emission file for each CityChem compound per line. If no emission file is available for a compound or if the compound has no emissions (like O₃) then an empty space in the first column of the line has to be entered. Emission file input is optional. However, if the name of an input file is provided in the list of area emission files, it will be checked if the file actually exists. If that is not the case, the model run will stop with a corresponding error message.

The list of emission files has to be given for each emission category, in the following order: 1) domestic heating, 2) solvent use and agriculture, 3) industrial and commercial sources, and 4) shipping.

*** Minimum windspeed for subgrid scale area source model**
0.4

Enter the value of the minimum wind speed for the area source sub-grid model. Not used by the program.

*** Output total area source emissions in netCDF file**
../OUTPUT/asrcmhour.nc

Location of the netCDF output file containing hourly 3-D area emissions (in g/s) for all CityChem compounds.

4.11 Point source data

The next section of the run script is the “POINT SOURCE MODEL DATA” section.

*** POINT SOURCES MODEL DATA**
*

*** Point sources subgrid model type (0 = Inject into grid after plume height calc.; 1 = Segmented Plume; 2,3,5 = Inpuff variants)**
1

Option for treatment of point source emissions. Enter ‘0’ for injecting point source emissions directly to the model grid at the height of the final plume rise calculated by the plume-rise algorithm. Enter ‘1’ for using the segmented plume model SEGPLUME, which is the sub-grid model for point sources.

*** Add point sources subgrid model to results**
***1 NB! Should not be changed!**
1

Option for adding point source emissions to sub-grid models. Enter ‘1’ (do not change).

*** Point source files**
*

*** Filename of one Point Source Data File (include all point sources with compound emis for every hour)**
../INPUT/emis/pointsrc_jul2013.txt

Location of the point source emission file. The point source emission file has to include emission values for all CityChem compounds, for each point source. See section 3.5.1 for the format of this input file.

4.12 Line source data

The next section of the run script is the “LINE SOURCE MODEL DATA” section.

* LINE SOURCES MODEL DATA

*

* Add line sources subgrid model to results

1 NB! Should not be changed!

Option for adding line source emissions to sub-grid models. Enter ‘1’ (do not change).

* Static data (meta data of the line sources)

../INPUT/emis/linesource_metadata.txt

Location of the line source meta data. See section 3.5.2 for the format of this input file.

* Variable emission data

```
emis_xxxxn/lsvr_traffic_line_20130701_20130731_o3.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_no.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_no2.txt
emis_xxxxn/line/lsvr_traffic_line_20130701_20130731_h2o2.txt
emis_xxxxn/line/lsvr_traffic_line_20130701_20130731_n2o5.txt
emis_xxxxn/line/lsvr_traffic_line_20130701_20130731_hno3.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_so2.txt
emis_xxxxn/lsvr_traffic_line_20130701_20130731_sulphate.txt
emis_xxxxn/lsvr_traffic_line_20130701_20130731_co.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_c2h6.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_hcho.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_ch3cho.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_c2h4.txt
emis_xxxxn/lsvr_traffic_line_20130701_20130731_pan.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_nc4h10.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_ch3coc2h5.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_c3h6.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_oxylen.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_pm25.txt
../INPUT/emis/lsvr_traffic_line_20130701_20130731_pm10.txt
```

Enter location of line source emission file for each CityChem compound per line. If no emission file is available for a compound or if the compound has no emissions (like O₃) then an empty space in the first column of the line has to be entered. Emission file input is optional. However, if the name of an input file is provided in the list of line emission files, it will be checked if the file actually exists. If that is not the case, the model run will stop with a corresponding error message.

* Data on line source is in the static metadata

*

* Emission scale factor for subgrid line source model, scales all above emissions

1.0

Scaling factor for line source emissions of NO_x. The emissions of all compounds from line sources will be scaled by this factor. Value 1.0 means that the emissions are taken as in the line source emission input files.

*** Street Canyon option**
*** 0: No Canyon (default)**
*** 1: Canyon (check land use value)**
0

Option to use the street canyon module for calculation of concentrations at receptors located at streets with the sub-grid line source model. Enter '1' for using the street canyon option. Requires presence of the land use input file.

*** Absolute maximum value for the "maximum influence distance" (in meters). ORG = 500.0**
*** 500.0**

*** Minimum distance from receptor to road side:**
*** 5.0**

*** Distance from road side to line source associated receptor point:**
*** 20.0**

Default parameter values for the treatment of line sources in relation to associated receptor points are used internally in the program. The corresponding lines in the run script have to be commented.

*** Minimum windspeed for subgrid scale line source model (m/s)**
1.0

Enter the value of the minimum wind speed (in m/s) that is allowed in the line source model. This is the last entry line in the run script.

5 Model Output

The program creates several output files in the subdirectory OUTPUT/. A list of all possible output files created by EPISODE-CityChem is shown in Table B1 in **Appendix B**. Output files indicated as mandatory need to be specified in the run script, all others do not have to be specified. The optional output files are in ASCII format and provide additional information for NO₂ at all stations, regular receptor points and the line-source associated receptor points, but not for the other pollutants. The most important output are the 2-D and 3-D concentration fields (main study grid and regular receptor grid) of the CityChem compounds which is written to netCDF output files in netcdf4 format. Compression (with compression level 9) is applied to the float variables in the netcdf4 files in order to reduce the disk space for the output.

Plotting with NCL

In the CityChem distribution package some tools for post-processing are provided, mainly for the graphical plotting of the output results. These plotting routines are written in NCL (NCAR Command Language). The NCL routines are located in the subdirectory postchem/. NCL can be downloaded from <https://www.ncl.ucar.edu/Download/linux.shtml> for various Linux systems.

NCL is available as package for Ubuntu Linux. To install on Ubuntu, type the following and you are ready to use NCL immediately:

```
{your-citychem}/OUTPUT$          sudo apt-get install ncl-ncarg
```

```
{your-citychem}/OUTPUT$          sudo ln -s /usr/share/ncarg /usr/lib/ncarg
```

After that, the NCARG_ROOT environment variable needs to be set in the bash profile (file in your home with name “.bash_profile”, “.profile” or “.bashrc”). At the end of this file add the following lines:

```
export NCARG_ROOT=/usr/lib/ncarg
export PATH=$NCARG_ROOT/bin:$PATH
```

You will need to logout and log back in for the changes to take effect. If you are not installing on Ubuntu Linux, follow the instructions at <https://www.ncl.ucar.edu/Download/install.shtml> for the proper installation of NCL.

The NCL's graphics default properties can be overwritten by creating the “.hluresfile” file in your home directory ({your-home}), see the description at <http://www.ncl.ucar.edu/Document/Graphics/hlures.shtml>.

Plotting with OpenAir

The OpenAir package for R provides further options to plot the CityChem output. Some useful plotting routines are provided as a R script and located in the subdirectory `postcchem/`. The R scripts can be used for additional graphical plotting of the output results and for the statistical evaluation of results. R scripts have been prepared for creating different plots, intended for additional analysis but not to replace the NCL scripts.

The use of `openair` requires the installation of R. For installation of R on Ubuntu Linux you first need to add the CRAN repository (<https://cran.r-project.org/>) for Ubuntu packages to `/etc/apt/sources.list`. This can be done by issuing the following in a terminal window:

```
{your-home}$      CODENAME=`grep CODENAME /etc/lsb-release | cut -c 18`
{your-home}$      sudo sh -c 'echo "deb http://cran.rstudio.com/bin/linux/ubuntu
CODENAME" >> /etc/apt/sources.list'
```

Then add the CRAN GPG key, which is used to sign the R packages for security:

```
{your-home}$      sudo apt-key adv --keyserver keyserver.ubuntu.com
--recv-keys E084DAB9
```

Then update and start the installation of R:

```
{your-home}$      sudo apt-get update
{your-home}$      sudo apt-get install r-base r-dev
```

After installation of R, you are ready to install `openair`. To use the OpenAir script for post-processing of CityChem output, you need the libraries “`ncdf4`” as well “`openair`” installed. If these are not installed yet, you can open R in your terminal install the libraries and quit the R console (change to directory `../postcchem/` in your CityChem installation):

```
{your-citychem}/postcchem$      R
> install.package("openair")
> install.package("ncdf4")
> quit()
```

Manipulation of netCDF output

Manipulation of the netCDF output can be done in a simple way using the NCO tools (<http://nco.sourceforge.net/nco.html>), a collection of utilities.

For example, to get the concentration differences between two simulation runs for the same study domain and time period from the monthly averaged receptor grid output files `statrecp_run1.nc` and `statrecp_run2.nc` type the following:

```
{your-citychem}/OUTPUT$ ncbo --op_type=- statrecp_run1.nc statrecp_run2.nc diffrecp.nc
```

Now, `diffrecp.nc` will contain concentration differences for all CityChem compounds from the two runs. To further calculate the concentration difference between the two runs relative to the concentration of run no. 1, type the following:

```
{your-citychem}/OUTPUT$ ncbo --op_type=/ diffrecp.nc statrecp_run1.nc diffrelrecp.nc
```

Now, `diffrelrecp.nc` will contain the fractional concentration differences for all CityChem compounds from the two runs relative to run no. 1. To further calculate the percentage difference for NO₂ (or any other CityChem compound), type:

```
{your-citychem}/OUTPUT$ ncap2 -s "dNO2=NO2*100" diffrelrecp.nc diffp_NO2.nc
```

Now, `diffp_NO2.nc` will contain the percentage concentration difference for NO₂ from the two runs in the (only) variable `dNO2`.

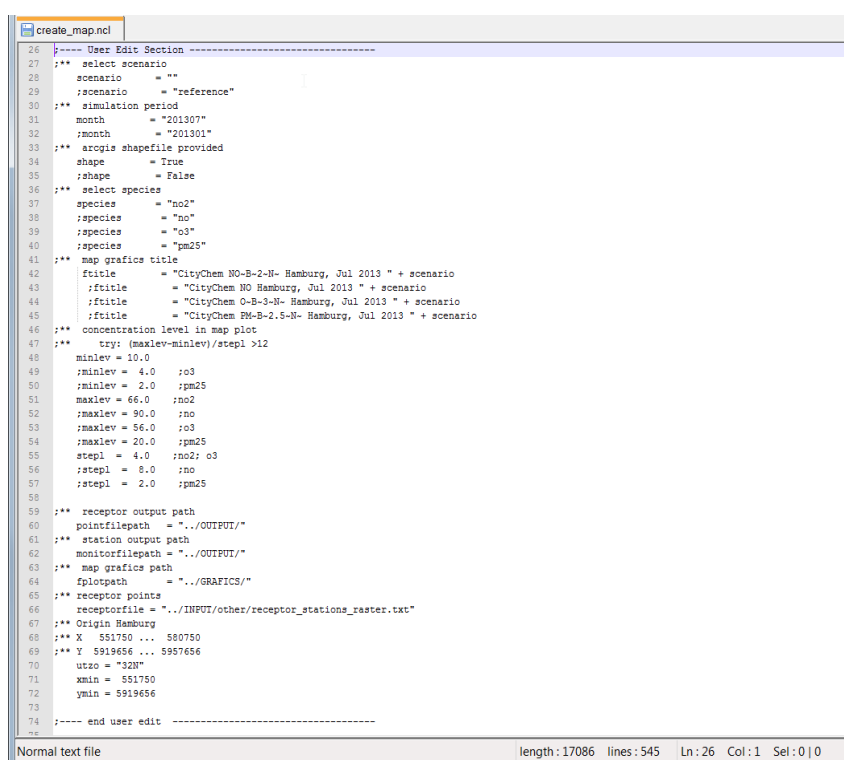
In order to concatenate the monthly averaged receptor grid output file from subsequent months for example of a full year series, the following can be typed:

```
{your-citychem}/OUTPUT$ nrcat statrecp_run1.nc statrecp_run2.nc statrecp_run3.nc monseries.nc
```

Now, `monseries.nc` will contain the time series of the monthly data. The same can be done to concatenate the hourly average receptor grid output (`statrecphour.nc`) files into a continuous hourly time series of several months.

5.1 Spatial lon-lat maps (NCL)

For plotting surface air concentration fields in the form of spatial maps, the NCL routine “create_map.ncl” is provided in subdirectory postcchem/. The routine reads the netCDF files **statrecp.nc** and **statmoni.nc** and produces image files (*.png) with spatial colour contour maps (with latitude and longitude coordinates) of the modelled receptor grid concentrations and the monitor station concentrations (as white circles). If an ArcGIS shape file of the water bodies and of the harbour area for the same geographic extent (as the model study domain) is provided, this will be used to overlay on the concentration map. In script “create_map.ncl” the ‘shape’ variable should be set to ‘True’ to activate the display of the shapefile polylines and polygons.



```

26  !----- User Edit Section -----
27  ;** select scenario
28  scenario = ""
29  ;scenario = "reference"
30  ;** simulation period
31  month = "201307"
32  ;month = "201301"
33  ;** arcgis shapefile provided
34  shape = True
35  ;shape = False
36  ;** select species
37  species = "no2"
38  ;species = "no"
39  ;species = "o3"
40  ;species = "pm25"
41  ;** map graphics title
42  ftitle = "CityChem NO-B-2-N- Hamburg, Jul 2013 " + scenario
43  ;ftitle = "CityChem NO Hamburg, Jul 2013 " + scenario
44  ;ftitle = "CityChem O-B-3-N- Hamburg, Jul 2013 " + scenario
45  ;ftitle = "CityChem PM-B-2.5-N- Hamburg, Jul 2013 " + scenario
46  ;** concentration level in map plot
47  ;** try: (maxlev-minlev)/stepl >12
48  minlev = 10.0
49  ;minlev = 4.0 ;o3
50  ;minlev = 2.0 ;pm25
51  ;maxlev = 66.0 ;no2
52  ;maxlev = 90.0 ;no
53  ;maxlev = 56.0 ;o3
54  ;maxlev = 20.0 ;pm25
55  stepl = 4.0 ;no2; o3
56  ;stepl = 8.0 ;no
57  ;stepl = 2.0 ;pm25
58
59  ;** receptor output path
60  pointfilepath = "../OUTPUT/"
61  ;** station output path
62  monitorfilepath = "../OUTPUT/"
63  ;** map graphics path
64  fplotpath = "../GRAFICS/"
65  ;** receptor points
66  receptorfile = "../INPUT/other/receptor_stations_raster.txt"
67  ;** Origin Hamburg
68  ;** X 551750 ... 580750
69  ;** Y 5919656 ... 5957656
70  utzo = "32N"
71  xmin = 551750
72  ymin = 5919656
73
74  !----- end user edit -----

```

Figure 11: Screenshot of the user edit section in routine create_map.ncl.

