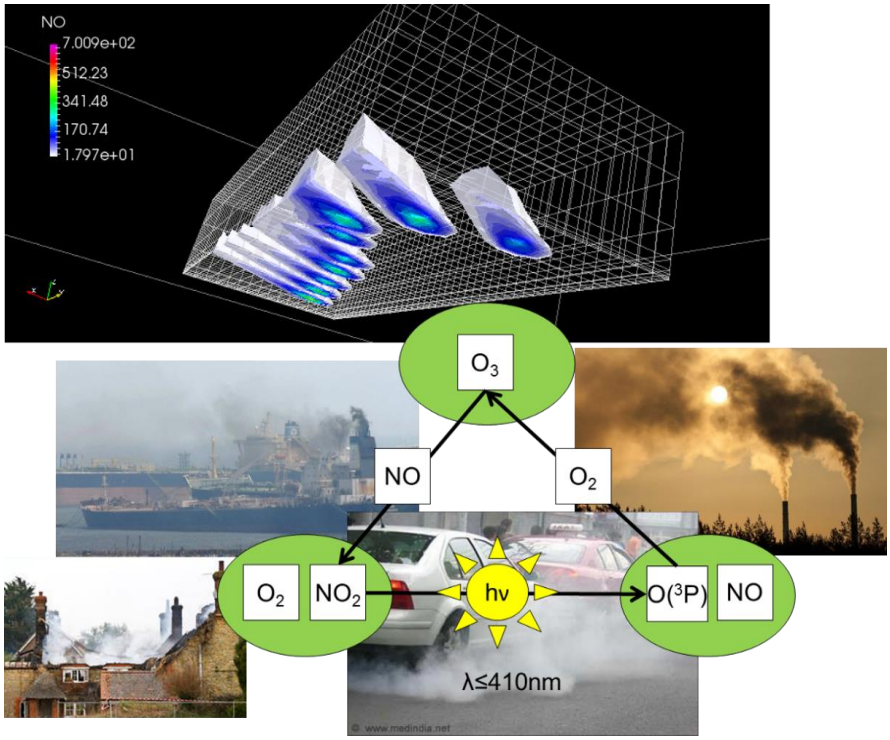
User's Guide

User's Guide EPISODE-CityChem version 1.8
City-scale Chemistry Transport Model

Matthias Karl[1] and Martin Ramacher[1]

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



Program manual

Helmholtz-Zentrum hereon

Preface

The CityChem extension to the EPISODE dispersion model has been developed by Dr. Matthias Karl (Helmholtz-Zentrum Hereon, Geesthacht). 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 Hereon.

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).

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, (4) 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.

For the setup of a simulation with EPISODE-CityChem, meteorological data and boundary concentration data is needed. Meteorological data can be either 1) hourly measurements, 2) output of the TAPM meteorological model, 3) output of a WRF simulation. Boundary concentration data (regional background concentrations) can be either 1) constant values, 2) hourly measurements, 3) output of the CMAQ regional AQ model. A fourth option for boundary concentrations is the CAMS regional ensemble reanalysis data. Contact the authors of this User's Guide in case you want to use CAMS reanalysis data.

EPISODE-CityChem has been applied for simulation of the air quality in Hamburg, Kiel, Rostock, Essen/Düsseldorf, Rotterdam, Liege, Gdansk, Riga, Marseille, Athens, and Singapore.

Contributions

Martin Ramacher, PhD (Hereon), 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. BSc Josefine Feldner (Leuphana Univ., Lüneburg) contributed an R script for the additional evaluation of model results with openair. BSc Marvin Lauenburg (Leuphana Univ., Lüneburg) helped to update the inventory of point source emissions for the Hamburg example.

Thanks to the attention of new model users, Eleni Athanasopoulou, PhD, Dimitris Karagiannis, MSc (both NOA, Athens), and Kang Pan, PhD (CARES, Cambridge) several minor issues in the model and the pre-processing utilities could be fixed.

Ronny Badeke, PhD (Hereon) contributed the vertical distribution function for area sources of ships. Stelios Myriokefalitakis, PhD (NOA, Athens) implemented the openMP parallelization of the Eulerian grid processes in EPISODE-CityChem.

Contents

Page

User's Guide for EPISODE-CityChem version 1.5

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 ($\leq$ 1x1 km^2) and simultaneously local ground air concentrations with high resolution (100x100 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 Hereon. 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 (O_3), nitrogen oxide (NO) and nitrogen dioxide (NO_2).

Aerosol particles can be treated in terms of primary emitted particulate matter (PM) and size-binned particle number (PN). Particulate matter is separated in $PM_{2.5}$ (PM with aerodynamic diameter $\leq$ 2.5 μm) and PM_{10} (PM with aerodynamic diameter $\leq$ 10 μm). Secondary particle formation is now implemented in EPISODE-CityChem. Starting with version 1.7 it is possible to calculate the formation of secondary organic aerosol (SOA). It is planned to include the formation of secondary inorganic aerosol (SIA) in the next version.

An important note to give in the beginning is that there are two different types of time steps in the model, first the simulation time step which is set to one hour and second the dynamical time step from the operator splitting, often referred to as “advection time step”. During one simulation time step (i.e. 1 hour), the emissions and meteorology are not changing. There is input of emissions and meteorology every new hour. The dynamical time step on the other hand is a integer fraction of the simulation time step (for example 10 minutes) which is calculated in the beginning of the simulation considering the time scales of the various processes on the Eulerian grid. The dispersion from line sources is treated with a Gaussian model and the point sources with a segmented plume model that behaves like a plume-in-grid.

EPISODE-CityChem has a restart option that allows to continue after a previous run using the state (concentrations, plume segments) from the end the last simulation hour as initial conditions for the continuation run. It is possible to change the position and emission rate of the emitting objects between the previous run and the restart run.

Embedded sub-grid models for near-source dispersion

The model considers three categories of emissions: area source emissions, line source emissions and point source emissions.

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 divers 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 ($C_{scdir,s}$) from the traffic plume plus a contribution from the recirculation of the traffic plume ($C_{screc,s}$) due to the vortex inside the canyon:

$$C_{line,s} = C_{scdir,s} + C_{screc,s} \qquad (1)$$

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.

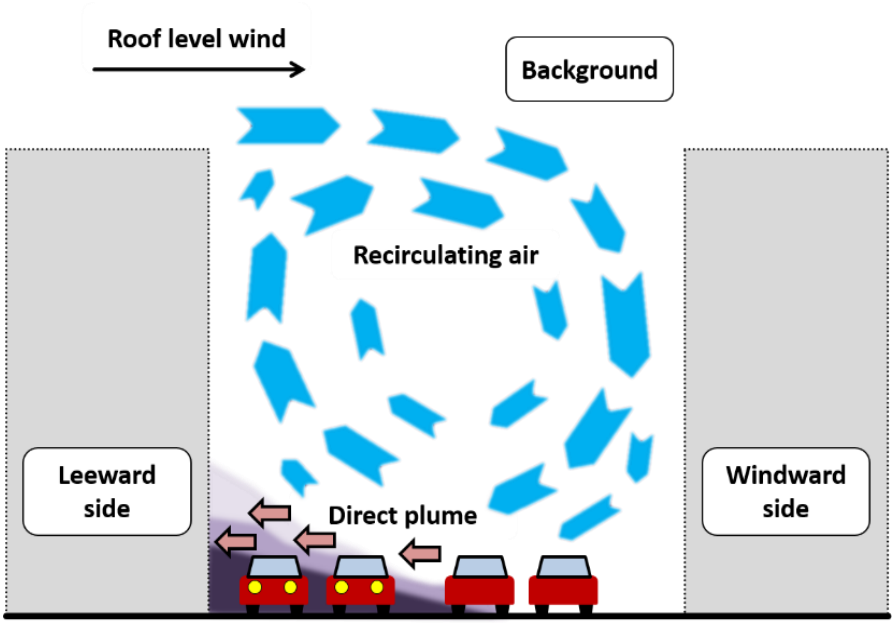


Figure 1: Illustration of the street canyon box model. In the figure, the recirculation zone extends through the whole street canyon, albeit the model can treat situations where this is not the case.

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. Each emitted plume segment has its own life cycle: release, growth, redirection, dissipation. The position, size, and pollutant mass content of each segment is recorded in one output file for every simulation hour. One point source can have several belonging segments, which can be in different grid cells and vertical layers; all of them form the entire plume. The contribution of the plume segments to the receptor points at the surface follow the assumption of a steady-state plume. Due to plume rise, the plume segments usually travel in the higher vertical levels.

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 (dynamical) 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 x 100 m^2, covering the entire model domain.

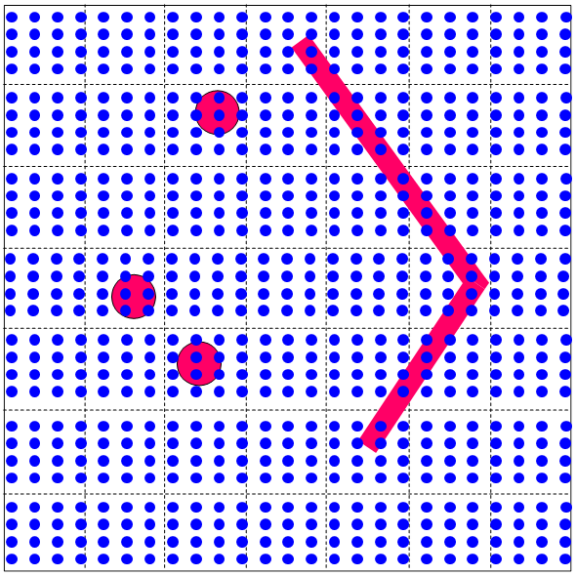


Figure 2: Illustration of the main grid and the regular receptor grid in EPISODE-CityChem. The black dotted lines represent the gridlines of the main grid while the blue dots represent the regular receptor grid. The objects in red are point source and line source objects, for which sub-grid dispersion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

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).

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:

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

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 line sources constantly emit into the Eulerian grid during one simulation time step. The receptor point concentration can still be described by Eq. (2). However, C_m (background concentration of the grid cell) is coming from the previous dynamical time step, i.e. not from the previous simulation time step. Therefore, the emitted mass from the line source is not accumulated over one hour but instead taken instantaneously from the current dynamical time step (as $\mathrm{d}C/\mathrm{d}t$). Note that the concentrations at the receptors, C_r, do not feed back into the grid cell during the simulation. Note again that there is no change in the wind field and in the line emissions during the simulation hour. Since C_m is taken from every previous dynamical time step, C_m in Eq. (2) considers the emission of the line source that has been emitted during the previous dynamical time step, but not the emission of the actual time step.

Chemistry options (Eulerian grid)

For the photochemistry on the Eulerian grid, four different options exist. The chemistry option can be selected in the run script (section 4.9). The options are:

1. Running as tracer model: all air pollutants are only dispersed as (chemically inert) passive tracers. Option “0” in the run script.
2. Simple photo-stationary equilibrium between O_3, NO_2 and NO, representing the photo-stationary state (PSS) simulation. Option “1” for grid chemistry in the run script and in addition option”1” for the local photochemistry (section 4.8). Note that with this option, PSS is only applied to the concentration field at the surface level.
3. Explicitly solving the photochemical reaction cycle between O_3, NO_2 and NO with the two-step solver. Option “2” in the run script.
4. Urban chemistry scheme EMEP45 developed at NILU (*Walker et al.*, 2003). The EMEP45 scheme was updated in recent years at Helmholtz-Zentrum Geesthacht for use in CityChem. The resulting scheme is called **EmChem03-mod** and consists of 45 gas-phase species, 51 thermal reactions and 16 photolysis reactions. The reaction scheme is listed in the Supplement of *Karl et al.* (2019). Option “3” in the run script.
5. Urban chemistry scheme EmChem09-mod. the EmChem09-mod scheme was developed based on the current EMEP chemistry mechanism,

EmChem09 (*Simpson et al.*, 2012). The reaction scheme is listed in the Supplement of *Karl et al.* (2019). EmChem09-mod includes reactions between organic peroxy radicals and the hydroperoxyl radical (HO_2) as well as other organic peroxy radicals; it is therefore appropriate for low NOx conditions in rural and suburban areas of the city domain. With EmChem09-mod the chemistry of biogenic volatile organic compounds (BVOCs), emitted from urban vegetation, can be simulated. EmChem09-mod was replaced by **EmChem09-HET** in EPISODE-CityChem v1.3, which also includes heterogeneous gas-phase reactions. Option "4" in the run script.

The urban chemistry scheme EmChem09-HET for the Eulerian grid will be further developed and extended. In EPISODE-CityChem v1.3, a light-dependent ground surface source of nitrous acid (HONO) was included, due to its importance as source of hydroxyl radicals (OH) in the urban atmosphere (*Vogel et al.*, 2003). In EPISODE-CityChem v1.4, additional heterogeneous reactions that occur on the surfaces of aerosol were included and the photo-oxidation of isoprene was completely revised based on a modified version of the Mainz Isoprene Mechanism (MIM, *Pöschl et al.*, 2000) published by *Karl et al.* (2006). EmChem09-HET now includes 80 chemical species and 78 chemical reactions and 28 photodissociation reactions.

Chemistry options (receptor grid)

For the local photochemistry on the receptor grid, three options are available:

1. Running as tracer model: all air pollutants are only dispersed as (chemically inert) passive tracers. Option "0" in the run script.
2. Approximation of the photostationary steady-state (PSSA) for the photochemical reaction cycle between O_3, NO_2 and NO. Option "1" in the run script.
3. **EP10-Plume** (*Karl et al.*, 2019), a small reaction scheme for use in the Gaussian models for the line and point sources. Option "3" in the run script.

EP10-Plume includes reactions of O_3, NO, NO_2, nitrous acid (HNO_3), carbon monoxide (CO) and formaldehyde (HCHO). EP10-Plume was shown to give very similar results as PSSA for O_3, NO and NO_2 concentrations up to a distance of 300 m from a line source (*Karl et al.*, 2019). In EPISODE-CityChem v1.5, the EP10-Plume scheme was modified based on the OSPM chemistry module (*Berkowicz et al.*, 1997) for the street canyons. The rate of change of the concentrations of NO, NO_2 and O_3 in a street canyon are extended by terms for direct emissions and background concentration:

$$\begin{aligned}\frac{d[NO_2]}{dt} &= -R_C + \frac{[NO_2]_V}{\tau} + \frac{[NO_2]_B - [NO_2]}{\tau} \\ \frac{d[NO]}{dt} &= -R_C + \frac{[NO]_V}{\tau} + \frac{[NO]_B - [NO]}{\tau} \\ \frac{d[O_3]}{dt} &= -R_C + \frac{[O_3]_B - [O_3]}{\tau}\end{aligned} \qquad (3)$$

The term R_C in Eq. (3) describes the sinks due to chemical reactions; the second term describes the effect of emissions (index V , no direct emissions for ozone). The last term describes the mixing as a function of concentration differences between the background station (index B) and the receptor point. Mixing processes and other sinks are parameterized via the mixing time, corresponding to the time constant τ, the residence time of pollutants in the street canyon. *Düring et al.* (2011) give values for the mixing time: $\tau = 100\ s$ (street canyons), $\tau = 40\ s$ (free dispersion). Different to the OSPM chemistry model, other reactions of the EP10-Plume scheme are also considered.

The heterogeneous reactions and isoprene reactions of EmChem09-het and the reactions of EP10-Plume are listed in Appendix A.

TUV photolysis module

In EPISODE-CityChem v1.8, the chemistry schemes for the Eulerian grid, EmChem03-mod and EmChem09-HET, were extended with the Tropospheric Ultraviolet and Visible (TUV) Radiation model (Lee-Taylor and Madronich, 2002) for calculation of photolysis rates. The implementation of the TUV module in EPISODE-CityChem is based on the standalone model code of TUV v5.4 (https://www2.acom.ucar.edu/modeling/tropospheric-ultraviolet-and-visible-tuv-radiation-model). Using the 2-Stream Delta Eddington radiative transfer scheme, the TUV module calculates actinic flux and photolysis rates in each vertical layer within a 1-D column of the Eulerian grid. The TUV module has its own set of cross-sections and quantum yields, which are mainly based on the JPL 2011 recommendation (Sander etal., 2011). The implemented wavelength grid ranges from 170-735 nm with 138 wavelength bins; out of 113 available photo-dissociation reactions of TUV, 31 photo-dissociation reactions were selected for the implemented TUV module, covering all relevant photolysis rates in the lower troposphere. Figure 3 shows photolysis rates of $O_3 \longrightarrow O_2 + O(^1D)$ and $NO_2 \longrightarrow NO + O(^3P)$ as function of zenith angle at surface.

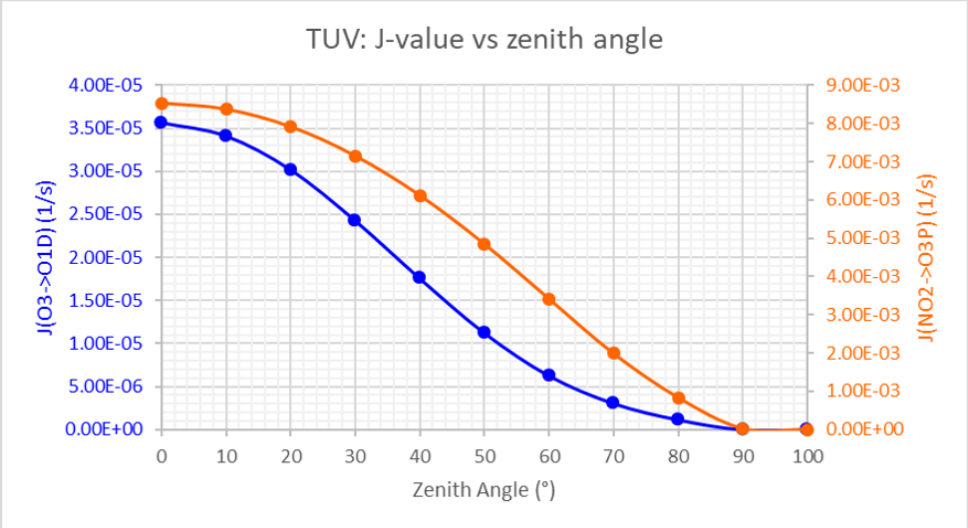


Figure 3: Photolysis rates $j(O_3)$ and $j(NO_2)$ as function of zenith angle at surface calculated with the TUV module.

EP10-Plume still uses the empirical schem for calculation of a reduced set of photolysis rates, listed in Appendix A.

Particle number size classes

In EPISODE-CityChem v1.5, the P8P parameterization scheme for particle number concentration calculation was implemented, based on the earlier parameterization scheme with three size classes developed by *Karl et al.* (2016). In the P8P scheme, eight particle size classes are defined, covering the particle diameter size range between 1 nm and 10 μm. For each size class, dry deposition and coagulation (between particles of the same size class) are considered, while coagulation between particles of different size categories is not taken into account. However, inter-modal coagulation is partly taken into account through the average coagulation coefficient derived from an aerosol dynamics model calculation that included coagulation between all size sections. The P8P scheme is adequate for implementation in the Eulerian grid model and the Gaussian (sub-grid) models. In EPISODE-CityChem v1.6, the particle number schemes was expanded by two more size classes. The properties of the new particle number scheme P8P+2 is shown in Table 1.

Table 1: Particle number scheme P8P+2 (in EPISODE-CityChem v1.6).

PN size class	Lower diameter (nm)	Upper diameter (nm)	Mean diameter (nm)	Band width	Dry deposition (cm/s)	Coagulation constant (cm^3/s)
PN1	1.5	4	2	1.3	1.000	6.30×10^{-9}
PN2	4	9	7	1.3	0.528	4.51×10^{-9}
PN3	9	21	17	1.5	0.355	9.76×10^{-9}
PN4	21	50	41	1.6	0.181	1.50×10^{-8}
PN5	50	120	98	1.6	0.068	5.40×10^{-9}
PN6	120	290	232	1.8	0.039	6.26×10^{-9}
PN7	290	670	551	1.8	0.031	4.27×10^{-9}
PN8	670	1000	850	1.8	0.023	2.28×10^{-9}
PN9	1000	2500	1500	2.0	0.110	8.69×10^{-10}
PN10	2500	10000	4000	2.0	0.200	8.00×10^{-10}

The accuracy of the implemented parameterization for aerosol processes for prediction of PN concentrations is limited by three factors: first, by the averaging of process parameters over a certain size range; second by the simplified treatment of coagulation; and third by neglecting condensation and evaporation. Compared to the fully size-resolved aerosol dynamics model, calculated total particle number (PN) concentrations have an error of approximately 10 % (*Karl et al.*, 2016). Depending on the background concentrations, fresh emissions and meteorological parameters, condensation might be relevant especially for sub-10 nm particles, causing a larger deviation from the accurate solution.

Aerosol modules

Aerosol modules for secondary formation of aerosols are currently implemented in EPISODE-CityChem. For the secondary aerosol formation on the Eulerian grid, three options will be available:

1. Running without aerosol module for secondary aerosol formation. Option "0" in the run script.
2. Secondary organic aerosol formation with MAFOR-SOA. Option "1" in the run script. Available since EPISODE-CityChem v1.7.
3. Secondary inorganic aerosol formation with ISORROPIA. Option "3" in the run script. This option will be available in the next version.

SOA formation with Option "1" requires that the particle scheme P8P+2 is activated by providing 10 particle size classes. In addition, the 6 tracers for SOA compounds need to be provided. For SOA formation, the aerosol dynamics solver for condensation/evaporation from MAFOR v2 (*Karl et al.*, 2022) was incorporated in EPISODE-CityChem v1.7. MAFOR v2 is a sectional aerosol model that solves the time evolution of the particle number and mass concentration distribution using the fixed sectional method. The Analytical Predictor of Condensation scheme (*Jacobson*, 2005) is employed to calculate the mass transfer of gas molecules to particles. The functional structure of the model with the two new components is illustrated in Figure 4.

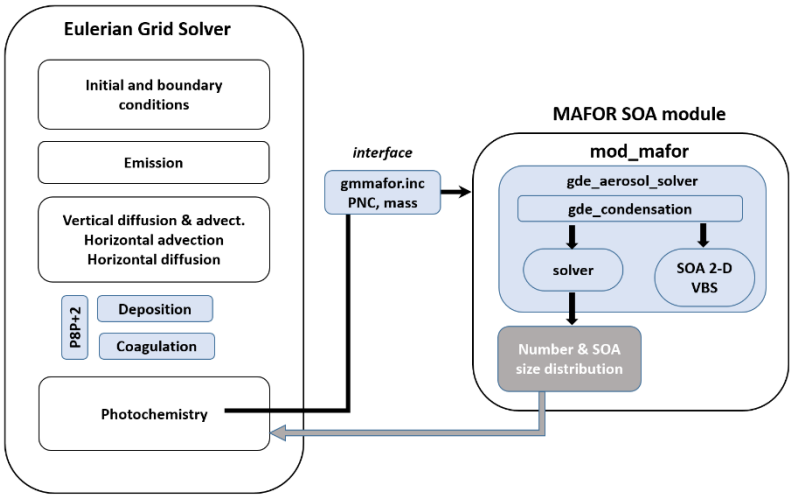


Figure 4: Block structure of the Eulerian grid solver with P8P+2 and the MAFOR-SOA module in EPISODE-CityChem. The new modules are in blue shaded blocks.

SOA formation is computed on the Eulerian grid in the following sequence. The grid photochemistry calls the gde_aerosol_solver of the MAFOR-SOA module after the gas-phase chemistry is solved and updated concentrations of SOA precursor gases and sulfuric acid are available. The SOA mass concentrations are distributed over the 8 size bins of P8P+2 (excluding classes 9 and 10) with a

default distribution in the interface gmmafor.inc before the call to gde_aerosol_solver. The routine soapartition (SOA 2-D VBS) is called in gde_condensation after calculation of fraction of absorptive material, O:C ratio and saturation pressure. Routine soapartition then calculates the activity coefficients in the non-ideal organic solution and returns the effective saturation concentration of SOA compounds. The numerical solution to SOA formation is non-iterative. The bin-wise masses are summed after the calculations in gde_aerosol_solver to give total mass per SOA compound.

The MAFOR-SOA module relies on the 2-D VBS framework (*Donahue et al.*, 2011) that uses the carbon oxidation state and the saturation concentration of the pure compound to define the organic aerosol composition in a two-dimensional space. A hybrid approach of condensation/evaporation and the absorptive partitioning into an organic liquid is used to treat condensation to an organic mixture considering non-ideal solution behavior of the organic mixture. Biogenic secondary, anthropogenic secondary (aromatic and aliphatic VOC) and primary volatile organics are represented by two SOA compounds each. Extremely low volatile SOA precursors are not considered yet to limit the number of SOA tracers.

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**
wrf4cc	**Interface to convert WRF meteorology output**
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 5 provides an overview of the components of the model system:

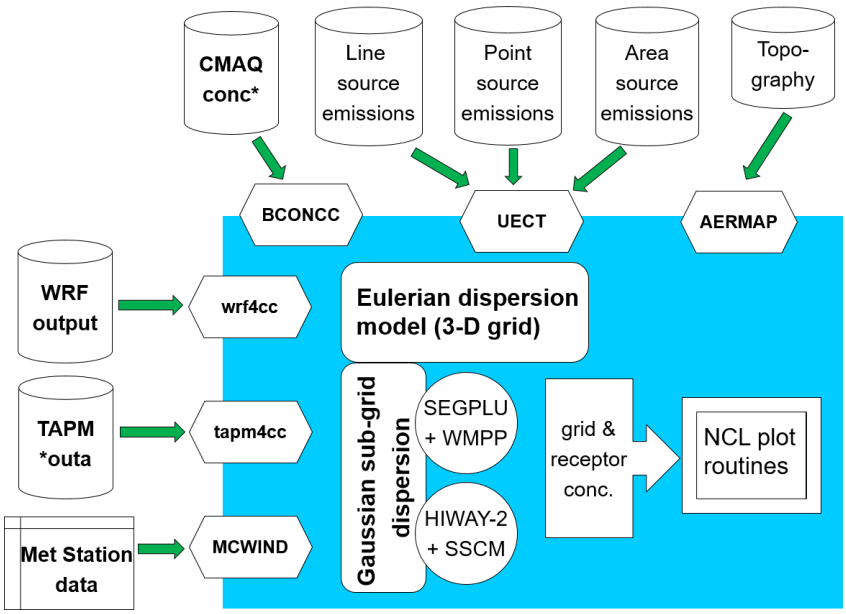


Figure 5:Overview of EPISODE-CityChem model components: utilities and interfaces.

