Chemistry model

Institute of Meteorology and Climate Research
Atmospheric Environmental Research
Campus Alpin

Institute of Meteorology
Tropospheric Environmental Research
Freie Universität Berlin

Klaus Ketelsen, Software developer

PALM group, IMUK, Leibniz Universität Hannover

Chemistry model

Simulating pollutant concentrations with PALM

Chemistry model

Content

A) Overview of available components

- Concept of chemistry model in PALM
- References
- Prognostic equations
- Gas-phase chemistry implementation
- Photolysis
- Deposition
- Aerosol
- Anthropogenic emission input
- Biogemic emissions
- Pollen emissions

B) Usage and application examples

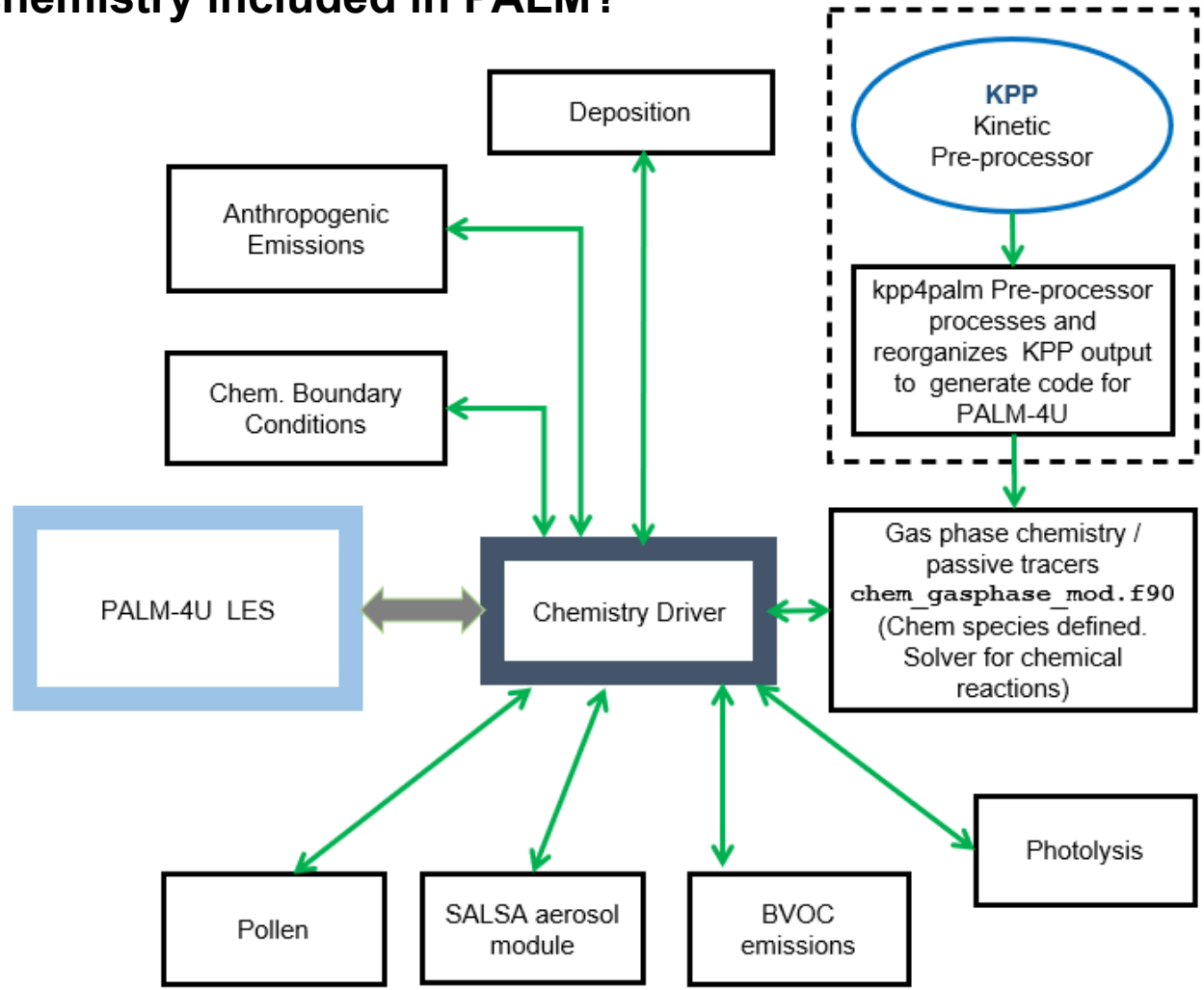
- Usage and namelist options
  - General settings
  - Boundary conditions
- Modifications to other namelists
- Special chemistry namelist sections
  - ➢ Point sources, BVOCs, Pollen,...
- The kpp4palm preprocessor
- Application examples
  - Typical application
  - New mechanism with kpp4palm
  - Nested chemistry runs
- Summary and outlook

How is chemistry included in PALM?

- Chemistry is coupled ’online’ into PALM
- Prognostic equations for chemistry compounds are solved consistently with equations for momentum, heat, and water constituents
- Flexible gas-phase chemistry implementation based on the **K**inetic **P**re-**P**rocessor (KPP)
  - Fortran subroutines solving rate equations of given gas phase chemistry mechanism are generated automatically
  - PALM includes a number of ready-to-use chemical mechanisms
- Anthropogenic emissions (surface emissions and point sources)
- Biogenic VOC emissions
- Pollen emissions
- Aerosol: Passive tracer, ISORROPIA, SALSA

Concept of chemistry model in PALM

How is chemistry included in PALM?



Chemistry source code

- Names of all chemistry related code starts with **`chem`**
- One exception is the aerosol module SALSA, which can be connected to the gas phase chemistry
- Driver routines are in **`chemistry_model_mod.f90`**
- Solution of chemical transformation equations occurs in **`chem_gasphase_mod.f90`** (created by kpp4palm)
- Further **`chem_*`** modules include anthropogenic emissions, BVOC emissions, photolysis and pollen

NAMELIST &chemistry_parameters

Chemistry is switched on by the presence of **`&chemistry_parameters`**

(For „commenting“ the entire chemistry section in the namelist the spelling of „**`&chemistry_parameters`**“ must be modified). Do not forget to switch off the output of the chemistry compounds when doing this.