The user can edit a part of “create_map.ncl” to select the pollutant, the contour levels for the display, output paths, as well as the geographic origin of the study domain (Figure 11). An example map of NO₂ monthly average concentration for Hamburg (using the example data, section 7) is shown in Figure 12. Run the *.ncl script as:

```
{your-citychem}/postcchem$ ncl create_map.ncl
```

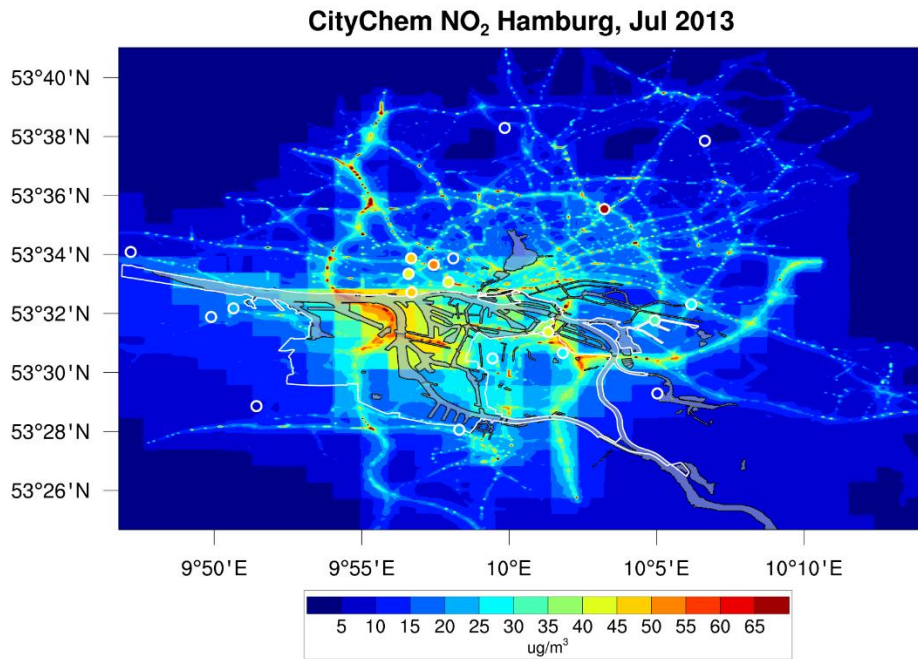


Figure 12: Spatial map of NO₂ monthly concentration average (in $\mu\text{g m}^{-3}$) for Hamburg city plotted with the routine `create_map.ncl`. White outline illustrates harbour areas and black outline illustrates water bodies. Open white circles denote stations.

5.2 Vertical profile sequence (NCL)

The NCL routine “`vertical_profile.ncl`” in subdirectory `postcchem/` can be used for plotting vertical profiles of pollutant concentrations as time series for inspecting the vertical exchange between the surface layer and higher layers in the air column above a specified grid cell. This routine reads the main grid 3-D output file **`concmhour.nc`** which contains hourly concentrations.

The user edit section of routine “`vertical_profile.ncl`” allows selecting the destination of the model output file and of the graphical plot file (Figure 13). Further, the main grid cell index (*i* and *j* value in `concmhour.nc`) and the start and end day of the period for plotting as well as the contour levels of the display can be specified. Figure 14 shows an example of the vertical profile plot of hourly NO₂ concentrations up to a height of 450 m. Run the `*.ncl` script as:

```
{your-citychem}/postcchem$ ncl vertical_profile.ncl
```



```

vertical_profile.ncl
13
14 ;--- User Edit Section -----
15
16 ;** CITYCHEM main grid hourly 3-D output
17   gridfilename = "../OUTPUT/concmhour.nc"
18
19 ;** graphics path and filename
20   fplotpath   = "../GRAFICS/vertical_no2_201307"
21
22 ;** simulation period
23   month       = "201307"
24
25 ;** main grid model cell
26   cellx      = 10
27   celly      = 16
28
29 ;** start/end day of month
30   startday   = 16
31   endday     = 20
32
33 ;** concentration range for contour plot
34   minlev     = 2.0
35   maxlev     = 28.0
36   stepl     = 2.0
37
38 ;--- end user edit -----
39
Normal text file      length: 3550  lines: 122  Ln: 1  Col: 1  Sel: 0 | 0  UNIX  UTF-8  INS

```

Figure 13: Screenshot of the user edit section in routine `vertical_profile.ncl`.

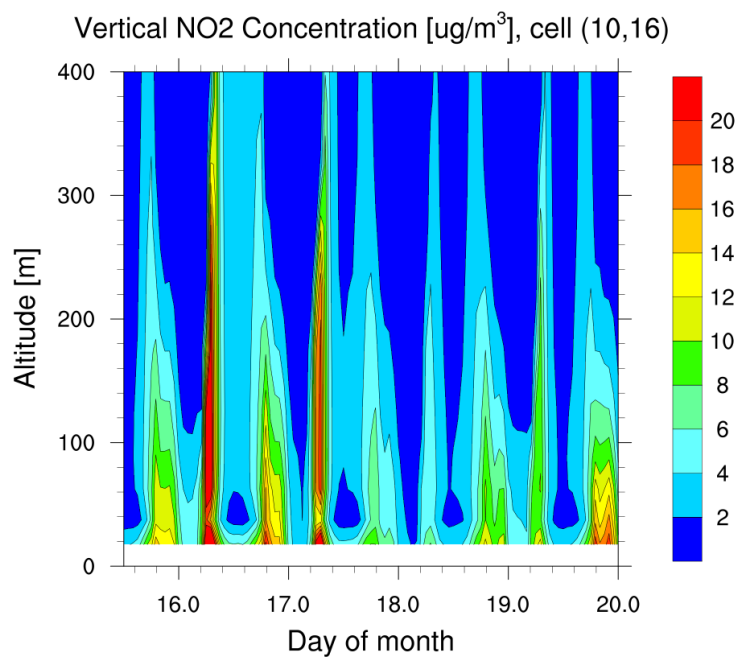


Figure 14: Hourly time slices of the vertical profile of NO₂ mixing ratio (in $\mu\text{g m}^{-3}$) for main grid cell (10,16) for 4.5 days in July 2013 (starting at noon of 15th July) as contour plot created by routine `vertical_profile.ncl`.

5.3 Concentration time series plots (NCL)

It is often useful to compare the computed concentration time series of an air pollutant to the monitored concentration time series. For this, measurements of hourly concentrations of the respective pollutants at an air quality monitoring station for the same time period are needed, if possible in CSV (comma separated values) formatted files. Example data for Hamburg is included in subdirectory `postchem/obsdat/`. Here we take the hourly averaged concentrations of NO₂ as example for plotting a time series comparison.

For plotting the time series of hourly modelled and measured air concentration of NO₂, the NCL routine “`plot_no2_series.ncl`” is provided in subdirectory `postchem/`. The routine reads the netCDF file `statmonihour.nc` and the monitoring data CSV file in `postchem/obsdat/` and produces image files (*.png) showing the time series of modelled and measured hourly averaged NO₂ concentrations as 6-hour running mean for four selected air quality monitoring stations.

Before starting the script, it is important to check the monitoring data in the CSV file (for the example data this has already been done). Apply the following steps:

1. Replace all comma by decimal point (if numerical values were given with comma instead of decimal point).
2. Remove all header lines at the beginning of the file (text information).
3. Replace any empty values ‘;’ by ‘;-99;’.

The user can edit a part of “`plot_no2_series.ncl`” to select the model output file, number of stations and simulation hours (has to match time dimension in `statmonihour.nc`) and the file destination of the output graphic (Figure 15).

Run the *.ncl script as:

```
{your-citychem}/postchem$ ncl plot_no2_series.ncl
```

```

19
20 ;---- User Edit Section -----
21
22 ; CityChem output file ("model")
23 ; File path and name
24     filename_mod = "../OUTPUT/statmonihour.nc"
25 ; Number of monitoring stations in output
26     nstat = 20
27 ; Number of hours in output
28     ntim = 745
29 ; Difference Local Time to UTC Time (UT-LT)
30     st     = -2 ; LT summer +2 hours ahead
31 ; Grafical plot path and name
32     wks1   = gsn_open_wks("png", "../GRAFICS/hamburg_series_no2_201307_pan1")
33
34 ;---- end user edit -----
35

```

Figure 15: Screenshot of the user edit section in routine `plot_no2_series.ncl`.

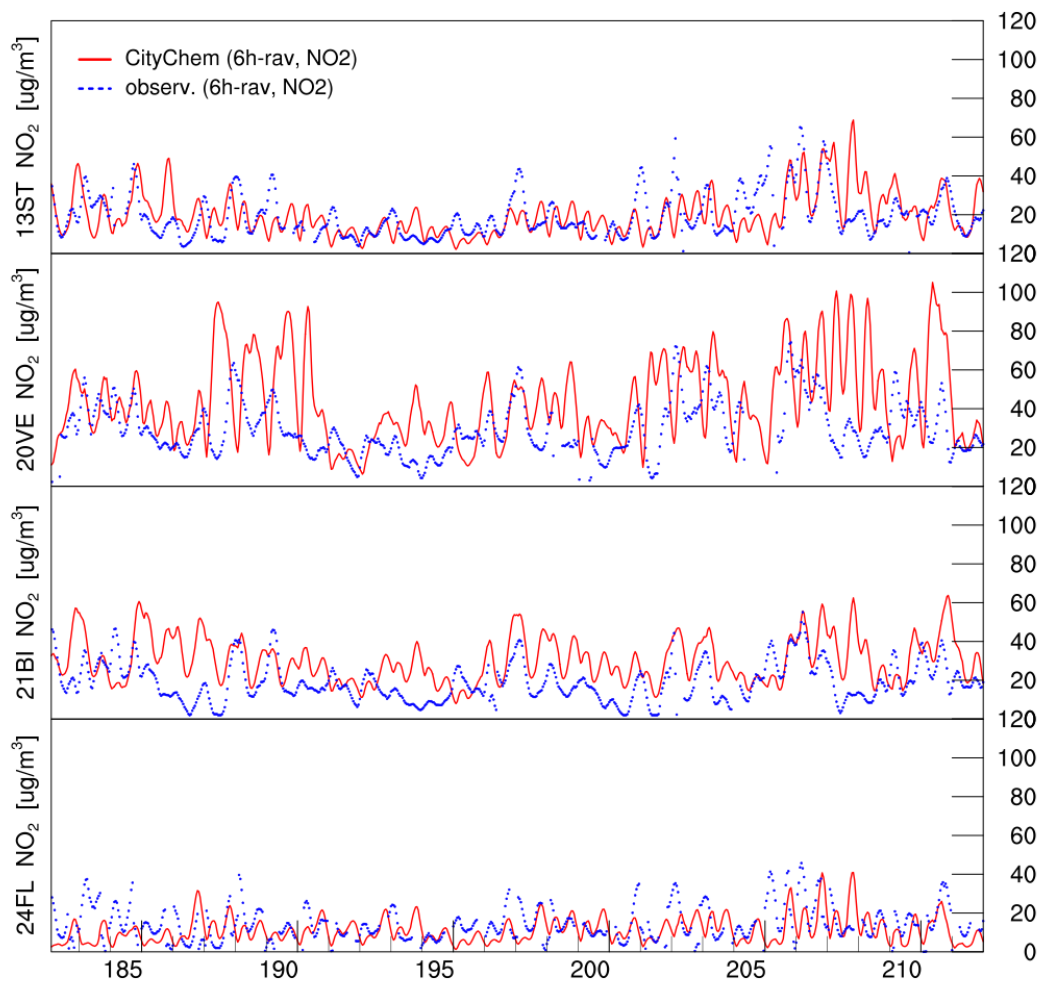


Figure 16: Time series of hourly averaged NO_2 concentrations in July 2013 (time given as day of year) plotted with the routine `plot_no2_series.ncl`. From top to bottom, the figure parts show model results (red lines) and measurements (blue dots) for the monitoring stations Sternschanze (13ST), Veddel (20VE), Billbrook (21BI) and Airport (24FL).

An example plot of the time series of NO_2 hourly average concentration for Hamburg in July 2013 is shown in Figure 16.

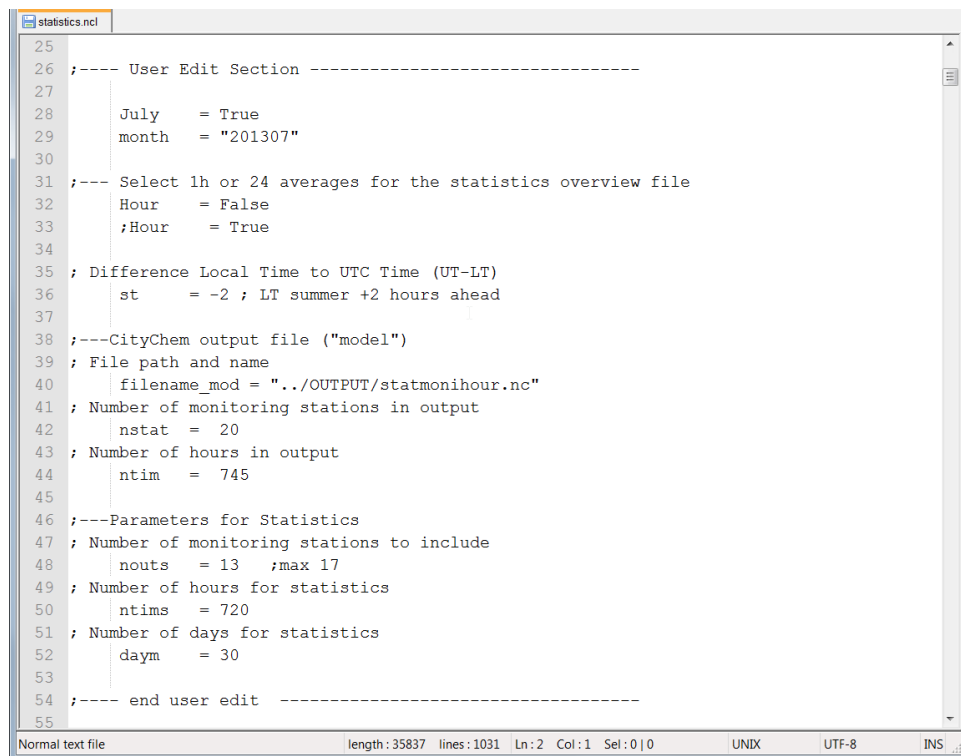
5.4 Statistical regression plots (NCL)

Having compared the concentration time series, an overview of the statistics of the comparison can be obtained with routine “statistics.ncl” provided in subdirectory postchem/. The routine produces a regression plot of the monthly and 24-hour concentration averages for all included monitoring stations and a statistical overview file for the 24-hour or hourly averages from linear regression. The routine reads the netCDF file **statmonihour.nc** and the monitoring data CSV file in postchem/obsdat/.

Before starting the script, it is important to check the monitoring data in the CSV file (for the example data this has already been done). Apply the following steps:

1. Replace all comma by decimal point (if numerical values were given with comma instead of decimal point).
2. Remove all header lines at the beginning of the file (text information).
3. Replace any empty values ‘;;’ by ‘;-99;’.

The user can edit a part of “statistics.ncl” to select the model output file, number of stations and simulation hours (has to match time dimension in statmonihour.nc), number of stations and simulation hours for inclusion in the regression, and the file destination of the output graphic (Figure 17). It can be chosen between output of using 24-hour averages (‘Hour = False’) or hourly averages (‘Hour = True’) for the statistical overview that is written to an ASCII file. Missing observation values are ignored in the statistical analysis.



```

25 ;
26 ;--- User Edit Section -----
27
28 July = True
29 month = "201307"
30
31 ;--- Select lh or 24 averages for the statistics overview file
32 Hour = False
33 ;Hour = True
34
35 ; Difference Local Time to UTC Time (UT-LT)
36 st = -2 ; LT summer +2 hours ahead
37
38 ;---CityChem output file ("model")
39 ; File path and name
40 filename_mod = "../OUTPUT/statmonihour.nc"
41 ; Number of monitoring stations in output
42 nstat = 20
43 ; Number of hours in output
44 ntim = 745
45
46 ;---Parameters for Statistics
47 ; Number of monitoring stations to include
48 nouts = 13 ;max 17
49 ; Number of hours for statistics
50 ntimes = 720
51 ; Number of days for statistics
52 daym = 30
53
54 ;--- end user edit -----
55

```

Figure 17: Screenshot of the user edit section in routine statistics.ncl.