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:
A public ftp address for download of the packages is made available on request by email to Matthias.Karl *at* hereon.de.

citychem-1.8.tar.gz
Contains model source code and pre-processing utilities and graphic scripts
[185 MB].

cc_example_data_new5_201307.tar.gz
Contains example data for simulation of Hamburg model domain
[388 MB].

Distribution license.
The CityChem extension package is distributed under the Open Reciprocal License, in sub-license of Hereon 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 (Hereon) 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.

Karl, M. Walker, S.-E., Solberg, S., and Ramacher, M. O. P. (2019) The Eulerian urban dispersion model EPISODE - Part 2: Extensions to the source dispersion and photochemistry for EPISODE-CityChem v1.2 and its application to the city of Hamburg, Geosci. Model Dev., 12, 3357-3399, https://doi.org/10.5194/gmd-12-3357-2019.

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 D.
- Three options exist for the preparation of meteorological input to the model. The first options requires the installation of the air pollution model TAPM on a Windows PC (see section 2.3). The second option is using output of the Weather Research and Forecasting WRF model (http://www.wrf-model.org/), see section 3.4.2. The third 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 or hourly varying 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. 3.7.3).

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-1.8 in your home directory (i.e. /home/user/), {your-home}, by typing in a terminal. Change to the new directory (i.e. /home/user/citychem-1.8/) and unpack the delivered tar-files into the new directory

```
{your-home}$ mkdir citychem-1.8
{your-home}$ cd citychem-1.8
{your-home}/citychem-1.8$ tar –zxvf citychem-1.8.tar.gz
{your-home}/citychem-1.8$ tar –zxvf cc_example_data_new4_201307.tar.gz
```

From now on the directory where CityChem is installed (i.e. **/home/user/citychem-1.8/)** will be referred to as {your-citychem}. Table 2 shows the directory structure that will be created in {your-citychem}.

Table 2: Subdirectories in {your-citychem} created by unpacking the CityChem download.

Name of subdirectory	Content
bin/	Location of the executable program
docu/	User's Guide document
GRAFICS/	Graphical plots of the output
testdata	Input files of background concentrations at the domain boundaries, produced from CMAQ model. TAPM *.outa file. From unpacking the Example Data tar package.
INPUT/emis	Input files of emission data
INPUT/tapm	Input files of meteorological variables produced from TAPM model
INPUT/other	Auxiliary static input files
INPUT/mcwind	Input files of meteorological variables produced from MCWIND (empty directory)
INPUT/wrf	Input files of meteorological variables produced from WRF model output (empty directory)
license/	License files
OUTPUT/	Output files generated by CityChem
postcchem/	Scripts for the visualization of CityChem output
preproc/	CityChem user input script (meta data) and utilities for preparation of input files
preproc/bconcc2.5	Scripts for preparation of chemical boundary conditions based on CMAQ output
preproc/uect2.8	UECT program (emission conversion tool)
preproc/mcwind1.3	MCWIND program (diagnostic wind field model)
preproc/tapm4cc2.2	Scripts for preparation of meteorological input based on TAPM output
preproc/wrf4cc1.3	Utility for preparation of meteorological input based on WRF output
preproc/auxiliary	Scripts for preparation of auxiliary input data
SIMU/	CityChem run scripts
src/	Source code of CityChem

2.5 Installation of EPISODE-CityChem

The installation is 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. A config.user template for compilation with either gfortran or Intel® Fortran is provided.

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 3.2 binary (default: Linux2_x86_64gfort)**
IOAPIDIR	**Full path of IOAPI 3.2 installation (IOAPI is included in the distribution)**

The IOAPIDIR path is normally the subdirectory preproc/bconcc2.5/ioapi3.2/ included in the distribution, unless you want to use a different or previously built IOAPI3.2 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").
Choose "bcon" for installation of the pre-processing utilities ("util") and for the installation of citychem, if you want to use a previous installation of IOAPI exists.
Choose "example" if the IOAPI library is not installed yet.

For "example" use:

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

For the self-made "bcon" files use:

./installcc.csh city bcon	**Install CityChem (normal executable)**
./installcc.csh cityopt bcon	**Install CityChem with high speed**
./installcc.csh util bcon	**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.

./installcc.csh cleanutil	**Remove utilities**
./installcc.csh cleancity	**Remove CityChem**
./installcc.csh cleanioapi	**Remove IOAPI library**
./installcc.csh cleandata	**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-1.8/):

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

And run the installation routine for CityChem as high speed executable:

```
{your-citychem}/$ ./installcc.csh cityopt 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 D) 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 running the EPISODE-CityChem model, it is recommended to use the advanced compilation option (cityopt). 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.

For parallelization with OpenMP, the OPENMP option has to be defined. For non-parallelized run it can simply be left empty.

```
# No parallelization
#OPENMP =
# OpenMP with gfortran
#OPENMP = -fopenmp
# OpenMP with intel fortran
OPENMP = -qopenmp
```

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-1.8/):

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

2.6 Running EPISODE-CityChem

All required information about physical and chemical options, input and output for an 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 B1 in Appendix B 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 3 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.

Table 3: 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

The EPISODE-CityChem model provides output for a certain number of chemical compounds (see Table 4), from here on called CityChem compounds. Parameter nc is the total number of CityChem compounds (by default **nc = 49**). Only for CityChem compounds, emission input can be provided. It is further possible to specify deposition velocities and wet scavenging coefficients for these compounds.
Volatile organic vapors that can partition to particles are given at index 25-30. These group of gases is involved in the chemical reaction of mechanism EmChem09-het. Their counterparts are the SOA compounds at index 31-36. All listed CityChem compounds with index 31 or higher are not participating in the chemistry. It is practical to list the particle number tracers at the end.

In the chemistry reaction mechanism EmChem09-HET, the SOA precursors are formed in the reactions of the VOC species oXylene (lumped surrogate of reactive aromatic VOC), C3H6 (propene and other alkenes with >4 C atoms) and nC4H10 (n-butane and other alkanes with >3 C atoms), isoprene, APIN (surrogate of slower reacting monoterpenes) and LIM (surrogate of faster reacting monoterpenes). The lumped SOA precursors from oxidation of oXylene, C3H6 and nC4H10 are ASOC and ALOC. The lumped SOA precursors from oxidation of isoprene, APIN and LIM are BSOC and BLOC. In addition two SOA precursors of different volatility are from direct emission (e.g. from ship and vehicles), PIOC and PSOC.

It is possible to use only a subset of the compounds in Table 4, however the point source files has to be adapted to the exact list with the order of compounds as they appear in the runscript (section 3.5.1).

Table 4: Chemical compounds with input and output in CityChem.

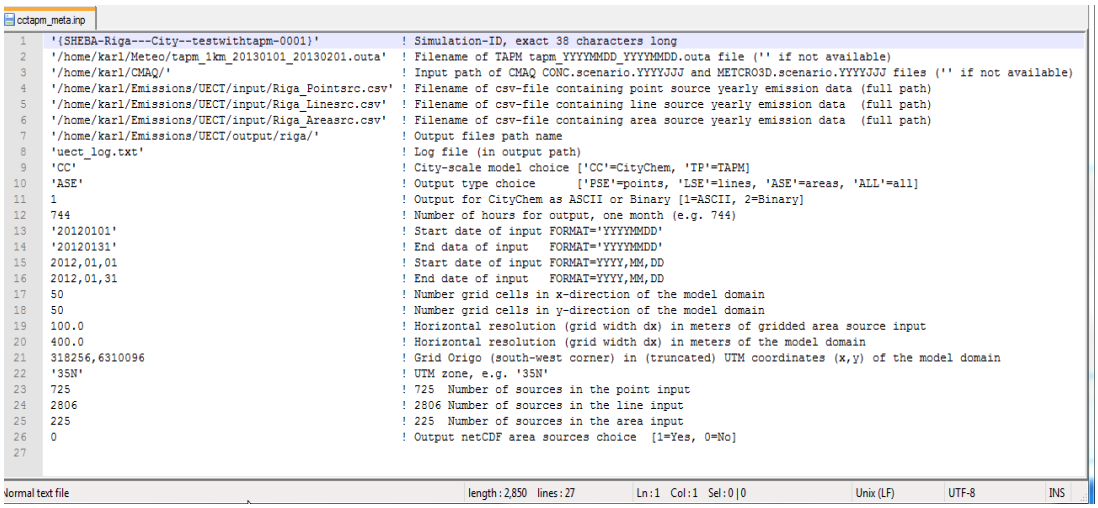
Index ic	Chemical compound	Full name
1	O3	Ozone
2	NO	Nitric oxide
3	NO2	Nitrogen dioxide
4	H2O2	Hydrogen peroxide
5	N2O5	Dinitrogen pentoxide
6	HNO3	Nitric acid
7	HONO	Nitrous acid
8	SO2	Sulphur dioxide
9	H2SO4	Sulphuric acid
10	CO	Carbon monoxide
11	C2H6	Ethane
12	HCHO	Formaldehyde
13	CH3CHO	Acetaldehyde (and higher aldehydes)
14	C2H4	Ethene
15	PAN	Peroxy acetyl nitrate
16	MACR	Methacrolein
17	MPAN	Peroxy methacryloyl nitrate
18	nC4H10	n-Butane (and other n>3 alkanes)
19	CH3COC2H5	Methyl ethyl ketone (and other ketones)
20	C3H6	Propene (and other n>4 alkenes)
21	oXylene	o-Xylene (and other aromatic HCs)
22	Isoprene	Isoprene
23	Apinene	Alpha-pinene (and other slow reacting terpenes)
24	Limonene	Limonene (and other fast reacting terpenes)
25	ASOC	Semi-volatile products from oXylene and alkanes
26	ALOC	Low-volatile products from oXylene and alkenes
27	BSOC	Semi-volatile biogenic oxidation product
28	BLOC	Low-volatile biogenic oxidation product
29	PIOC	Primary intermediate volatility organic compound
30	PSOC	Primary semi-volatile organic compound
31	ASOA	Semi-volatile SOA from oXylene and alkenes
32	ALOA	Low-volatile SOA from oXylene and alkenes
33	BSOA	Semi-volatile SOA from biogenic VOC
34	BLOA	Low-volatile SOA from biogenic VOC
35	PIOA	Intermediate volatile SOA from primary organics
36	PSOA	Semi-volatile SOA from primary organics
37	NH3	Ammonia
38	PM2.5	Fine Particulate Matter, Dp ≤ 2.5 μm
39	PM10	Particulate Matter; Dp ≤ 10 μm
40	pnc1	Particle number; Dp: 1.5− 4 nm
41	pnc2	Particle number; Dp: 4− 9 nm
42	pnc3	Particle number; Dp: 9− 21 nm
43	pnc4	Particle number; Dp: 21− 50 nm
44	pnc5	Particle number; Dp: 50− 120 nm
45	pnc6	Particle number; Dp: 120− 290 nm
46	pnc7	Particle number; Dp: 290− 670 nm
47	pnc8	Particle number; Dp: 670− 1000 nm
48	pnc9	Particle number; Dp: 1000− 2500 nm
49	pnc10	Particle number; Dp: 2500−10000 nm

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 6 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.


cctapm_meta.inp

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

Normal text file | length : 2,850 lines : 27 | Ln : 1 Col : 1 Sel : 0 | 0 | Unix (LF) | UTF-8 | INS

Figure 6: 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 ‘’.
point_path	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 ‘’.
line_path	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 ‘’.
area_path	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 ‘’.
log_file	File name of the log file of UECT. The log file will be written to the directory where UECT was executed.
model	City-scale model option used by UECT to produce emission input files. Enter ‘CC’ for CityChem or ‘TP’ for TAPM.
source	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.
ep_fmt	Output file format option. Select *ep_fmt* = 1 for ASCII output files or *ep_fmt* = 2 for binary output (input files for CityChem) or *ep_fmt* = 3 for netCDF output (only enabled for tapm4cc.exe).
hmonth	Numbers of simulation hours (for one month).
start_string	Start date as string. Format has to be ‘YYYYMMDD’.
end_string	End date as string. Format has to be ‘YYYYMMDD’.
startdate	Start date separated by comma. Format: YYYY,MM,DD.
enddate	End date separated by comma. Format: YYYY,MM,DD.
nx	Number of grid cells of the model domain (main grid) in x-direction.
ny	Number of grid cells of the model domain (main grid) in y-direction.
dxarea	Grid width dx of the cells in the area source emission file.
dx	Grid width dx of the cells of the model domain (main grid).
utmx, utmy	UTM coordinates (x, y) of the model grid origin (south-west corner of the rectangular domain). UTM x-coordinate *utmx* has to be truncated, i.e. not containing the two leading digits which indicate the UTM zone.
utm_zone	UTM zone as string, for example: ‘35N’. If the zone is on the northern hemisphere, third sign is ‘N’, if the zone is on the southern hemisphere, third sign is ‘S’.
nsopp	Number of point sources in the point source input file to UECT.
nsoll	Number of line sources in the line source input file to UECT.
nsoaa	Number of area sources in the area source input file to UECT.
ncout	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 B1 in Appendix B gives an overview of the mandatory and optional meteorological input files for simulations with EPISODE-CityChem. There are three options for creating input files for meteorology:

1. 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.
2. Use 2-D and 3-D fields of meteorological variables computed by the WRF model. Meteorological data is converted to the same format as TAPM, but uses the vertical dimensions of the WRF output (see section 6.4).
3. If neither model data from TAPM or WRF is available, one can 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).

3.4.1 Setup with TAPM

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 7).

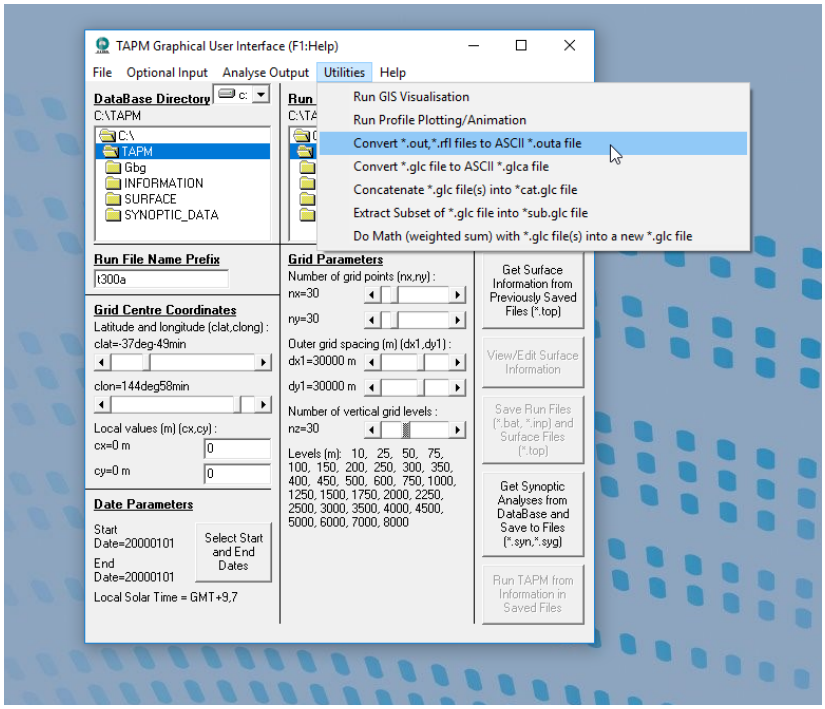


Figure 7: TAPM Graphical User Interface, selection of outa-file creation in the Utilities menu.

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.

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 outa-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*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/tapm4cc2.2/ and edit in Makefile.tapm the variables INCLUDES and LIBS. Here the correct path of the netCDF library (see Appendix D) 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/tapm4cc2.2/$        make -f Makefile.tapm
```

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

```
{your-citychem}/preproc/$        ln -s ./tapm4cc2.2/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
```

3.4.2 Setup with WRF

Output of the Weather Research and Forecasting WRF model (http://www.wrf-model.org/), usually from the innermost nest (d03), can be used to set up the meteorology for EPISODE-CityChem. WRF output files should be merged into daily files containing 25 hours, each including the last hour of the previous day. The WRF4CC utility (section 6.4) is used for conversion of WRF output files. It produces the same meteorological files as the TAPM converter. The land use scheme applied in WRF should be the MODIS-NOAH with 20 or 21 categories or the USGS24 land use scheme with 24 categories. A list of required meteorological variables in the WRF output is given in Table 5.

Table 5: Required meteorological variables in the WRF output.

Variable	Field name
U	Horizontal wind component u
V	Horizontal wind component v
W	Vertical wind component w
PH	Perturbation geopotential
PHB	Base state geopotential
P	Perturbation pressure
PB	Base state pressure
T	Perturbation potential temperature
QVAPOR	Water vapour mixing ratio
QCLOUD	Cloud water mixing ratio
QRAIN	Rain water mixing ratio
CLDFRA	Cloud fraction
MU	Perturbation dry air mass in column
MUB	Base state dry air mass in column
T2	Air temperature at 2 m
Q2	Specific humidity at 2 m
PSFC	Surface pressure
MAPFAC_M	Map scale factor on mass grid
MAPFAC_U	Map scale factor on u-grid
MAPFAC_V	Map scale factor on v-grid
HGT	Terrain height
RAINC	Accumulated total cumulus precipitation
RAINNC	Accumulated total grid scale precipitation
SWDOWN	Downward short wave radiation at ground surface
GLW	Downward long wave flux at surface
LU_INDEX	Land use category
LANDMASK	Land mask (1 for land, 0 for water)
HFX	Upward heat flux at the surface
LH	Latent heat flux at the surface
UST	u_* in similarity theory
TSK	Surface skin temperature
ALBEDO	Albedo
SNOWC	Snow coverage flag
SEAICE	Sea ice flag
SNOWH	Physical snow depth
ZNU	Eta values on half (mass) levels
ZNW	Eta values on full (w) levels

3.4.3 Required meteorological fields

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

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

  read("(2a10.2i3") TEXT1, TEXT2, nx, ny
  read(*) ((ii,jj,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/WRF processors.
  Read format (ascii):

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

  read("(2a10.2i3") TEXT1, TEXT2, nx, ny
  read(*) ((ii,jj,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/WRF processors.
  Read format (ascii):
  read("(2a10.2i3") TEXT1, TEXT2, nx, ny
  read(*) ((ii,jj,aero(ix,iy),ix=1,nx),iy=1,ny)

- Mixing height:
  Mixing height field (hmix in 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/m^2) 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/m^2) 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.
  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
```

- 3-D Specific humidity:
  For each vertical layer, starting with lowest layer, field of specific humidity (shum in kg/kg) for every simulation hour. For TAPM, the 3-D field of relative humidity is converted to specific humidity with the tapm4cc utility.

```
do iz=1,nz
  read("(2a10.2i3")  TEXT1, TEXT2, nx, ny
  read(*)  ((ii,jj,shum(ix,iy,iz),ix=1,nx),iy=1,ny)
enddo
```

- 3-D Cloud water:
  For each vertical layer, starting with lowest layer, field of cloud water mixing ratio (clwater in kg/kg) for every simulation hour. The variable is also known as liquid water content (LWC). This field can be obtained from WRF4CC processor. This input is optional.

```
do iz=1,nz
  read("(2a10.2i3")  TEXT1, TEXT2, nx, ny
  read(*)  ((ii,jj,clwater(ix,iy,iz),ix=1,nx),iy=1,ny)
enddo
```

- 3-D Pressure:
  For each vertical layer, starting with lowest layer, field of atmospheric pressure (pres in Pa) for every simulation hour. Note pressure has to be in Pascal. This field can be obtained from WRF4CC processor. This input is optional.

```
do iz=1,nz
  read("(2a10.2i3")  TEXT1, TEXT2, nx, ny
  read(*)  ((ii,jj,pres(ix,iy,iz),ix=1,nx),iy=1,ny)
enddo
```

- TAPM/WRF 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/WRF 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/(m^2s)) for every simulation hour. This field is not produced by TAPM/WRF processors.
  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/m^2) and second field of surface momentum stress in y-direction (taus_y in N/m^2) 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/m^2) 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.
- 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 (NOx), non-methane volatile organic compounds (NMVOC), carbon monoxide (CO), sulphur dioxide (SO_2), ammonia (NH_3), particulate matter $PM_{2.5}$ and PM_{10}, as well as total particle number (PN) 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 8 illustrates the three emission source objects.

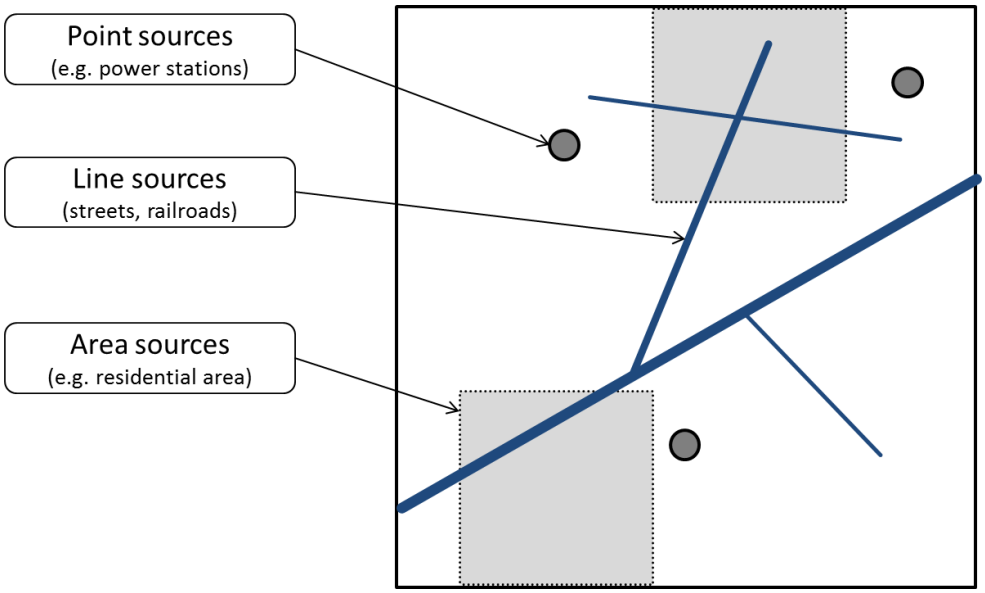


Figure 8: Illustration of the three emissions source objects.

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 nx × ny cells per model layer) and also for the vertical model layer that is on top of the highest model layer.

There are three options for using background concentrations in the simulation:

1) The first option is to use one constant concentration value for each CityChem compound.
2) The second option is to use hourly varying concentrations from measurements. The minimum requirement is to provide concentration time series of NO_2, O_3 and $PM_{2.5}$.
3) The third 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, 2 for hourly measured values, and to 3 for 3-D BCON files (section 4.9).

The input of background concentration is mandatory for each CityChem compound, except for the semi-volatile vapour compounds (ASOC, ALOC, BSOC, BLOC, PIOC, PSOC) and their related SOA compounds (ASOA, ALOA, BSOA, BLOA, PIOA, PSOA).

Constant background concentrations:
The option to use constant values should be chosen if no CMAQ model concentration data is available. This option requires only one background concentration value per CityChem compound (in unit µg/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.

Hourly varying concentrations:
Hourly concentrations of pollutants can be provided as input files. This option is new in EPISODE-CityChem v1.3. The minimum set of boundary concentrations is for NO_2, O_3 and $PM_{2.5}$. The format of the input files is the file format of measurements in the EBAS database (http://ebas.nilu.no). Data downloaded from the EBAS database through its web-portal is provided in a format based on the NASA Ames 1001 format, called the 'EBAS-Nasa Ames' format. Measurements should be selected from a regional background station of the EMEP monitoring network that is situated outside of the model domain and usually in the same

region of Europe. The EBAS files can be downloaded for each month and used in the model after removing the header information (all lines at the beginning of the file above the columns with numerical values). A concentration value is required for each hour of the simulation. Required format of the hourly background concentration file is given below.

Format of the file:
For every simulation hour:
Read format (ascii):

```
do h = 1, nhours
    read(*)   starttime, endtime, pollutant, flag_pollutant
enddo
```

Where nhours is the number of hours of the respective simulation period (usually one month).

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*nbcy)
```

Where nx_D=nx+2*nbcx and ny_D=ny+2*ncby, with nbcx and ncby 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 6 shows a list of the auxiliary input files.

Table 6: 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.

The plume segment file is explained in section 3.7.1 and building the receptor-stations raster file is explained in section 3.7.2.

The 2-D auxiliary fields are described in section 3.7.3.

There are three different methods for producing input files of the 2-D auxiliary fields:

1. Using the output file from the TAPM model (section 3.7.4).
2. Using the converter of WRF model output (section 3.7.5).
3. Using AERMAP and other tools if no TAPM/WRF model output is available (section 3.7.6).

3.7.1 Old plume segments

This auxiliary file contains the plume segments from the previous time step. This file can also be used for the restart of an interrupted simulation.

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:

Start column:	*property*	*default value*
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

3.7.2 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.