Publications and References

General description of the gas phase chemistry module

Khan et al., 2021, Development of an atmospheric chemistry model coupled to the PALM model system 6.0, *Geosci. Model Dev.*, 14, https://doi.org/10.5194/gmd-14-1171-2021

Salsa-Papers

Kurppa et al, 2019, Implementation of the sectional aerosol module SALSA2.0 into the PALM model system, *Geosci. Model Dev.,* 12, https://doi.org/10.5194/gmd-12-1403-2019

Kurppa et al, 2020, Sensitivity of spatial aerosol particle distributions to the boundary conditions..., *Geosci. Model Dev.*, 13, https://doi.org/10.5194/gmd-13-5663-2020

PALM joint issue: https://gmd.copernicus.org/articles/special_issue999.html

KPP and kpp4palm

http://people.cs.vt.edu/~asandu/Software/Kpp

Sandu and Sander, 2006: Simulating chemical systems in Fortran90

and Matlab with the kinetic preprocessor KPP-2.1, *Atmos. Chem. Phys.*, 6, 187–195, 2006, http://www.atmos-chem-phys.org/acp/6/187/

Jöckel et al, 2010, Development cycle 2 of the Modular Earth Submodel System (MESSy2), *Geosci. Model Dev.,* 3, https://doi.org/10.5194/gmd-3-717-2010

Prognostic equation for concentrations of ith chemical compound

$$\frac{\partial \overline{c_i}}{\partial t} = -\frac{1}{\rho}\frac{\partial}{\partial x_j}\left(\rho \overline{u_j c_i}\right) - \frac{1}{\rho}\frac{\partial}{\partial x_j}\left(\rho \overline{u_j'' c_i''}\right) + \left(\frac{\partial c_i}{\partial t}\right)_{chem} + S_i$$

with j=1,2,3 (sum over j)

$\frac{\partial \overline{c_i}}{\partial t}$: local change of c_i

i : index of chemical compound (i=1,...,n)

c_i : concentration of chemical compound i

$\left(\frac{\partial c_i}{\partial t}\right)_{chem}$: change of c_i due to chemical reactions (is solved in **`chem_gasphase_mod.f90`**)

S_i : other sources and sinks of chemical compound i (e.g. emissions)

Gas phase chemistry related code in `packages/palm/model/src`

- **`chemistry_model_mod.f90`**
  - **`chem_prognostic_equations`**
    compution of concentration change due to advection, diffusion
  - **`chem_non_advective_processes`**
    driver for gas phase chemistry (**`chem_integrate`**), deposition
  - Initialisation, boundary conditions, output ...

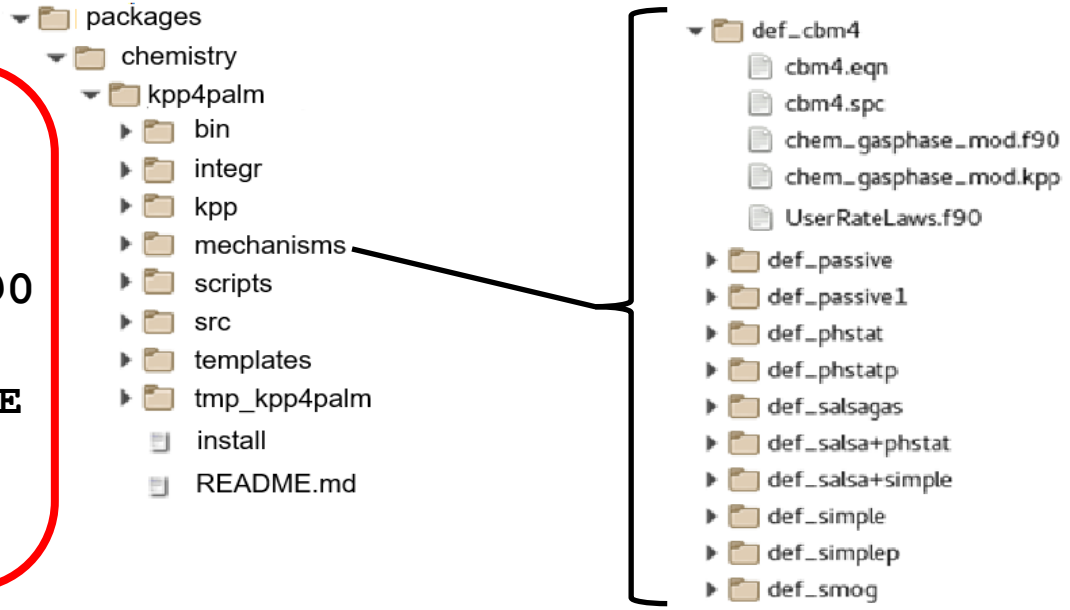
- **`chem_gasphase_mod.f90`** (from **`kpp4palm`** preprocessor)
  compution of concentration change $\left(\frac{\partial c_i}{\partial t}\right)_{chem}$ due to chemical reactions

- **`kpp4palm`** permits to create **`chem_gasphase_mod.f90`** for any gas phase mechanism and an any number of passive tracers
  More about the application of **`kpp4palm`** later in this talk

Available ready-to-use gas phase chemistry mechanisms

- The default mechanism, which is included in **`packages/palm/model/src`** is the photostationary equilibrium between NO_2, NO, and ozone plus one passive tracer named PM10 (mechanism ‚phstatp‘)
- Ready-to-use Fortran code (file **`chem_gasphase_mod.f90`**) is supplied for some mechanisms of different complexity in subdirectories of **`packages/chemistry/kpp4palm/mechanisms`**

- To apply one of these mechanisms, copy the respective **`chem_gasphase_mod.f90`** file into **`JOBS/my_setup/USER_CODE`** (where ‚my_setup‘ stands for the job identifier)