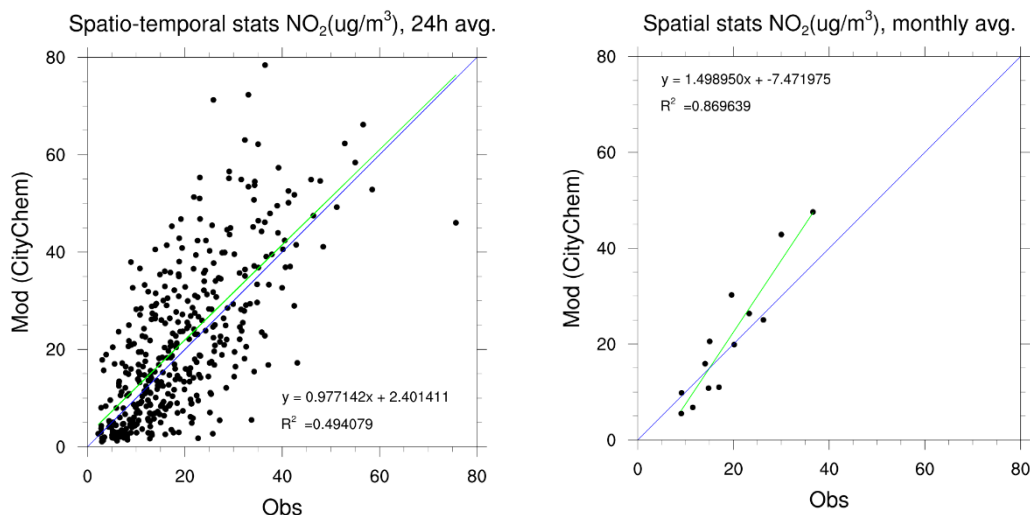


Figure 18: Regression plots of modelled NO₂ concentration versus monitored NO₂ concentration when including 13 monitoring stations in Hamburg using (a) 24-hour averages and (b) monthly averages. Blue line denotes the 1:1 line and the green line is the regression line. Regression parameters are included in the plots.

Run the *.ncl script as:

```
{your-citychem}/postcchem$ ncl statistics.ncl
```

The graphical output of “statistics.ncl” is a scatter plot of the regression for the 24-hour average (Figure 18a) and for the monthly average (Figure 18b) of the NO₂ concentration for Hamburg in July 2013.

The text file output of “statistics.ncl” is a summary of some statistical indicators for all stations (text file: “hh_no2_stat_201307.txt”) and the hourly mean concentration data in columns with first observed and second modelled value for all stations (text file: “hh_no2_timeseries_201307”).

5.5 OpenAir script for additional analysis

The script “openAir_evaluation.R” provided in subdirectory postcchem reads the CityChem output netCDF file **statmonihour.nc** as well as observations for NO₂ concentrations in HH_NO2_monit_2013.csv (in postcchem/obsdat/) to compare modelled and observed data. The observation file can also be exchanged by other observations for other domains, but it is prepared to work with the delivered example for July 2013 in Hamburg.

First, make sure openair library is installed:

```
{your-citychem}/postcchem$ R
> require("openair")
> quit()
```

If openair is not yet installed, please refer to the beginning of section 5.

The user can edit a part of “openAir_evaluation.R” to select the directory that contains the **statmonihour.nc** output file. Here it is useful to follow to the general directory structure of CityChem. In the delivered script, the path is set to “./OUTPUT/”, which is the designated path for the Hamburg July 2013 example output.

Now, the script can be started and the plots will be written to “./GRAFICS”.

Run the script as:

```
{your-citychem}/postcchem$ Rscript openAir_evaluation.R
```

The scripts generates a CSV file (“openAir_modStats.csv”) with statistics of the model-observation comparison and the following six graphical plots:

1. openAir_timePlot_hourly.png
2. openAir_timePlot_daily.png
3. openAir_timeVariation_13ST.png
4. openAir_TaylorDiagram.png
5. openAir_scatterplot_daily.png
6. openAir_scatterplot_monthly.png

The first two plots of “openAir_evaluation.R” are created by the timePlot command. It gives an overview of the data over the modelled timeline. There are options to group the data by the stations. An example plot is given in Figure 19.

The third plot is created by timeVariation command. It gives a panel with four subplots to depict different aspects of the temporal variation of modelled versus measured data at one specific station. One plots the given data for each hour of a day, one plots the average hourly distribution for one day, one plots the variation over one year and one plots the average daily variation over one week. An example plot is given in Figure 20.

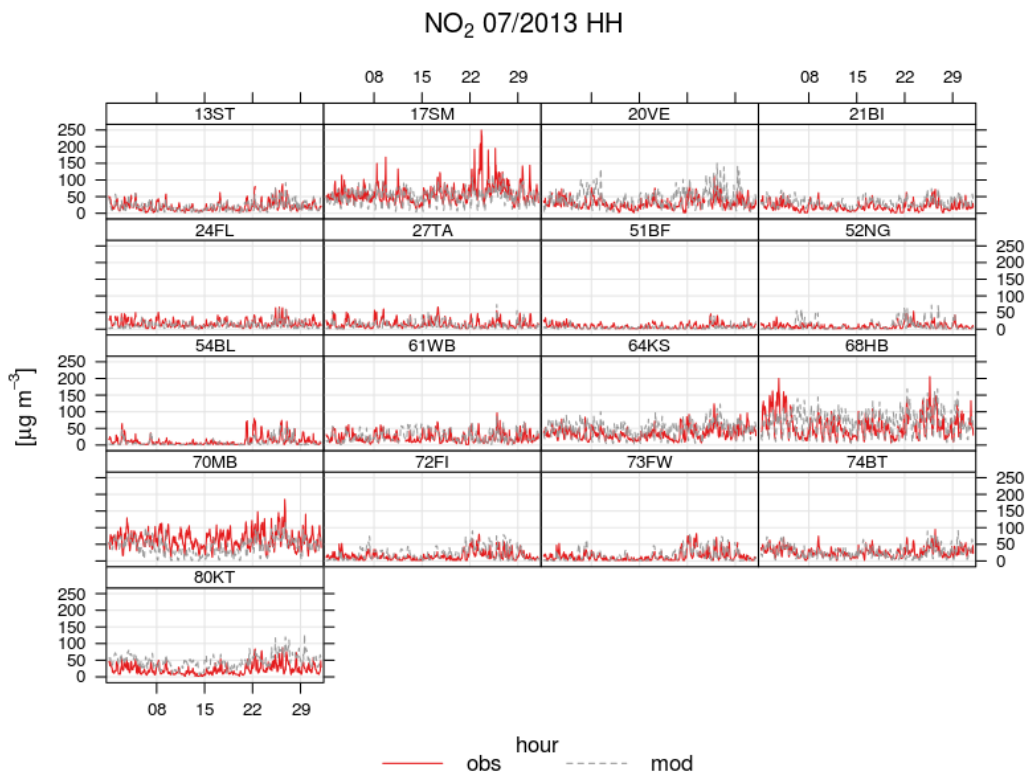


Figure 19: OpenAir time series plots generated by timePlot for observed and modelled data grouped by station.

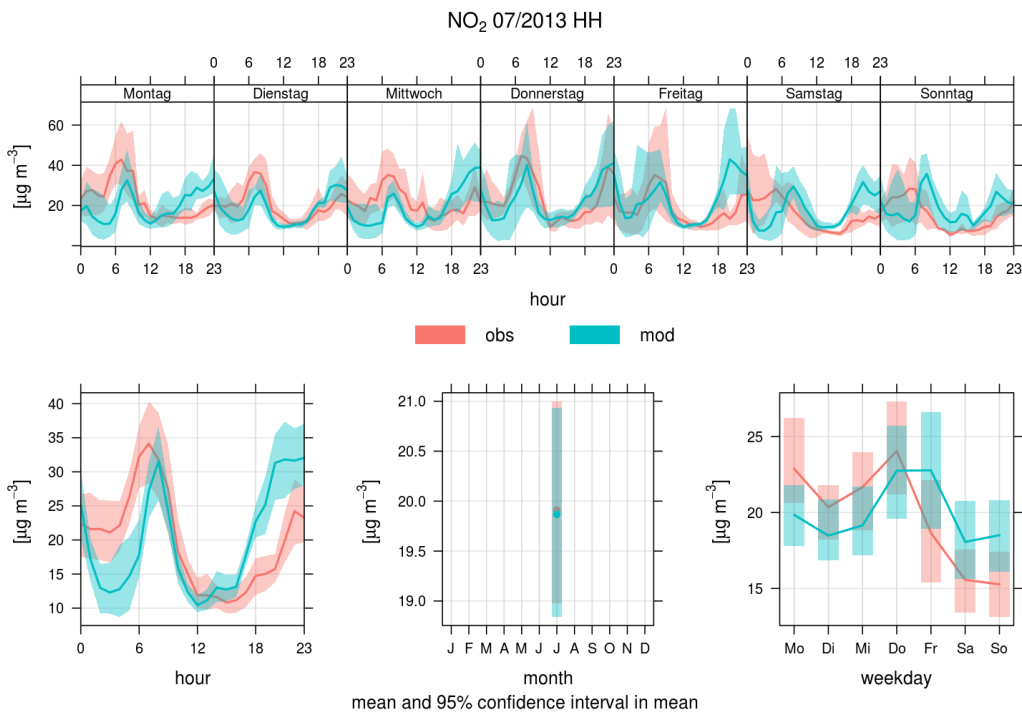


Figure 20: OpenAir temperial variation plots (diurnal and weekly profiles) generated by timeVariation for the modelled and observed data.

The time variation plot can be produced for different stations, by changing the keyword “site” in the timeVariation call (possible stations are: 80KT, 21BI, 74BT, 54BL, 51BF, 73FW, 72FI, 24FL, 68HB, 64KS, 70MB, 52NG, 79SU, 13ST, 17SM, 27TA, 20VE, 61WB).

The “openAir_evaluation.R” script provides statistical information of the model-observation comparison. The modStats command calculates common model evaluation statistics, amongst others the Pearson correlation coefficient, the coefficient of efficiency and the Index of Agreement. All calculations require complete pairs of modelled and observed data points. The modStats are written to openAir_modstats.csv.

The fourth plot is a Taylor diagram, drawn by the command TaylorDiagram. The Taylor diagram is a useful tool for model evaluation. It depicts the correlation coefficient, the standard deviation and the (centred) root-mean-square error simultaneously. Figure 21 shows an example diagram which is grouped by the modelled stations.

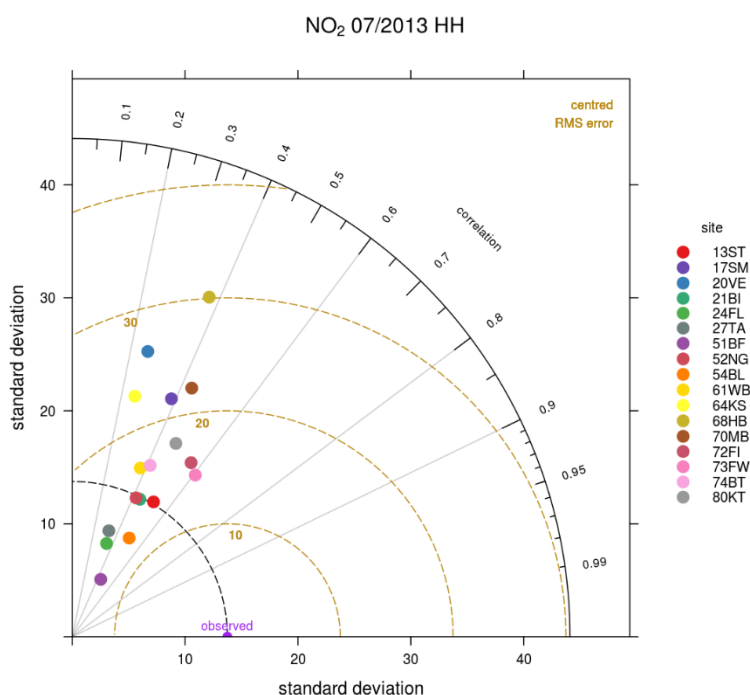


Figure 21: OpenAir generated Taylor diagram of observed and modelled data, grouped by stations.

The fifth and sixth plot is a scatterplot of modelled versus observed data at the stations. The scatterplot function is used for this. Figure 22 (top panel) shows an example plot with monthly averaged data. The bottom panel shows an example plot with daily averaged data, grouped by station.

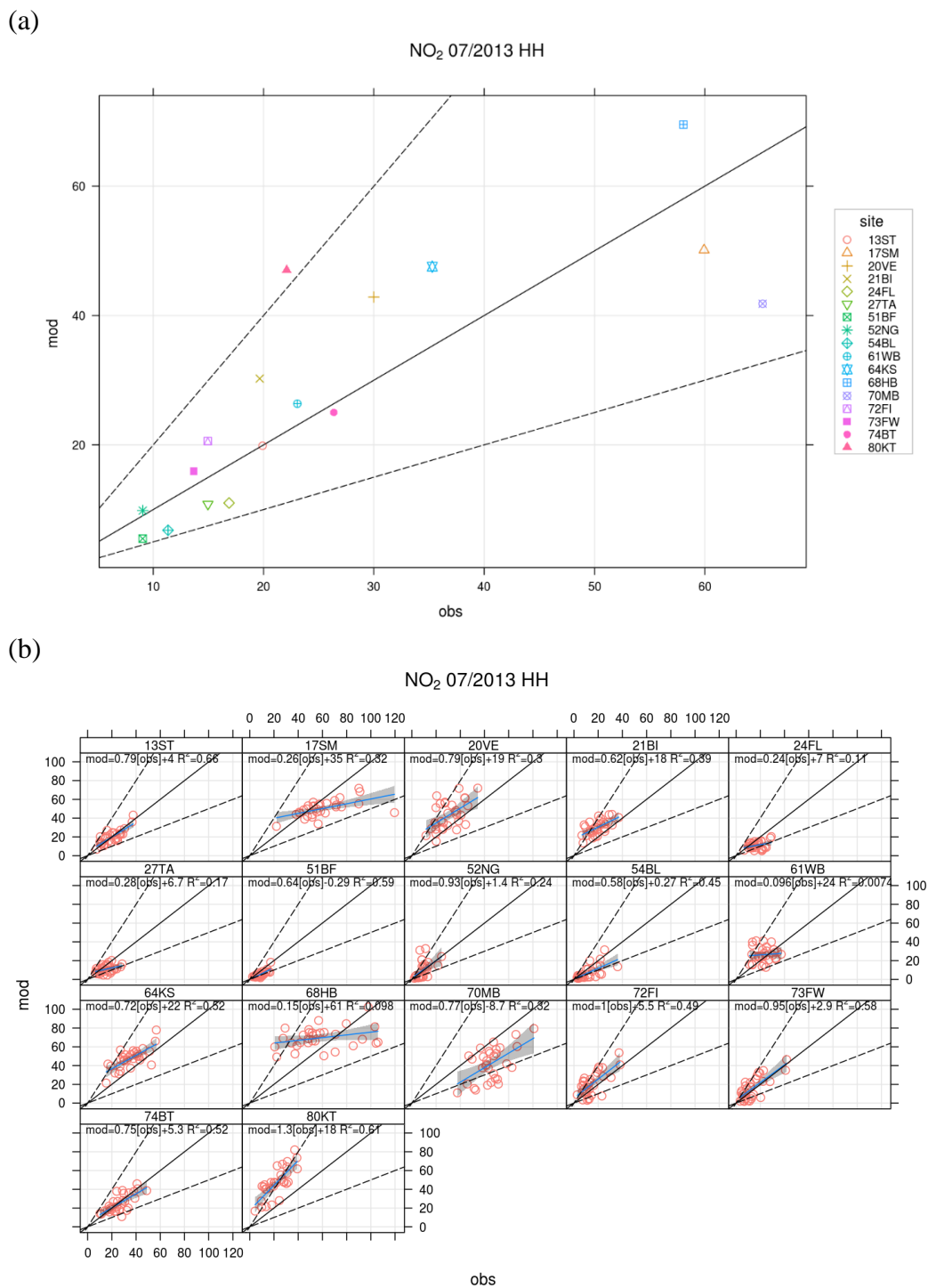


Figure 22: OpenAir scatterplot of (a) monthly averaged data and (b) daily averaged data, grouped by station.