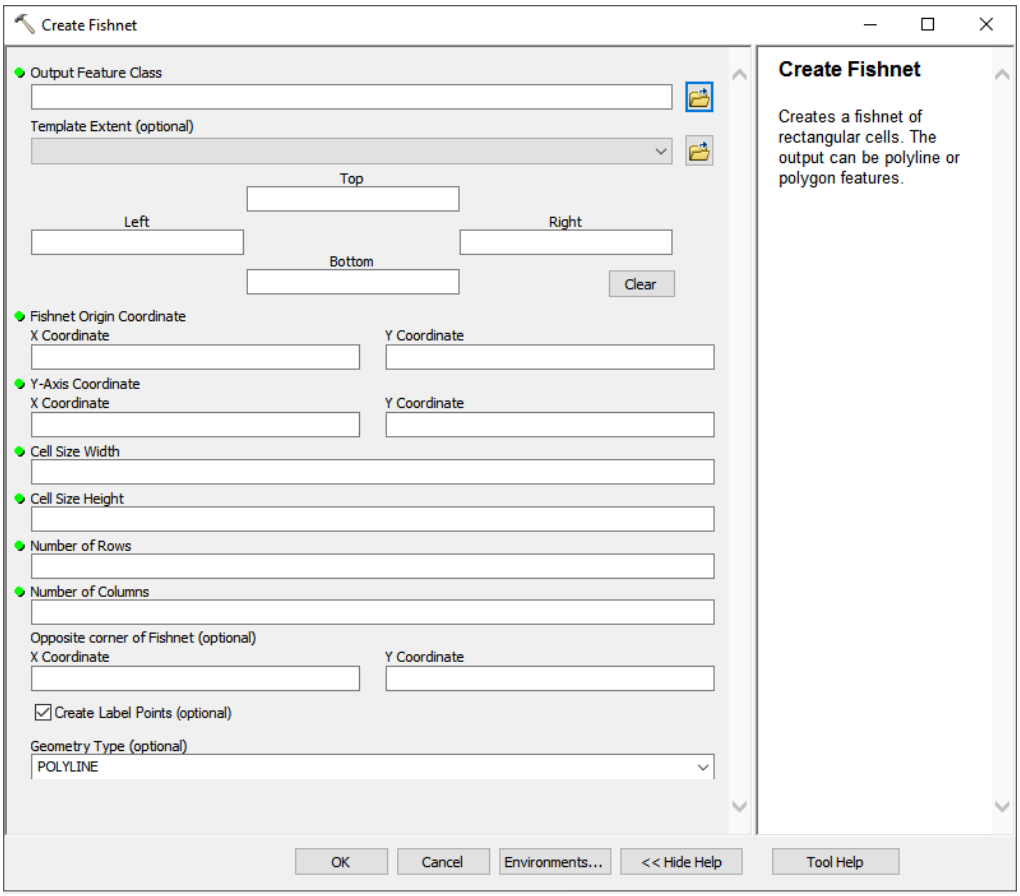


Figure 9: Screenshot of the Esri® ArcMap™ tool “Create Fishnet (Data Management)”.

- To create a grid of receptor points within ArcGIS, the ‘Fishnet’ tool will be used. Tools in ArcMap are opened by clicking on the menu item ‘Geoprocessing’, selecting ‘Search for Tools’ and entering the name of the tool in the search field. The ‘Fishnet’ 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 9 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 10 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. The output format is *.dbf by default and doesn’t need to be changed
- 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.
- Next the tool ‘Table to Excel (Conversion)’ is opened. Enter the exported fishnet and start the conversion into an Excel worksheet by clicking ‘OK’.

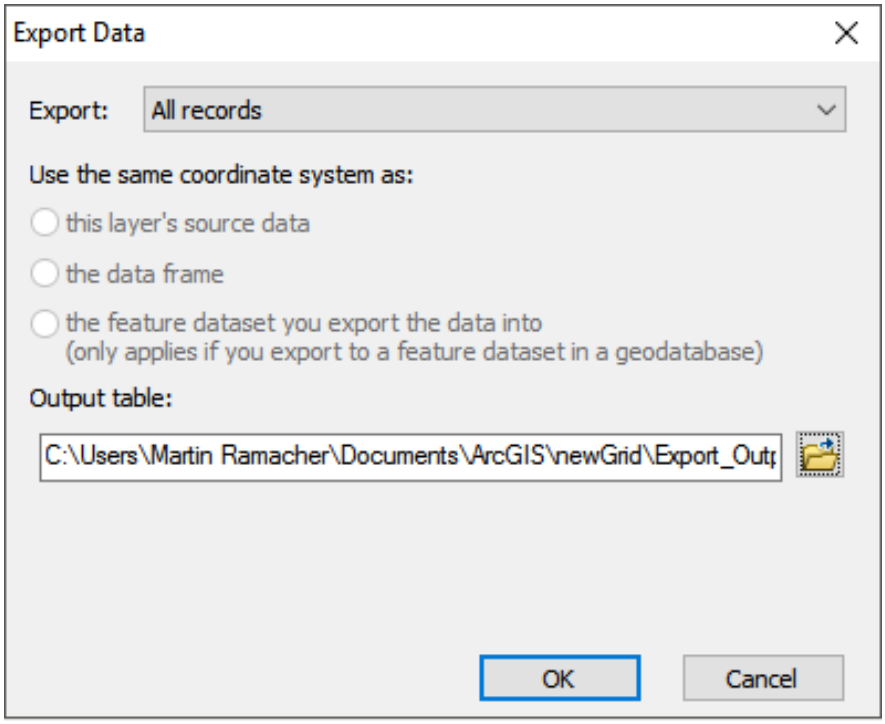


Figure 10: Screenshot of the Esri® ArcMap™ pop-up window "Export Data".

Finally, the Excel worksheet is opened and the remaining columns (fields sim_id, zcor, rcmax 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.

3.7.3 Auxiliary 2-D fields

Mandatory 2-D fields 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. The latter is needed by EPISODE-CityChem if the street canyon option is activated.

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(*) ((ii,jj,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(*) ((ii,jj,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.

3.7.4 Auxiliary 2-D fields from TAPM

This section describes the procedure of creating input files of auxiliary 2-D fields from TAPM model 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 7 lists the TAPM land use classes together with vegetation height (h_f in m) and surface roughness length (z0 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 (z0 in m) is obtained from the vegetation height by division with 10. For urban land use, z0 is defined in Table 2 of the technical documentation of TAPM (*Hurley*, 2008).

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 × y)/10] lines (with x and y as raster dimensions, given in the first line of the *.top file) and ends after of x × y values or (x × 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 followed by an extra space (e.g. replace '4' by '4 '). All values must then 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 z0 values based on Table 7. 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. The program **z0top4cc.exe** is built by the automated installation routine. Alternatively, build the tool as follows:

```
{your-citychem}/preproc/auxiliary/$  make –f Makfile.zo
{your-citychem}/preproc/auxiliary/$  cd ..
{your-citychem}/preproc/$  ln –s ./auxiliary/bin/z0top4cc.exe
```

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

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

Table 7: 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

3.7.5 Auxiliary 2-D fields from WRF

The WRF4CC utility will produce all four auxiliary 2-D fields, i.e. topography, land use classes, surface roughness and cloud fraction, in the format required for EPISODE-CityChem.

It is expected that the land use MODIS-NOAH with 20/21 categories was used in the WRF simulation. The land use categories of MODIS-NOAH are mapped to the land use categories of TAPM when producing the land use input file. It is also possible to apply the USGS24 land use scheme with 24 categories in WRF4CC.

Details on installation and use of WRF4CC are given in section 6.4.

3.7.6 Auxiliary 2-D fields with AERMAP

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

The only modification compared to the original AERMAP source code is in aermap.f, subroutine WRITRZ, for writing of 5 values instead of 6 values per line to the output file AERMAP.REC.

A separate installation of AERMAP is normally not required, since it becomes installed when you issued:

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

If this is not working for any reason, use Makefile.aermap in subdirectory directory {your-citychem}/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 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 file can be located according to "NlatElon.hgt". 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.

- Extract the selected *.hgt 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.

- The shell script "srtm_generate_hdr.sh" in subdirectory preproc/auxiliary is used to convert from *.hgt to GeoTIFF. The shell script is from grassbock.org (http://www.grassbook.org/wp-content/uploads/neteler/rs/srtm/)

- 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 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. For large domains (e.g. 50 km x 50 km) or domains that are crossing a UTM zone border, it might be necessary to subtract 1500 m from UTME and from UTMN of the lower left corner.
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 UMTN 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 11 shows a screenshot of "aermap.inp" input file of AERMAP.
- Figure 12 gives an illustration of how to find the domain coordinates DOMAINXY and ANCHORXY (for use in the "aermap.inp" input file of AERMAP) relative to the corners of the chosen city domain.

aermap.inp

```
CO STARTING
   TITLEONE  Using 3 arc-second SRTM File for N60E024
   TITLETWO  With NAD83-Equivalent Anchor Point
   DATATYPE  NED

** Use 3 arc-second SRTM File for Helsinki area (WGS84)
** N60E024 and N60E025 needed to cover Helsinki area
   DATAFILE  "../geotiff/N60E024.tif"   tiffdebug
   DATAFILE  "../geotiff/N60E025.tif"   tiffdebug

** Request ALL Debug Output Files
   DEBUGOPT  ALL

**          x-500 m                 x+500m
**          Lower Left corner       Upper right corner
**          LL UTME  LL UTMN   Zn  UR UTME  UR UTMN   Zn
   DOMAINXY  378455.0 6661198.0 35  394455.0 6676198.0 35

** NAD83 anchor point: (use NAD84 for WGS84)
** NAD code in TABLE 3-1 of AERMAP User Guide
** WGS84 = 3
** Mid-point = x1+7.5km, y1+7.5km
**          X-point  Y-point     UTME     UTMN   Zn  NAD
   ANCHORXY  0.0      0.0     386455.0 6668698.0  35  3

   NADGRIDS  ..\
   RUNORNOT  RUN
CO FINISHED

** Output elevation on a Cartesian grid
**                       xinit xnum  dx    yinit  ynum  dy
RE STARTING
   GRIDCART CART1 STA
                 XYINC  -7500.0 150 100.0  -7500.0 150 100.0
   GRIDCART CART1 END
RE FINISHED
```

Normal text file length : 1370 lines : 43 Ln : 2 Col : 54 Sel : 0 | 0 Dos\Windows UTF-8 INS

Figure 11: Screenshot of the "aermap.inp" configuration file for AERMAP.

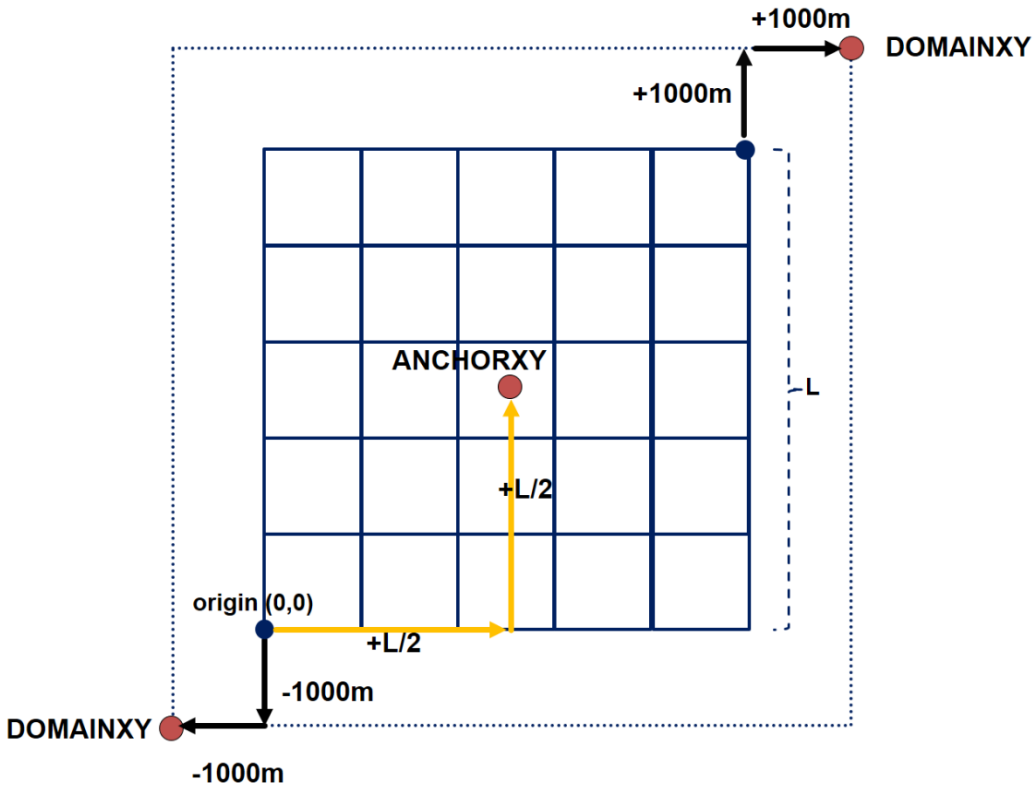


Figure 12: Illustration of the grid definition in AERMAP.

- 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/**

- It is recommended to check the *.REC file in a text editor. If you find a message like:

  ** NOTE:
  ** This run includes: 2 Receptor(s) located in GAPS INSIDE file(s);
  ** with a total of: 0 Receptor(s) assigned elevations based on subsequent file(s).
  ** This run includes: 2 Receptor(s) assigned missing elevations and hill heights (-9999.0)!

  then replace the missing values (-9999.0) with 0.0

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

An example of the elevation map produced from SRTM3 data using the AERMAP terrain pre-processor is shown in Figure 13.

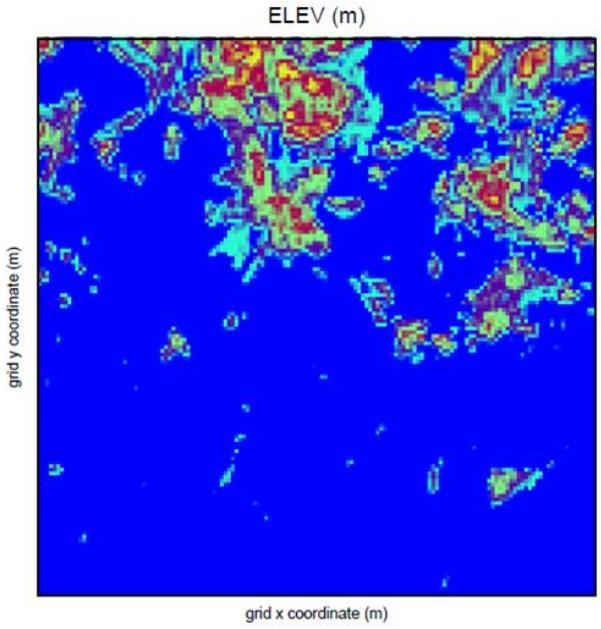


Figure 13: Terrain elevation map (heights in m) of the Helsinki bay with a grid resolution of 100 m, produced from SRTM3 data using the AERMAP terrain pre-processor. Blue colour is sea surface, yellow to red colours show elevations up to 50.9 m.

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
   testbecnh_log_is_not_created.txt
```

The testbench log file is not created by the program. Do not change this line

```
* Run log file (written for airquis)
* this log-file is not created by CityChem
../OUTPUT/ CITYCHEM_Yourmachine_log.txt
```

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

```
* LHS example of new INFO-file
 ../OUTPUT/CITYCHEM_INFO_masstime.txt
```

This info file records the mass balance of all compounds at the four sides of the model domain. It also records the computation timings. The info file is not obligatory, therefore a space can be inserted at the beginning of the line.

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 angel (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
1000.0,1000.0,17.50,20.00,25.00,25.00,37.50,50.00,50.00,50.00,50.00,50.00,50.00,50.00,75.00,125.00,20
0.00,250.00,250.00,250.00,250.00,250.00,250.00,375.00,500.00,500.00
```

Enter grid cell width in x-direction (in m), grid cell width in y-direction (in m), and the individual layer thickness (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 3.7.3 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 3.7.3 how to produce this file.

```
* Surface Albedo file/value (1 = Total reflection; 0 = Black body) Default 0.3?
 inp_xxxnn/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 3.7.2 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_xxxnn/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 ‘pisiteexternaldata’ 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 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. For use of WRF it is referred to section 3.4.2. The converted output from WRF is used here with option ‘1’ as well.

```
* 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))
2
```

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_xxxnn/sigmavw.fld

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

*** TAPM aerodynamic resistance file**
*** missing data is tolerated**
inp_xxxnn/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 3.7.3. 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 Specific humidity (kg/kg) (or NZ-values)**
*** missing data is tolerated**
../INPUT/tapm/shum.fld

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

The specific humidity field is required since CityChem v1.5.

*** WRF 3D Cloud water mixing ratio (kg/kg) (or NZ-values)**
*** optional input**
inp_xxxnn/clwater3d.fld

Enter the path of the location of the file containing the 3-D cloud water mixing ratio field. The file can be produced from WRF and COSMO models using the WRF4CC converter. The 3-D cloud water file is optional, leave empty line if not available.

*** WRF 3D Pressure (Pa) (or NZ-values)**

```
* optional input
 inp_xxxnn/press3d.fld
```

Enter the path of the location of the file containing the 3-D pressure field. The file can be produced from WRF and COSMO models using the WRF4CC converter. The 3-D pressure file is optional, leave empty line if not available.

```
* 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_xxxnn/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_xxxnn/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
49
```

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
19
```

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
19
```

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 is the id in EmChem09-HET
*
```

*Id-nr	Compound	Unit	Conl	DDepI	WDepI	MolW	DDepV	WDepSR	THalf/Kcoag
17	'O3 '	'ug/m3 '	1	1	1	48	0.2E+00	0.0E+00	0.0E+00
18	'NO '	'ug/m3 '	1	0	0	30	0.0E+00	0.0E+00	0.0E+00
19	'NO2 '	'ug/m3 '	1	1	1	46	0.2E+00	0.0E+00	0.0E+00
65	'H2O2 '	'ug/m3 '	0	1	0	34	0.5E+00	1.0E+00	0.0E+00
23	'N2O5 '	'ug/m3 '	0	0	0	108	0.0E+00	0.0E+00	0.0E+00
25	'HNO3 '	'ug/m3 '	1	1	1	63	1.0E+00	1.0E+00	0.0E+00
26	'HONO '	'ug/m3 '	1	1	1	47	1.0E+00	1.0E+00	0.0E+00
73	'SO2 '	'ug/m3 '	0	1	1	64	0.0E+00	0.3E+00	0.0E+00
74	'H2SO4 '	'ug/m3 '	1	1	1	96	0.1E+00	1.0E+00	0.0E+00
71	'CO '	'ug/m3 '	1	0	0	28	0.0E+00	0.0E+00	0.0E+00
46	'C2H6 '	'ug/m3 '	0	0	0	30	0.0E+00	0.0E+00	0.0E+00
44	'HCHO '	'ug/m3 '	1	0	0	30	0.0E+00	0.0E+00	0.0E+00
45	'CH3CHO '	'ug/m3 '	0	0	0	44	0.0E+00	0.0E+00	0.0E+00
48	'C2H4 '	'ug/m3 '	0	0	0	28	0.0E+00	0.0E+00	0.0E+00
20	'PAN '	'ug/m3 '	1	0	1	121	0.0E+00	0.0E+00	0.0E+00
29	MACR	'ug/m3 '	0	0	0	70	0.0E+00	0.0E+00	0.0E+00
21	MPAN	'ug/m3 '	1	0	1	147	0.0E+00	0.0E+00	0.0E+00
47	'nC4H10 '	'ug/m3 '	1	0	0	58	0.0E+00	0.0E+00	0.0E+00
36	'CH3COC2H5 '	'ug/m3 '	0	0	0	72	0.0E+00	0.0E+00	0.0E+00
49	'C3H6 '	'ug/m3 '	0	0	0	42	0.0E+00	0.0E+00	0.0E+00
50	'oXylene '	'ug/m3 '	0	0	0	106	0.0E+00	0.0E+00	0.0E+00
51	'Isoprene '	'ug/m3 '	0	0	0	68	0.0E+00	0.0E+00	0.0E+00
52	'Apinene '	'ug/m3 '	0	0	0	136	0.0E+00	0.0E+00	0.0E+00
53	'Limonene '	'ug/m3 '	0	0	0	136	0.0E+00	0.0E+00	0.0E+00
77	'ASOC '	'ug/m3 '	0	0	0	150	0.0E+00	0.0E+00	0.0E+00
78	'ALOC '	'ug/m3 '	0	0	0	150	0.0E+00	0.0E+00	0.0E+00
76	'BSOC '	'ug/m3 '	0	0	0	180	0.0E+00	0.0E+00	0.0E+00
75	'BLOC '	'ug/m3 '	0	0	0	180	0.0E+00	0.0E+00	0.0E+00
79	'PIOC '	'ug/m3 '	0	0	0	280	0.0E+00	0.0E+00	0.0E+00
80	'PSOC '	'ug/m3 '	0	0	0	280	0.0E+00	0.0E+00	0.0E+00
81	'ASOA '	'ug/m3 '	0	0	0	150	0.1E+00	0.1E+00	0.0E+00
82	'ALOA '	'ug/m3 '	0	0	0	150	0.1E+00	0.1E+00	0.0E+00
83	'BSOA '	'ug/m3 '	0	0	0	180	0.1E+00	0.1E+00	0.0E+00
84	'BLOA '	'ug/m3 '	0	0	0	180	0.1E+00	0.1E+00	0.0E+00
85	'PIOA '	'ug/m3 '	0	0	0	280	0.1E+00	0.1E+00	0.0E+00
86	'PSOA '	'ug/m3 '	0	0	0	280	0.1E+00	0.1E+00	0.0E+00
87	'NH3 '	'ug/m3 '	0	1	1	17	0.2E+00	0.1E+00	0.0E+00
88	'PM2.5 '	'ug/m3 '	1	1	1	90	0.2E+00	1.0E+00	0.0E+00
89	'PM10 '	'ug/m3 '	1	1	1	90	0.2E+00	1.0E+00	0.0E+00
90	'pnc1 '	'num/cm3 '	1	1	1	-9900	1.000	0.4E+00	6.30E-09
91	'pnc2 '	'num/cm3 '	1	1	1	-9900	0.528	0.4E+00	4.51E-09
92	'pnc3 '	'num/cm3 '	1	1	1	-9900	0.355	0.4E+00	9.76E-08
93	'pnc4 '	'num/cm3 '	1	1	1	-9900	0.181	0.4E+00	1.50E-08
94	'pnc5 '	'num/cm3 '	1	1	1	-9900	0.068	0.4E+00	5.40E-09
95	'pnc6 '	'num/cm3 '	1	1	1	-9900	0.039	0.4E+00	6.26E-09
96	'pnc7 '	'num/cm3 '	1	1	1	-9900	0.031	0.4E+00	4.27E-09
97	'pnc8 '	'num/cm3 '	1	1	1	-9900	0.023	0.4E+00	2.28E-09
98	'pnc9 '	'num/cm3 '	1	1	1	-9900	0.110	0.4E+00	8.69E-10
99	'pnc10 '	'num/cm3 '	1	1	1	-9900	0.200	0.4E+00	8.00E-10

A list of the CityChem compounds.

The non-reactive species have to be listed at the end of the list of CityChem compounds. Optionally, other reactive species that are part of the chemical reaction mechanism (see *Karl et al.* (2019), Supplementary Materials) can be included (before PM2.5) but then the number of compounds needs to be increased corrrespondingly and emission/boundary condition input files have to be provided. In CityChem v1.8, 'Sulphate' was renamed to 'H2SO4' since the compound refers to gaseous sulphuric acid.

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 $\mu g/m^3$, 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.

Simulation of particle number concentration in eight size categories following the concept described by *Karl et al.* (2016).

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.ie 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_xxxnn/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_xxxnn/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
*
* deposition output new in CC v1.6
*
* Main grid dry deposition file
../OUTPUT/ddepmhour.nc
*
* Main grid wet deposition file
../OUTPUT/wdepmhour.nc
```

Location of the netCDF file containing hourly 2-D model dry deposition and wet deposition output for all CityChem compounds.
The output of dry depositon and wet deposition is available since CityChem v1.6.

```
* Receptor point dry deposition file
 res_xxxnn/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_xxxnn/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_xxxnn/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_xxxnn/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_xxxnn/statcmhighh_no2.txt
*
* Receptors nhighh-concentrations file
* res_xxxnn/statcrhighh_no2.txt
*
* Line source receptors nhighh-concentrations file
* res_xxxnn/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 coloumn)
* res_xxxnn/statcmhighd_no2.txt
*
* Receptors nhighd-concentrations file
* res_xxxnn/statcrhighd_no2.txt
*
* Line source receptors nhighd-concentrations file
* res_xxxnn/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.

4.8 Local photochemistry data

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

```
* PHOT MODEL DATA (Defined inside EPISODE if "defined airquis" and nc == 3. See RPHOTO)
*
* Photochemical scheme (0 or 1)
* this is used for receptor and line source points
* can be combined with EMEP schemes for Eulerian grid below
* 0 for NOx-simulation
* 1 for NO2,NO,O3 simple Photochemical equilibrium
* 3 for EP-10 basic photochemistry solver
3
```

Option to calculate photochemistry at the receptor points and at the line source associated receptor points. Enter ‘0’ to switch off chemical transformation at the receptor points. Enter ‘1’ for calculation of simple photochemical equilibrium (photo-stationary equilibrium between O_3, NO_2 and NO). The choice of the chemistry option for the receptor points is independent of the chemistry option for the Eulerian model grid in the following “GRID MODEL DATA” section.

4.9 Grid model data

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

*** GRID MODEL DATA**
*****
*** Add grid model to results**
1

Option for adding concentration and deposition calculated for the Eulerian model grid to the output concentration and deposition. Recommended to use ‘1’.

*** Horizontal numerical advection scheme(s) 1:nc**
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 NB! (0/1/2) means (no horizontal advection/pos def Bott/pos.def. and monotone Bott) included!

Enter the option for the horizontal advection scheme. Enter ‘0’ to switch off horizontal advection. Enter ‘1’ to apply horizontal advection using a positive definite 4th degree Bott scheme (recommended option). Enter ‘2’ to apply horizontal advection using a positive definite and monotone 4th degree Bott scheme. The flag has to be entered for each CityChem compound, separated by comma.