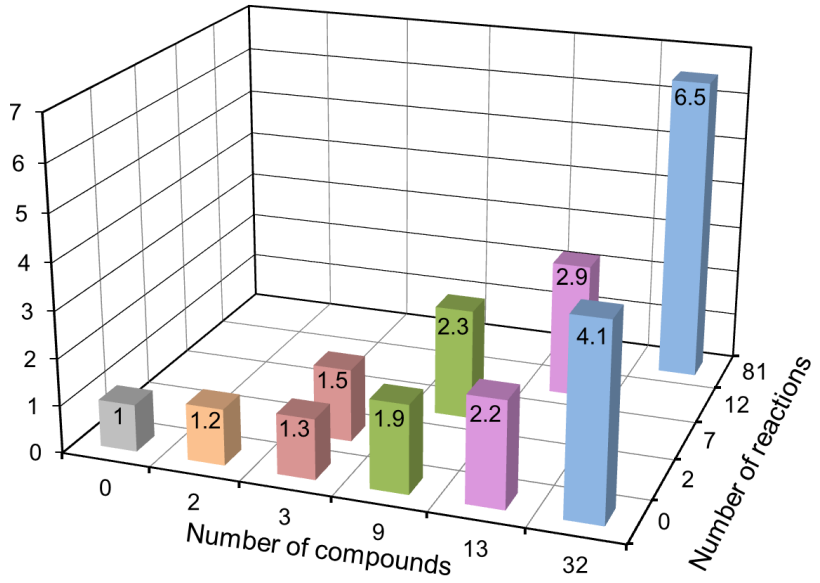


Ready-to-use mechanisms: Computational demand

- CBM4: Carbon Bond Mechanism (Gery et al. 1989, 32 compounds, 81 reactions)
- SMOG: Small photochemical mechanism (13 compounds, 12 reactions)
- SIMPLE: Simplified version of SMOG (9 compounds, 7 reactions)
- PHSTAT: Photo-stationary state (3 compounds, 2 reactions)
- PASSIVE: Two passive tracers (2 compounds, 0 reactions)
- Reference: Meteorology only

Resources required for PALM-4U chemistry run:

CPU time requirement for a PALM run using the different provided mechanisms relative to a meteorology only run (test for one day with Intel compiler for 96 x 96 x 320 grid points)



Photolysis frequencies are supplied for the ready-to-use mechanisms

Photolysis can be calculated from radiation model

$$J = \int_0^\infty q(\lambda)\sigma(\lambda)I(\lambda)d\lambda$$

I: radiation flux on sphere (radiation model)
σ: absorption cross section (depending on molecule)
q: quantum yield (depending on molecule)

Currently just a simple parameterization depending on solar zenith angle ϑ from MCM (Master Chemical Mechanism) is included in PALM-4U

$$J = l(\cos\vartheta)^m \exp(-n\sec\vartheta)$$

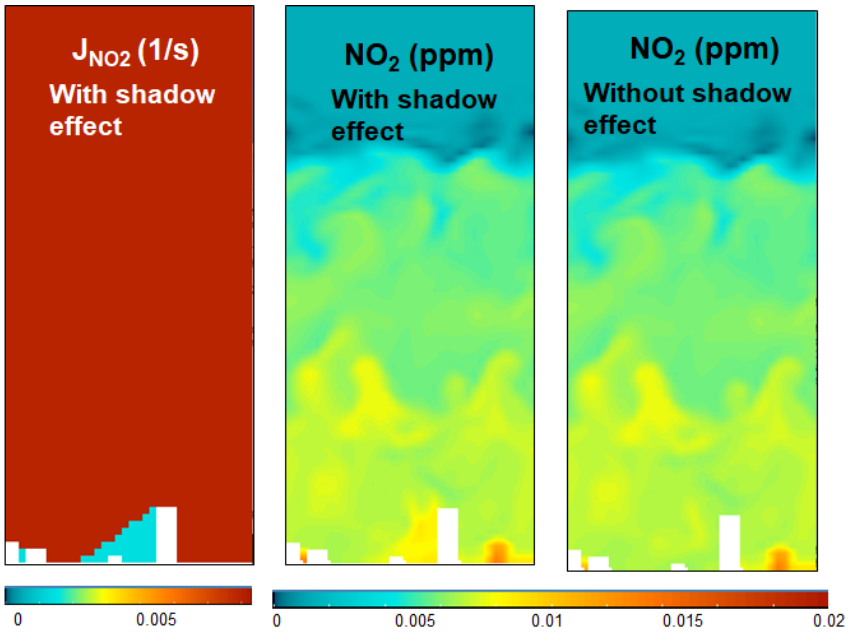
http://chmlin9.leeds.ac.uk/MCM/parameters/photolysis_param.htt or

https://www.atmos-chem-phys.net/3/161/2003/

Photolysis model based on a radiation model e.g. FAST-J (Wild et al., 2000, Prather, 2015) will be implemented later

Building shadows for photolysis

Parametrisation based on height of the shaded area. Diffuse contribution in shadow is assumed to be 20% of the value for full sun (should be improved later).



Resistance approach

- Dry deposition is major sink of atmospheric pollutant concentrations
- A resistance approach has been implemented
  - Exchange flux is result of a concentration difference between atmosphere and earth surface and the resistance between them
  - Several pathways exist for this flux

$$F = v_d \cdot C \quad \text{with}$$

$$v_d = \frac{1}{R_a + R_b + R_c}$$

F : deposition flux
v_d : deposition velocity
C : concentraion
$R_{a/b/c}$: resistances of different pathways a, b and c

! The dry deposition calculation requires the use of the LSM/USM/BSM !

- For gases: Resistences modelled following **Simpson et al. (2003)** and **van Zanten et al. (2010)**
- For particles: Resistences modelled following **Zhang et al. (2001)** including explicit dependence on aerosol size

Chemistry model

Wet Deposition

In- and below cloud scavenging

- Wet deposition is major sink of atmospheric pollutant concentrations
- Approach using scavenging coefficients following Berge und Jakobsen (1998)
  - In-cloud

$$S_{in} = -\chi \frac{W_{in} P}{h_s \rho_w}$$

W_{in}: In-cloud scavenging ratio
P [kg m^{-2} s^{-1}]: precipitation rate
h_s [m]: characteristic scavenging depth
ρ_w [kg m^{-3}]: water density
X: species mixing ratio

  - Below-cloud
    - Gases

$$S_{sub}^{gas} = -\chi \frac{W_{sub} P}{h_s \rho_w}$$

W_{sub}: Below-cloud scavenging ratio

    - Particles

$$S_{sub}^{aer} = -\chi \frac{A\ P}{V_{dr}}\ \overline{E}$$

E: size-dependent collection efficiency
A (=5.2 m^3 kg^{-1} s^{-1}): empirical coefficient assuming a Marshall-Palmer size distribution for rain drops
V_{dr} [m s^{-1}]: raindrop fall speed

- Scavenging ratios are taken from Simpson et al. (2012)

Three options for aerosol description

1) PM10 and PM2.5 as passive tracer

- Some gas phase mechanisms include aerosol as a passive tracer, e.g. mechanisms **`passive, phstatp, simplep`**