5.6 Importing to ArcGIS

Spatial maps of the receptor grid concentration of the CityChem compounds in the netCDF output file **statrecp.nc** can be obtained by importing the file into the ArcGIS software. It is also possible to import **statrecphour.nc** and to select a specific hourly time step for display. This way, the concentration map can be used as a graphical layer together with other geo-spatial information.

To import the **statrecp.nc** file as raster layer in ArcGIS, open ArcMap™ or ArcPro™.

For the correct projection of the netCDF raster shape, make sure the layers in the opened project have the same projected coordinate system as used within the creation of the netCDF files (UTM projection WGS84). Search for the tool ‘Make NetCDF Raster Layer’ and open it (see Figure 23). Choose the input netCDF file and select the variable of interest. Select the correct x and y dimension variables and define a name for the Output raster shape. By clicking ‘OK’ the netCDF raster layer will be created and displayed in the coordinate system.

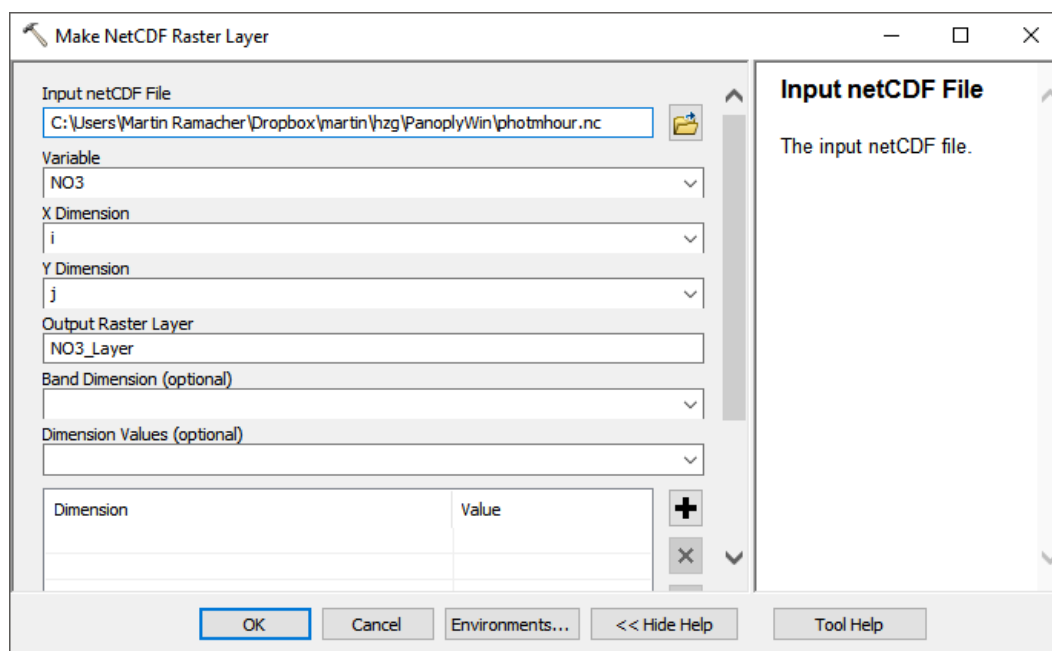


Figure 23: Make netCDF file tool with exemplary inputs.

To change the properties of the created raster layer right click on the created layer in the ‘Table of Contents’ section and open ‘Properties’. In the Tab ‘NetCDF’ other Variables and time steps can be selected (for **statrecphour.nc**) and will be displayed by clicking ‘OK’.

6 Utilities

6.1 MCWIND

If no installation of TAPM is available or if meteorology based on observational data is preferred, MCWIND is an alternative to generate meteorological input for EPISODE-CityChem. MCWIND has been developed at NILU and is a software that produces a diagnostic wind field. MCWIND adjusts a first guess wind field to a given topography in such manner that it becomes mass consistent, i.e. fulfilling the non-divergence criterion. This is achieved by use of a variational calculus technique.

To use MCWIND, meteorological observations from a meteorological tower (temperature at two heights, wind speed and direction, rel. humidity, precipitation, cloud cover) and wind observations (at surface) from at least one other meteorological stations are required.

Build the MCWIND program

The program **MCWIND.exe** is built by the automated installation routine:

```
{your-citychem}/$ ./installcc.csh util example
or (for self-made bcon files),
{your-citychem}/$ ./installcc.csh bcon example
```

Alternatively, separate installation is possible: change to subdirectory preproc/mcwind/src/ and type in the terminal:

```
{your-citychem}/preproc/mcwind/$ make -f Makfile.mc
```

This will create **MCWIND.exe** in the subdirectory bin/. Now create a symbolic link to the executable in subdirectory preproc/MCWIND/:

```
{your-citychem}/preproc$ ln -s ./mcwind/bin/MCWIND.exe
```

If you want to refresh the built at a later time, proceed as follows:

```
{your-citychem}/preproc/mcwind/$ make -f Makfile.mc clean
{your-citychem}/preproc/mcwind/$ make -f Makfile.mc
```

Input files for MCWIND

MCWIND is configured with the run script “run_file.asc”. In the run script the path and filenames of the input files have to be specified. The run script “run_file.asc” has to be placed in the same folder where mcwind.exe is run (here: {your-citychem}/preproc/MCWIND).

MCWIND requires four input files:

1. Topography input in CityChem format (see section 3.7).
2. Land use input in CityChem format (see section 3.7).
3. Surface roughness input in CityChem format (see section 3.7).
4. Meteorological input from observations as ASCII or CSV text file.

The option to use a file containing the applied vertical profile wind observations at the meteorological tower or a file containing the applied geostrophic (upper air) wind observations has been deactivated in MCWIND. The vertical wind profile is calculated internally in MCWIND applying surface similarity profiles according to Monin-Obukhov theory to calculate the wind speed and wind direction at a user-defined reference height (either a constant height above ground or the height of the mid-point of the first model layer).

Meteorological input file for MCWIND

The most important input to MCWIND is the meteorological input file. It is an ASCII text file in table-style format containing first wind speed and direction, temperature, temperature gradient (dT/dz), precipitation, cloud fraction, and pressure at the meteorological tower, second, wind speed and direction at one or more additional meteorological station(s) for each hour. Missing measurement data has to be indicated by -9900 (missing value).

Figure 24 shows an example of the meteorological input file for MCWIND.

	*	yyyy	mm	dd	hh	FF1	DD1	T25m	DT	RH%	PP_mm	Cloud	Press	FF2	DD2
1		2011	1	1	0	5.00	150	-2.60	0.007	94.00	0	0.89	995.00	7.00	170
2		2011	1	1	1	5.33	146	-2.67	0.007	93.33	0	0.89	994.13	7.67	166
3		2011	1	1	2	5.67	143	-2.73	0.117	92.67	0	0.89	993.27	8.33	163
4		2011	1	1	3	6.00	140	-2.80	0.048	92.00	0	0.89	992.40	9.00	160
5		2011	1	1	4	6.00	133	-2.87	0.026	92.67	0	0.89	991.80	8.67	148
6		2011	1	1	5	6.00	127	-2.93	0.022	93.33	0	0.89	991.20	8.33	134
7		2011	1	1	6	6.00	120	-3.00	0.019	94.00	0	0.89	990.60	8.00	120
8		2011	1	1	7	6.00	107	-3.20	0.016	93.67	0	0.89	990.83	8.67	112
9		2011	1	1	8	6.00	93	-3.40	0.014	93.33	0	0.89	991.07	9.33	106
10		2011	1	1	9	6.00	80	-3.60	0.012	93.00	0	0.89	991.30	10.00	100
11		2011	1	1	10	6.67	72	-4.57	0.009	91.67	0	0.89	991.67	10.67	85
12		2011	1	1	11	7.33	65	-5.53	0.008	90.33	0	0.89	992.03	11.33	71
13		2011	1	1	12	8.00	60	-6.50	0.008	89.00	0	0.89	992.40	12.00	60
14		2011	1	1	13	7.33	55	-7.20	0.009	88.33	0	0.89	993.17	10.67	58
15		2011	1	1	14	6.67	48	-7.90	0.011	87.67	0	0.89	993.93	9.33	54
16		2011	1	1	15	6.00	40	-8.60	0.014	87.00	0	0.89	994.70	8.00	50
17		2011	1	1	16	5.67	40	-8.57	0.018	87.00	0	0.89	995.30	7.33	47
18		2011	1	1	17	5.33	40	-8.53	0.018	87.00	0	0.89	995.90	6.67	44
19		2011	1	1	18	5.00	40	-8.50	0.018	87.00	0	0.89	996.50	6.00	40
20		2011	1	1	19	5.33	36	-8.30	0.018	87.33	0	0.89	997.03	6.00	37
21		2011	1	1	20	5.67	33	-8.10	0.022	87.67	0	0.89	997.57	6.00	33
22		2011	1	1	21	6.00	30	-7.90	0.027	88.00	0	0.89	998.10	6.00	30
23		2011	1	1	22	5.67	30	-8.27	0.035	88.33	0	0.89	998.60	6.33	26
24		2011	1	1	23	5.33	30	-8.63	0.035	88.67	0	0.89	999.10	6.67	23
25		2011	1	2	0	5.00	30	-9.00	0.036	89.00	0	0.89	999.60	7.00	20
26		2011	1	2	1	5.00	23	-9.37	0.036	88.00	0	0.89	1000.10	6.67	20
27		2011	1	2	2	5.00	17	-9.73	0.049	87.00	0	0.89	1000.60	6.33	20
28		2011	1	2	3	5.00	10	-10.10	0.064	86.00	0	0.89	1001.10	6.00	20
29		2011	1	2	3	5.00	10	-10.10	0.064	86.00	0	0.89	1001.10	6.00	20

Figure 24: Screenshot of a meteorological input file for MCWIND.

Format of the file:

Read format (ascii):

```
do n = 1, tend
  read(*) yyyy, mm, dd, hh, FF1, DD1, T25, DT, RH, PP, CLF, PRS, FF2, DD2, FF4, DD4
end do
```

Where the variables represent the following

<i>yyyy</i>	Year in format YYYY
<i>mm</i>	Month in format M (no leading zero)
<i>dd</i>	Day in format D (no leading zero)
<i>hh</i>	Hour in format H (no leading zero)
<i>FF1</i>	Wind speed at meteorological tower (in m/s)
<i>DD1</i>	Wind direction at meteorological tower (in degrees)
<i>T25</i>	Air temperature at user-defined reference height at meteorological tower (in °C)
<i>DT</i>	Temperature gradient dT at meteorological tower (in K/m), calculated as temperature difference between an upper and a lower measurement height.
<i>RH</i>	Relative humidity at meteorological tower (in %)
<i>PP</i>	Precipitation rate (in mm/h)
<i>CLF</i>	Cloud fraction at meteorological tower (fraction 0 ... 1)
<i>PRS</i>	Sea surface level pressure (hPa or mbar)
<i>FF2</i>	Wind speed at additional station 1 (m/s)
<i>DD2</i>	Wind direction at additional station 1 (degrees)
<i>FF4</i>	Wind speed at additional station 2 (m/s)
<i>DD4</i>	Wind direction at additional station 2 (degrees)

Running MCWIND to produce input for CityChem

Before running diagnostic wind field model MCWIND, it is necessary to configure the run script file “run_file.asc”, a line-based ASCII file. The name of the run script must not be changed, in order to be recognized as configuration file by the MCWIND program. The first four lines are for the paths and filenames of the input files. Then two lines follow with path and filenames which are not used by MCWIND. They should not be changed. From line 8 to 24 the path and filenames of the MCWIND output files are entered. These are the meteorological input files for CityChem-EPISODE; the output path should be the subdirectory INPUT/mcwind. Figure 25 shows this part of “run_file.asc”.

```

1 './input_mcwind_helsinki/topo.asc'           ! Filename of file containing the Topography.
2 './input_mcwind_helsinki/landuse_episode.asc' ! Filename of file containing the Land-Use information.
3 './input_mcwind_helsinki/surrough_episode.asc' ! Filename of file containing the Surface Roughness in meters.
4 './input_mcwind_helsinki/mcwind_input_helsinki_2011.txt' ! Filename of file containing the applied surface wind observations
5 'met_profile.txt'                            ! NOT USED Filename of file containing the applied profile wind observations
6 'met_geostrophic.txt'                       ! NOT USED Filename of file containing the applied geostrophic wind observations
7 './output_mcwind_helsinki/LOG_McWIND.txt'    ! The name of the log file for this run.
8 './output_mcwind_helsinki/top_file.asc'      ! File containing the topography and the model-depth fields
9 './output_mcwind_helsinki/Res_U_V_and_W fld' ! Result-file containing u, v and w in the lambda points.
10 './output_mcwind_helsinki/temp_episode.asc' ! Result-file containing T and DT/DS in the lambda points.
11 './output_mcwind_helsinki/rhum_episode.asc' ! Result-file containing u, v and w in the lambda points.
12 './output_mcwind_helsinki/prec_episode.asc' ! Result-file containing T and DT/DS in the lambda points.
13 './output_mcwind_helsinki/clou_episode.asc' ! Result-file containing u, v and w in the lambda points.
14 './output_mcwind_helsinki/ts_episode.asc'    ! Time series of surface layer grid values at containing obs stations.
15 './output_mcwind_helsinki/ts_alpha_episode.asc' ! Time series of the applied alfa-value.
16 './output_mcwind_helsinki/ts_preproc_episode.asc' ! Time series of domain mixing height.
17 './output_mcwind_helsinki/z0_file.asc'       ! File containing the applied z0 (for momentum) information
18 './output_mcwind_helsinki/landuse_file.asc' ! File containing the applied land-use information.
19 './output_mcwind_helsinki/tsrad_episode.asc' ! Result-file containing TSRAD in the lambda points.
20 './output_mcwind_helsinki/albedo_episode.asc' ! Result-file containing ALBEDO in the lambda points.
21 './output_mcwind_helsinki/nrad_episode.asc' ! Result-file containing NRAD in the lambda points.
22 './output_mcwind_helsinki/press_episode.asc' ! Result-file containing pressure in the lambda points.
23 './output_mcwind_helsinki/tsmet_episode.asc' ! Result-file containing TSMET in the lambda points.

```

Figure 25: Screenshot of the input/output part of the “run_script.asc” configuration file for MCWIND.