*** Horizontal numerical diffusion scheme(s) 1:nc**
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1 NB! (0/1) means that horizontal diffusion (is not/is) included!

Enter the option for the horizontal diffusion scheme. Enter ‘0’ to switch off horizontal diffusion. Enter ‘1’ to apply horizontal diffusion using a fully explicit forward Euler scheme. The flag has to be entered for each CityChem compound, separated by comma.

*** Vertical numerical adv./diff. scheme(s) 1:nc**
*** option 1 is combined up-stream and diffusion, takes long**
*** option 2 is very time consuming**
3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3 NB! (0 or 1/2/3) means that vertical advection and diffusion (is not/is) included!

Enter the option for the vertical advection and diffusion scheme. Enter ‘0’ to switch off vertical transport. Enter ‘1’ to apply a vertical transport using a combined vertical advection and up-stream diffusion solution. Enter ‘2’ to apply vertical transport using a split vertical diffusion and vertical advection solution. Enter ‘3’ to apply vertical transport using an approach where vertical advection is solved with an up-stream scheme and vertical diffusion is solved with a Crank-Nicholson scheme (recommended).

*** Dry deposition scheme(s) 1:nc**
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

Enter the option for dry deposition. Enter’0’ to switch off dry deposition and ‘1’ to include dry deposition. The selected value (0|1) has to be the same for all compounds.

*** Wet deposition scheme(s) 1:nc**

1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

Enter the option for dry deposition. Enter’0’ to switch off dry deposition and ‘1’ to include dry deposition.

*** Photochemical scheme [only used in AirQuis version], give same number as above**
*** 0 for PMxx-, NOx- etc. dispersion w/o photochemistry (AirQuis default)**
*** 1 for standard NO2/NO/O3-photoequilibrium simulation (not used)**
*** 2 for EMEP03 (only work with nc=3, O3 NO NO2)**
*** 3 for EmChem03mod**
*** 4 for EmChem09mod (new: EmChem09-HET)**

4

Enter the option for chemical transformation on the model grid. Enter '0' to switch off chemical transformation. Enter '1' for the simple photo-stationary equilibrium between O_3, NO_2 and NO. This only works if the option for local photochemistry (section 4.8) is also set to '1'. Note that with this option, PSS is only applied to the concentration field at the surface level. We therefore discourage the use of this option. Enter '2' for applying the two-step chemical solver to a simple reaction scheme between O_3, NO_2 and NO. Enter '3' for applying the two-step chemical solver with the detailed chemistry scheme EmChem03-mod. Enter '4' to use the new EmChem09-het chemistry scheme extending the previous one. Option '4' is the default.

*** Aerosol modules**
*** 0 for no aerosol module**
*** 1 for MAFOR aerosol dynamics (only condensation): requires 10 PNC compounds**
*** 2 for ISORROPIA (secondary inorganic aerosol). Not possible now!**

1

Enter the option for secondary aerosol formation on the model grid. Enter '0' to run without aerosol module. Enter '1' for the MAFOR-SOA module to produce secondary organic aerosol. This option requires 10 PNC compounds and 6 SOA tracers in the list of chemical compounds. The mass of SOA compounds is not added to PM2.5 and PM10. To obtain the total primary and secondary PM, the SOA mass has to be added in the post-processing. It is only recommended to activate this option, if there are emissions of VOC and emissions of primary organic vapours (PIOC and PSOC) at least from the road traffic sector. Enter '2' for the ISORROPIA module to produce secondary inorganic aerosol. This option is currently not available and the program will stop.

*** INPUT Background concentration unit: 1 = ug/m3, 2 = molecules/cm3, 4 = num/cm3**
***4**
*** simply use same UNIT as in the list of chemicals**

Possibility to use different concentration units for the background concentration input. The line in the run script has to be commented.

By default, $\mu g/m^3$ is used as default unit for the background concentration.

*** INPUT Background concentrations file format: 1 = Hourly scalar, 2 = Hourly vertical vector, 3 = Hourly 3D-field**
*** 1: constant value**
*** 2: ebas BC file**
*** 3: 3D BC file**
3

Option for the input of background concentrations. Enter '1' to provide one constant background concentration value for each CityChem compound.

Enter ‘2’ to provide hourly concentration measurements.

Enter ‘3’ to provide 3-D concentration field of background concentrations.

If constant value option was chosen in the previous line, enter one background concentration value for each CityChem compound per line.

If hourly value option was chosen in the previous line, enter the path and name of the background concentration file for each CityChem compound per line. The minimum set of background files is for NO_2, O_3 and $PM_{2.5}$. When no background data is available for a compound, leave the line empty.

The format of the input file of hourly background concentrations is explained in section 3.6. Note that all hours need to have real concentration values, missing values and NaN values have to be replaced.

Below follows the example when 3-D input files are chosen.

```
* 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_hono.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_macr.fld
../INPUT/cmaq/cbg_mpan.fld
../INPUT/cmaq/cbg_nc4h10.fld
../INPUT/cmaq/cbg_ch3coc2h5.fld
../INPUT/cmaq/cbg_c3h6.fld
../INPUT/cmaq/cbg_oxylen.fld
../INPUT/cmaq/cbg_isoprene.fld
../testdata/cmaq/cbg_apinene.fld
../testdata/cmaq/cbg_limonene.fld
../testdata/cmaq/cbg_asoc.fld
../testdata/cmaq/cbg_aloc.fld
../testdata/cmaq/cbg_bsoc.fld
../testdata/cmaq/cbg_bloc.fld
  ../cbg_pioc.fld
  ../cbg_psoc.fld
  ../cbg_asoa.fld
  ../cbg_aloa.fld
  ../cbg_bsoa.fld
  ../cbg_bloa.fld
  ../cbg_pioa.fld
  ../cbg_psoa.fld
../testdata/cmaq/cbg_nh3.fld
../testdata/cmaq/cbg_pm25.fld
../testdata/cmaq/cbg_pm10.fld
../testdata/cmaq/cbg_pnc1.fld
../testdata/cmaq/cbg_pnc2.fld
../testdata/cmaq/cbg_pnc3.fld
../testdata/cmaq/cbg_pnc4.fld
../testdata/cmaq/cbg_pnc5.fld
../testdata/cmaq/cbg_pnc6.fld
../testdata/cmaq/cbg_pnc7.fld
../testdata/cmaq/cbg_pnc8.fld
../testdata/cmaq/cbg_pnc9.fld
../testdata/cmaq/cbg_pnc10.fld
```

If 3-D input files option was chosen in the previous line, enter one BCON file path for each CityChem compound per line. The input files are mandatory, except for the semi-volatile vapour compounds (ASOC, ALOC, BSOC, BLOC, PIOC, PSOC) and the SOA compounds (ASOA, ALOA, BSOA, BLOA, PIOA, PSOA).

```
* OFFSET Ozone (O3) in BCON file in ug/m^3
0.0
* OFFSET PM (PM2.5 & PM10) in BCON file in ug/m^3
0.0
```

Adding a fix positive offset to the boundary concentration of O_3 (in µg/m^3) and PM (in µg/m^3) 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 CO**
*** 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, HO_2, CH_3O_2, NO_3), 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_20130701_20130731_o3.txt
../INPUT/emis/asrc_domestic_20130701_20130731_no.txt
../INPUT/emis/asrc_domestic_20130701_20130731_no2.txt
  emis_xxxnn/asrc_domestic_20130701_20130731_h2o2.txt
  emis_xxxnn/asrc_domestic_20130701_20130731_n2o5.txt
  emis_xxxnn/asrc_domestic_20130701_20130731_hno3.txt
  emis_xxxnn/asrc_domestic_20130701_20130731_hono.txt
../INPUT/emis/asrc_domestic_20130701_20130731_so2.txt
  emis_xxxnn/asrc_domestic_20130701_20130731_sulphate.txt
  emis_xxxnn/asrc_domestic_20130701_20130731_co.txt
../INPUT/emis/asrc_domestic_20130701_20130731_c2h6.txt
../INPUT/emis/asrc_domestic_20130701_20130731_hcho.txt
../INPUT/emis/asrc_domestic_20130701_20130731_ch3cho.txt
../INPUT/emis/asrc_domestic_20130701_20130731_c2h4.txt
  emis_xxxnn/asrc_domestic_20130701_20131231_pan.txt
  emis_xxxnn/asrc_domestic_20130701_20131231_macr.txt
  emis_xxxnn/asrc_domestic_20130701_20131231_mpan.txt
../INPUT/emis/asrc_domestic_20130701_20130731_nc4h10.txt
../INPUT/emis/asrc_domestic_20130701_20130731_ch3coc2h5.txt
../INPUT/emis/asrc_domestic_20130701_20130731_c3h6.txt
../INPUT/emis/asrc_domestic_20130701_20130731_oxylen.txt
  emis_xxxnn/asrc_domestic_20130701_20131231_isoprene.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_apinene.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_limonene.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_asoc.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_aloc.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_bsoc.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_bloc.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_pioc.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_psoc.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_asoa.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_aloa.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_bsoa.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_bloa.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_pioa.txt
 emis_xxxnn/area/asrc_domestic_uple_20130701_20130731_psoa.txt
../INPUT/emis/asrc_domestic_nh3_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pm2.5_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pm10_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc1_20130701_20130731.txt
```

../INPUT/emis/asrc_domestic_pnc2_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc3_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc4_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc5_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc6_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc7_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc8_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc9_20130701_20130731.txt
../INPUT/emis/asrc_domestic_pnc10_20130701_20130731.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 O_3) 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.

```
* Old plume segments file
../INPUT/other/old_plume.dat
```

Location of the file containing the parameters of plume segments that existed before the start time of the simulation. See section 3.7 for the format of this input file.

```
* New plume segments file
../OUTPUT/plume_segments.dat
```

Location of the output ASCII file containing the hourly plume segments.

```
* Plume segment redirection limit
30.0
```

Limit value for the angle of the wind direction change (in degrees). Default value is 30^0. If the wind changes more than given by this limit angle, the direction of the trajectory of a plume segment will be changed.

```
* Maximum plume segment fractions YLF,ZLF(1:NZ)
* NB! Should be changed when changing vertical resolution!
0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.2
5,0.25,0.25,0.25
```

Maximum fraction of a grid cell area covered by the plume segment. Once the extent of the plume segment exceeds this fraction limit, the transported material of the plume segment will be “dumped” into the grid cell. The fraction value has to be given for each vertical layer of the model domain, separated by comma.

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

Enter the value of the minimum wind speed (in m/s) that is allowed in the point source model.

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_xxxnn/lsrv_traffic_line_20130701_20130731_o3.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_no.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_no2.txt
   emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_h2o2.txt
   emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_n2o5.txt
   emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_hno3.txt
   emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_hono.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_so2.txt
   emis_xxxnn/lsrv_traffic_line_20130701_20130731_sulphate.txt
   emis_xxxnn/lsrv_traffic_line_20130701_20130731_co.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_c2h6.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_hcho.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_ch3cho.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_c2h4.txt
   emis_xxxnn/lsrv_traffic_line_20130701_20130731_pan.txt
   emis_xxxnn/lsrv_traffic_line_20130701_20130731_macr.txt
   emis_xxxnn/lsrv_traffic_line_20130701_20130731_mpan.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_nc4h10.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_ch3coc2h5.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_c3h6.txt
../INPUT/emis/lsrv_traffic_line_20130701_20130731_oxylen.txt
    emis_xxxnn/lsrv_traffic_line_20130701_20130731_isoprene.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_apinene.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_limonene.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_asoc.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_aloc.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_bsoc.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_bloc.txt
../INPUT/emis/linesrc_traffic_pioc_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_psoc_20130701_20130731.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_asoa.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_aloa.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_bsoa.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_bloa.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_pioa.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_psoa.txt
    emis_xxxnn/line/lsrv_traffic_line_20130701_20130731_nh3.txt
../INPUT/emis/linesrc_traffic_pm2.5_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pm10_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc1_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc2_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc3_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc4_20130701_20130731.txt
```

```
../INPUT/emis/linesrc_traffic_pnc5_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc6_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc7_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc8_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc9_20130701_20130731.txt
../INPUT/emis/linesrc_traffic_pnc10_20130701_20130731.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 O_3) 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, for NOx (HH_standard: 1.20)
1.20
* Emission scale factor for subgrid line source model, for PM (HH_Default: 1.00)
1.00
```

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 C1 in Appendix C. 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 NO_2 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 postcchem/. If you want to install NCL/NCARG on Linux, follow the instructions at https://www.ncl.ucar.edu/Download/install.shtml for the proper installation, in short:

1. Extract the file where you want the software to be installed.
2. Set the NCARG_ROOT and PATH environment variables to point to where the software was installed.
3. Set the DISPLAY environment variable to indicate where to display graphics (this step likely not needed)
4. Put a ".hluresfile" file in your home directory.
5. Test your NCL/NCARG installation.

Below we give some more details for the installation on Linux systems.

NCL packages are now hosted at the Climate Data Gateway of NCAR, https://www.earthsystemgrid.org/dataset/ncl.html. From there you can select between different versions of NCL. It is recommended to choose version 6.3.0 or higher. After clicking on the version you have the option to download OPeNDAP-enabled precompiled binaries or not OPeNDAP-enabled precompiled binaries. We recommend to choose the latter (i.e. *_nodap_*). Next, after clicking on “Download options”, you get to a download page with several options. There are several pre-compiled binaries as *.tar.gz files for different Linux systems. The binary filenames contain the type of Linux it was built on (RedHat/RHEL, Debian, CentOS, SuSE Linux), the “bitness” (32-bit or 64-bit) of your machine, and which version of the GNU or Intel compilers it was compiled with. For the installation on Ubuntu Linux, we recommend the Debian (*_nodap_*), preferably an older version, to avoid conflicts with newer system libraries.

Then download and unpack the selected pre-compiled binary. You find some additional hints for running NCL under Linux systems here: https://www.ncl.ucar.edu/Download/linux.shtml.

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
```

Replace /usr/lib/ncarg in the above statement with the corresponding path where you installed the downloaded binary. You will need to logout and log back in for the changes to take effect. You can check by typing in the console:

```
{your-home}$    echo $NCARG_ROOT
```

if the changes have come into effect.

Sometimes it is necessary to change the path of the NCARG libraries which are in the first lines of the NCL script. In case that it is not possible to define the NCARG_ROOT environment variable, the first three command lines in each NCL script (starting with "load") need to be changed to provide the correct path of the user-contributed NCL-scripts (gsn_code.ncl, gsn_csm.ncl, WRFUserARW.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 R

Several R scripts are distributed with CityChem, providing additional options to plot and analyse 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'
```

Here CODENAME is the name of the Ubuntu distribution.

Then enable the CRAN repository and add the trusted key of CRAN, which is used to sign the R packages for security:

```
{your-home}$    sudo apt-key adv --keyserver keyserver.ubuntu.com
                 --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9

{your-home}$    sudo add-apt-repository 'deb [arch=amd64,i386]
                https://cran.rstudio.com/bin/linux/ubuntu CODENAME/'
```

Then update and start the installation of R:

```
{your-home}$    sudo apt-get update

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

{your-home}$    R --version
```

After installation of R, you are ready to install openair. Install the build-essential package which contains the tools required for compiling R packages:

```
{your-home}$    sudo apt install build-essential
```

To use the OpenAir script for post-processing of CityChem output, you need the libraries “ncdf4” as well “openair” installed. For plotting spatial maps, you further need the libraries “rgdal”, “ggmap” and “sf”. If these are not installed yet, you can open R in your terminal install the libraries as root and quit the R console (change to directory ../postcchem/ in your CityChem installation). The installation of the R-package “ncdf4” requires the netcdf library, which can be installed by:

```
{your-home}$    sudo apt-get install libnetcdf-dev
```

Then start the installation of R-packages by opening the R console as root (to install packages):

```
{your-citychem}/postcchem$    sudo -r R

    > install.packages(“openair”)
    > install.packages(“ncdf4”)
    > install.packages(“rgdal”)
    > install.packages(“ggmap”)
    > install.packages(“sf”)

    > quit()
```

The R packages are installed by default to the directory ‘/usr/local/lib/R/site-library’. This site-wide path is available because we ran R as root (by using ‘sudo’) and is the correct location to make the package available to all users of the computer.

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 to 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_typ=- 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_typ=/  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$ ncrcat  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.

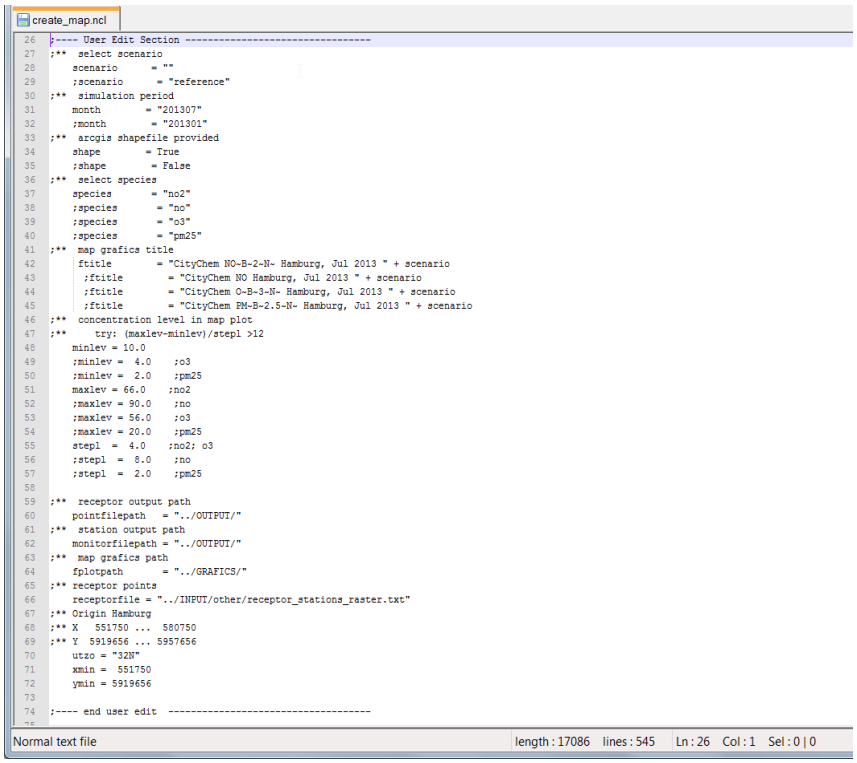


Figure 14: Screenshot of the user edit section in routine create_map.ncl.

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 14). An example map of NO_2 monthly average concentration for Hamburg (using the example data, section 7) is shown in Figure 15. Run the *.ncl script as:

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

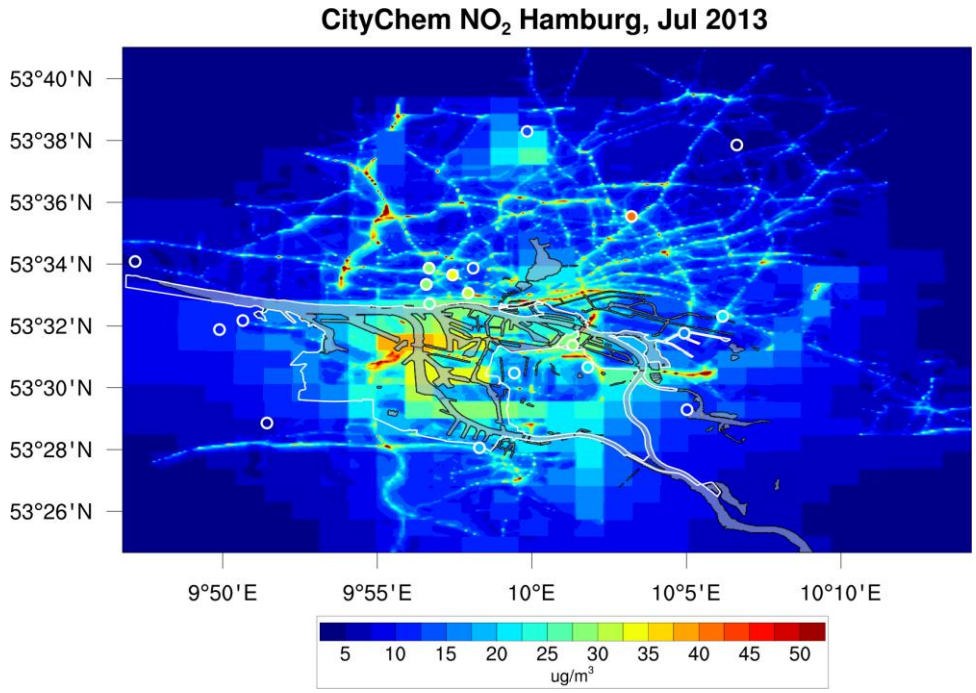


Figure 15: Spatial map of NO_2 monthly concentration average (in µg m^{-3}) for Hamburg city plotted with the routine create_map.ncl. White outline illustrates harbour areas and black outline illustrates water bodies. Open white circles denote stations.

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 16). 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 17 shows an example of the vertical profile plot of hourly NO_2 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 ;**  grafics 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 16: Screenshot of the user edit section in routine vertical_profile.ncl.

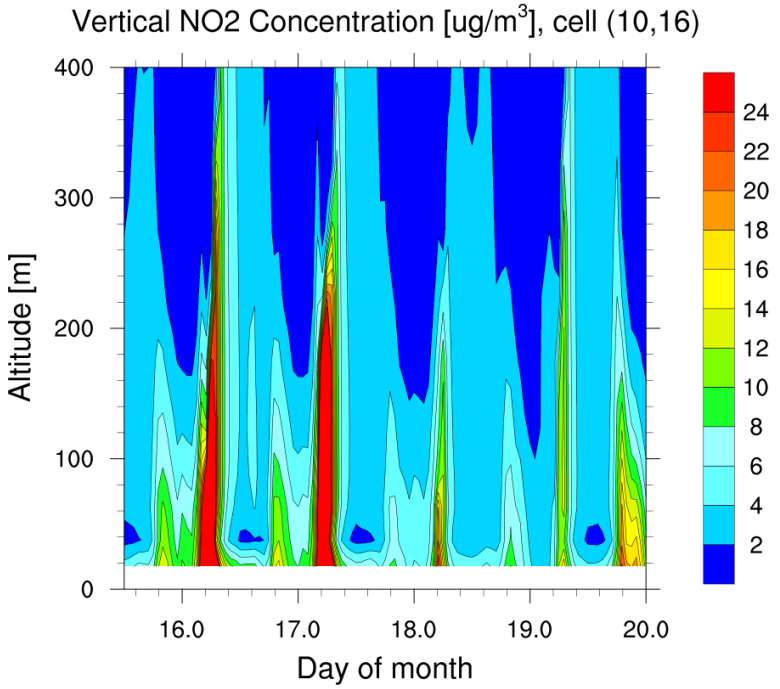


Figure 17:Hourly time slices of the vertical profile of NO_2 mixing ratio (in $\mu g\ m^{-3}$) for main grid cell (10,16) for 4.5 days in July 2013 (starting at noon of 15th July) as contour plot created by routine vertical_profile.ncl.

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 postcchem/obsdat/. Here we take the hourly averaged concentrations of NO_2 as example for plotting a time series comparison.

For plotting the time series of hourly modelled and measured air concentration of NO_2, the NCL routine "plot_no2_series.ncl" is provided in subdirectory postcchem/. The routine reads the netCDF file **statmonihour.nc** and the monitoring data CSV file in postcchem/obsdat/ and produces image files (*.png) showing the time series of modelled and measured hourly averaged NO_2 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 18).

Run the *.ncl script as:

{your-citychem}/postcchem$ **ncl plot_no2_series.ncl**

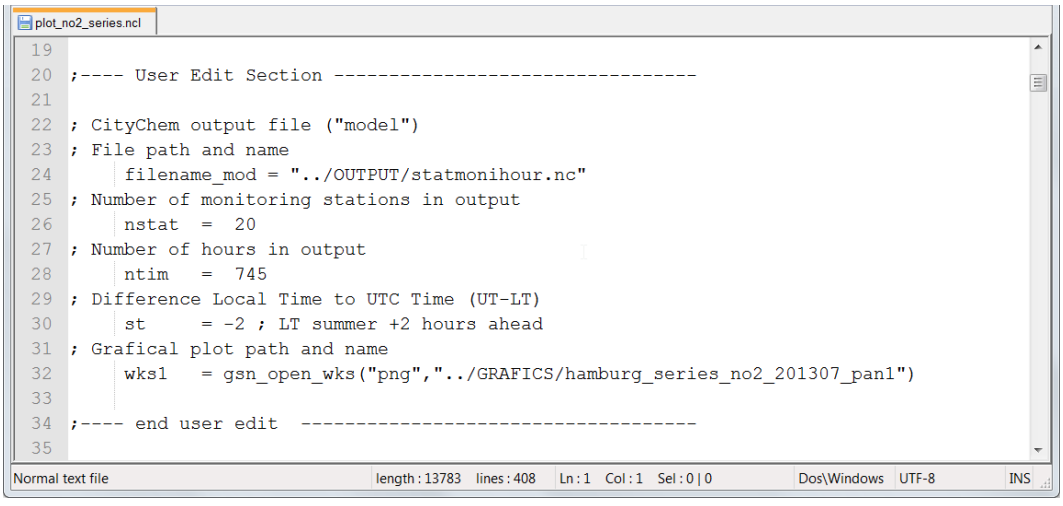


Figure 18: Screenshot of the user edit section in routine plot_no2_series.ncl.