2) Sectional aerosol module SALSA2.0 in PALM

- Sectional aerosol model including coagulation, condensational growth, nucleation, (Kurppa et al, 2019, Geosci. Model Dev., 12)
- Compounds: OC, BC, H_2SO_4, HNO_3, NH_3/NH_4, SVOC and NVOC
- Optional coupling of H_2SO_4, HNO_3, NH_3, SVOC and NVOC to gas phase chemistry module; transport in gas phase
- Currently available gas phase chemistry mechanisms coupled to SALSA:
  - **`salsagas`**
  - **`salsa+phstat`**
  - **`salsa+simple`**

3) Bulk description for secondary inorganic aerosols (ISORROPIA)

- A large part of the particle dry mass is inorganic (25–50%) with ammonium, sodium, sulfate, bisulfate, nitrate and chloride being the most important contributors to the dry inorganic PM2.5
- Inorganic species may be in the form of aqueous ions, or in the form of precipitated solids, in thermodynamic equilibrium with atmospheric gases and humidity

➢ For modelling the formation of secondary inorganic aerosols, the user can choose from the third-party aerosol thermodynamic equilibrium models

  - ISORROPIA (Nenes et al., 1998, Aquatic Geochem., 4, 123–152) and
  - ISORROPIA II (Fountoukis and Nenes, 2007, ACP, 7, 4639–4659)

- Activation in chemistry namelist
- ISORROPIA (II) must currently be compiled as a standalone shared library (.so) with the same toolchain used in building PALM
- Find more info and possible namelist options in PALM model system on GitLab:
  palm_model_system-v23.04\packages\palm\model\docs\content\Guide\Modules\Chemistry

Two implementations for anthropogenic emission input

- Emission input implemented as surface fluxes
  - LOD 0
    - ➢ Traffic emissions via namelist input
  - LOD 1
    - ➢ Gridded data, annual emission values via netCDF file
  - LOD 2
    - ➢ Gridded data, temporally disaggregated emission values via netCDF file

! Supplying surface emissions to PALM requires the use of the LSM/USM/BSM !

- Emission input implemented as volume sources
  - LOD 0
    - ➢ Emissions from point sources via namelist input
  - LOD 2
    - ➢ Gridded data, temporally disaggregated emission values via netCDF file

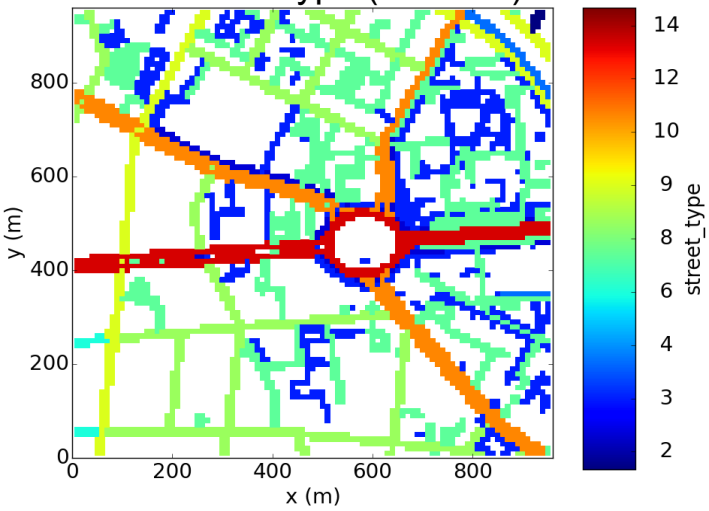
LOD 0 surface fluxes

- So far only implemented for the traffic sector
- Classification of streets, closely following the Open Street Map classification
  - ➢ Flexible split into MAIN and SIDE roads
- Emission scaling factor for MAIN and SIDE roads
- Emission values in µmol m^{-2} d^{-1} (gas) or kg m^{-2} d^{-1} (PM) ↔ chem. mechanism
- Standard week-day inner city time profile used to temporally disaggregate daily emissions to hourly values

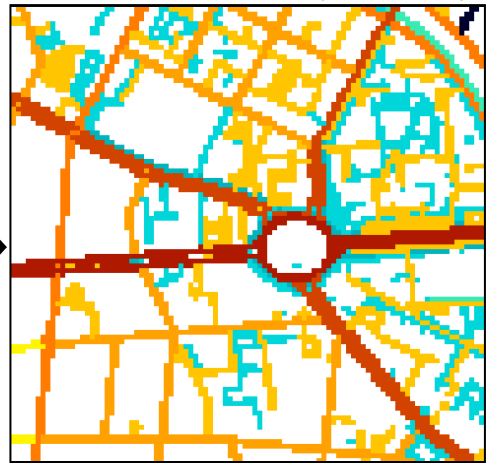
Classification

street_type	Description
1	unclassified
2	cycleway
3	footway / pedestrian
4	path
5	track
6	living street
7	service
8	residential
9	tertiary
10	tertiary link
11	secondary
12	secondary link
13	primary
14	primary link
15	trunk
16	trunk link
17	motorway
18	motorway link
19	raceway

Street type (10x10m)



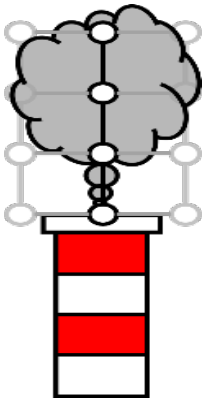
Main/side roads (10x10m)

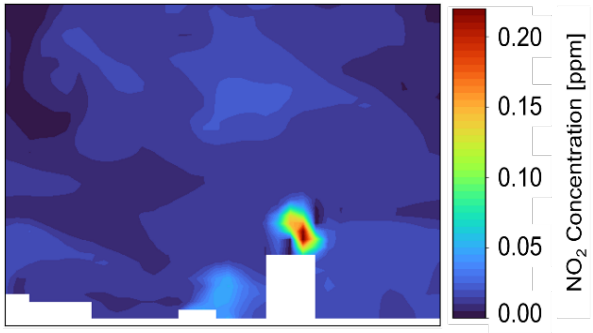


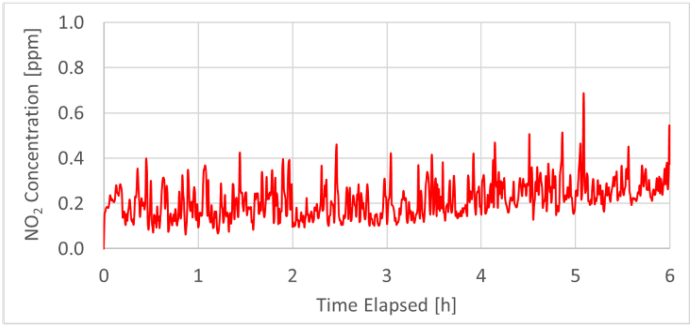
main road
side road
other

LOD 0 volume sources

- Implemented for emissions from point sources e.g. Power Plants as included in E-PRTR
  - Activation (emis_pt_source=.TRUE.) and specification in chemistry namelist
  - Species names and point source locations defined by user
    - ➢ Up to 200 point sources possible
  - Annual emission per point source and per species defined by user (mol/year for gases and kg/year for PMs)