The lines which follow the input/output part of “run_script.asc” are:

CLAT, CLON	Latitude and longitude of the model domain mid-point, 1-degree accuracy is sufficient (space separated)
TSHIFT	Time shift between universal time (UT or GMT) and the local time (LT) calculated as $TSHIFT = UT - LT$. For Hamburg winter time this is -1.
TEND	Number of hours to be computed
IM, JM, KM	Number of grid cells in x-direction, y-direction and z-direction (space separated). The grid dimension should be chosen in accordance with the planned CityChem run and should also agree with the dimensions used for producing the auxiliary input files (topography, land use, surface roughness)
DX, DY	Grid cell width in meter (space separated)
FLD, STR	First layer depth (in m) and stretch factor (dim.-less) (space-separated).

The next KM lines are the layer depths (LDepth in m) of all layers in ascending order from ground to domain top layer. These layer-specific values are only used by MCWIND if STR is a negative value.

```

do k = 1, KM
  LDepth
end do

```

MTE	Maximum topography elevation (in m) for test setup
U0, V0, W0	Wind velocities for test setup (space separated)
Z0C	Constant surface roughness parameter (in m)
STAB_CLASS	Stability class (neutral = 4)
PROF	Surface profile method (1 = power law formula, 2 = Zilitinkevich)

The lines following this part are not intended to be changed. It continues in line 73 of “run_script.asc” with the part on observation stations.

NSURF	Number of meteorological stations at surface
NAME1	Name of the main station (meteorological tower)
X1, Y1	x-position and y-position of main station (in km) from the model domain origin, i.e. the south west corner of the model domain (space separated)
HW1	Height (m above ground) for wind measurement at main station
HT1	Height (m above ground) for temperature measurement at main station
UHT1	Upper height (m above ground) for DT measurement (vertical temperature gradient) at main station
LHT1	Lower height (m above ground) for DT measurement at main station
Z01	Surface roughness at main station (do not change)
PWR1	Surface PWR value at main station (do not change)
SCL1	Surface SCALE value at main station (do not change)
NAME2	Name of the first additional station (station 2)
X2, Y2	x-position and y-position of station 2 (in km) from the model domain origin, i.e. the south west corner of the model domain (space separated)
HW2	Height (m above ground) for wind measurement at station 2
Z02	Surface roughness at station 2 (do not change)
PWR2	Surface PWR value at station 2 (do not change)
SCL2	Surface SCALE value at station 2 (do not change)
<i>NAME3</i>	<i>Name of the second additional station (station 3)</i>
<i>X3, Y3</i>	<i>x-position and y-position of station 3 (in km) from the model domain origin, i.e. the south west corner of the model domain (space separated)</i>
<i>HW3</i>	<i>Height (m above ground) for wind measurement at station 3</i>
<i>Z03</i>	<i>Surface roughness at station 3 (do not change)</i>
<i>PWR3</i>	<i>Surface PWR value at station 3 (do not change)</i>
<i>SCL3</i>	<i>Surface SCALE value at station 3 (do not change)</i>

(Continue the same sequence for any further additional surface stations.)

NGEO	Number of geostrophic meteorological stations. NGEO is set to zero since the use of geostrophic (upper air) stations is deactivated.
-------------	--

After configuration of “run_script.asc” **MCWIND.exe** can be run. File “run_script.asc” has to be placed in the same directory where MCWIND is run. The output files will be written to subdirectory INPUT/mcwind/. To start MCWIND type:

```
{your-citychem}/preproc/mcwind/$ ./MCWIND.exe
```

The run will calculate the diagnostic wind field and other meteorological fields for the defined model grid for each hour of the period defined by TEND. The produced files (see Appendix A, Table A1) will be in ASCII format, except of the file containing the u, v, and w wind components, which will be in binary format.

6.2 UECT

The Urban Emission Conversion Tool (UECT) is a utility for preparing emission input files with hourly emission values based on the yearly emission table for an urban study region. UECT produces emission input files for point sources, line sources and area sources for use with EPISODE-CityChem. For area and line sources, emission files for all CityChem compounds will be produced. For point sources, one emission file is produced that contains emissions of all compounds. UECT output files have a time stamp with begin and end date as part of the filename.

UECT has an interface to produce output files for TAPM (*.pse, *.lse, and *.ase files). A typical composition of volatile organic compounds is assumed when converting to the TAPM compound Rsmog ($[Rsmog] = 0.0067[TVOC]$). Carbon monoxide (CO) is not considered in TAPM emission files. UECT *starting with v2.0* supports the production of TAPM emission input files.

Point sources:

Information about point sources has to include data on stack parameters, effluent parameters and adjacent building. Missing data will be replaced by default parameter values, depending on SNAP emission category. UECT *starting with v2.1* produces two point source files, one including isoprene and one without.

Line sources:

Line sources have emissions uniformly distributed along a straight line using the start and end points specified.

Area sources:

UECT produces area source emission files which are only including the area emissions inside the model domain. Input area emissions have to be on a regular grid, which can have a smaller or equal grid width as the chosen model grid width. UECT creates four categories of area emissions: **(I)** residential heating (SNAP 2), **(II)** commercial and industrial combustion and processes (SNAP 1, 3, 4, 5, 9), **(III)** solvent use and agriculture (SNAP 6, 7, 10), and **(IV)** ship traffic (SNAP 8).

TAPM area/volume source emission file (*.ase file). In TAPM, area sources can have a vertical extent. In CityChem, area/volume sources are areal sources at the ground with a quadratic area, the vertical distribution of area emissions is dealt with in the model and depends on the area source category.

Input is yearly total emissions with geospatial reference (x/y-coordinates):

- The geo-reference for point sources is the (x,y)-coordinate of the point.
- The geo-reference for lines sources is the start (x,y)-coordinate together with the end (x,y)-coordinate of the line, and the geo-reference.
- For area sources is the (x,y)-coordinate of the lower left (southwest) corner together with the (x,y)-coordinate of the upper right (northeast) corner of the quadratic area cell.

Output is CityChem emission input files in the format as described in section 3.5.

Build the UECT program

The UECT program is built by the automated installation routine:

```
{your-citychem}/$ ./installcc.csh util example
or (for self-made bcon files),
{your-citychem}/$ ./installcc.csh bcon example
```

Alternatively, separate installation of the UECT program is possible. Change to subdirectory `preproc/uct2.2/`. Edit “`Makefile.uct`” in a text editor of your choice. Edit `NCPATH`, which is the path of the netCDF library and include files on your computer. By default this is:

```
NCPATH = /usr/local/netcdf4
```

Save the file and type in the terminal:

```
{your-citychem}/preproc/uct2.2$ make -f Makfile.uct
```

This will create the UECT executable program in the subdirectory `bin/`. Now create a symbolic link to the executable in subdirectory `preproc/`:

```
{your-citychem}/preproc/uct2.2$ cd..
{your-citychem}/preproc$ ln -s ./uct/bin/uct.exe
```

If you want to refresh the built at a later time, proceed as follows:

```
{your-citychem}/preproc/uct2.2$ make -f Makfile.uct cleanall
{your-citychem}/preproc/uct2.2$ make -f Makfile.uct
```

Meta information on emissions sources

Before emission input files can be created with UECT, the basic meta information on the emission sources and file locations has to be included in the user metadata file (“`cctapm_meta.inp`”, see section 3.3). The file has to be located in subdirectory `preproc`. Some information is required for all types: simulation-ID (`simid`), name of the UECT log file (`log_file`), city-scale model (`model`), output format (`ep_fmt=1`) for ASCII, number of hours (`hmonth`), start and end dates (`start_string`, `end_string`, `startdate`, `enddate`), city domain dimensions and projection (`nx`, `nz`, `dx`, `utmx`, `utmy`, `utm_zone`). Depending on source type (`point`, `line`, `area`) different additional information has to be provided.

Point sources:

Set emission type to point sources (source='PSE') and edit the total number of point emission sources in the input to UECT (nsopp). Provide the full path and file name of the input file of point sources (point_path). The input file has to be in CSV format containing the yearly emission totals for each point source.

Line sources:

Set emission type to line sources (source='LSE') and edit the total number of line emission sources in the input to UECT (nsoll). Provide the full path and file name of the input file of line sources (line_path). The input file has to be in CSV format containing the emission rate (in g/s) for each line source.

Area sources:

Set emission type to line sources (source='ASE') and edit the total number of area emission sources in the input to UECT (nsoaa). Provide the full path and file name of the input file of area sources (area_path). The input file has to be in CSV format containing the yearly emission totals for each area source. In addition, provide the grid width (dxarea) of the area emission grid in the area emission input file. The area cells have to be quadratic and on a regular grid. The orientation of the area grid can be different from the model grid, but it has to be in the same UTM zone (WGS 1984). An optional netCDF file of the area source emission output of the first 3 days of the simulation period can be produced by UECT. In this case set ncout to 1.

Area emissions of residential heating:

Area emissions of residential heating (SNAP 2) can be generated with dependence on the daily average temperature (heat degree days method). For this procedure, the program **tapm4cc.exe** (section 3.4) has to be run using the to write netCDF output (*ep_fmt* = 3) in the user metadata file ("cctapm_meta.inp"). This generates a netCDF file that contains air temperature at ground and vertical temperature gradient (T_and_dtdz_*.nc) in the given output file path. It is suggested to move this netCDF file (from subdirectory INPUT/tapm/) to subdirectory testdata/ncfiles/. Then edit "cctapm_meta.inp" in preproc/ by choosing *ep_fmt* = 1 and providing the file path ('./testdata/ncfiles/') in the line of *tapm_path*. For the generation of the area sources follow the steps above ("Area emissions"). After that check the user log-file (by default "user_log.txt"). If the temperature correction of residential heating emissions was done properly the message will be shown in the user log-file: "UECT: Air temperature is used to calculate SNAP2 emissions".

Area emissions of isoprene:

Area emissions of isoprene can be created by selecting SNAP 11 as emission sector and entering the emission amount into the cell for NMVOC emissions in the input file (see below). The provided annual emission total should be an isoprene emission estimate for standard conditions (T=303 K and optimum solar radiation, i.e. gamma_P=1 and gamma_LAI=1) in units kgC/year. UECT, **starting with v2.1**, will then automatically detect the area emission of isoprene and modulate them with a diurnal profile of sunlight and computes the temperature dependence based on the actual daily mean temperature. The netCDF file that contains air temperature at ground and vertical temperature gradient

(T_and_dtdz_*.nc) has to be provided; the procedure is the same as for the area emissions of residential heating. It is cautioned that this implementation is only for testing and that resulting hourly isoprene emissions might be up to a factor of 10 higher than the real-world isoprene emissions in mid-latitude and high-altitude regions.

All sources (points, lines and areas):

Set emission type to all emission sources (source='ALL'). All information specified in the points above has to be provided for a UECT run that produces all emission output (points, lines and areas).

Format of emission input files for UECT

Emission data for the different source types has to be given to UECT as ASCII files in CVS format (UTF-8 encoded). The emission input files should contain one header line followed line-by-line with the data for each source.

Data values for one source have to be separated by ','.

The format for the different source types is described in the following. Figure 26 gives an overview of the input format of point, line and area source emission input files. Missing values have to be denoted by -999.

Point source emission input:

Contains the total number of point sources (np) for which exhaust stack and emission data is provided.

Read format (ascii):

read(*) headerline

The first line of the point source CSV file is a header line with a descriptive text.

Then follows one line per point source object ip:

```
do ip = 1, np
  read(*) snap(ip),xcor(ip),ycor(ip),Hi(ip),Vi(ip),Ti(ip),radi(ip),
           NOx(ip),NMVOC(ip),CO(ip),SO2(ip),NH3(ip),PM2.5(ip),PM10(ip)
enddo
```

Each entry for one source is separated by ‘,’ or ‘;’. The following list gives a description of each entry for point sources:

<i>snap</i>	SNAP emission sector code (1, ..., 10)
<i>xcor</i>	west-east (UTMx) local coordinate of the stack (m). Give as truncated UTM x-coordinate (without the two leading digits for UTM zone).
<i>ycor</i>	south-north (UTMy) local coordinate of the stack (m).
<i>Hi</i>	stack height above the ground (m).
<i>Vi</i>	stack exit velocity (m/s).
<i>Ti</i>	stack exit temperature (°C).
<i>radi</i>	internal stack radius (m).
<i>NOx</i>	emission rate of nitrogen oxides, NOx (kg/year).
<i>NMVOC</i>	emission rate of non-methane volatile organic compounds, NMVOC (kg/year).
<i>CO</i>	emission rate of carbon monoxide, CO (kg/year).
<i>SO2</i>	emission rate of sulphur dioxide, SO ₂ (kg/year).
<i>NH3</i>	emission rate of ammonia, NH ₃ (kg/year).
<i>PM2.5</i>	emission rate of particulate matter with aerodynamic diameter ≤ 2.5 μm, PM _{2.5} (kg/year).
<i>PM10</i>	emission rate of particulate matter with aerodynamic diameter ≤ 10 μm, PM ₁₀ (kg/year).

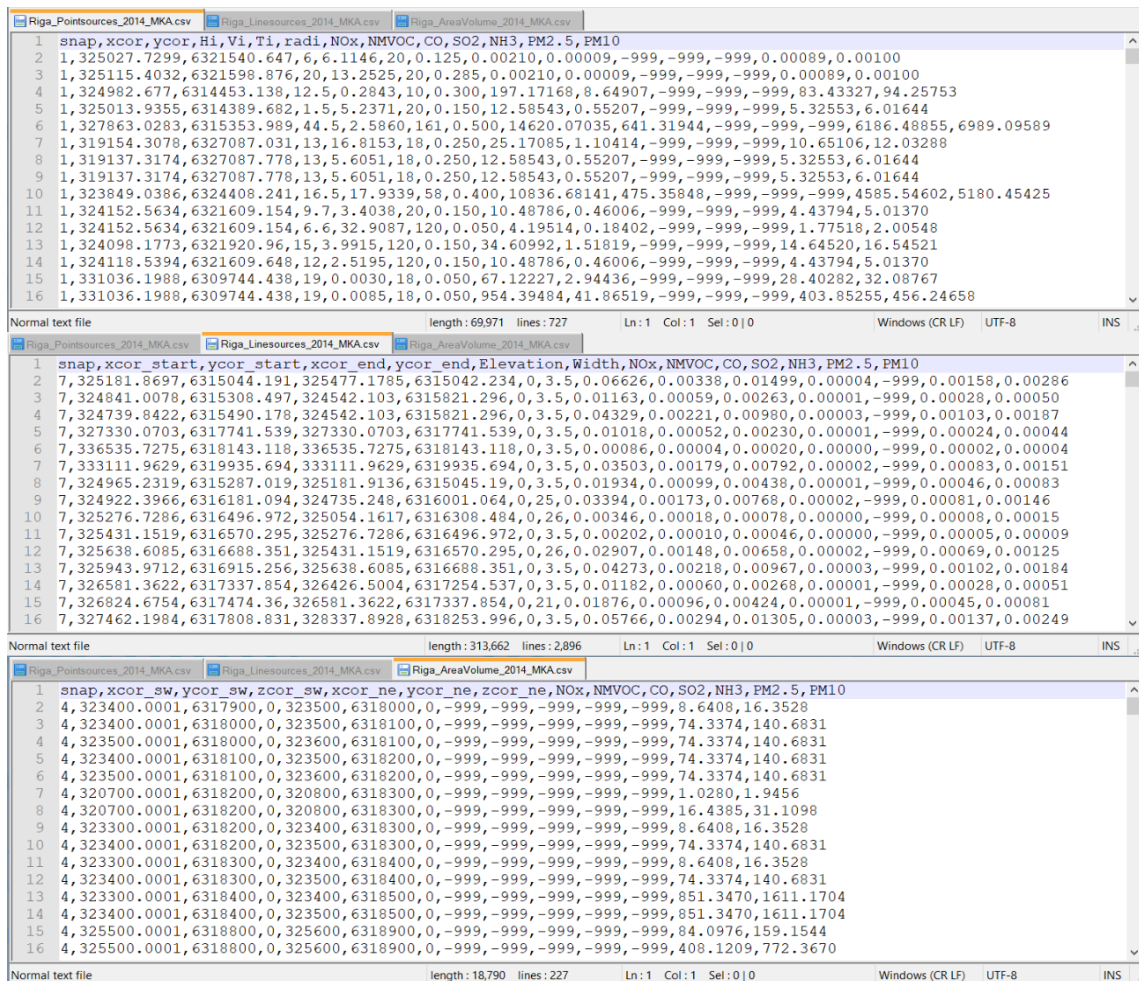


Figure 26: Screenshot of excerpts from the different emission input files for UECT: point source emissions, line source emissions, and area source emission (from top to bottom).