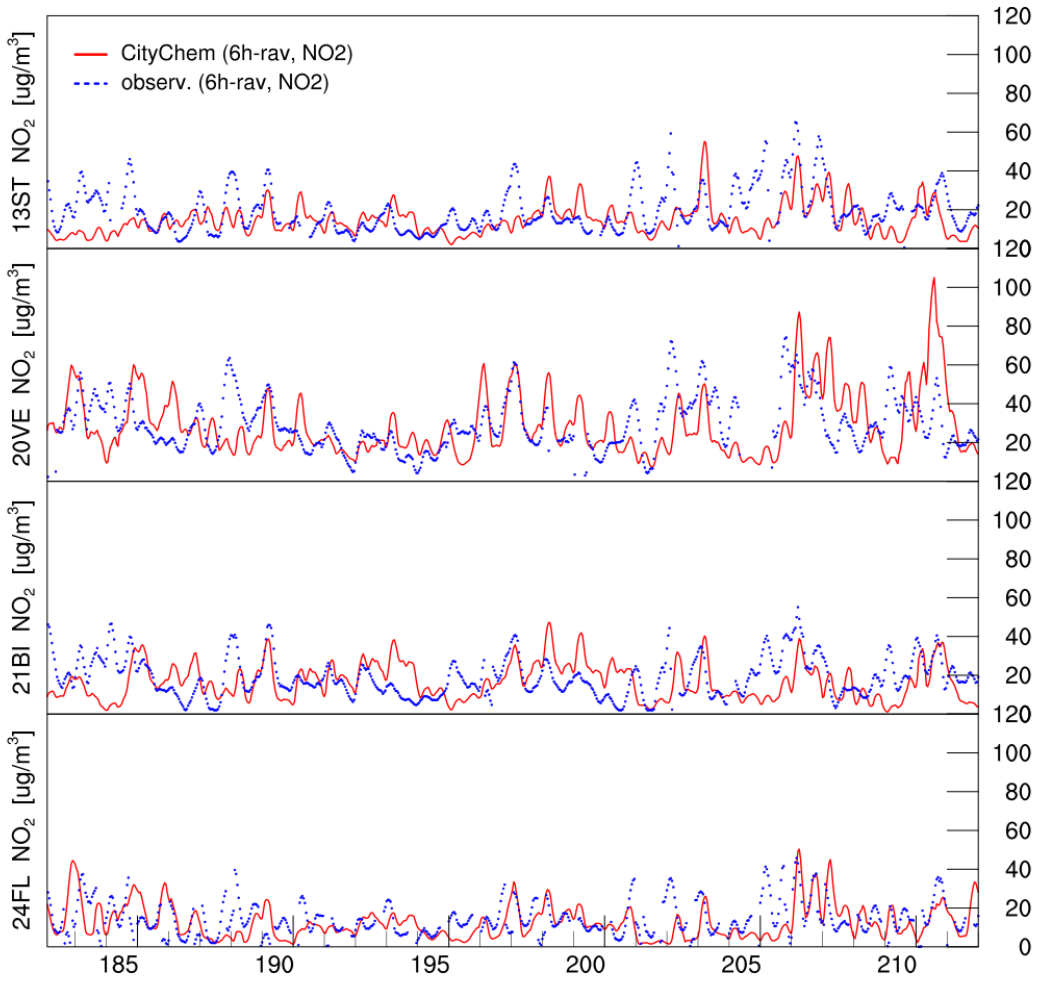


Figure 19: Time series of hourly averaged NO_2 concentrations in July 2013 (time given as day of year) plotted with the routine plot_no2_series.ncl. From top to bottom, the figure parts show model results (red lines) and measurements (blue dots) for the monitoring stations Sternschanze (13ST), Veddel (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 19.

An alternative for plotting the modelled time series in an Excel spreadsheet or other table calculation software (LibreOffice Calc, OpenOffice Calc) is to open the netCDF file **statmonihour.nc** in Panoply netcdf Data Viewer (https://www.giss.nasa.gov/tools/panoply/). Then select variable "NO2" and export as *.csv file, see Figure 20. This export the NO2 concentration values of all stations line-by-line to a *csv file. This can be opened in Excel (or other table calculation software). Every second line is empty and has to be deleted. Each column represents a station; the order of the station is exactly the same as defined in the station/receptor raster file, receptor_stations_raster.txt (section 3.7.2).

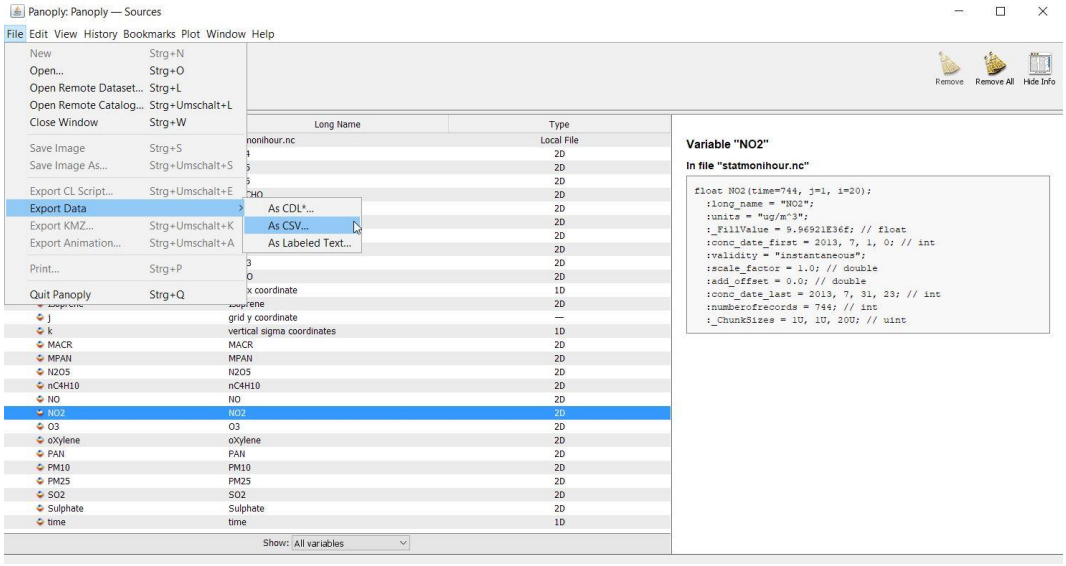


*Figure 20: Export of NO2 as *csv file in Panoply.*

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 postcchem/. 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 postcchem/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 21). 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.

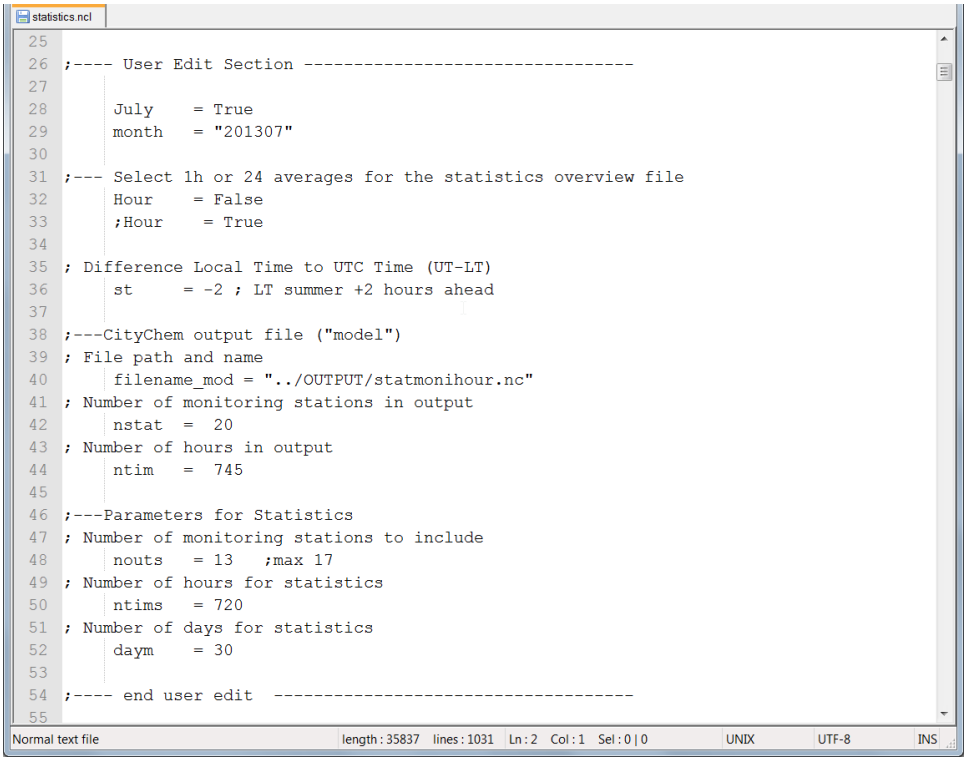


Figure 21: Screenshot of the user edit section in routine statistics.ncl.

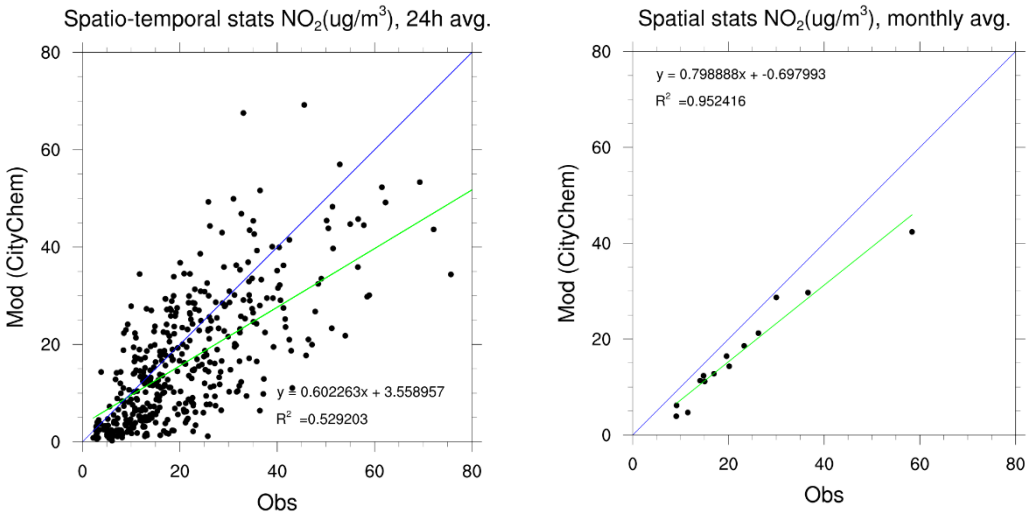


Figure 22: Regression plots of modelled NO_2 concentration versus monitored NO_2 concentration when including 14 monitoring stations in Hamburg using (a) 24-hour averages and (b) monthly averages. Blue line denotes the 1:1 line and the green line is the regression line. Regression parameters are included 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 22a) and for the monthly average (Figure 22b) of the NO_2 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 (R)

The script "openAir_evaluation.R" provided in subdirectory postcchem reads the CityChem output netCDF file **statmonihour.nc** as well as observations for NO_2 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 23.

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 24.

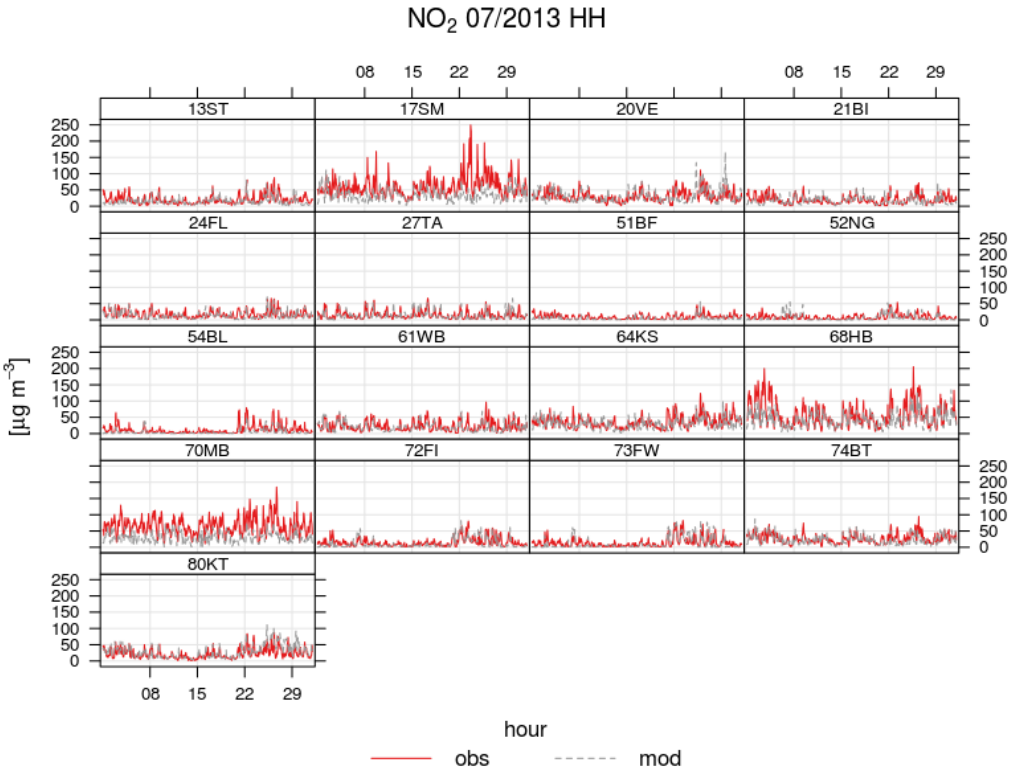


Figure 23: OpenAir time series plots generated by timePlot for observed and modelled data grouped by station.

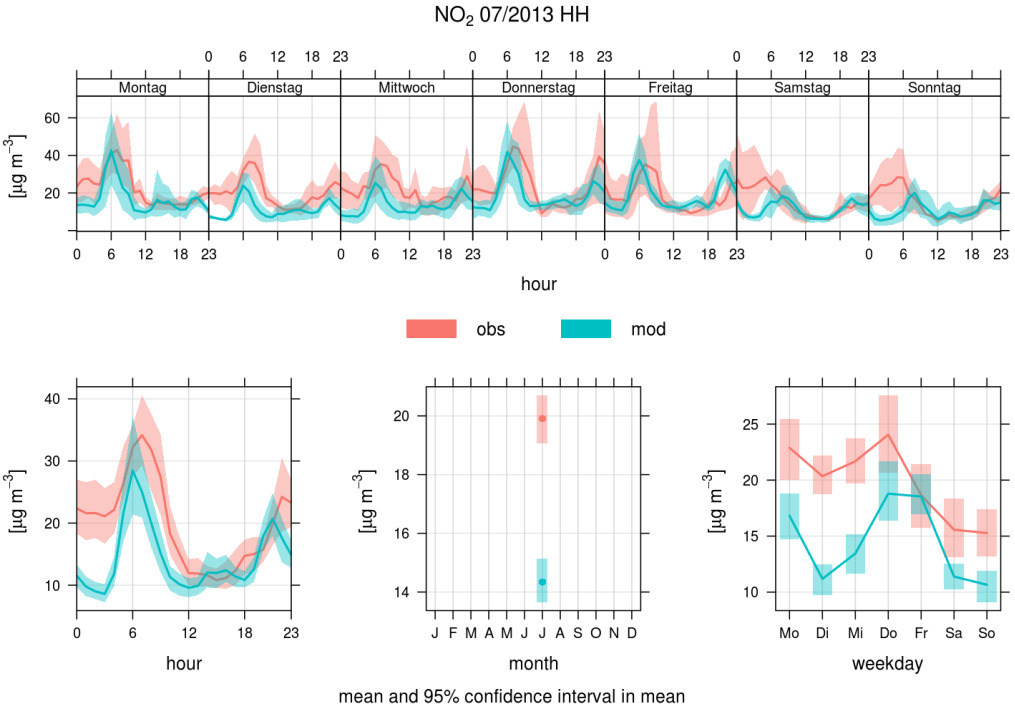


Figure 24: OpenAir temporal variation plots (diurnal and weekly profiles) generated by timeVariation for the modelled and observed data.

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 25 shows an example diagram which is grouped by the modelled stations.

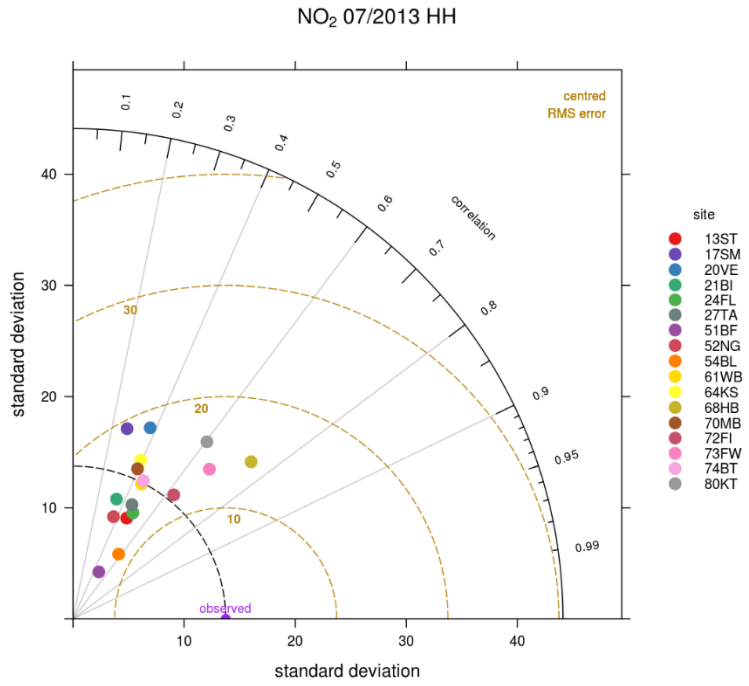
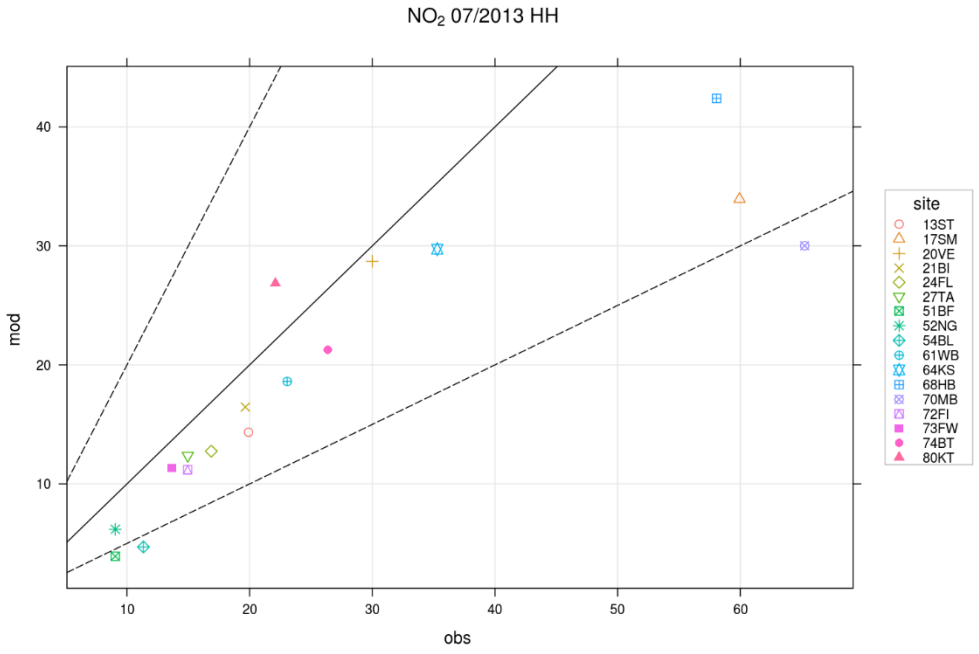


Figure 25: OpenAir generated Taylor diagram of observed and modelled data, grouped by stations.

The fifth and sixth plot is a scatterplot of modelled versus observed data at the stations. The scatterplot function is used for this. Figure 26 (top panel) shows an example plot with monthly averaged data. The bottom panel shows an example plot with daily averaged data, grouped by station.

(a)



(b)

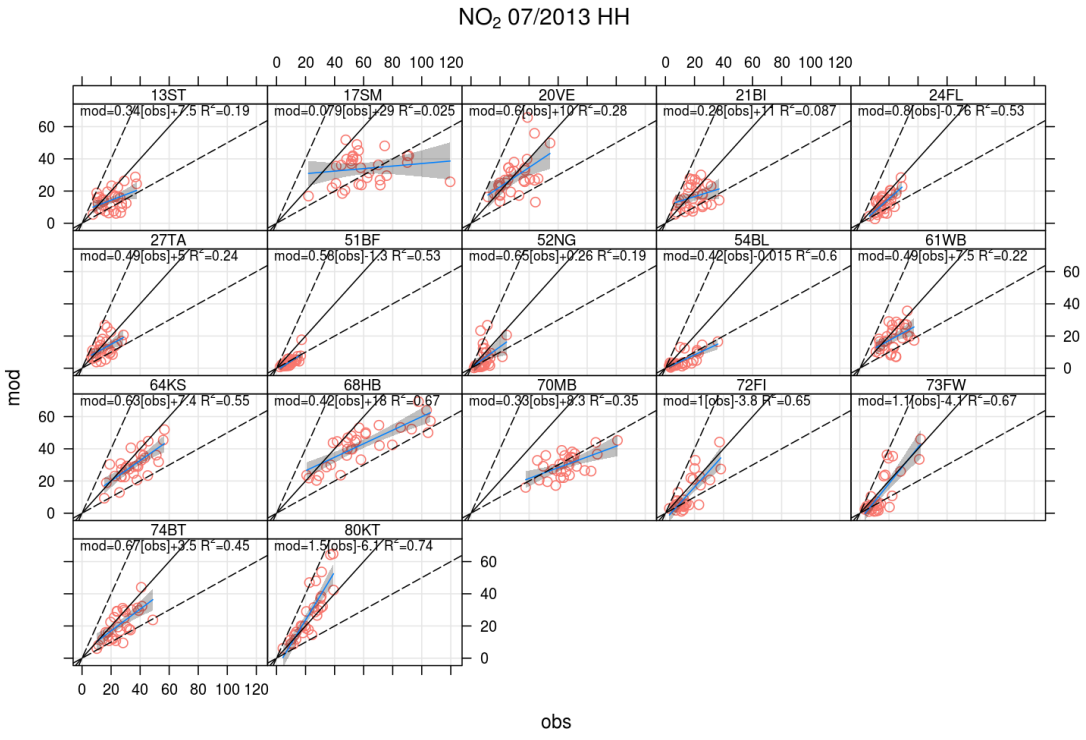


Figure 26: OpenAir scatterplot of (a) monthly averaged data and (b) daily averaged data, grouped by station. The analysis with OpenAir includes all available 17 monitoring stations in Hamburg.

5.6 Spatial lon-lat maps (R)

For plotting surface air concentration fields in the form of spatial maps, the R routine "create_ggmap.R" is provided in the subdirectory postcchem/. The routine reads the netCDF files **statrecp.nc** (or **statmain.nc**) and **statmoni.nc**, and produces image files (*.png) showing spatial colour contour maps (with latitude and longitude coordinates) of the modelled receptor grid concentrations and the monitor station concentrations (as black circles). Additionally, the background map from OpenStreetMap (OSM) underlays the partially transparent colour contours of the concentration map.

Moreover, it is optionally possible to display single or several land use land cover classes (short: LULC) for European cities from the Copernicus Urban Atlas, such as port areas, water bodies or urban green areas, to be applied as overlay on the concentration map.

The Urban Atlas provides harmonised land cover data for ~700 cities in the EU28 and EFTA countries. To apply specific LULC as overlay on the concentration map, it is first necessary to check if the city of interest is available at the Urban Atlas website, and then to download a GIS shapefile for the city containing the LULC data. The download URL is: https://land.copernicus.eu/local/urban-atlas/urban-atlas-2012 (Make sure that the shapefile path contains all additive file types (*.cpg, *.dpf, …), distributed with the *.shp file, in the same folder!).

Before starting the "create_ggmap.R" script, several paths and options have to be edited in the user input section at the beginning of the script (lines 1−27). Open the script in a text editor of your choice and edit the following:

1. The full path of the statrecp.nc (or statmain.nc) output file (line 4)
2. The full path of the statmoni.nc output file (line 7)
3. The full path of the receptor raster file (*.txt) as used in the model setup (line 10)
4. The number of monitoring stations in statmoni.nc (line 13)
5. The pollutant selected for plotting, e.g. "NO2", "O3", "PM25" (line 17)
6. The full path of the output folder for the maps to be plotted (line 19), usually this is the subdirectory GRAFICS/
7. If overlaying the plot with one or more LULC from the Urban Atlas is desired, the full path of the downloaded Urban Atlas shapefile needs to be specified in line 15. If not, the variable "ua_2012" needs to be set to FALSE.

After the editing, the script can be started and the plots will be saved to the entered output folder.

Run the script as:

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

The script generates a PNG file with a title containing the name of the selected pollutant name, month, and year as defined in the model output file. Figure 27 shows the plotted map without the Urban Atlas LULC as overlays.

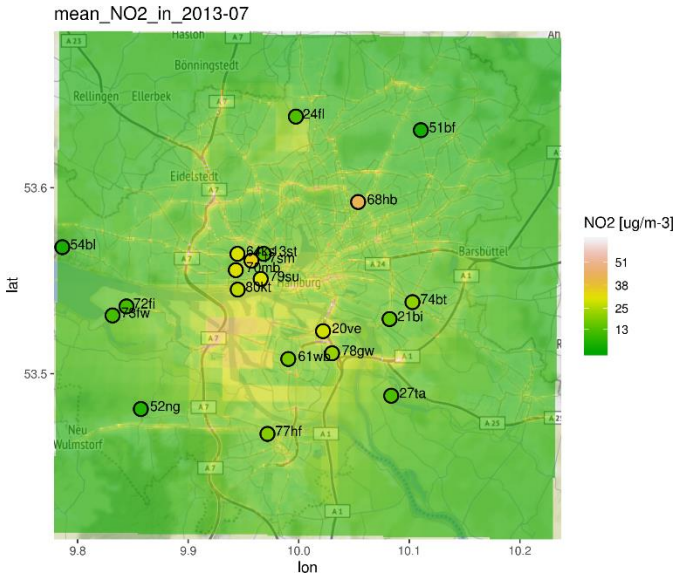


Figure 27: Spatial map of NO_2 monthly concentration average (in µg m^{-3}) for Hamburg city plotted with the routine create_ggmap.R. Open black circles denote stations.