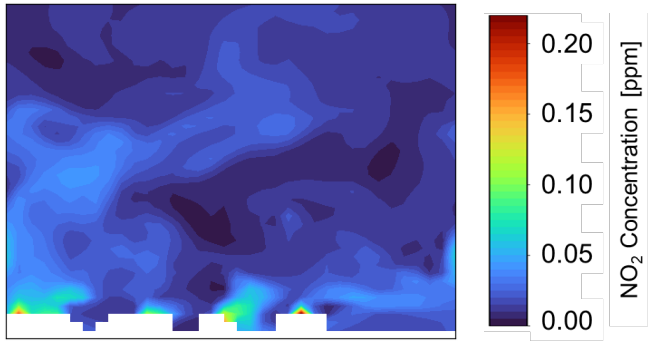


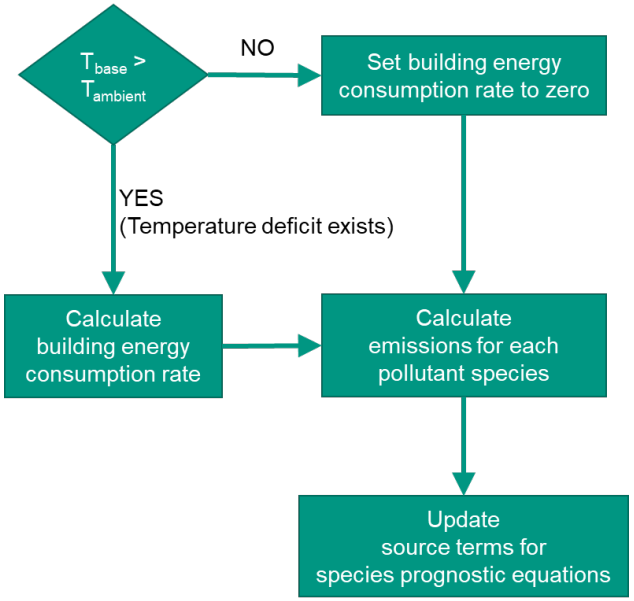
Chemistry model

Anthropogenic emission input

LOD 0 volume sources

- Emissions from domestic heating
  - Based on Struschka and Li (2019)
  - Activation (emis_domestic=.TRUE.) and specification in chemistry namelist
  - Species and furnace type (using corresponding emisson factor) defined by user





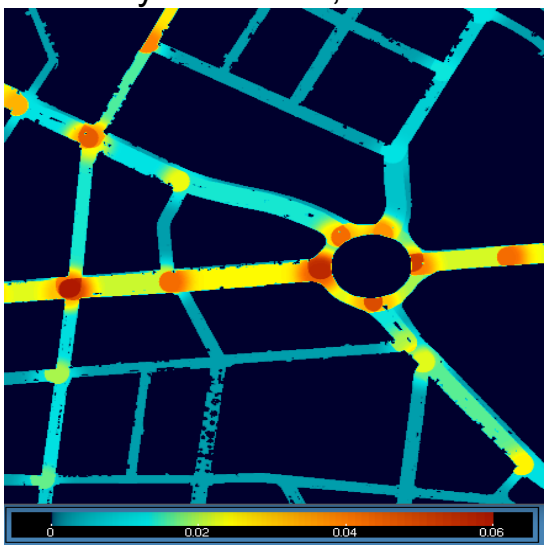
  - Stack position defined using automated algorithm based on geometric center of building roof (Script in PALM download soon!)
    - ➢ Integrated into the static driver

Exemplary representation of chimney position location (orange) on the respective buildings (yellow)

LOD 2 surface fluxes or volume sources

- Emission data given via netCDF *_chemistry input file
  - Surface fluxes (e.g. traffic emissions)
    - https://palm.muk.uni-hannover.de/trac/wiki/doc/app/iofiles/pids/chem
  - Volume sources (domestic heating, traffic or „generic“ sector, e.g. point source emissions)
    - See PALM model system on GitLab: palm_model_system-v23.04\packages\palm\model\docs\content\Guide\Modules\Chemistry
- Gridded spatially distributed emission data
  - ➢ Pre-processed emission values on grid per chemical species
  - ➢ (Splitted) temporally disaggregated
    - ➢ Flexible temporal resolution possible

Traffic emissions
January 17th 2017, 7:00am

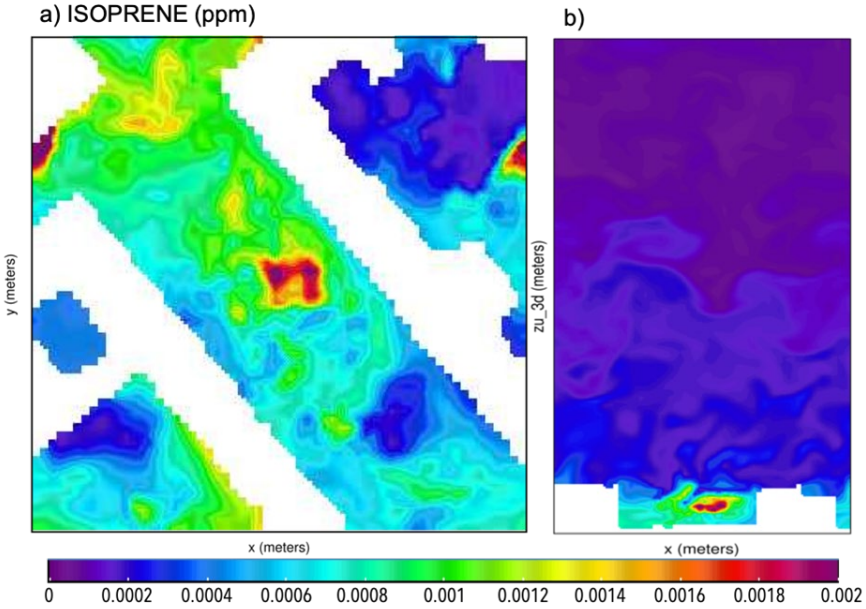


NO (in g/m²/h)

BVOC emissions

Currently only one option based on parametrisations by Guenther et al. (2012, https://doi.org/10.5194/gmd-5-1471-2012) is available.
Dependence on radiation, temperature, soil moisture, season (but no consideration of temperature during the previous days).