Line source emission input:

Contains the total number of traffic line sources (nql) for which street data and emission data is provided. Line source emissions have to be in unit g/s, which means that emissions in unit g/s/m have to be multiplied by the road link length (m) and that yearly total emissions in kg/year have to be multiplied by the factor $1.e3/(365*24*3600)$.

Read format (ascii):

read(*) headerline

The first line of the line source CSV file is a header line with a descriptive text.

Then follows one line per line source object L:

```
do L = 1, nql
  read(*) snap(L), xcor_start(L), ycor_start(L), xcor_end(L), ycor_end(L), elevation(L),
          width(L), NOx(L), NMVOC(L), CO(L), SO2(L), NH3(L), PM2.5(L), PM10(L)
enddo
```

Each entry for one source is separated by ‘,’ or ‘;’. The following list gives a description of each entry for line sources:

<i>snap</i>	SNAP emission code for traffic (7 or 8).
<i>xcor_start</i>	west-east (UTMx) local coordinate of the line source (m) as start coordinate of the road link. Give as truncated UTM x-coordinate (without the two leading digits for UTM zone).
<i>ycor_start</i>	south-north (UTMy) local coordinate of the line source (m) as start coordinate of the road link.
<i>xcor_end</i>	west-east (UTMx) local coordinate of the line source (m) as end coordinate of the road link. Give as truncated UTM x-coordinate (without the two leading digits for UTM zone).
<i>ycor_end</i>	south-north (UTMy) local coordinate of the line source (m) as end coordinate of the road link.
<i>elevation</i>	elevation of the street (m). By default elevation should be 0. A value >0 indicate treatment of the line source as elevated level source. UECT will automatically set elevation to zero if missing values or values > 0 are provided.
<i>width</i>	width of the street, including all lanes (m).
<i>NOx</i>	emission rate of nitrogen oxides, NO _x (g/s).
<i>NMVOC</i>	emission rate of non-methane volatile organic compounds, NMVOC (g/s).
<i>CO</i>	emission rate of carbon monoxide, CO (g/s).
<i>SO2</i>	emission rate of sulphur dioxide, SO ₂ (g/s).
<i>NH3</i>	emission rate of ammonia, NH ₃ (g/s).
<i>PM2.5</i>	emission rate of particulate matter with aerodynamic diameter ≤ 2.5 μm, PM _{2.5} (g/s).
<i>PM10</i>	emission rate of particulate matter with aerodynamic diameter ≤ 10 μm, PM ₁₀ (g/s).

Area source emission input:

Contains the total number of are sources (nar) for which emission data is provided. Area/volume sources have emissions uniformly distributed within a volume or x-y plane area (if heights are the same) with sides aligned along the Cartesian coordinate axes and side lengths $|x2v-x1v|$ and $|y2v-y1v|$ in the x- and y-directions, respectively.

Read format (ascii):

read(*) headerline

The first line of the area source CSV file is a header line with a descriptive text.

Then follows one line per are source object ia:

do ia = 1, nar

**read(*) snap(ia),xcor_sw(ia),ycor_sw(ia),zcor_sw(ia),xcor_ne(ia),ycor_ne(ia),zcor_ne(ia),
NOx(ia),NMVOC(ia),CO(ia),SO2(ia),NH3(ia),PM2.5(ia),PM10(ia)**

enddo

Each entry for one source is separated by ‘,’ or ‘;’. The following list gives a description of each entry for area sources:

<i>snap</i>	SNAP emission sector code (1, ..., 10). Area sources with SNAP code 7, 8 or 9 are not allowed.
<i>xcor_sw</i>	west-east (UTMx) local coordinate (m) of the lower left corner (SW-corner) of the area emission cell. Give as truncated UTM x-coordinate (without the two leading digits for UTM zone).
<i>ycor_sw</i>	south-north (UTMy) local coordinate (m) of the lower left corner (SW-corner) of the area emission cell.
<i>zcor_sw</i>	elevation above ground (m) of the lower left corner (SW-corner) of the area emission cell.
<i>xcor_ne</i>	west-east (UTMx) local coordinate (m) of the upper right corner (NE-corner) of the area emission cell. Give as truncated UTM x-coordinate (without the two leading digits for UTM zone).
<i>ycor_ne</i>	south-north (UTMy) local coordinate (m) of the upper right corner (NE-corner) of the area emission cell.
<i>zcor_ne</i>	elevation above ground (m) of the upper right corner (NE-corner) of the area emission cell.
<i>NOx</i>	emission rate of nitrogen oxides, NO _x (kg/year).
<i>NMVOC</i>	emission rate of non-methane volatile organic compounds, NMVOC (kg/year).
<i>CO</i>	emission rate of carbon monoxide, CO (kg/year).
<i>SO2</i>	emission rate of sulphur dioxide, SO ₂ (kg/year).
<i>NH3</i>	emission rate of ammonia, NH ₃ (kg/year).
<i>PM2.5</i>	emission rate of particulate matter with aerodynamic diameter $\leq 2.5 \mu\text{m}$, PM _{2.5} (kg/year).
<i>PM10</i>	emission rate of particulate matter with aerodynamic diameter $\leq 10 \mu\text{m}$, PM ₁₀ (kg/year).

Limitations:

In CityChem only area sources in a plane area are allowed. Vertical distribution of area emissions is done using default profiles by the model. The model user cannot change the vertical distribution of area emissions. The four emission categories that can be used in CityChem are: 1) residential heating, 2) industrial and commercial sources, 3) solvent use and agriculture, and 4) shipping. UECT allocates all other SNAP sectors to one of these categories. UECT does not deal with SNAP8 emissions other than shipping; which means that area emissions from railway and off-road machinery have to be included in other SNAP sectors to avoid that they are included in the ship emission files. It is recommended to treat railway as line sources and to treat off-road machinery as SNAP 10 (agriculture) area sources.

Running UECT

After preparation of emission data for the one or all source types in CSV format (UTF-8 encoded) and configuration of the basic meta information in the user metadata file “cctapm_meta.inp” (in subdirectory preproc), a run with UECT can be started. Change to subdirectory preproc and type in a terminal:

```
{your-citychem}/preproc$ ./uct.exe
```

UECT will produce a log file (in subdirectory preproc) with information about the status of completion of the program and about plausibility of the emission output. The emission input files for EPISODE-CityChem will be placed in the directory indicated in the user metadata file (point_path, line_path, area_path). The optional netCDF file for inspecting area source emissions will be in the directory specified by area_path.

UECT may stop processing in two cases: 1) invalid entries in the emission CSV file or in “cctapm_meta.inp” or 2) the emission CSV file cannot be opened or read by the program. In the first case, it is advisable to consult the UECT log file for hints about the invalid entry. In the second case, either the file path of the emission CSV file was wrong or the file does not exist. If the file exists and its path is correct, open the CSV file with a text editor (Notepad++ is recommended) in order to find invalid data or line breaks (such as MS-DOS carriage return). Numerical values have to be with decimal point always. Sometimes it helps to save the file in Linux as Unix file and to make sure that it is UTF-8 encoded. It may also happen, that the CSV file does not include all data columns or that values are in wrong order. The user has to follow strictly the ASCII format for the data entries in point source, line source and area source files for UECT.

6.3 BCONCC

In this section it is explained how to prepare the 3-D boundary conditions field for EPISODE-CityChem starting from the concentration output from CMAQ for a geographical region that includes the city model domain. For a description of performing simulations with CMAQ it is referred to the CMAQ manuals at <https://www.epa.gov/cmaq/>.

The conversion of CMAQ concentration output into CityChem BCON input requires two stages:

- 1) In the first stage the city's domain extent is cut out from a CMAQ domain, interpolating the hourly concentrations to CityChem's horizontal main grid resolution.
- 2) In the second stage BCON files containing background concentrations of all individual CityChem compounds are created for the defined model domain in the required input format.

1. Tool for cutting CMAQ output and linear interpolation

The first step requires the IOAPI 3.1 library to be installed on your system. A modified Fortran program from IOAPI's m3tools is used for cutting out the EPISODE-CityChem model domain and for bilinear interpolation of hourly CMAQ concentrations to the required horizontal resolution of the model domain.

IOAPI3.1 has to be installed using the gfortran compiler (e.g. using ioapi's Makeinclude.Linux2_x86_64gfort). The subdirectory preproc/bconcc2.0/ioapi3.1 contains all files required for the installation of IOAPI3.1 on your computer.

A) Build the IOAPI library

The installation of the IOAPI version 3.1 library needs to be done only one time, when BCONCC is installed the first time. The Models-3/EDSS Input/Output Application Programming Interface (I/O API) is an easy-to-learn, easy-to-use programming library for data storage and accesses, available from both Fortran and C. It is the standard data access library for both the NCSC/CMAS's EDSS project and EPA's Models-3, including CMAQ. The IOAPI provides a variety of data structure types for organizing the data, and a set of access routines which off selective direct access to data (<https://www.cmascenter.org/ioapi>).

If the IOAPI library has not been built by the installation script installcc.csh or does not exist from a previous installation, the following needs to be done. Change to subdirectory /preproc/bconcc2.1/ioapi3.1/. In this folder you need to edit the Makefile. The following editing needs to be done in Makefile:

- Edit BASEDIR in Makefile to the path on your computer, e.g.:

```
BASEDIR = /home/yourname/citychem1.2/preproc/bconcc2.1/ioapi3.1
```

- Edit NC_LIB path, path of the netcdf4 library on your computer, e.g.:

```
NC_LIB = -L/usr/local/netcdf4/lib
```

Then build the IOAPI library by typing:

```
{your-citychem}/preproc/bconcc2.1/ioapi3.1$ make all
```

If you want to refresh the built, type:

```
{your-citychem}/preproc/bconcc2.1/ioapi3.1$ make clean
```

```
{your-citychem}/preproc/bconcc2.1/ioapi3.1$ make all
```

B) Create m3cpleplus.exe

The tool for cutting the study domain, interpolating between the different resolutions and transforming to UTM projection is **m3cpleplus.exe**. If the program has not been installed by the automated routine `installcc.csh`, the below needs to be done, otherwise you can continue with step C).

To create `m3cpleplus.exe` type:

```
{your-citychem}/preproc/bconcc2.1/ioapi3.1$ make m3p
```

Then change to next higher directory,

```
{your-citychem}/preproc/bconcc2.1/ioapi3.1$ cd ..
```

And create a symbolic link to the executable:

```
{your-citychem}/preproc/bconcc2.1$ ln -s ./ioapi3.1/bin/m3cpleplus.exe
```

C) Create and edit a GRIDDESC file for the city

For each city model domain a GRIDDESC file has to be created containing the information about the geographic projection, domain extent and resolution.

Change the subdirectory `preproc/bconcc2.1/city_grids/`. Create a new text file with name `GRIDDESC_Yourcity`. Edit in Notepad++ or other text editor of your choice. The following information has to be entered:

Line 1:

```
' '
```

Line 2:

string with user-defined name of the mother grid

Line 3:

1st value: 5 (= UTM projection, do not change)

2nd value: 32.000 (= UTM zone)

3rd, 4th value: coordinate of the reference of the mother grid (YORIG and XORIG)

Line 4:

```
' '
```

Line 5:

'SHEBA1AUTM' (user-defined name of the city grid)

Line 6:

1st value: string with name of the mother grid (as in Line 2)
 2nd value: UTM x coordinate (truncated) of the Origin (southwest corner) of the city domain, after subtracting the grid width dx to account for the one boundary cell.

3rd value: UTM y coordinate of the Origin (southwest corner) of the city domain, after subtracting the grid width dy (=dx) to account for the one boundary cell

4th, 5th value: grid width dx and dy (=dx) in meters

6th, 7th value: number of cells in x- and y-direction of the city domain plus 2 cells for the boundaries

8th value: 1 (= UTM projection, do not change)

Line 7

' '

Below an example for the city GRIDDESC file of Hamburg:

```
''
'UTM_EUROPE3'
 5  32.000  60.000  10.000  0.000  0.000
''
'SHEBA1AUTM'
'UTM_EUROPE3'  550750.0  5917656.0  1000.000  1000.000  32 32  1
''
```

D) Convert METCRO3D files and re-project to city

The meteorology file of the CMAQ simulation, METCRO3D, is needed to extract 3-D fields of temperature and pressure. METCRO3D has to be converted by cutting out the geographic extent of the city study domain and re-projecting the cut-out to UTM coordinate projection. The C-Shell Unix script run_m3cple_city_metcro.csh is used for the conversion of METCRO3D files.

The script can be found in subdirectory preproc/bconcc2.1/. Some items in the script have to be edited before use for a specific city domain. The following editing needs to be done in run_m3cple_city_metcro.csh:

- Edit GRIDDESC variable by specifying the path and filename of the GRIDDESC file for the city model domain:, e.g.:

```
setenv GRIDDESC ./city_grids/GRIDDESC_Hamburg
```

- Edit INFILE_SH variable by specifying the path where the METCRO3D files are located and the name (last three chars are days), e.g.:

```
setenv INFILE_SH = /home/yourname/CMAQ/data/meteo/METCRO3D_yyyy${strDAYin}
```

- Edit OUTFILE_SH variable by specifying the path where the output files should be written and the name of the files (last three chars are days), e.g.:

```
setenv OUTFILE_SH = /home/yourname/CMAQ/Yourcity/m3meteo/metc_yyyy${strDAYin}
```

- Edit DAYin (start day) and Run_Id_end (end day) to define the time period of the METCRO3D files to convert, e.g.:

```
set DAYin = 1
set Run_Id_end = 365
```

Run the script on a Linux/Unix computer by typing:

```
{your-citychem}/preproc/bconcc2.1$ ./run_m3cple_city_metcro.csh
```

E) Convert CONC files and re-project to city

The 3-D concentration output of the CMAQ simulation, CONC, contains the hourly concentrations of gases and particles. CONC files have to be converted by cutting out the geographic extent of the city study domain and re-projecting the cut-out to UTM coordinate projection. The C-Shell Unix script run_m3cple_city_conc.csh is used for the conversion of CONC files. This step is time-consuming – depending on the chosen size of the model domain – because the conversion is done for each simulation hour.