When a Urban Atlas shapefile is defined in line 25 of create_ggmap.R, the spatial map will show port areas and water bodies (Figure 28). To change the LULC to be plotted as overlay, the desired LULC can be edited in lines 112 and 114.

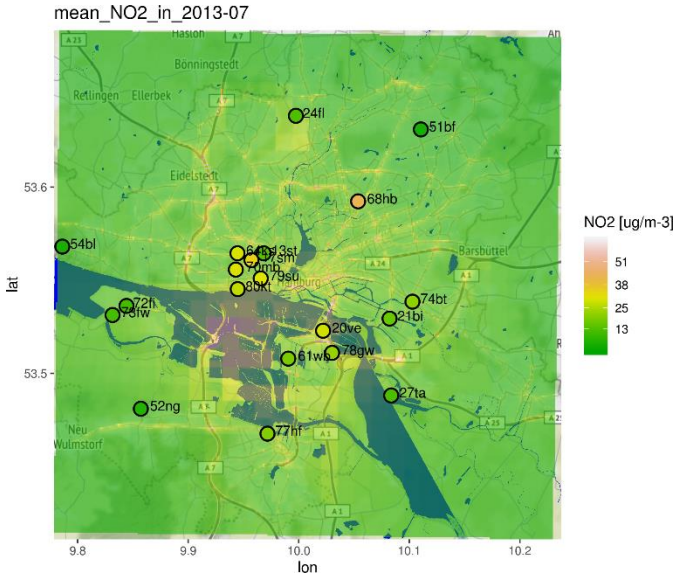


Figure 28: Spatial map of NO_2 monthly concentration average (in µg m^{-3}) for Hamburg city plotted with the routine create_ggmap.R. Open black circles denote stations. Blue areas in the background display port areas and water bodies, which are derived from Urban Atlas.

5.7 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 29). 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.

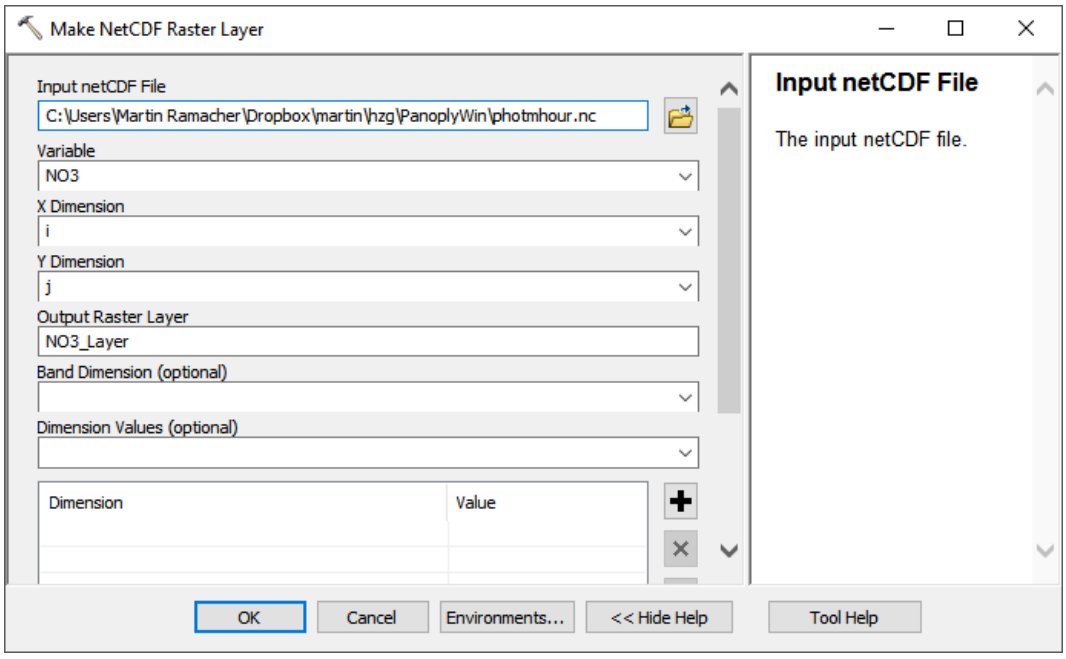


Figure 29: Make netCDF file tool with exemplary inputs.

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 v1.3

MCWIND is an alternative to generate meteorological input for EPISODE-CityChem, if installation of a meteorological model (TAPM, WRF) is not possible or if meteorology based on observational data is preferred. 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. Starting with MCWIND v1.2, observations from more than two additional surface stations and observation from one geostrophic station can be utilized.

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:

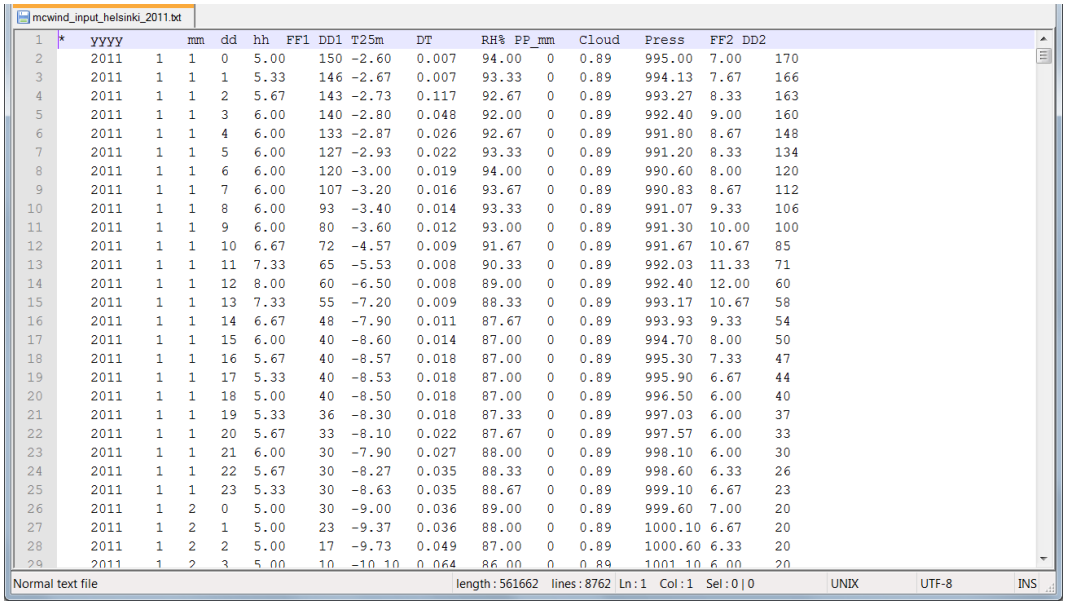
1. Topography input in CityChem format (see section 3.7).
2. Land use input in CityChem format (see section 3.7).
3. Surface roughness input in CityChem format (see section 3.7).
4. Meteorological input from observations as ASCII or CSV text file.

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 between two heights), 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 30 shows an example of the meteorological input file for MCWIND.



mcwind_input_helsinki_2011.txt

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

Normal text file | length : 561662 | lines : 8762 | Ln : 1 | Col : 1 | Sel : 0 | 0 | UNIX | UTF-8 | INS

Figure 30: 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, FF3, DD3
end do
```

Where the variables represent the following

yyyy	Year in format YYYY
mm	Month in format M (no leading zero)
dd	Day in format D (no leading zero)
hh	Hour in format H (no leading zero)
FF1	Wind speed at meteorological tower (in m/s)
DD1	Wind direction at meteorological tower (in degrees)
T25	Air temperature at user-defined reference height (the upper height) at a meteorological tower (in °C)
DT	Temperature gradient *dT* at meteorological tower (in K/m), calculated as temperature difference between an upper and a lower measurement height.
RH	Relative humidity at meteorological tower (in %)
PP	Precipitation rate (in mm/h)
CLF	Cloud fraction at meteorological tower (fraction 0 … 1)
PRS	Sea surface level pressure (hPa or mbar)
FF2	Wind speed at additional station 2 (m/s)
DD2	Wind direction at additional station 2 (degrees)
FF3	Wind speed at additional station 3 (m/s)
DD3	Wind direction at additional station 3 (degrees)

Continue with wind speed and wind direction for any further surface stations:

do *n* = 4, *NSURF*	
FFn	Wind speed at additional station n (m/s)
DDn	Wind direction at additional station n (degrees)
end do	

Running MCWIND to produce input for EPISODE-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 31 shows this part of “run_file.asc”.

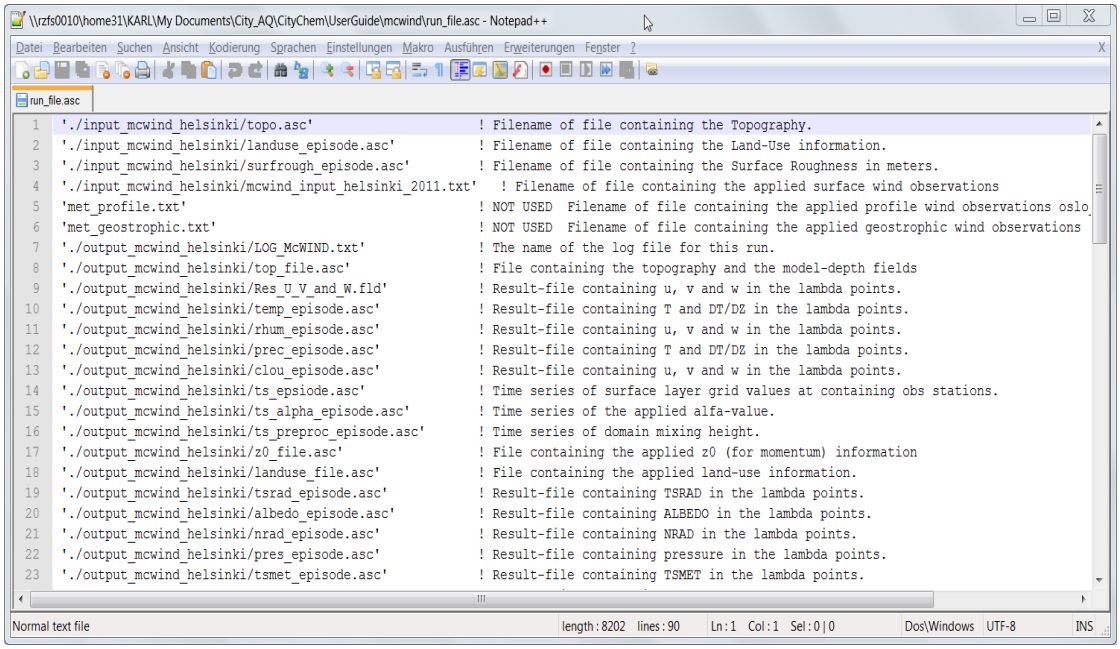


Figure 31: Screenshot of the input/output part of the "run_script.asc" configuration file for MCWIND.

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

The entries in the lines that follow below the layer depths should not be changed.

It continues 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 (upper height).
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)
NAME3	*Name of the second additional station (station 3)*
X3, Y3	*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)*
HW3	*Height (m above ground) for wind measurement at station 3*
Z03	*Surface roughness at station 3 (do not change)*
PWR3	*Surface PWR value at station 3 (do not change)*
SCL3	*Surface SCALE value at station 3 (do not change)*

Continue the same sequence for any further additional surface stations.

NAMEn	*Name of the second additional station (station n)*
Xn, Yn	*x-position and y-position of station n (in km) from the model domain origin, i.e. the south west corner of the model domain (space separated)*
HWn	*Height (m above ground) for wind measurement at station n*
Z0n	*Surface roughness at station n (do not change)*
PWRn	*Surface PWR value at station n (do not change)*
SCLn	*Surface SCALE value at station n (do not change)*

As last station, one geostrophic upper air station can be added optionally. The position of the geostrophic stations should be chosen so that it is closest to the surface station that is the least influenced by the topography. The geostrophic station does not have observations, it is merely used in the construction of the initial wind field.

NGEO	Number of geostrophic meteorological stations. NGEO is set to 1, if a geostrophic upper air station is applied
NAMEGEO	Name of the geostrophic meteorological station ('Geostrophic' would be a good choice)
XGEO, YGEO	x- and y-position of the geostrophic station in km from model origin, i.e. the south west corner of the model domain (space separated)
PWRGEO	Geostrophic station's PWR-value for the IDW procedure, typical value: 2.0
SCLGEO	Geostrophic station's SCALE-value, typical value: 1.0
VWPWR	Geostrophic station's VERTICAL_WEIGHT_PWR value, typical value 0.5 (1 = linear interpolation)

An example of "run_script.asc" with multiple surface stations and belonging observation data (May 2019) for Hamburg is provided in subdirectory preproc/mcwind1.3/example_hamburg.

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 B) 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 v2.8

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 (Figure 32).

Emission data on nitrogen oxides (NO_x), total VOC (NMVOC), carbon monoxide (CO), sulphur dioxide (SO_2), ammonia (NH_3), $PM_{2.5}$, PM_{10}, and total particle number (PN) can be provided. 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 creates hourly varying emission input for point sources, line sources and area source categories using sector specific temporal profiles and vertical profiles, based on annual totals of emissions. Temporal profiles from the SMOKE-EU model are applied (*Bieser et al.*, 2011).

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.

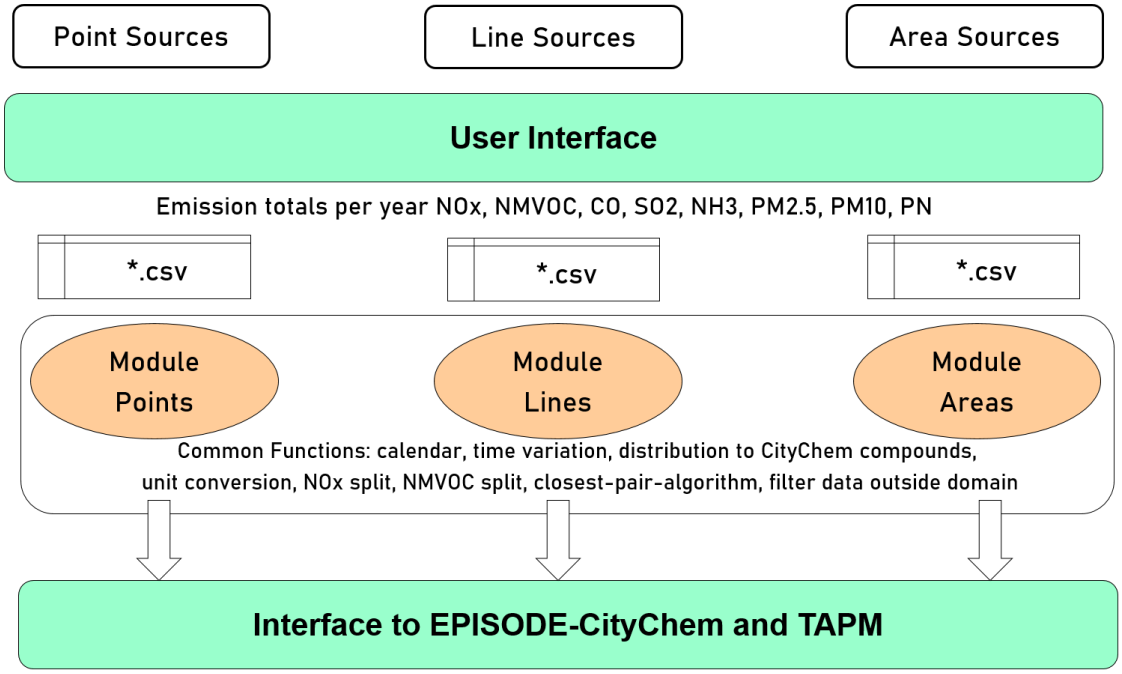


Figure 32: UECT Workflow.

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.

UECT ***starting with v2.7*** applies different temporal profiles for the monthly variation of emissions of energy production (SNAP 1) and road traffic (SNAP 7) for major European cities (Rotterdam, Athens, Helsinki, Oslo, Warschaw, Hamburg, Marseille). The chosen monthly profile for a given city depends on the geographic area shown in Figure 33. For cities outside the coloured region, monthly profiles of Hamburg are chosen as default (as in earlier versions of UECT).

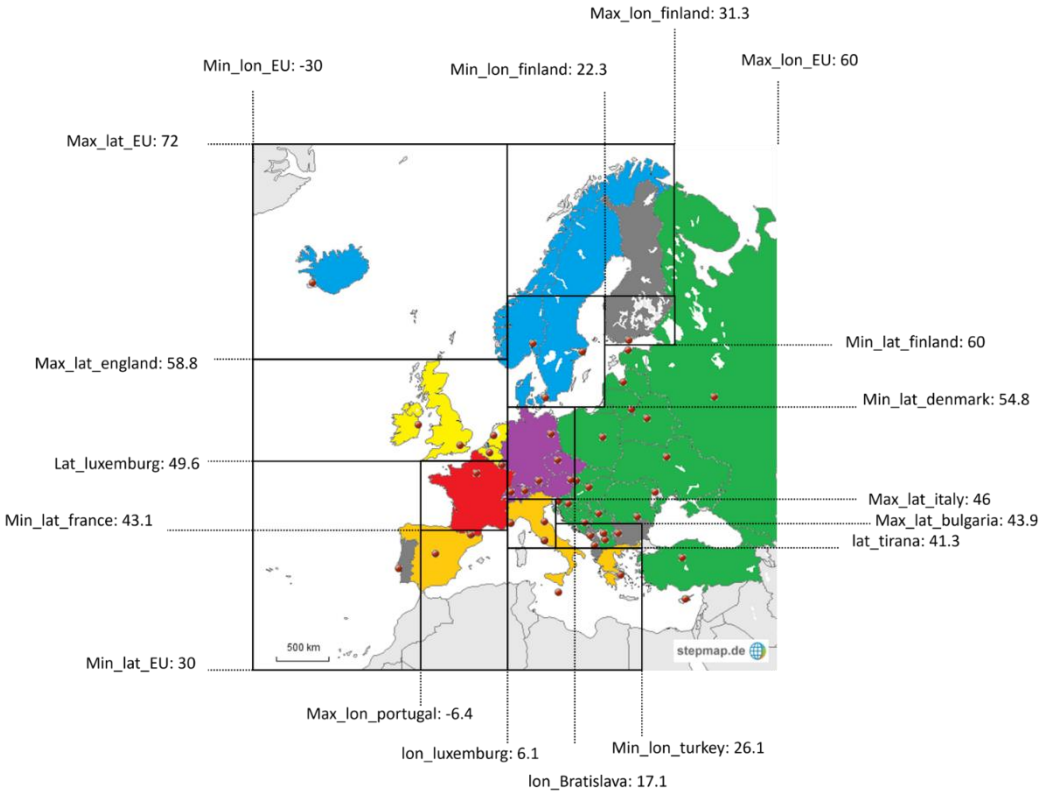


Figure 33: Areas of Eeuropean cities for attribution of monthly factors.

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/uect2.8/. Edit "Makefile.uect" 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/uect2.8$ make –f Makfile.uect
```

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/uect2.8$ cd..
{your-citychem}/preproc$ ln –s ./uect/bin/uect.exe
```

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