Required PALM components for application of BVOC emissions
a) chemistry_model_mod
b) land_surface_model
c) radiation_model_mod
d) urban_surface_model
e) plant_canopy_model



Horizontal (a) and vertical (b) cross sections of isoprene concentration in small domain with 2 m grid spacing at 14:00 h. Emissions from birch, chestnut and beech trees.

Documentation (currently) located in packages/palm/model/docs/content/Guide/Modules/Chemistry/Emissions

Pollen emissions

- Based on EMPOL 1.0
  (Zink et al., 2013; https://doi.org/10.5194/gmd-6-1961-2013
- Pollen as passive tracers
- Currently implemented: - Birch
  - Alder
  - Grass

  (Ambrosia is not yet implemented – also not (yet) included PALM's species list for vegetation)
- Tree types and vegetation type (for grass) must be included in static file
- Available "mechanism": empol1.0 must be used

Documentation currently (Sept. 2022) located in PALM model system on GitLab in directory:
palm_model_system-v23.04\packages/palm/model/docs/content/Guide/Modules/Chemistry

How is chemistry started in PALM?

- Chemistry is switched on by providing the additional namelist **chemistry_parameters** in the NAMELIST file (_p3d):

```
&chemistry_parameters
          . . .
```

- By default, PALM includes a **chem_gasphase_mod.f90** describing the photostationary equilibrium between O_3, NO, and NO_2

  The mechanism is indicated in the third line of **chem_gasphase_mod.f90**

  The reactions for this mechanism can be seen in the array **eqn_names** in **chem_gasphase_mod.f90** and in more detail in the KPP input file **phstatp.eqn** in directory **packages/chemistry/kpp4palm/mechanisms/def_phstatp:**

```
NO2 + hv = NO + O3 :          phot(j_no2)   ;
NO + O3 = NO2 :               arr2(1.8E-12_dp, 1370.0_dp, temp)  ;
PM10 = PM10 :                 1.0_dp ;
```

Example chemistry_parameters NAMELIST

```
&chemistry_parameters

! Chemistry choice
               chem_mechanism             = "phstatp",
               chem_gasphase_on           = .T.,                       ! .F. = no chemical reactions
! KPP solver parameters (removal will result in higher computational time)
               icntrl(3)                  = 1,                         ! solver ros2
               icntrl(4)                  = 500,                       ! max number of chem-substeps
               rcntrl(3)                  = 0.1,                       ! Hstart, integration step size
! Processes
               photolysis_scheme          = "simple",
               deposition_dry             = .T.,
! Emission choice
               emissions_anthropogenic    = .TRUE.,
               emiss_lod                  =  0,
               surface_csflux_name        = 'NO',   'NO2', 'PM10',     ! Only for parameterized emissions
               surface_csflux             = 4745.0, 1326.0, 2700e-09,  ! surface emission of NO, NO2 in umol m-2 d-1 (PMs in kg m-2 d-1),
               emiss_factor_main          = 1.667, 1.667, 1.667,
               emiss_factor_side          = 0.334, 0.334, 0.334,
               main_street_id             = 11,
               side_street_id             =  8,
               max_street_id              = 19,
! Initial values at surface (valid at all heights in case of no other values)
               cs_name                    = 'O3',  'NO',  'NO2', 'PM10',
               cs_surface                 = 0.020, 0.000, 0.010, 2.0e-09,    !units- chem spcs in ppm, and PM10 in ug/m**2
! Initial profiles (see cs_names for order of variables)
               cs_profile(1,:)            =  0.015, 0.022, 0.050,            !units for gases ppm
               cs_profile(2,:)            =  0.000, 0.000, 0.000,            !units for gases ppm
               cs_profile(3,:)            =  0.010, 0.005, 0.002,            !units for gases ppm
               cs_profile(4,:)            =  12.0e-09, 5.0e-09, 1.0e-09,     ! PM kg/m**3

               cs_heights(1,:)            =  0.0, 105.0, 295.0,
               cs_heights(2,:)            =  0.0,  85.0, 295.0,
               cs_heights(3,:)            =  0.0,  85.0, 295.0,
               cs_heights(4,:)            =  0.0, 125.0, 255.0,
! Top and bottom boundary conditions
               bc_cs_b                    = 'neumann',    ! If emissions_anthropogenic=.T.
               bc_cs_t                    = 'neumann',
 / ! end of chemistry_parameters namelist
```

chemistry_parameters NAMELIST

```
&chemistry_parameters

! Chemistry choice
        chem_mechanism                  = "phstatp",
        chem_gasphase_on                = .T.,   ! .F. = no chemical reactions


! KPP solver parameters (removal will result in higher computational time)
        icntrl(3)                       = 1,     ! solver ros2
        icntrl(4)                       = 500,   ! max number of chem-substeps
        rcntrl(3)                       = 0.1,   ! Hstart, integration step size


! Processes
     photolysis_scheme              = "simple",
     deposition_dry                 = .T.,
```

chemistry_parameters NAMELIST (cont.)

```
&chemistry_parameters

! Chemistry choice
        chem_mechanism              = "phstatp",
        chem_gasphase_on            = .T.,   ! .F. = no chemical reactions


! KPP solver parameters (removal will result in higher computational time)
        icntrl(3)                   = 1,     ! solver ros2
        icntrl(4)                   = 500,   ! max number of chem-substeps
        rcntrl(3)                   = 0.1,   ! Hstart, integration step size


! Processes
    photolysis_scheme           = "simple",
    deposition_dry              = .T.,
```

chemistry_parameters NAMELIST (cont.)

```
&chemistry_parameters

! Chemistry choice
        chem_mechanism                = "phstatp",
        chem_gasphase_on              = .T.,   ! .F. = no chemical reactions


! KPP solver parameters (removal will result in higher computational time)
        icntrl(3)                     = 1,     ! solver ros2
        icntrl(4)                     = 500,   ! max number of chem-substeps
        rcntrl(3)                     = 0.1,   ! Hstart, integration step size


! Processes
    photolysis_scheme             = "simple",
    deposition_dry                = .T.,
```

chemistry_parameters NAMELIST (cont.)

```
! Emission parameter

     emissions_anthropogenic    = .TRUE.,  ! Do NOT omit, else concentrations
                                           ! will 'explode'
     emiss_lod                  = 0,   ! Parameterized emissions


     surface_csflux_name        = 'NO',    'NO2', 'PM10',     ! Only for LOD 0
        surface_csflux                = 4745.0, 1326.0, 2700e-09,   ! surface
emission of NO, NO2 in umol m-2 d-1 (PMs in kg m-2 d-1)

     emiss_factor_main          = 1.667, 1.667, 1.667,
     emiss_factor_side          = 0.334, 0.334, 0.334,
     main_street_id             = 11,
     side_street_id             =  8,
     max_street_id              = 19,
```