The script can be found in subdirectory preproc/bconcc2.1/. Some items in the script have to be edited before use for a specific city domain. The following editing needs to be done in run_m3cple_city_conc.csh:

- Edit GRIDDESC variable by specifying the path and filename of the GRIDDESC file for the city model domain:, e.g.:

```
setenv GRIDDESC ./city_grids/GRIDDESC_Hamburg
```

- Edit INFILE_SH variable by specifying the path where the CONC files are located and the name (last three chars are days), e.g.:

```
setenv INFILE_SH = /home/yourname/CMAQ/output/CONC_yyyy${strDAYin}
```

- Edit OUTFILE_SH variable by specifying the path where the output files should be written and the name of the files (last three chars are days), e.g.:

```
setenv OUTFILE_SH = /home/yourname/CMAQ/Yourcity/m3bcon/conc_yyyy${strDAYin}
```

- Edit DAYin (start day) and Run_Id_end (end day) to define the time period of the CONC files to convert, e.g.:

```
set DAYin = 1
set Run_Id_end = 365
```

Run the script on a Linux/Unix computer by typing:

```
{your-citychem}/preproc/bconcc2.1$ ./run_m3cple_city_conc.csh
```

2. Tool for creating 3-D background concentrations

In the second stage the netCDF concentration file and the meteorology file are merged and a 3-D BCON file is generated containing the 3-D boundary conditions for all compounds. The Fortran program **cmaq4cc.exe** converts CMAQ concentrations of gas-phase compounds to unit $\mu\text{g}/\text{m}^3$, summarizes concentrations from particulate compounds in CMAQ to PM2.5 (based on CMAQ's AERO5 aerosol model), and interpolates the vertical model layers of CMAQ to the corresponding vertical layers of EPISODE-CityChem. The tool creates one BCON file in Fortran binary format for the specified simulation month.

A) Build the cmaq4cc tool

Building the program **cmaq4cc.exe** requires the IOAPI 3.1 library and the netcdf4 library. See stage 1 description for installation of the IOAPI 3.1 library. If the program has not been installed by the automated routine `installcc.csh`, the following needs to be done. Change to subdirectory `/preproc/bconcc2.1/cmaq4cc/`.

```
{your-citychem}/preproc/bconcc2.1$ cd cmaq4cc
```

In this folder you need to edit the Makefile. The following editing needs to be done in Makefile:

- Edit `NC_LIB` path, path of the netcdf4 installation on your computer, e.g.:

```
NC_LIB = /usr/local/netcdf4/
```

Then build **cmaq4cc.exe** by typing:

```
{your-citychem}/preproc/bconcc2.1/cmaq4cc$ make
```

Change to the next higher directory and create a symbolic link to the tool:

```
{your-citychem}/preproc/bconcc2.1/cmaq4cc$ cd ..
{your-citychem}/preproc/bconcc2.1$ ln -s ./cmaq4cc/cmaq4cc.exe
```

B) Edit run_cmaq4cc.csh

The script can be found in subdirectory `/preproc/bconcc2.1/`. Some items in the script have to be edited before use for a specific city domain. The following editing needs to be done in `run_cmaq4cc.csh`:

- Edit `INPUT_FILE1` variable by specifying the path m3bcon input files (created in step E of stage 1) where the last 7 chars are the date (format `yyyymmdd`), e.g.:

```
setenv INPUT_FILE1 /home/yourname/CMAQ/Yourcity/m3bcon/conc_${DATE}
```

- Edit INPUT_FILE2 variable by specifying the path m3meteo input files (created in step D of stage 1) where the last 7 chars are the date (format yyyyddd), e.g.:

```
setenv INPUT_FILE2 /home/yourname/CMAQ/Yourcity/m3meteo/metc_${DATE}
```

- Edit STDATE and ENDATE to set the start and end date (format: yyyyddd) for the processing:), e.g.:

```
set STDATE = 2012001
set ENDATE = 2012032
```

Consider using one extra day for the one-month BCON files, i.e. start data is first of current month and end date is first of next month.

C) Edit user metadata file

The run of **cmaq4cc.exe** is controlled by the user metadata file (cctapm_inp.dat). The user metadata file for the city simulation needs to be copied to the subdirectory /preproc/bconcc2.1/. The following edits have to be made in cctapm_inp.dat for use with **cmaq4cc.exe**:

Line 3:
edit Input path of CMAQ CONC.surface.yyyyddd files. Note: the path is currently not used, so the placeholder " " is sufficient.

Line 7:
Set output directory path (full path name). This will be the directory where the 3-BCON files are written to.

Line 8:
Edit name of log file.

Line 11:
Chose binary option for CityChem input (2).

Line 13:
Edit start of input string.

Line 14:
Edit end of input date string.

Line 15:
Edit start date.

Line 16:
Edit end date.

Line 17:
Edit number of cells in x-direction (*nx*)

Line18:
Edit number of cells in y-direction (*ny*)

D) Running cmaq4cc tool

After the changes in the script `run_cmaq4cc.csh` and in the user metadata file adapting them to a specific city and time period, the `cmaq4cc` tool can be run.

Note, that the output directory given in the user metadata file must be empty before running `cmaq4cc.csh`.

For running the script, type:

```
{your-citychem}/preproc/bconcc2.1$ ./run_cmaq4cc.csh
```

Figure 27 shows a screenshot of the successfully completion of the script for one day of boundary conditions. After completion of the run the 3-D BCON files (located in the output directory specified in `cctapm_inp.dat` have to be copied to the input directory for a city simulation with EPISODE-CityChem, subdirectory `INPUT/cmaq/`.

Note: The corresponding 3-D BCON of C_3H_6 will not be produced by using the `cmaq4cc` tool. It is recommended to make a copy of the BCON file for C_2H_4 .

```
matthias@matthias-Precision-Tower-7910: ~/CITYCHEM/citychem-1.2/preproc/bconcc2.0
matthias@matthias-Precision-Tower-7910:~/CITYCHEM/citychem-1.2/preproc/bconcc2.0$ ./run_cmaq4cc.csh

This program uses the EPA-AREAL/MCNC-EnvPgms/BAMS Models-3
I/O Applications Programming Interface, [I/O API] which is
built on top of the netCDF I/O library (Copyright 1993, 1996
University Corporation for Atmospheric Research/Unidata
Program) and the PVM parallel-programming library (from
Oak Ridge National Laboratory). Copyright (C) 1992-2002 MCNC
and Carlie J. Coats, Jr., and 2003-2006 Baron Advanced
Meteorological Systems, LLC and released under the GNU LGPL
License, version 2.1. See URL

http://www.baronams.com/products/loapl/LGPL.txt

for conditions of use.

Library release tag: $JDate: 2010068 $

$Id: @(#) loapl library version 3.1 $
Version with PARAMS3.EXT/PARAMETER:HXVARS3= 2048
netCDF version 4.6.1 of Mar 29 2018 12:51:17 $

EXECUTION_ID: Linux2_x86_64gfort

Value for PROMPTFLAG: N returning FALSE
Value for IOAPI_CHECK_HEADERS not defined;returning default: FALSE

"INPUT_FILE1" opened as OLD:READ-ONLY
File name "/home/matthias/CHAQ/Hamburg/n3bcon/conc_utm_cd04a_2012183"
File type GRDDED3
Execution ID "0001"
Grid name "SHEBA1AUTM"
Dimensions: 32 rows, 32 cols, 30 lays, 135 vbles
NetCDF ID: 65530 opened as READONLY
Starting date and time 2012183:000000 (0:00:00 July 1, 2012)
Timestep 010000 (1:00:00 hh:mm:ss)
Maximum current record number 25

"INPUT_FILE2" opened as OLD:READ-ONLY
File name "/home/matthias/CHAQ/Hamburg/n3meteo/metc_utm_cd04a_2012183"
File type GRDDED3
Execution ID "0001"
Grid name "SHEBA1AUTM"
Dimensions: 32 rows, 32 cols, 30 lays, 16 vbles
NetCDF ID: 131072 opened as READONLY
Starting date and time 2012183:000000 (0:00:00 July 1, 2012)
Timestep 010000 (1:00:00 hh:mm:ss)
Maximum current record number 25
Finished writing!
```

Figure 27: Screenshot of terminal window with output from the `cmaq4cc` tool.

7 Example data

With the example data included in the distribution it is possible to set up and configure your own simulation with EPISODE-CityChem. The example is the one-month simulation of the city of Hamburg in July 2013 with the run script file `citychem_hamburg_new2_201307.txt` provided in subdirectory `SIMU/`. In order to work with the example data you should first follow the steps described in section 2 “Getting Started”.

This practical example shows how to create input files for EPISODE-CityChem with the distributed pre-processing utilities.

The user input file `cctapm_meta.inp` in subdirectory `preproc/` is used to control the generation of input files.

First unpack the example data package (`cc_example_data_new_201307.tar.gz`; see section 2.1 and section 2.4).

The run script `citychem_hamburg_new2_201307.txt` for the example run is located in the subdirectory `SIMU/`.

A) *Unpacking the example data*

The example data package includes the TAPM output file and the BCON files for the example runs. After unpacking the example data package, the TAPM `*.outa` file can be found in subdirectory `testdata/`. The 3-D BCON files from CMAQ boundary concentrations for the Hamburg domain are located in the subdirectory `testdata/cmaq`. The files are ready for use with the model. A guidance for creating 3-D BCON files from CMAQ output files (`CONC*`) with the utility `BCONCC` is given in section 6.3.

B) *Starting with the meteorology input*

The utility `tapm4cc.exe` should be build according to section 3.4. With a symbolic link it becomes executable from subdirectory `preproc/`. If the link does not exist, issue the following:

```
{your-citychem}/preproc/$      ln -s ./tapm4cc2.0/bin/tapm4cc.exe
```

In the user input file, change output files path name to `../INPUT/tapm/` and the output format to ASCII (`ep_fmt = 1`). Then run `tapm4cc.exe` inside subdirectory `preproc/`. This will create the required meteorological input files in ASCII format for the simulation with EPISODE-CityChem in the subdirectory `INPUT/tapm/`.

C) *Preparing emission input data*

The utility `UECT` should be build according to section 6.2. With a symbolic link it becomes executable from subdirectory `preproc/`. If the link does not exist, issue the following:

```
{your-citychem}/preproc/$      ln -s ./uct2.2/bin/uct.exe
```

The *.csv files with the annual emission totals of point, line, and area emission sources for Hamburg are located in subdirectory preproc/uct/input/. Note that this is arbitrary emission data and should not be used in research publication.

Area emissions of residential heating (SNAP2) are generated with dependence on the daily average temperature (heat degree days method). For this procedure, the program **tapm4cc.exe** (section 3.4) has to be run using the to write netCDF output (*ep_fmt* = 3) and output files path name ‘./INPUT/tapm/’ in the user input file. This generates a netCDF file that contains air temperature at ground and vertical temperature gradient (T_and_dtdz_*.nc) in subdirectory INPUT/tapm/. The other generated netcdf files can be deleted. Move this netCDF file (from subdirectory INPUT/tapm/) to subdirectory testdata/ncfiles/. Then edit the user input file in preproc/ again, by choosing *ep_fmt* = 1 and providing the file path (./testdata/ncfiles/) in the line of *tapm_path*. Finally, for the area source emissions edit the user input file; output source type is ‘ASE’, number of area sources is 2397, and output files path name is ‘./INPUT/emis/’. Run **uct.exe** for the area sources. After that check the user log-file (by default “user_log.txt”). If the temperature correction of residential heating emissions was done properly the message will be shown in the user log-file: “UCT: Air temperature is used to calculate SNAP2 emissions”. Running uct.exe will create the emission input files for the simulation in the subdirectory INPUT/emis.

Now the point source emissions, edit the user input file: output source type is ‘PSE’ and number of point sources is 121, (./testdata/ncfiles/) in the line of *tapm_path*, output files path name is ‘./INPUT/emis/’. Then run **uct.exe** inside subdirectory preproc/. Continue with the line source emissions and edit the user input file as follows: output source type is ‘LSE’, number of line sources is 15850, and output files path name is ‘./INPUT/emis/’. Again run **uct.exe**.

Note: The ‘LSE’ run takes a long time to calculate and it is not possible to see the progress; please be patient. For the ‘ASE’ run the processing step for every day is shown in the terminal window.

D) Preparing auxiliary input data (first variant)

There are two variants to create the auxiliary input files. The first variant uses the topography file (topo.asc) generated by TAPM (see step B) and the land use information extracted with the TAPM GUI. The first variant is the most consistent procedure when TAPM has been used to generate the meteorology input. The tool **z0top4cc.exe** has to be built as described in section 3.7.2. With a symbolic link it becomes executable from subdirectory preproc/. If the link does not exist, issue the following:

```
{your-citychem}/preproc/$ ln -s ./auxiliary/bin/z0top4cc.exe
```

In the user input file, change output files path name to ‘./INPUT/other/’. Then run **z0top4cc.exe** inside subdirectory preproc/. It will read the file landuse.top located in subdirectory ‘./INPUT/other/’ and write the auxiliary input files landuse_episode.asc, surfrough_episode.asc, and clou_episode.asc.

E) Preparing auxiliary input data (second variant)

The second variant extracts topography with the AERMAP utility and uses static information for the other auxiliary files. AERMAP utility is first used to extract the topography of the study area and then AERMAP output is converted to an topo.asc file.

The utility AERMAP should be build according to section 3.7.2. With a symbolic link it becomes executable from subdirectory preproc/auxiliary/. If the link does not exist, issue the following:

```
{your-citychem}/preproc/auxiliary/$ ln -s ./bin/aermap.exe
```

The SRTM3 elevation data for Hamburg is already included in subdirectory preproc/auxiliary/srtm3/ in tif format (N53E009.tif and N53E010.tif). The AERMAP input file aermap.inp is already prepared for the study domain of Hamburg (30 km x 30 km area). Running AERMAP produces a terrain elevation output file (AERMAP.REC) in ASCII format. Other output files from AERMAP can be ignored (list file, plot file, save file):

```
{your-citychem}/preproc/auxiliary/$ ./aermap.exe
```

```
{your-citychem}/preproc/auxiliary/$ mv *.OUT *.out *.dbg ./output/
```

```
{your-citychem}/preproc/auxiliary/$ mv *.REC * ../INPUT/other/
```

Now, the tool **static4cc.exe** is used to convert AERMAP.REC. Before building **static4cc.exe**, it is possible to change the value of the surface roughness length for urban land use, z0 that can be found approximately in line 224 of aermap_reader.for in subdirectory preproc/auxiliary/src/ (the default value is 0.4 m).

After that **static4cc.exe** has to be built as described in section 3.7.2. With a symbolic link it becomes executable from subdirectory preproc/. If the link does not exist, issue the following:

```
{your-citychem}/preproc/$ ln -s ./auxiliary/bin/static4cc.exe
```

Before using the tool change the output file path in the user metadata file to ‘../INPUT/other’. Running **static4cc.exe** inside subdirectory preproc/ produces the auxiliary input files topo.asc, landuse_episode.asc, and surfrough_episode.asc. It does not produce the clou_episode.asc file. For getting the cloud fraction file, it is necessary to run the MCWIND utility with observation data as described in section 6.1. The other files that **static4cc.exe** produces are required as input for MCWIND.