```
{your-citychem}/preproc/uect2.8$ make –f Makfile.uect cleanall
{your-citychem}/preproc/uect2.8$ make –f Makfile.uect
```

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, projection (nx, nz, dx, utmx, utmy, utm_zone), and emission type for output of emission files (source). Enter 'PSE' for point source input, 'LSE' for line source input, 'ASE' for area source input or 'ALL' for all source types. Depending on source type (point, line, area) different additional information has to be provided.

Point sources

Edits in the user metadata file: 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 (in kg/year) for each point source.

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. The default parameter values are displayed in Table 8.

UECT produces a point source file with the standard configuration of CityChem compounds in the order given in Table 4. The point source file is a mandatory input in EPISODE-CityChem simulations.

Table 8: Default values of point source parameters. The values were derived from average stack parameters of ca. 8700 point sources in North Rhine Westphalia, Germany.

Point source parameter	SNAP 1	SNAP 3	SNAP 4	SNAP 5	SNAP 6	SNAP 9	SNAP 10
Stack height (m) if NOx emissions are > 15000 t/y	20.0	18.0	17.0	27.7	12.0	16.0	8.0
Stack height (m) if NOx emissions are < 15000 t/y	29.0	18.0	17.0	27.7	12.0	16.0	8.0
Stack exit velocity (m/s) if NOx emissions are < 3500 t/y	10.0	7.63	8.58	9.47	8.21	10.3	7.56
Stack exit velocity (m/s) if NOx emissions are > 3500 t/y	10.0	7.63	8.58	9.47	8.21	16.0	7.56
Exhaust gas temperature (°C) if NOx emissions are < 3500 t/y	147.8	160.1	51.99	42.07	56.56	213.5	22.38
Exhaust gas temperature (°C) if NOx emissions are > 3500 t/y	147.8	160.1	51.99	42.07	56.56	271.7	22.38
Stack radius (m)	0.22	0.29	0.26	0.4	0.3	0.17	0.75

Line sources

Edits in the user metadata file: 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.

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

For the lines source emissions, different diurnal profiles are applied for weekdays and weekends. The composition of the vehicle fleet assumes a fraction of 10% heavy duty and commercial vehicles. An NO_2-to-NOx ratio of 0.3 is applied to recalculate NO_2 emissions because of the expected higher real-world NO_2 emissions from diesel vehicles. Mobile source emissions are an important source of HONO in the atmosphere. HONO-to-NOx ratios in vehicle exhausts are found within a range of 0.29–1.7% (*Kurtenbach et al.*, 2001). UECT ***starting with v2.3*** applies a HONO-to-NOx emission ratio of 2.0 % for all vehicles.

Figure 34 shows the weekday profile of line source emissions.

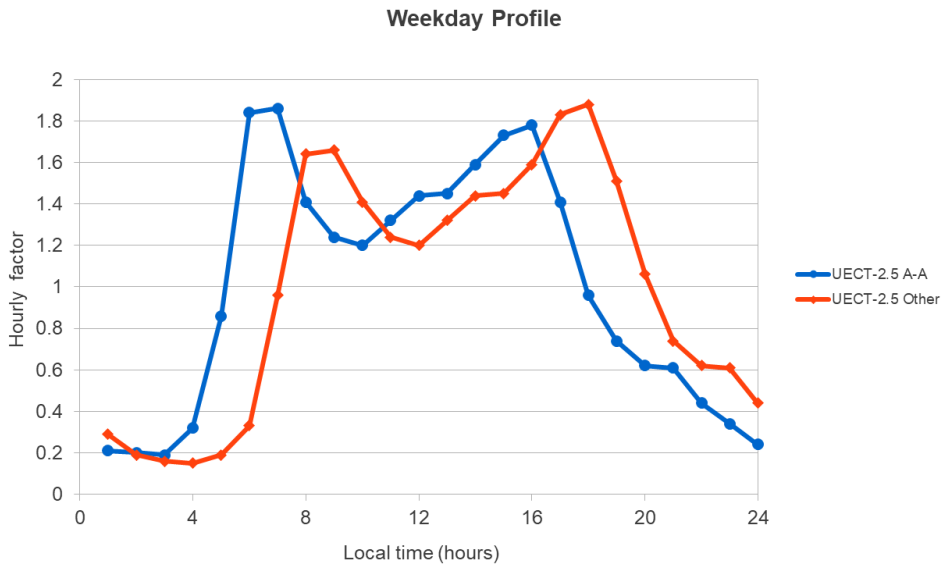


Figure 34: Weekday profile of line source emissions. In UECT v2.3 and later versions, the months April to August are treated with the blue profile. In v2.4 and later versions, the weekday profile for the remaining months of the year are treated with the red profile. In UECT v.2.2 and earlier versions, all months had the same weekday profile.

The emission of particle number (PN) from line sources is built into UECT (since v.2.5). Yearly total PN emissions can be created by the model user by applying a constant ratio between the emission factor (EF) of total particle number and the emission factor of NO_X of $3.0x10^{14}$ particles/gNO_X, which is the ratio given by *Ketzel et al.* (2003) for workdays, multiplied by 1.5, to account for possible underestimation of NO_X emissions. A temperature dependence of the particle number emissions based on the observation study of *Klose et al.* (2009) is implemented as follows:

$$EF_{PN}(T) = EF_{PN}(T_{ref}) \cdot (1.28 - 0.02T(\,^{o}C)) \qquad (4)$$

The particle emissions from line sources are distribution over the ten size classes (Table 1), where the percentage fractions in the classes between PN1 and PN10 are 2%, 4%, 14.2%, 56.8%, 15%, 4.8%, 1.6%, 1.5%, 0.1%, and 0%, respectively. The size fractionation is based on the typical traffic size distribution given in *Karl et al.* (2016).

A cold start emission factor is calculated to adjust the temporal profile of vehicle emissions in UECT (since v.2.6). The cold start adjustment and the (non-temperature dependent) monthly time factor profiles are adopted from CAMS-TEMPO v3.1 (Guevara et al., 2021; ECCAD 2022). For the cold start adjustment it is assumed that a share of 25% of the vehicle fleet is in cold start mode at any time. Table 9 gives an overview of the monthly time factors for SNAP7.

Table 9: Monthly time factors for SNAP7 line source emissions for months 1-12 (Jan to Dec).

Data source	1	2	3	4	5	6	7	8	9	10	11	12
UECT v.2.7 calculated for 2012	1.16	1.21	1.09	1.00	0.93	0.84	0.79	0.82	0.89	1.04	1.20	1.02
CAMS-TEMPO V3.1 w/o cold start	1.13	1.12	1.15	1.00	0.95	0.82	0.77	0.79	0.97	1.10	1.32	0.89
CAMS-TEMPO V2.1 with cold start	1.17	1.16	1.12	1.02	0.91	0.85	0.84	0.83	0.89	0.99	1.07	1.14
SMOKE-EU (until UECT v.2.5)	0.88	0.92	0.98	1.03	1.05	1.06	1.01	1.02	1.06	1.05	1.01	0.93

Area sources

Edits in the user metadata file: set emission type to area 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 (in kg/year) 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.

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.

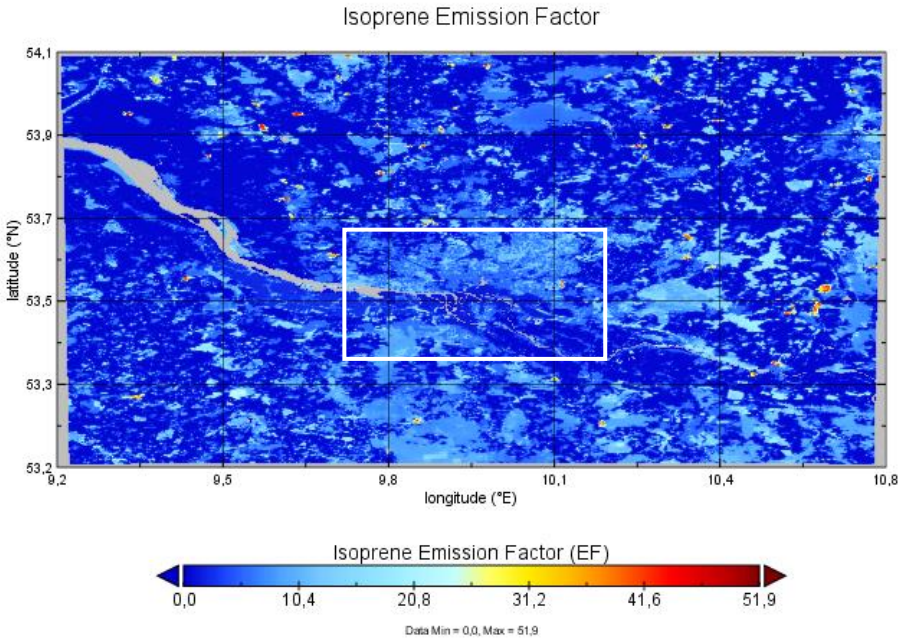
Area emissions of residential heating:
Area emissions of residential heating (SNAP 2) can be generated with dependence on the daily average temperature (heat degree days method). For this procedure, the program **tapm4cc.exe** (section 3.4) has to be run using the option write netCDF output (*ep_fmt* = 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, *Format of emission input files for UECT*). 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.

Area emissions of biogenic VOC for the Hamburg example:
As additional option, UECT, ***starting with v2.3***, provides specific emission of isoprene and monoterpenes for the city domain of Hamburg (see example in section 7). The emission potentials and foliar density for the Hamburg domain is provided as netCDF file "BVOC_4_UECT_HH.nc" which is included in the *example* testdata zip file (see Figure 35). The netCDF file containing air temperature at ground and vertical temperature gradient (T_and_dtdz_*.nc) has to be provided. In order to activate the isoprene emissions in UECT, it is necessary to create an area source file for SNAP11 that includes all grid cells of the domain. NCL script "snap11grid.ncl" in the raster folder of UECT creates an area source file for SNAP11 in the required format. The created file has to be appended to the area source CSV file (see below, *Format of emission input files for UECT*) to produce the area emissions of biogenic VOC in UECT for Hamburg. In addition to area emissions of isoprene, area emissions of alpha-pinene (representing T-dependent monoterpenes) and limonene (representing LT-dependent monoterpenes) are produced.

The netCDF files with emission potentials and foliar density for Rhein-Ruhr area (Germany) and Marseille (France) are available upon request.

a)



b)

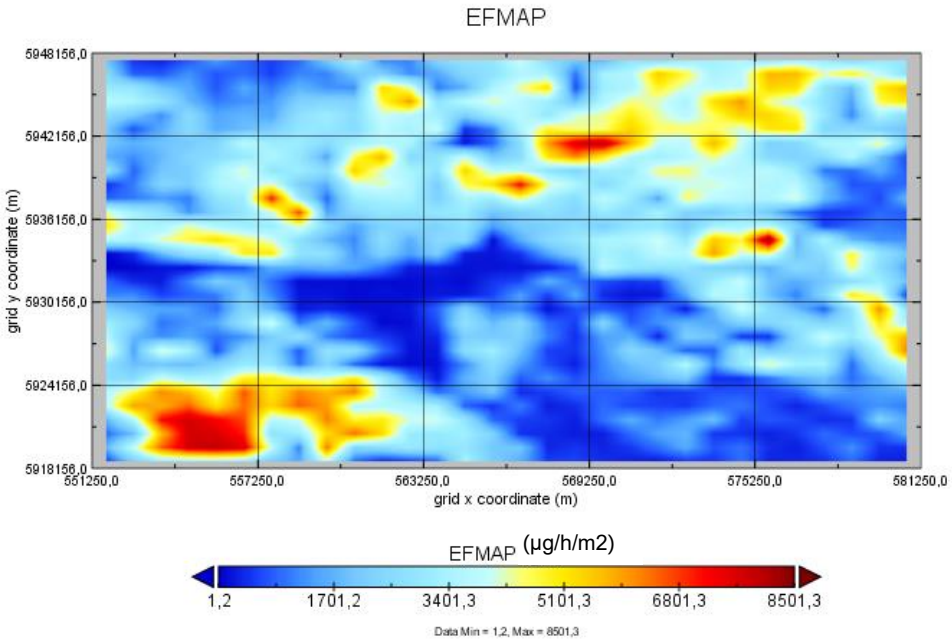


Figure 35: Map of isoprene emission factors for the region of Hamburg: a) isoprene standard emission potential (unit: µg/g/hr) map created based on Copernicus Tree Cover Density and Corine Land Cover; resolution of 100 m; b) interpolated field of emission factor x foliar density (unit: µg/hr/m^2) calculated in UECT; resolution of 1000 m in zoom region of a).

All sources (point, lines and areas)

Edits in the user metadata file: set emission type to all emission sources (source='ALL') if you want to produce all emission files at once. All information specified above has to be provided for a UECT run that produces all emission output (points, lines and areas). This option can be used in UECT, ***starting with v.2.4***.

Format of emission input files for UECT

Emission data for the different source types has to be given to UECT as ASCII files in CSV format (UTF-8 encoded). The emission input files should contain one header line followed line-by-line with the data for each source. The format for the different source types is described in the following. Figure 36 gives an overview of the input format of point, line and area source emission input files.

Data values for one source have to be separated by ',', Missing values have to be denoted by -999.

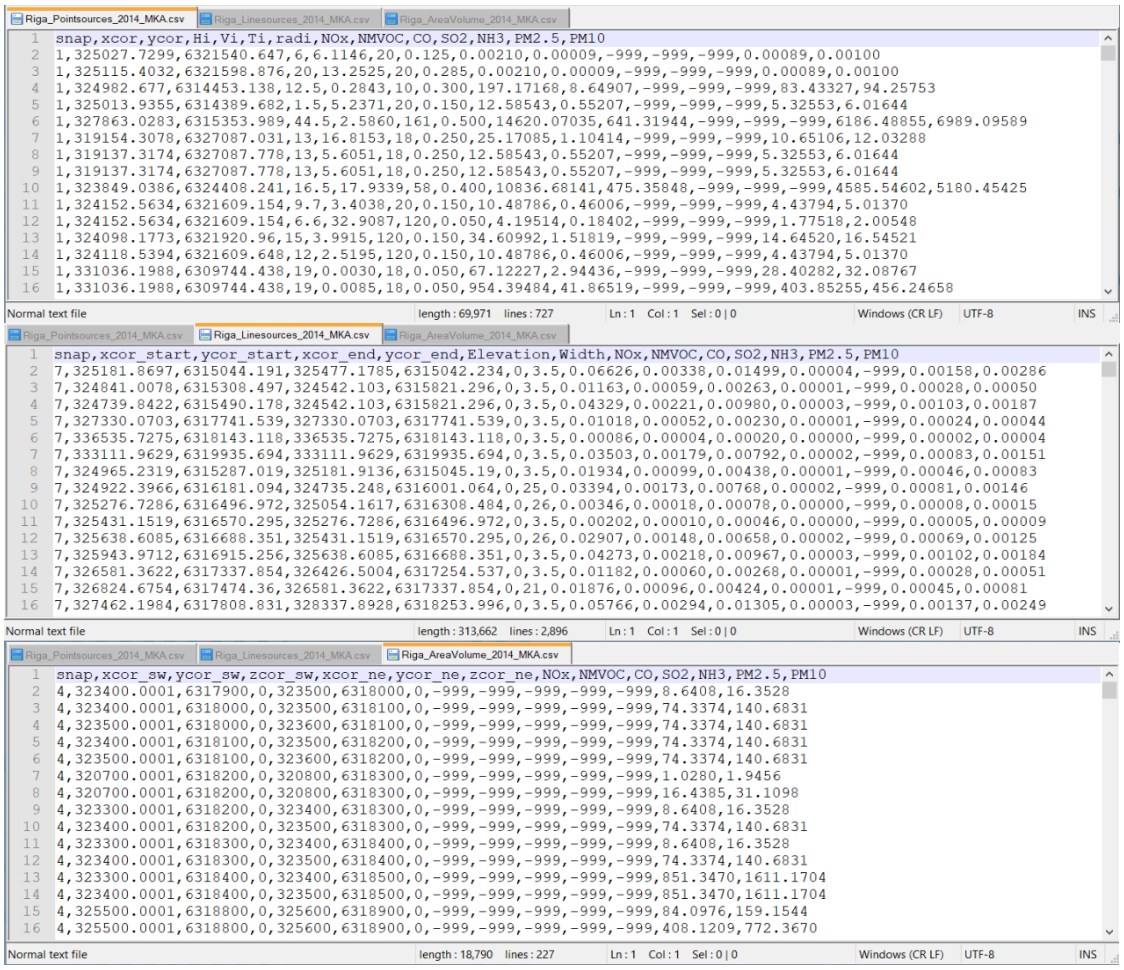


Figure 36: Screenshot of excerpts from the different emission input files for UECT: point source emissions, line source emissions, and area source emission (from top to bottom).

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),PN(ip)
enddo
```

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

snap	SNAP emission sector code (1, …, 10)
xcor	west-east (UTMx) local coordinate of the stack (m). Give as truncated UTM x-coordinate (without the two leading digits for UTM zone).
ycor	south-north (UTMy) local coordinate of the stack (m).
Hi	stack height above the ground (m).
Vi	stack exit velocity (m/s).
Ti	stack exit temperature (°C).
radi	internal stack radius (m).
NOx	emission rate of nitrogen oxides, NOx (kg/year).
NMVOC	emission rate of non-methane volatile organic compounds, NMVOC (kg/year).
CO	emission rate of carbon monoxide, CO (kg/year).
SO2	emission rate of sulphur dioxide, SO_2 (kg/year).
NH3	emission rate of ammonia, NH_3 (kg/year).
PM2.5	emission rate of particulate matter with aerodynamic diameter ≤ 2.5 μm, $PM_{2.5}$ (kg/year).
PM10	emission rate of particulate matter with aerodynamic diameter ≤ 10 μm, PM_{10} (kg/year).
PN	total particle number emission rate (particles/year).

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 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),PN(L)
enddo
```

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

snap	SNAP emission code for traffic (7 or 8).
xcor_start	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).
ycor_start	south-north (UTMy) local coordinate of the line source (m) as start coordinate of the road link.
xcor_end	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).
ycor_end	south-north (UTMy) local coordinate of the line source (m) as end coordinate of the road link.
elevation	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.
width	width of the street, including all lanes (m).
NOx	emission rate of nitrogen oxides, NOx (g/s).
NMVOC	emission rate of non-methane volatile organic compounds, NMVOC (g/s).
CO	emission rate of carbon monoxide, CO (g/s).
SO2	emission rate of sulphur dioxide, SO_2 (g/s).
NH3	emission rate of ammonia, NH_3 (g/s).
PM2.5	emission rate of particulate matter with aerodynamic diameter ≤ 2.5 μm, $PM_{2.5}$ (g/s).
PM10	emission rate of particulate matter with aerodynamic diameter ≤ 10 μm, PM_{10} (g/s).
PN	total particle number emission rate (particles/year)

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),PN(ia)
enddo
```

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

snap	SNAP emission sector code (1, …, 10).
xcor_sw	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).
ycor_sw	south-north (UTMy) local coordinate (m) of the lower left corner (SW-corner) of the area emission cell.
zcor_sw	elevation above ground (m) of the lower left corner (SW-corner) of the area emission cell.
xcor_ne	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).
ycor_ne	south-north (UTMy) local coordinate (m) of the upper right corner (NE-corner) of the area emission cell.
zcor_ne	elevation above ground (m) of the upper right corner (NE-corner) of the area emission cell.
NOx	emission rate of nitrogen oxides, NOx (kg/year).
NMVOC	emission rate of non-methane volatile organic compounds, NMVOC (kg/year).
CO	emission rate of carbon monoxide, CO (kg/year).
SO2	emission rate of sulphur dioxide, SO_2 (kg/year).
NH3	emission rate of ammonia, NH_3 (kg/year).
PM2.5	emission rate of particulate matter with aerodynamic diameter ≤ 2.5 μm, $PM_{2.5}$ (kg/year).
PM10	emission rate of particulate matter with aerodynamic diameter ≤ 10 μm, PM_{10} (kg/year).
PN	total particle number emission rate (particles/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$ ./uect.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 Installation of IOAPI

The utilities **WRF4CC** (section 6.4) and **BCONCC** (section 6.5) require the installation of the **IOAPI 3.2 library** on your system.

A modified Fortran program from IOAPI's m3tools is used in both utilities for cutting out the EPISODE-CityChem model domain and for bilinear interpolation of hourly fields to the required horizontal resolution of the model domain.

IOAPI3.2 should be installed with the same compiler as the CityChem executable, i.e. either with gfortran (e.g. using ioapi's Makeinclude.Linux2_x86_64gfort) or with intel fortran (e.g. using ioapi's Makeinclude.Linux2_ x86_64ifort). The subdirectory preproc/bconcc2.5/ioapi3.2 contains all files required for the installation of IOAPI3.2 on your computer. Note that from BCONCC v2.2, we use IOAPI 3.2 while earlier BCONCC versions used IOAPI 3.1.

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).

The installation of the IOAPI library needs to be done only one time. In the automated installation with the installation script installcc.csh (section 2.5) the following parameters are set in the configuration file (config.user):

IOAPI	**IOAPI 3.2 binary (default: Linux2_x86_64gfort)**
IOAPIDIR	**Full path of IOAPI 3.2 installation (IOAPI is included in the distribution)**

The IOAPIDIR path is normally the subdirectory preproc/bconcc2.5/ioapi3.2/ included in the distribution, unless you want to use a different or previously built IOAPI3.2 installation.

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/bconcc2.5/ioapi3.2/. 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/citychem1.8/preproc/bconcc2.5/ioapi3.2

- 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/bconcc2.5/ioapi3.2$ **make all**

If you want to refresh the built, type:
{your-citychem}/preproc/bconcc2.5/ioapi3.2$ **make clean**
{your-citychem}/preproc/bconcc2.5/ioapi3.2$ **make all**

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.

To create m3cpleplus.exe type:

{your-citychem}/preproc/bconcc2.5/ioapi3.2$ **make m3p**

Then change to next higher directory,

{your-citychem}/preproc/bconcc2.5/ioapi3.2$ **cd ..**

And create a symbolic link to the executable:

{your-citychem}/preproc/bconcc2.5$ **ln -s ./ioapi3.2/bin/m3cpleplus.exe**

And:

{your-citychem}/preproc/wrf4cc1.2$ **ln -s ../bconcc2.5/ioapi3.2/bin/m3cpleplus.exe**

6.4 WRF4CC v1.3

In this section it is explained how to prepare the 2-D and 3-D meteorological fields and the auxiliary files for EPISODE-CityChem based on output of the Weather Research and Forecasting WRF model (http://www.wrf-model.org/). WRF output from the innermost nest (d03) should be used, merged into daily files containing 25 hours, each including the last hour of the previous day. A list of required meteorological variables from WRF is given in section 3.4.2.

The WRF post-processor requires the **IOAPI 3.2 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 WRF output fields to the required horizontal resolution of the model domain. For Installation of IOAPI and the m3tools program, refer to section 6.3.

The conversion of WRF output is done in three steps:

A) Conversion of WRF output using the Meteorology-Chemistry Interface Processor (MCIP) processor.
B) Re-projection to the city domain and interpolation to UTM projection.
C) Producing the meteorological input files for EPISODE-CityChem.

A) Conversion of WRF output using the MCIP processor

The Meteorology-Chemistry Interface Processor (MCIP) from US EPA (https://www.epa.gov/cmaq/meteorology-chemistry-interface-processor) ingests output from the Weather Research and Forecasting (WRF) Model to prepare the meteorology files that can be used within the CMAQ Modelling System. Where possible, MCIP uses data directly from the meteorological model to maximize consistency with the CMAQ Modelling System. When specific atmospheric fields are not explicitly output by WRF, MCIP uses scientific algorithms to create those fields for CMAQ. The output files from MCIP are then further converted in steps B) and C). Version 5.0 of the MCIP processor is included in the EPISODE-CityChem distribution.

If the MCIP program has not been installed by the automated routine installcc.csh, the following needs to be done. Change to subdirectory /preproc/wrf4cc1.3/mcip5.0/src/. In this folder you need to edit the Makefile. The following editing needs to be done in Makefile:

- Edit the Fortran compiler name, e.g.:

```
FC = gfortran
```

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

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

- Edit IOAPI_ROOT path, path of the ioapi installation on your computer, e.g.:

```
IOAPI_ROOT = {your-citychem}/preproc/bconcc2.5/ioapi3.2/
```

- Edit BIN, the binary of the ioapi installation on your computer, e.g.:

```
BIN = Linux2_x86_64gfort
```

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

```
{your-citychem}/preproc/wrf4cc1.3/mcip5.0/src/$ make
```

Change to the preproc/wrf4cc1.3 directory and create a symbolic link to the tool:

```
{your-citychem}/preproc/wrf4cc1.3/mcip5.0/src$ cd ../..
{your-citychem}/preproc/wrf4cc1.3$ ln –s ./mcip5.0/bin/mcip.exe
```

After installation of MCIP, change to subdirectory /preproc/wrf4cc1.3/scripts and edit the MCIP run script run_mcip.csh. Again, WRF output files should be from the d03 nest and contain 25 hours including the last hour of the previous day. The script can be used for multiple consecutive days of the same month.

The naming convention of the WRF output file is:
'wrfout_d03_YYYY-MM-DD_00_00_00_25h'
where YYYY is the year, MM the month (01, …) and DD (01, …) the day as Gregorian date.

The user needs to edit the following parameters:

CoordName:	name of the WRF d03 grid (16-character maximum)
GridName:	default is d03
DataPath:	path of the WRF output files
YEAR:	year of the WRF simulation
MM:	month of the first WRF output file
MM2	same as MM
DDAY	day of first WRF output file
MCIP_DAY	start date in Julian date format (YYYYDDD)
MCIP_DAYEND	end date in Julian date format (YYYYDDD)
InMetFiles	list of consecutive WRF daily output files, separated by '\'

The following directory has to be created (mkdir) in the specified WRF output path ($DataPath) for the output files from MCIP:
$DataPath/mcip/$GridName

Then run the script on a Linux/Unix computer by typing:

```
{your-citychem}/preproc/wrf4cc1.3/scripts$ ./run_mcip.csh
```

This will create several output files for each day in the specified output directory.

The relevant output files of MCIP include GRIDCRO2D*, METCRO2D*, METCRO3D*, METDOT3D* and GRIDDESC.

B) Re-projection to the city domain

After successfully producing the MCIP output files, the specific city domain needs to be cut out from the d03 grid and bilinear interpolation has to be performed to obtain the required horizontal resolution for the meteorological fields. The tool for this step is **m3cpleplus.exe**. See section 6.3 for installation of this program.

For each city model domain a GRIDDESC file has to be created containing the information about the geographic projection, domain extent and resolution. This file is based on the GRIDDESC output from the MCIP run.

Change to subdirectory preproc/wrf4cc1.3/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 (same as CoordName in the MCIP run script).
Line 3:
1st value: 5 (= UTM projection, do not change)
2nd value: 34.000 (= UTM zone)
3rd value: Latitude coordinate of the d03 grid centre; corresponds to the 6th value of the third line in GRIDDESC file of the MCIP output, global attribute YCENT in METCRO3D)
4th value: Longitude coordinate of the d03 centre; corresponds to the 5th value of the third line in GRIDDESC file of the MCIP output, global attribute XCENT in METCRO3D)
5th, 6th value: coordinate of the grid centre of the Cartesian coordinate system, here the Meridian (0.000 degrees) and the Equator (0.000 degrees) (do not change)
Line 4:
‘ ‘
Line 5:
‘ATHENSAUTM’ (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.
3rd value: UTM y coordinate of the Origin (southwest corner) of the city domain.

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
8th value: 0 (= do not change)
Line 7
‘ ‘

Below an example for the city GRIDDESC file of Athens:

```
' '
'LAM_GREECE'
 5     34.000      45.000     24.000     0.000    0.000
' '
'ATHENSUTM'
'LAM_GREECE'    716397.0     4191261.0   1000.000    1000.000  45  45  0
' '
```

Change to subdirectory preproc/wrf4cc1.3/city_input. Edit the file input_conc.txt in Notepad++ or other text editor of your choice. It contains the following lines:

```
Y
NONE
INFILE_SH
ATHENSUTM
OUTFILE_SH
00000
10000
24
OUTFILE_SH
```

If you have changed the name of the city UTM grid in the file GRIDDESC_Yourcity, you need to change the entry in the 4th line ("ATHENSUTM") to the same name.

Then change to subdirectory /preproc/wrf4cc1.0/scripts and edit the m3cple run script run_m3cple_city_wrf.csh.

- Edit ProgDir variable by specifying the path of the subdirectory preproc/wrf4cc1.3, e.g. (not needed if automated installation worked):

```
set ProgDir   =  ${your-citychem}/preproc/wrf4cc1.3
```

- 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_Athens
```

- Edit INFILE_SH variable by specifying the path where the output files of MCIP (GRIDCRO2D, METCRO2D, METCRO3D, METDOT3D) are located and their names.

- Edit OUTFILE_SH variable of the four output files by specifying the path where the output files should be written and the name of the files (replace yyyy by year of the simulation), e.g.:

```
setenv OUTFILE_SH  ${ProgDir}/city_output/gridwrf_utm_2d_yyyy${strDAYin}
setenv OUTFILE_SH  ${ProgDir}/city_output/metwrf_utm_2d_yyyy${strDAYin}
setenv OUTFILE_SH  ${ProgDir}/city_output/metwrf_utm_3d_yyyy${strDAYin}
setenv OUTFILE_SH  ${ProgDir}/city_output/metwrf_utm_3dot_yyyy${strDAYin}
```

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

```
set DAYin     = 1
set Run_Id_end = 31
```

- Edit NEWDATE to define the year of the MCIP files to convert, e.g.:

```
setenv NEWDATE      yyyy${strDAYin}
```

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

```
{your-citychem}/preproc/wrf4cc1.3$ ./run_m3cple_city_wrf.csh
```

This creates four output files for each day in the city_output directory, containing all required variables for producing the meteorological and auxiliary input files for the model simulation.

C) Producing the meteorological input files

After successfully producing the output files with re-projected fields for the specific city domain, the **wrf4cc.exe** program is used to produce the meteorological input and the auxiliary input files for EPISODE-CityChem model simulations. The program also performs vertical interpolation from the WRF model layers to the standard vertical dimension of EPISODE-CityChem (see Table 3). Note that the program applies cubic spline smoothing of the horizontal wind fields (u- and v-wind).

If the program has not been installed by the automated routine installcc.csh, the following needs to be done. Change to subdirectory /preproc/wrf4cc1.3/wrf4cc/. In this folder you need to edit the makefile Makefile.wrfp. The following editing needs to be done in Makefile.wrfp:

- Edit the Fortran compiler name, e.g.:

```
FC = gfortran
```

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

```
NETCDF    = -L/usr/local/netcdf4/lib -lnetcdf -lnetcdff
```

(keep '-L' and '-lnetcdf –lnetcdff')

- Edit NETCDFINC path, the include path of the netcdf4 installation on your computer, e.g.:

```
NETCDFINC = -I/usr/local/netcdf4/include
```

(keep '-I')

- Edit IOAPIDIR path, path of the ioapi installation on your computer, e.g.:

```
IOAPIDIR = {your-citychem}/preproc/bconcc2.4/ioapi3.2/
```

- Edit BIN, the binary of the ioapi installation on your computer, e.g.:

```
BIN = Linux2_x86_64gfort
```

Change the land use class definition:
If the WRF simulation was done with the MODIS-NOAH land use scheme, nothing needs to be changed. If instead the WRF simulation employed the USGS24 land use scheme, the Fortran source code wrf4cchem.f90 in subdirectory /preproc/wrf4cc1.3/wrf4cc/src/ needs to be edited in line 269, by setting the logical lu_usgs to ".true.".

If LM-MCIP instead of MCIP is used, e.g. when using the meteorological model COSMO, then the Fortran source code wrf4cchem.f90 in subdirectory /preproc/wrf4cc1.3/wrf4cc/src/ needs to be edited in line 270, by setting the logical cosmo to ".true.".

Then build **wrf4cc.exe** by typing:

```
{your-citychem}/preproc/wrf4cc1.3/wrf4cc/$ make –f Makefile.wrfp
```

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

```
{your-citychem}/preproc/wrf4cc1.3/wrf4cc$ cd ..
{your-citychem}/preproc/wrf4cc1.3$ ln –s ./wrf4cc/bin/wrf4cc.exe
```

After installation of **wrf4cc.exe**, change to the run script for this step, run_wrf4cc.csh. The script can be found in subdirectory /preproc/wrf4cc1.3/. 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_wrf4cc.csh:

- Edit M3HOME variable by specifying the path of the subdirectory preproc/wrf4cc1.3, e.g.:

```
set M3HOME = ${your-citychem}/preproc/wrf4cc1.1
```

- Edit INPUT_FILE1 to INPUT_FILE4 variables by specifying the path of output files from step B), e.g.:

```
setenv INPUT_FILE1 ${M3HOME}/city_output/gridwrf_utm_2d_${DATE}
```

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

```
set STDATE = 1995001
set ENDATE = 1995031
```

The run of **wrf4cc.exe** is controlled by the user metadata file (cctapm_inp.dat). The user metadata file for the city simulation needs to be located in subdirectory /preproc/wrf4cc1.3/. The following edits have to be made in cctapm_inp.dat for use with **wrf4cc.exe**:

Line 7:
Set output directory path (full path name). This will be the directory where the converted WRF files are written to (by default this should be '../../INPUT/wrf/').

Line 8:
Edit name of log file.

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

Line 12:
Edit number of hours for output (not used in wrf4cc).

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*)

After the changes in the script run_wrf4cc.csh and in the user metadata file adapting them to a specific city and time period, the wrf4cc tool can be run.

Note, that the output directory given in the user metadata file must be empty before executing run_wrf4cc.csh.

For running the script, type:

```
{your-citychem}/preproc/wrf4cc1.3$ ./run_wrf4cc.csh
```

After completion of the run the meteorological and auxiliary 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/wrf/.

6.5 BCONCC v2.5

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 3.2 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. For Installation of IOAPI and the m3tools program, refer to section 6.3.

A) 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/bconcc2.5/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 value: Latitude coordinate of the mother grid's reference; corresponds to the global attribute YCENT in METCRO3D)
4th value: Longitude coordinate of the mother grid's reference; corresponds to the global attribute XCENT in METCRO3D)
5th, 6th value: coordinate of the grid centre of the Cartesian coordinate system, here the Meridian (0.000 degrees) and the Equator (0.000 degrees)
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
' '
```

B) 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/bconcc2.4/. 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/bconcc2.5$ ./run_m3cple_city_metcro.csh
```

C) 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/bconcc2.4/. 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/bconcc2.5$ ./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 µg/m^3, summarizes concentrations from particulate compounds in CMAQ to $PM_{2.5}$, 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.

The cmaq4cc tool of BCONCC v2.3 (and earlier versions) interfaces with the list of chemical species of chemistry mechanism CB05TUCL_AE5_AQ in CMAQ v5.0.1; particulate species are based on CMAQ's AERO5 aerosol model. Starting with BCONCC v2.4, the cmaq4cc tool interfaces with the list of chemical species of CB05TUCL_AE6_AQ in CMAQ v5.2; particulate species are based on AERO6 aerosol model. When a different chemical mechanism is used in CMAQ, the variable indices is cmaq4cchem.f90 have to be changed. Please get in contact with the authors of this User's Guide for support in adaption of **cmaq4cc.exe** to different chemical mechanisms.

A) Build the cmaq4cc tool

Building the program **cmaq4cc.exe** requires the IOAPI 3.2 library and the netcdf4 library. See description in section 6.3 for installation of the IOAPI 3.2 library. If the program has not been installed by the automated routine installcc.csh, the following needs to be done. Change to subdirectory /preproc/bconcc2.5/cmaq4cc/.