For **emiss_lod = 2** see exercise

chemistry_parameters NAMELIST (cont.)

```
! Initial values at surface (valid at all heights in case of no other values)
        cs_name                      = 'O3',   'NO',   'NO2',    'PM10',
        cs_surface                   = 0.020, 0.000, 0.010,     2.0e-09,
                                    !units- chem spcs in ppm, and PM10 in ug/m**2

! Initial profiles (see cs_names for order of variables)
        cs_profile(1,:)          =  0.015, 0.022, 0.050,   !units for gases ppm
        cs_profile(2,:)          =  0.000, 0.000, 0.000,   !units for gases ppm
        cs_profile(3,:)          =  0.010, 0.005, 0.002,   !units for gases ppm
        cs_profile(4,:)          =  12.0e-09, 5.0e-09, 1.0e-09,  ! PM kg/m**3

        cs_heights(1,:)          =  0.0, 105.0, 295.0,
        cs_heights(2,:)          =  0.0,  85.0, 295.0,
        cs_heights(3,:)          =  0.0,  85.0, 295.0,
        cs_heights(4,:)          =  0.0, 125.0, 255.0,

! Top and bottom boundary conditions
        bc_cs_b                  = 'neumann', ! If emissions_anthropogenic=.T.
        bc_cs_t                  = 'neumann',

 / ! end of chemistry_parameters namelist
```

chemistry_parameters NAMELIST (cont.)

```
! Initial values at surface (valid at all heights in case of no other values)
        cs_name                      = 'O3',   'NO',   'NO2',    'PM10',
        cs_surface                   = 0.020, 0.000, 0.010,     2.0e-09,
                                    !units- chem spcs in ppm, and PM10 in ug/m**2

! Initial profiles (see cs_names for order of variables)
        cs_profile(1,:)          =  0.015, 0.022, 0.050,   !units for gases ppm
        cs_profile(2,:)          =  0.000, 0.000, 0.000,   !units for gases ppm
        cs_profile(3,:)          =  0.010, 0.005, 0.002,   !units for gases ppm
        cs_profile(4,:)          =  12.0e-09, 5.0e-09, 1.0e-09,  ! PM kg/m**3

        cs_heights(1,:)          =  0.0, 105.0, 295.0,
        cs_heights(2,:)          =  0.0,  85.0, 295.0,
        cs_heights(3,:)          =  0.0,  85.0, 295.0,
        cs_heights(4,:)          =  0.0, 125.0, 255.0,

! Top and bottom boundary conditions
        bc_cs_b                  = 'neumann', ! If emissions_anthropogenic=.T.
        bc_cs_t                  = 'neumann',

 / ! end of chemistry_parameters namelist
```

Chemical boundary conditions

Lateral boundary conditions

- Lateral boundary conditions are specified by **`bc_cs_l, bc_cs_r, bc_cs_n,`** and **`bc_cs_s:`**
  - If not specified in the chemistry namelist, the values for the meteorology variables, **`bc_lr`** and **`bc_ns,`** are used.
  - Cyclic boundary conditions are applied by default in PALM. This holds also for the chemistry variables.
  - Possible issues: unrealistically high concentrations at inflow boundary
- Options for by **`bc_cs_l, bc_cs_r, bc_cs_n,`** and **`bc_cs_s`**: *'dirichlet'* (required for nested runs; in case of no nesting constant concentrations over the entire simulation as given by the initial profiles), *'neumann'* (zero concentration gradient), and *'cyclic'* .
  - Possible issues with ´**`dirichlet`**‘ without nesting: Fixed values, no diurnal course. Not really convincing - in particular not for ozone.

Chemical boundary conditions

Lateral boundary conditions (cont.)

- **Offline nesting**: PALM can read and handle chemistry variables if they are included in the **`_DYNAMIC`** file. This holds for initial conditions as well as lateral and top boundary conditions.
- For offline nesting **`bc_cs_l`**, **`bc_cs_r`**, **`bc_cs_n`**, and **`bc_cs_s`** must be set to **`'dirichlet'`**.
- So far, no universal tools for inclusion of chemistry variables into the **`_DYNAMIC`** file are available.
  Tools available in **`packages/dynamic_driver:`**
  - **`inifor`** can only process meteorological variables. A work-around exists.
  - **`wrf_interface`** allows processing of WRF and CAMx output (chemistry mapping for CAMx v6.50 with CB05 + PM (CF,SOAP2.1,ISORROPIA))

  Further Python based pre-processors for WRF, which could be extended to chemistry, e.g.: **`WRF4PALM`** (https://gmd.copernicus.org/articles/14/2503/2021/gmd-14-2503-2021.html)

Chemical boundary conditions

Bottom and top boundary condition

```
! Top and bottom boundary conditions
    bc_cs_b                    = 'neumann', ! If emissions_anthropogenic=.T.
    bc_cs_t                    = 'neumann',

/ ! end of chemistry_parameters namelist
```

- When a constant surface concentration flux is used (**surface_csflux**) or emissions are applied (**emissions_anthropogenic = .T.**), then **bc_cs_b = 'neumann'** must be used.