F) Running CityChem

The run script file citychem_hamburg_201307_new.txt is provided in subdirectory SIMU/. The default configuration of the run script assumes that you

have used the first variant in *D*). [Change the file paths and names for the auxiliary files inside the run script if the second variant in *E*\) has been used.](#) To start the test simulation, open a terminal window and enter:

```
{your-citychem}/SIMU$ ./citychem.exe citychem_hamburg_new2_201207.txt
```

To continue the run the session on your computer has to remain open and the terminal window must not be closed. Main debugging information is written during runtime to the log file (in the example: CITYCHEM_Yourmachine_log.txt) in the subdirectory OUTPUT/.

G) Plotting model output

The plotting of model output with NCL (NCAR Command Language) scripts and an OpenAir script available in subdirectory postcchem/ is explained in section 5.

8 References

Appel, K. W., Pouliot, G. A., Simon, H., Sarwar, G., Pye, H. O. T., Napelenok, S. L., Akhtar, F., and Roselle, S. J. (2013) Evaluation of dust and trace metal estimates from the Community Multiscale Air Quality (CMAQ) model version 5.0, *Geosci. Model Dev.*, 6, 883-899, doi:10.5194/gmd-6-883-2013.

Berkowicz, R., Hertel, O., Larsen, S. E., Sørensen, N. N., and Nielsen, M. (1997): Modelling traffic pollution in streets, Ministry of Environment and Energy, National Environmental Research Institute, Roskilde, Denmark, available at: http://www2.dmu.dk/1_viden/2_Miljoe-tilstand/3_luft/4_spredningsmodeller/5_OSPM/5_description/ModellingTrafficPollution_report.pdf.

Byun, D. and Ching, J. (1999) Science Algorithms of the EPA Models-3 Community Multiscale Air Quality Modeling System. Technical report, U.S. Environmental Protection Agency, Office of Research and Development, Washington DC.

Byun, D. and Schere, K. (2006) Review of the Governing Equations, Computational Algorithms, and Other Components of the Models-3 Community Multiscale Air Quality (CMAQ) Modeling System, *Applied Mechanics Reviews* 59, 51-77.

Bøhler, T. and Sivertsen, B. (1998) A modern air quality management system used in Norway. Norwegian Institute for Air Research, Report NILU F 4/98, Kjeller, Norway.

Denby, B. R., Sundvor, I., Schneider, P., and Thanh, D. V. (2014) Air quality maps of NO₂ and PM₁₀ for the region including Stavanger, Sandnes, Randaberg and Sola. Norwegian Institute for Air Research, NILU TR 01/2014, Kjeller, Norway, 2014.

Hertel, O. and Berkowicz, R. (1989) Modelling Pollution from Traffic in a Street Canyon, Evaluation of Data and Model Development, National Environmental Research Institute, Roskilde, Denmark.

Hurley, P. J. (2008) TAPM V4. Part 1: Technical description. CSIRO Marine and Atmospheric Research Paper No. 25, Aspendale, Australia.

Karl, M., Castell, N., Simpson, D., Solberg, S., Starrfelt, J., Svendby, T., Walker, S.-E., and Wright, R. F. (2014) Uncertainties in assessing the environmental impact of amine emissions from a CO₂ capture plant, *Atmos. Chem. Phys.* 14, 8533-8577, www.atmos-chem-phys.net/14/8533/2014/.

Karl, M., Kukkonen, J. Keuken, M. P., Lützenkirchen, S., Pirjola, L., and Hussein, T. (2016) Modeling and measurements of urban aerosol processes on the neighborhood scale in Rotterdam, Oslo and Helsinki, *Atmos. Chem. Phys.* 16, 4817-4835, www.atmos-chem-phys.net/16/4817/2016/.

Karl, M. Walker, S.-E., Solberg, S., and Ramacher, M. O. P. (2019) The Eulerian urban dispersion model EPISODE. Part II: Extensions to the source dispersion and photochemistry for EPISODE-CityChem and its application to the city of Hamburg, *Geosci. Model Dev.*, <https://doi.org/10.5194/gmd-2018-325>, in review, 2019.

Petersen, W. B. (1980) User's guide for HIWAY-2: A highway air pollution model. U.S. Environmental Protection Agency, Report EPA-600/8-80-018, Research Triangle Park, NC.

Rodriguez, E., Morris, C. S., Belz, J. E., Chapin, E. C., Martin, J. M., Daffer, W., and Hensley, S. (2005) An assessment of the SRTM topographic products, Technical Report JPL D-31639, Jet Propulsion Laboratory, Pasadena, California, 143 pp.

Slørdal, L. H., Solberg, S., and Walker, S. E. (2003) The Urban Air Dispersion Model EPISODE applied in AirQUIS2003, Technical description, Norwegian Institute for Air Research, NILU TR 12/03, Kjeller, Norway.

Slørdal, L. H., McInnes, H., Krognnes, T. (2008): The Air Quality Information System AirQUIS. *Info. Techn. Environ. Eng.*, 1, 40-47.

Walker, S. E. and Grønskei, K. E. (1992) Spredningsberegninger for on-line overvåking i Grenland. Programbeskrivelse og brukerveiledning (in Norwegian). Norwegian Institute for Air Research, Report NILU OR 55/92, Kjeller, Norway.

U.S. EPA (2004) User's Guide for the AMS/EPA Regulatory Model - AERMOD. EPA-454/B-03-001. U.S. Environmental Protection Agency, Research Triangle Park, NC.

Appendix A

Meteorological input files

Table A1: Mandatory and optional input files for meteorology.

Filename (without suffix)	Variable	Required	TAPM	MCWIND
temp_episode or T_and_dtdz	Air temperature at ground and vertical temperature gradient	yes	yes	yes
Res_U_V_and_W wind	Wind u-, v-components	yes	yes	yes
sigmavw	Turbulence sigma-vw	no	no	no
aeroresist	Aerodynamic resistance	no	no	no
hmix	Mixing height	no	yes	no
ustar	Surface friction velocity scale	no	yes	no
sens_heatfl	Sensible heat flux	no	yes	no
evap_heatfl	Latent heat flux	no	yes	no
lanu	Land use category	yes	no	no
temp	3-D air temperature	no	yes	no
pot_T	Potential temperature	no	yes	no
3D_EPISODE_z_abg	TAPM 3D mid-layer geopotential height equivalent	no	yes	no
mflx_episode	Surface water vapour flux	no	no	no
sfws_episode	Surface momentum stress	no	no	no
ptstar	Potential temperature scale	no	yes	no
pvstar	Potential virtual temperature scale	no	yes	no
wstar	Convective velocity scale	no	yes	no
tsrad_episode or tot_solar_rad	Total solar radiation (global radiation)	no	yes	yes
T_surf	2D ground temperature	no	yes	no
prec_episode or prec	Precipitation	yes	yes	yes
rhum_episode or RH_screen	Relative humidity	yes	yes	yes
clou_episode	Cloud cover	yes	no	yes

Appendix B

Model output appendix

Table B1: Output files generated by CityChem

Filename in the run script	Content	Mandatory	File format
CITYCHEM_Yourmachine_log.txt	Log file with debug information from the simulation.	yes	ASCII
concmhour.nc	Instantaneous hourly 3-D model grid concentration output for all CityChem compounds. Uses horizontal resolution of the main domain grid.	no	netcdf4
concr_no2.asc	Hourly NO ₂ concentration at all stations and regular receptor grid points.	no	ASCII
concl_no2.asc	Hourly NO ₂ concentration at the receptor points associated with line sources.	no	ASCII
ddepr_no2.prn	Hourly dry deposition flux of NO ₂ at all stations and regular receptor grid points.	no	ASCII
wdepr_no2.prn	Hourly wet deposition flux of NO ₂ at all stations and regular receptor grid points.	no	ASCII
ddepl_no2.prn	Hourly dry deposition flux of NO ₂ at the receptor points associated with line sources.	no	ASCII
wdepl_no2.prn	Hourly wet deposition flux of NO ₂ at the receptor points associated with line sources.	no	ASCII
statmainhour.nc	Hourly 2-D model grid ground concentration output for all CityChem compounds.	no	netcdf4
statrecphour.nc	Hourly 2-D concentration output from the regular receptor grid for all CityChem compounds.	no	netcdf4
statmonihour.nc	Hourly concentration output at monitoring station points for all CityChem compounds.	no	netcdf4

Table B1 continued.

Filename in the run script	Content	Mandatory	File format
statclaved_no2.txt	Daily average NO ₂ concentration at the receptor points associated with line sources.	yes	ASCII
statmain.nc	2-D Model grid concentration averaged over the simulation period for all CityChem compounds.	no	netcdf4
statrecp.nc	2-D Concentration averaged over the simulation period from the regular receptor grid for all CityChem compounds.	no	netcdf4
statmoni.nc	Concentration averaged over the simulation period at monitoring stations for all CityChem compounds.	no	netcdf4
photmhour.nc	Hourly 2-D model grid surface fields of radical concentrations, selected reaction rate coefficients, photolysis frequencies, and meteorological variables.	no	netcdf4
asrcmhour.nc	Hourly total area emissions (sum of all source categories) as 3-D field for all CityChem compounds.	no	netcdf4
icmhour_YYYYMMDD.nc	3-D model grid concentration output for all CityChem compounds of the last simulation hour. The file can be used as initial concentration field for the next simulation period.	if restart option is set to '1'	netcdf4
plume_segments.dat	Hourly plume segments and plume parameter values.	yes	ASCII

Appendix C

Installation of NetCDF

This is a documentation for the installation of the NetCDF libraries on Ubuntu Linux. The installation on other Linux systems is similar, where probably different package managers (https://en.wikipedia.org/wiki/Package_manager) have to be used. The step-by-step NetCDF installation instruction below is partly based on the tutorial published on the Tux Coder Blog (<https://tuxcoder.wordpress.com/2015/02/02/install-netcdf4-with-hdf5-in-ubuntu-linux>). The below instruction will install Netcdf4 along with Hdf5 on Ubuntu Linux 64-bit. For the installation described here, the following files were downloaded:

```
zlib-1.2.8.tar.gz
gzip-2.1.tar.gz
hdf5-1.8.17.tar
netcdf-4.3.0.tar.gz
netcdf-fortran-4.2.tar.gz
```

There might be newer versions of some of the libraries which can also be used. For netcdf-C and netcdf-fortran libraries always the latest release versions should be used.

If the gfortran/gcc Compiler is not installed on your computer this has to be done before installing netCDF:

```
{/your-ubuntu-home}/$ sudo apt-get update
```

```
{/your-ubuntu-home}/$ sudo apt-get install gfortran -y
```

For the installation of the netCDF libraries it is recommended to first create a directory 'netcdf4' in your home directory (named 'your-ubuntu-home' in the following). All downloaded packages should be saved to the 'netcdf4' directory.

A) Install the zlib library

Download latest source of zlib from here <http://www.zlib.net/> (e.g. zlib-1.2.8.tar.gz) and extract it to /usr/local/. Rename the extracted directory to 'zlib'. Change to the new directory /usr/local/zlib/.

```
{/usr/local/zlib}/$ ./configure
```

```
{/usr/local/zlib}/$ make
```

```
{/usr/local/zlib}/$ sudo make install
```

B) Install the szip library

Download szip from <http://www.hdfgroup.org/ftp/lib-external/szip/2.1/src/szip-2.1.tar.gz> and extract it to your home into the ‘netcdf4’ directory. Change to the new directory.

```
{/your-ubuntu-home/netcdf4/szip-2.x.x}/$ ./configure --prefix=/usr/local/szip
```

```
{/your-ubuntu-home/netcdf4/szip-2.x.x}/$ make
```

```
{/your-ubuntu-home/netcdf4/szip-2.x.x}/$ sudo make install
```

C) Install the hdf5 library

In the configure of hdf5, -with-pthread option must point to the directory that contains the POSIX Threads library install ([DIR]/includes/pthread.h, [DIR]/lib/libpthread*).

```
{/your-ubuntu-home}/$ locate pthread.h -> /usr/include/
```

```
{/your-ubuntu-home}/$ locate libpthread.a -> /usr/lib/x86_64-linux-gnu/
```

```
{/your-ubuntu-home}/$ locate libpthread.so -> /usr/lib/x86_64-linux-gnu/
```

Download hdf5 latest source from <http://www.hdfgroup.org/HDF5/release/obtainsrc.html>. Extract the tar package to your home into the ‘netcdf4’ directory. Change to the new directory.

```
{/your-ubuntu-home/netcdf4/hdf5-1.8.x}/$ ./configure --prefix=/usr/local/hdf5 --enable-fortran --enable-cxx --with-szlib=/usr/local/szip --enable-threadsafe --with-pthread=/usr/include/,/usr/lib/x86_64-linux-gnu/ --enable-hl --enable-shared --enable-unsupported
```

```
{/your-ubuntu-home/netcdf4/hdf5-1.8.x}/$ make
```

```
{/your-ubuntu-home/netcdf4/hdf5-1.8.x}/$ sudo make check install
```

D) Install the netCDF-C library

Download latest netCDF-C from <http://www.unidata.ucar.edu/downloads/netcdf/index.jsp>. It is found in the section “The Latest Stable netCDF-C Release, tar.gz form”. It is recommended to download the latest stable version of netCDF-C.

Extract the package (e.g. netcdf-4.4.0.tar.gz) to your home into the ‘netcdf4’ directory and change to the new directory.

```
{/your-ubuntu-home/netcdf4/netcdf-4.x.x}/$ ./configure --prefix=/usr/local/netcdf4 --enable-netcdf-4 --enable-shared --disable-dap CPPFLAGS="-I/usr/local/szip/include -I/usr/local/hdf5/include" LDFLAGS="-L/usr/local/szip/lib -L/usr/local/hdf5/lib"
```

```
{/your-ubuntu-home/netcdf4/netcdf-4.x.x}/$ make
```

```
{/your-ubuntu-home/netcdf4/netcdf-4.x.x}/$ make check
```

The output should look like this:

```
=====
Testsuite summary for netCDF 4.x.x
=====
# TOTAL: 2
# PASS: 2
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
```

If the test is passed without errors, type:

```
{/usr/local/netcdf4/netcdf-4.x}/$ sudo make install
```

E) Install the netCDF Fortran library

Download latest netCDF-Fortran at <http://www.unidata.ucar.edu/downloads/netcdf/index.jsp>. It is found in the section “The Latest Stable netCDF-Fortran Release, tar.gz form”. Extract the package (e.g. netcdf-fortran-4.2.tar.gz) to your home into the ‘netcdf4’ directory and change to the new directory.

```
{/your-ubuntu-home/netcdf4/netcdf-fortran-4.x}/$ ./configure --prefix=/usr/local/netcdf4 --enable-shared CPPFLAGS="-I/usr/local/zip/include -I/usr/local/hdf5/include -I/usr/local/netcdf4/include" LDFLAGS="-L/usr/local/zip/lib -L/usr/local/hdf5/lib -L/usr/local/netcdf4/lib"
```

```
{/your-ubuntu-home/netcdf4/netcdf-fortran-4.x}/$ make
```

```
{/your-ubuntu-home/netcdf4/netcdf-fortran-4.x}/$ make check
```

```
{/your-ubuntu-home/netcdf4/netcdf-fortran-4.x}/$ sudo make install
```

F) Adjust your .profile file

Change to your home directory on Ubuntu Linux and the .profile file (or .bash_profile):

```
{/your-ubuntu-home}/$ emacs .profile &
```

The LD_LIBRARY_PATH has to be changed. Place the following at the end of your .profile file:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib:/usr/local/netcdf4/lib
export LD_LIBRARY_PATH
```

Save and exit emacs.

Restart the computer.

The netCDF library should now be installed on your Ubuntu Linux computer.