```
{your-citychem}/preproc/bconcc2.5$ 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/bconcc2.5/cmaq4cc$ make
```

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

```
{your-citychem}/preproc/bconcc2.5/cmaq4cc$ cd ..
{your-citychem}/preproc/bconcc2.5$ ln –s ./cmaq4cc/cmaq4cc.exe
```

B) Edit run_cmaq4cc.csh

The script can be found in subdirectory /preproc/bconcc2.5/. 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 C of stage 1) where the last 7 chars are the date (format yyyyddd), 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 B 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/bconcc2.5/. 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/bconcc2.5$ ./run_cmaq4cc.csh
```

Figure 37 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/.

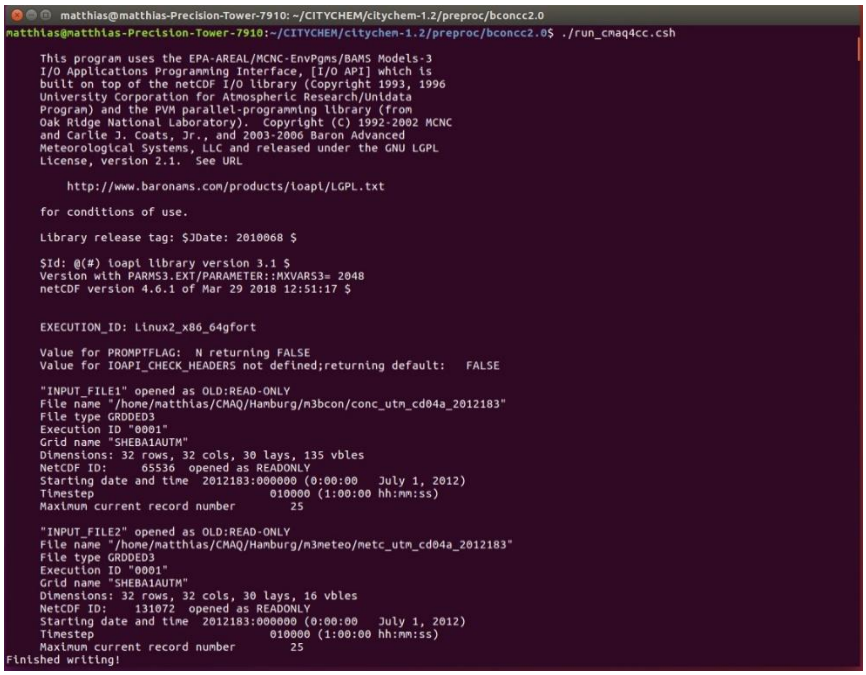


Figure 37: Screenshot of terminal window with output from the cmaq4cc tool.

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_new8_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_new5_201307.tar.gz; see section 2.1 and section 2.4).

The run script citychem_hamburg_new8_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 run. 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 6.5.

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 ./tapm4cc2.2/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). The entry of the TAPM *.outa file (*tapm_path*) should be ‘../testdata/tapm_1km_20130701_20130801.outa’.

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 ./uect2.8/bin/uect.exe

The *.csv files with the annual emission totals of point, line, and area emission sources for Hamburg are located in subdirectory preproc/uect/input/. Note that this is arbitrary emission data and should not be used in research publication.

Area sources:
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 write netCDF output option (*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/.
The netCDF file "BVOC_4_UECT_HH.nc" with the emission potentials of biogenic VOC and foliar density for the Hamburg domain has to be located in subdirectory testdata/ in order to generate area emissions of isoprene. 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 8531, and output files path name is '../INPUT/emis/'.
Run **uect.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: "UECT: Air temperature is used to calculate SNAP2 emissions". If the BVOC emission potentials were correctly provided the following message will be shown in the user log-file: "UECT: BVOC emission factors are used to calculate SNAP11". Running uect.exe will create the emission input files for the simulation in the subdirectory INPUT/emis.

Point sources:
Now the point source emissions, edit the user input file: output source type is 'PSE' and number of point sources is 748, ('../testdata/ncfiles/') in the line of *tapm_path,* output files path name is '../INPUT/emis/'. Then run **uect.exe** inside subdirectory preproc/.

Line sources:
Continue with the line source emissions and edit the user input file as follows: output source type is 'LSE', number of line sources is 15838, and output files path name is '../INPUT/emis/'. Again run **uect.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 3.7.3. 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 3.7.3. 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 3.7.3. 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_new8_201307.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_new8_201307.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/.

The simulation time of the one-month run on a Linux desktop computer (Intel® Xeon® CPU E5-2637 v3 at 3.50 GHz with 64GB of RAM, 4 cores and 8 threads) is around **12.5 hours** when using the advanced compilation option for building EPISODE-CityChem and parallelization with OpenMP.

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.

Bieser, J., Aulinger, A., Matthias, V., Quante, M., Builtjes, P. (2011): SMOKE for Europe – adaptation, modification and evaluation of a comprehensive emission model for Europe, Geosci. Model Dev., 4, 47-68, doi:10.5194/gmd-4-47-2011.

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 NO_2 and PM_{10} for the region including Stavanger, Sandnes, Randaberg and Sola. Norwegian Institute for Air Research, NILU TR 01/2014, Kjeller, Norway.

Donahue, N. M., Epstein, S. A., Pandis, S. N. and Robinson, A. L. (2011) A two-dimensional volatility basis set: 1. organic-aerosol mixing thermodynamics. Atmos. Chem. Phys., 11, 3303-3318, doi:10.5194/acp-11-3303-2011.

Düring, I., Bächlin, W., Ketzel, M., Baum, A., Friedrich, U., and Wurzler, S. (2011) A new simplified NO/NO_2 conversion model under consideration of direct NO_2-emissions, Meteorologische Zeitschrift, 20 (1), 67-73.

ECCAD (2021) CAMS-TEMPO V3.1, CAMS global and regional temporal profiles in ECCAD, https://eccad.aeris-data.fr.

Guevara, M., Jorba, O., Tena, C., Denier van der Gon, H., Kuenen, J., Elguindi, N., Darras, S., Granier, C., and Pérez García-Pando, C. (2021) Copernicus

Atmosphere Monitoring Service TEMPOral profiles (CAMS-TEMPO): global and European emission temporal profile maps for atmospheric chemistry modelling, Earth System Science Data, 13, 367-404, doi:10.5194/essd-13-367-2021.

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.

Jacobson, M. Z. (2005) Fundamentals of Atmospheric Modeling, 2nd edn., Cambridge University Press, Cambridge, UK, ISBN: 0521548659, p. 542-545.

Karl, M., Dorn, H.-P., Holland, F., Koppmann, R., Poppe, D., Rupp, L., Schaub, A., and Wahner, A. (2006) Product study of the reaction of OH radicals with isoprene in the atmosphere simulation chamber SAPHIR, J. Atmos. Chem., 55, 167-187, doi:10.1007/s10874-006-9034-x.

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, doi:10.5194/acp-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, doi:10.5194/acp-16-4817-2016.

Karl, M. Walker, S.-E., Solberg, S., and Ramacher, M. O. P. (2019) The Eulerian urban dispersion model EPISODE - Part 2: Extensions to the source dispersion and photochemistry for EPISODE-CityChem v1.2 and its application to the city of Hamburg, Geosci. Model Dev., 12, 3357-3399, doi:10.5194/gmd-12-3357-2019.

Karl, M., Pirjola, L., Grönholm, T., Kurppa, M., Anand, S., Zhang, X., Held, A., Sander, R., Dal Maso, M., Topping, D., Jiang, S., Kangas, L., and Kukkonen, J. (2022) Description and evaluation of the community aerosol dynamics model MAFOR v2.0, Geosci. Model Dev., 15, 3969-4026, doi:10.5194/gmd-15-3969-2022.

Ketzel, M., Wahlin, P., Berkowicz, R., and Palmgren, F. (2003) Particle and trace gas emission factors under urban driving conditions in Copenhagen based on street and roof-level observation, Atmos. Environ., 37, 2735-2749, doi:10.1016/S1352-2310(03)00245-0, 2003.

Klose, S., Birmili, W., Voigtländer, J., Tuch, T., Wehner, B., Wiedensohler, A., and Ketzel, M. (2009) Particle number emissions of motor traffic derived from street canyon measurements in a Central European city, Atmos. Chem. Phys. Discuss., 9, 3763-3809, www.atmos-chem-phys-discuss.net/9/3763/2009/.

Kurtenbach, R., Becker, K. H., Gomes, J. A. G., Kleffmann, J., Lörzer, J. C., Spittler, M., Wiesen, P., Ackermann, R., Geyer, A., and Platt, U. (2001) Investigations of emissions and heterogeneous formation of HONO in a road traffic tunnel, Atmos. Environ., 35(20), 3385-3394, doi:10.1016/S1352-2310(01)00138-8.

Lee-Taylor, J., and Madronich, S. (2002) Calculation of actinic fluxes with a coupled atmosphere–snow radiative transfer model, J. Geophys. Res., 107(D24), 4796, doi:10.1029/2002JD002084, 2002.

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.

Pöschl, U. R., von Kuhlmann, R., Poisson, N., and Crutzen, P. J. (2000) Development and intercomparison of condensed isoprene oxidation mechanisms for global atmospheric modeling, J. Atmos. Chem., 37, 29-52, doi: 10.1023/A:1006391009798.

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.

Sander, S. P., Abbatt, J., Barker, J. R., Burkholder, J. B., Friedl, R. R., Golden, D. M., Huie, R. E., Kolb, C. E., Kurylo, M. J., Moortgat, G. K., Orkin, V. L., and Wine, P. H. (2011) Chemical Kinetics and Photochemical Data for Use in Atmospheric Studies, Evaluation No. 17, JPL Publication 10-6, Jet Propulsion Laboratory, Pasadena, California, https://jpldataeval.jpl.nasa.gov/pdf/JPL%2010-6%20Final%2015June2011.pdf

Simpson, D., Benedictow, A., Berge, H., Bergström, R., Emberson, L. D., Fagerli, H., Flechard, C. R., Hayman, G. D., Gauss, M., Jonson, J. E., Jenkin, M. E., Nyíri, A., Richter, C., Semeena, V. S., Tsyro, S., Tuovinen, J.-P., Valdebenito, Á., and Wind, P. (2012) The EMEP MSC-W chemical transport model – technical description, Atmos. Chem. Phys., 12, 7825-7865, doi:10.5194/acp-12-7825-2012.

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., and Krognes, T. (2008) The Air Quality Information System AirQUIS. Info. Techn. Environ. Eng., 1, 40-47, doi:10.1007/978-3-540-71335-7_6.

Vogel, B., Vogel, H., Kleffmann, J., Kurtenbach, R. (2003) Measured and simulated profiles of nitrous acid – Part II. Model simulations and indications for a photolytic source, Atmos. Environ., 37, 2957-2966, doi:10.1016/S1352-2310(03)00243-7.

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.

Walker, S.-E., Solberg, S., and Denby, B. (2003) Development and implementation of a simplified EMEP photochemistry scheme for urban areas in EPISODE, Norwegian Institute for Air Research, NILU TR 13/2003, 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

Chemical reactions in EmChem09-HET

Table A1: Heterogeneous gas-phase reactions in EmChem09-HET.

Heterogeneous reactions				
				function
HE-1	NO2 + {NO2} + {H2O}	⟶	HONO + {HNO3}	k_{ground} (T,TSRAD)
HE-2	NO2 + {NO2} + {H2O}	⟶	HONO + {HNO3}	k_{uptake} (T,RH,MW,GAMMA,PM2.5)
HE-3	N2O5 + {H2O}	⟶	2.0 HNO3	k_{uptake} (T,RH,MW,GAMMA,PM2.5)
HE-4	HO2 + {H2O}	⟶	0.5 H2O2 + 0.5 {O2} + {H2}	k_{uptake} (T,RH,MW,GAMMA,PM2.5)
HE-5	O3 + {H2O}	⟶	HO2 + {O2}	k_{uptake} (T,RH,MW,GAMMA,PM2.5)

Reactants in brackets are not accounted for when solving the chemical equations.

Table A2: Gas-phase degradation of isoprene in EmChem09-HET.

Isoprene chemistry				
IS-1	C5H8 + OH	⟶	ISOPO2	2.7E-11 exp(390/T)
IS-2	C5H8 + NO3	⟶	ISOPNO3	3.03E-12 exp(-446/T)
IS-3	C5H8 + O3	⟶	0.26 MVK + 0.39 MACR + 0.90 HCHO + 0.10 MVKO2 + CH3COO2 + CH3O2 + 0.09 H2O2 + 0.25 HO2 + 0.25 OH	7.86E-15 exp(-1913/T)
IS-4	ISOPO2 + NO	⟶	0.34 MVK + 0.22 MACR + 0.34 CAR4 + 0.63 HCHO + 0.05 ISNI + HO2 + NO2	2.54E-12 exp(360/T)
IS-5	MVK + OH	⟶	MVKO2	2.6E-12 exp(610/T)
IS-6	MACR + OH	⟶	0.5 MVKO2 + 0.5 MACO3	8.0E-12 exp(380/T)
IS-7	CAR4 + OH	⟶	MVKO2	4.52E-11
IS-8	MVKO2 + NO	⟶	NO2 + 0.25 CH3COCH2OH + 0.25 CO +0.5 CH3COCHO + 0.75 HCHO + 0.75 HO2	2.54E-12 exp(360/T)
IS-9	ISOPO2 + HO2	⟶	0.857 ISRO2H	0.706 × 2.91E-13 exp(1300/T)
IS-10	ISRO2H + OH	⟶	OH + CAR4	1.00E-10
IS-11	ISOPNO3 + NO	⟶	1.1 NO2 + 0.8 HO2 + 0.85 ISNI + 0.1 MACR + 0.15 HCHO + 0.05 MVK	2.54E-12 exp(360/T)
IS-12	ISNI + OH	⟶	CH3COCH2OH + NALD	1.3E-11
IS-13	MVKO2 + HO2	⟶	MVKO2H	0.625 × 2.91E-13 exp(1300/T)
IS-14	MVKO2H + OH	⟶	MVKO2	3.0E-11
IS-15	NALD + OH	⟶	HCHO + CO + NO2	5.6E-12 exp(270/T)
IS-16	MVK + O3	⟶	0.90 CH3COCHO + 0.32 HO2 + 0.22 CO + 0.19 OH + 0.10 CH3COO2	7.51E-16 exp(-1521/T)
IS-17	MACR + O3	⟶	0.90 CH3COCHO + 0.32 HO2 + 0.22 CO + 0.19 OH + 0.10 CH3COO2	1.36E-15 exp(-2112/T)
IS-18	CAR4 + O3	⟶	0.90 CH3COCHO + 0.32 HO2 + 0.22 CO + 0.19 OH + 0.10 CH3COO2	2.4E-17
IS-19	MPAN + OH	⟶	CH3COCH2OH + CO + NO2	2.9E-11
IS-20	MACO3 + NO	⟶	CH3COO2 + HCHO + NO2	8.7E-12 exp(290/T)
IS-21	MACO3 + NO2	⟶	MPAN	ktr (CH3OO2 + NO2)
IS-22	MPAN	⟶	MACO3 + NO2	ktr (PAN)

ktr (CH3OO2 + NO2): k0/M = 2.7E-28 (300/T)7.1, k∞ = 1.2-11 (300/T)0.9, Fc = 0.3
ktr (PAN): k0/M = 4.9E-3 (300/T)-12100, k∞ = 5.4E16 exp(-13830/T), Fc = 0.3

Table A3: Chemical and photo-dissociation reactions of the EP10-Plume scheme.

In-plume reactions			
IN-1	OP + O2 + M	⟶ O3	5.67E-34 × M × O2 × (T/300)$^{-2.8}$
IN-2	OD + M	⟶ OP	1.8E-11 exp(107/T) × N2 + 3.2E-11 exp(67/T) × O2
IN-3	OP + NO + M	⟶ NO2	*ktr* (NO + OP)
IN-4	OD + H2O	⟶ 2.0 OH	2.2E-10 × H2O
IN-5	O3 + NO	⟶ NO2 + O2	1.4E-12 exp(-1310/T)
IN-7	O3 +OH	⟶ HO2 + O2	1.7E-12 exp(-940/T)
IN-8	O3 + HO2	⟶ OH + 2 O2	2.03E-16 × (300/T)$^{-4.57}$ exp(693/T)
IN-10	NO + HO2	⟶ NO2 + OH	3.6E-12 exp(270/T)
IN-13	NO2 + OH + M	⟶ HNO3	*ktr* (NO2 + OH)
IN-23	OH + HONO	⟶ NO2	2.5E-12 exp(-260/T)
IN-24	OH + NO	⟶ HONO	*ktr* (OH + NO)
MA-8	OH + HCHO	⟶ CO + HO2 + HO2	1.25E-17 × T^2 × exp(615/T)
MA-10	OH + CO	⟶ HO2 + CO2	1.44E-13 + 3.43E-33 × M
PH-1	O3	⟶ OD	7.10E-05; 0.700; 0.86; 0.33
PH-2	O3	⟶ OP	8.94E-04; 0.4500; 0.92; 0.41
PH-3	NO2	⟶ OP + NO	1.37E-02; 0.500; 0.91; 0.38
PH-5	HNO3	⟶ NO2 + OH	3.00E-06; 1.250; 0.87; 0.33
PH-6	HCHO	⟶ CO + 2 HO2	5.40E-05; 0.790; 0.88; 0.34
PH-7	HCHO	⟶ CO + H2	6.65E-05; 0.600; 0.89; 0.35
PH-16	HONO	⟶ OH + NO	3.22E-03; 0.400; 0.91; 0.38

ktr (NO+OP): k0/M = 1.0E-31 (300/T)1.6, k∞ = 3.0E-11 (300/T)-0.3, Fc = 0.85
ktr (NO2+OH): k_0/M = 3.3E-30 (300/T)$^{3.0}$, k_∞ = 4.1E-11, F_c = 0.40
ktr (OH+NO): k_0/M = 7.4E-31 (300/T)$^{2.4}$, k_∞ = 3.3E-11 (300/T)$^{0.3}$, F_c = exp(-T/1420)

Table A4: Calculation of photodissociation rates in the EP10-Plume scheme and in the photochemical reaction cycle between O_3, NO_2 and NO (Option "2" in the run script).

$j_i = CLF_i \cdot A_i \cdot \exp(B_i / \cos(\theta))$ for $\theta < 60°$

$j_i = CLF_i \cdot A_i \cdot \exp(B_i \alpha(\theta))$ for $60° \leq \theta < 89°$

$j_i = CLF_i \cdot A_i \cdot \exp(B_i \alpha(\theta=89°))$ for $\theta \geq 89°$

where θ is the zenith angle and α denotes the optical air mass for large zenith angles, i.e. when the sun is low, and CLF_i is the cloud correction factor for reaction number *i* calculated as follows:

$CLF_i = (1{,}0 - CL/0{,}2) + CL1_i \, CL/0{,}2$ for $CL \leq 0{,}2$

$CLF_i = CL1_i + (CL - 0{,}2)\,(CL2_i - CL1_i)/0{,}6$ for $CL > 0{,}2$

Appendix B

Meteorological input files

Table B1: 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 or 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	3-D potential temperature	no	yes	no
shum	3-D specific humidity (derived from TAPM 3-D relative humidity field)	no	yes	no
Clwater	3-D cloud water mixing ratio (from WRF or COSMO)	no	no	no
pres	3-D pressure (from WRF or COSMO)	no	no	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 C

Model output appendix

Table C1: 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 NO2 concentration at all stations and regular receptor grid points.	no	ASCII
concl_no2.asc	Hourly NO2 concentration at the receptor points associated with line sources.	no	ASCII
ddepmhour.nc	Hourly 2-D model grid dry deposition output for all CityChem compounds at surface. Uses horizontal resolution of the main domain grid.	no	netcdf4
wdepmhour.nc	Hourly 2-D model grid wet deposition output for all CityChem compounds at surface. Uses horizontal resolution of the main domain grid.	no	netcdf4
ddepr_no2.prn	Hourly dry deposition flux of NO2 at all stations and regular receptor grid points.	no	ASCII
wdepr_no2.prn	Hourly wet deposition flux of NO2 at all stations and regular receptor grid points.	no	ASCII
ddepl_no2.prn	Hourly dry deposition flux of NO2 at the receptor points associated with line sources.	no	ASCII
wdepl_no2.prn	Hourly wet deposition flux of NO2 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

Table C1 continued.

Filename in the run script	Content	Mandatory	File format
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
statclaved_no2.txt	Daily average NO2 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 3-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 D

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
szip-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/szip/include -I/usr/local/hdf5/include -I/usr/local/netcdf4/include" LDFLAGS="-L/usr/local/szip/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**

The netCDF library should now be installed on your Ubuntu Linux computer.

F) Adjust your .profile file and test the installation

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.

After restart, check in a terminal, if the netCDF libraries are loaded, by typing:

```
{/your-ubuntu-home}/$  echo $LD_LIBRARY_PATH
```

The selected installation path, /usr/local/netcdf4/lib, should appear in the list of libraries that you see when you issue the above command. If this is not the case, check whether .profile or .bash_profile or a different profile start file is loaded by default. The lines regarding LD_LIBRARY_PATH have to be inserted. It is also possible to directly source the .profile file in the terminal:

```
{/your-ubuntu-home}/$  source ~/.profile
```

If you change to directory /usr/local/netcdf4/lib/ and type ‘ls’, it should look like:

```
matthias@matthias-Precision-Tower-7910:~$ cd /usr/local
matthias@matthias-Precision-Tower-7910:/usr/local$ ls
bin  etc  games  hdf5  include  lib  man  ncl-6.6.2  netcdf4  sbin  share  src  szip  zlib
matthias@matthias-Precision-Tower-7910:/usr/local$ cd netcdf4
matthias@matthias-Precision-Tower-7910:/usr/local/netcdf4$ ls
bin  include  lib  share
matthias@matthias-Precision-Tower-7910:/usr/local/netcdf4$ cd lib
matthias@matthias-Precision-Tower-7910:/usr/local/netcdf4/lib$ ls
libbzip2.la  libmisc.so    libnetcdff.la    libnetcdff.so.6.1.1  libnetcdf.so          libnetcdf.so.7
libbzip2.so  libnetcdf.a   libnetcdff.so    libnetcdf.la         libnetcdf.so.13       libnetcdf.so.7.2.0
libmisc.la   libnetcdff.a  libnetcdff.so.6  libnetcdf.settings   libnetcdf.so.13.1.1   pkgconfig
matthias@matthias-Precision-Tower-7910:/usr/local/netcdf4/lib$
```

where libnetcdf.so.x and libnetcdff.so.x may have different numbers.


Helmholtz-Zentrum
hereon