List of all chemistry parameters:
https://palm.muk.uni-hannover.de/trac/wiki/doc/app/chemistry_parameters

Modifications to other namelist sections

- **Namelist runtime_parameters**
- **Output of chemistry variables** is handled in the same namelist sections as for the meteorological variables
- Necessary modifications to namelist **runtime_parameters:**
  - Output for chemistry variables must be set, e.g. for mechanism ‘phstatp’

```
data_output      =  …,
                    'kc_NO‘, 'kc_NO2‘, 'kc_O3','kc_PM10‘,
```

  - Output of concentrations is named '**kc_**’ followed by the name of the compound as given the **.eqn** file. Names can also be looked up under **eqn_names** in **chem_gasphase_mod.f90.**
  - Similar for averages, xy, xz or yz slices or masked output

Modifications to other namelist sections

- **Namelist initialization_parameters**:
- Initial values (similar settings as for meteorology)

```
initializing_actions = 'read_from_file set_constant_profiles',
```

Values from dynamic file are used if included in the _dynamic file, constant profiles for all other variables (e.g. **`read_from_file`** for meteorology, constant profiles for chemistry)

- Further required namelists:
  - **radiation_parameters** (for photolysis: zenith angle and shading due to buildings)
  - **land_surface_parameters, urban_surface_parameters** (for deposition, emissions)

Building shadows for photolysis

- Default is no shading effects
- Activation by **`photolysis_shading =.TRUE.,`**
  in the **`&chemistry_parameters`** namelist
- Requires setting **`radiation_volumetric_flux = .TRUE.,`**
  in the **`&radiation_parameters`** namelist !

Point source emissions via namelist (= LOD 0)

Activation by

`emis_pt_source =.TRUE.` (Default is **`.FALSE.`**)

in the **`&chemistry_parameters`** namelist

Example with all options included:

```
! Leap year or no leap year
emis_pt_source_leap_year     = .FALSE.,
! Z-level spread of emissions (default is 3 levels)
emis_pt_source_k_spread      = 4,
emis_pt_source_k_weights(:)  = 0.1, 0.2, 0.5, 0.2,
! Emitted species
emis_pt_source_species_names(:) = 'HNO3', 'NO2', 'NO',
! Point source location in domain
emis_pt_source_locations_ijk(1,:) = 10, 12, 15,
emis_pt_source_locations_ijk(2,:) = 20, 14,  8,
emis_pt_source_locations_ijk(3,:) =  8,  4,  5,
! Annual emission values per point source per spec. (mol/year, kg/year)
emis_pt_source_annual_values(1,:) = 1.0e9, 1.0e9, 1.0e9,
emis_pt_source_annual_values(2,:) = 2.0e6, 2.0e9, 2.0e6,
emis_pt_source_annual_values(3,:) = 3.0e6, 3.0e9, 3.0e6,
```

Emissions from domestic heating via namelist (= LOD 0)

Activation by

`emis_domestic =.TRUE.` (Default is `.FALSE.`)

in the `&chemistry_parameters` namelist

Example for PM10 and furnace type centralized oil:

```
emis_domestic_lod                          =  0,
emis_domestic_species_names(:)             = 'PM10',
emis_domestic_species_emission_factors(:) =  0.34,
```

- Here, emission factors were chosen for furnace type „Centralized Oil“. Emission factors for different furnace types are proposed in the PALM documentation in GitLab (under palm_model_system-v23.04\packages\palm\model\docs\content\Guide\Modules\Chemistry) but can also be defined differently by the user

BVOC emissions

Activation by

`emis_biogenic =.TRUE.` (Default is `.FALSE.`)

`emis_generic = .TRUE.` (Required for BVOC emíssions)

in the `&chemistry_parameters` namelist

Resolved vegetation only (Single trees & patches [described by vegetation type])

Emission of isoprene, 8 terpenes, 14 OVOC, 7 oxigenated compounds
Currently only connected to mechanisms CBM4 and SMOG
Compounds should be selected matching with the mechanism, e.g.

```
ebio_emis_name   = 'ISOP','HCHO','CO','ETH','TOL',
```

Emission factors for tree types (ebio_ef_tree) for single trees (currently 86 types) and emission factors for plant functional types (ebio_ef_pft) for patches (currently 18 PFTs) can be supplied by the user. Otherwise, emission factors for plant functional types from MEGAN will be used as default.

Pollen emissions

- Activation by **`emis_pollen = .TRUE.`** in **`&chemistry_parameters`**
- Required mechanism: **`empol1.0`**

Example for 20th April (peak emission day of Birch trees) :

```
chem_mechanism              = "empol1.0",
emis_pollen                 = .TRUE.,
epol_ignore_precip          = .TRUE.,      !< Only T is accounted for
epol_ignore_solar           = .TRUE.,      !< Only T is accounted for.
epol_update_interval        =  60.0,
epol_pool_reset_hour        =  0,          !< reset, replenish pools
epol_model                  = 'zink',      !< only option currently
epol_tke_scheme             = 'default',   !< 'dynamic', 'adhoc',
epol_tke_sgs_fraction       = 0.10,        !<  range 0.0 - 1.0
epol_specs_names            = 'POL_ALNU', 'POL_POAC', 'POL_BETU',
epol_seasonal_factors       =   0.0,         0.0,          0.9,
epol_tuning_factors         =   1.0,        0.01,           10,
! relate epol_specs_names to single tree and veg. type in static file
epol_tree_specs             =   999,         999,           7,
epol_vegetation_specs       =   999,           3,         999,
```

From reaction scheme to PALM subroutines: kpp4palm and KPP

- **`chem_gasphase_mod.f90`** is specific for each chemistry mechanism and is created automatically by the **`kpp4palm`** preprocessor.
- **`kpp4palm`** makes use of the Kinetic PreProcessor **KPP**
- For details see file **README.md** in directory **`packages/chemistry/kpp4palm/`**
- **kpp4palm** (by Klaus Ketelsen and Jöckel et al, 2010, GMD, 3, 717–752)
  Starts the Kinetic PreProcessor KPP and converts KPP output to PALM4U module **`chem_gasphase_mod.f90`**
- Kinetic PreProcessor **KPP**
  http://people.cs.vt.edu/~asandu/Software/Kpp
  Atmos. Chem. Phys., 6, 187–195, 2006
  KPP translates a list of chemical reactions for a given chemical mechanism into Fortran77, Fortran90, C, or Matlab code
  - **Note**: For running KPP, the flex library and bison are required!

From reaction scheme to PALM subroutines: kpp4palm and KPP

$$NO_2 \xrightarrow{J} NO + O_3$$
$$NO + O_3 \xrightarrow{k} NO_2,$$

Fortran code in file **`chem_gasphase_mod.f90`** solves prognostic equations for

$$\left(\frac{\partial c_i}{\partial t}\right)_{chem}$$

with c_i standing in this example for NO_2, NO, O_3

Reaction rates

- Rate constants: Mostly Arrhenius equations

$$k = A \exp\left(\frac{E}{RT}\right)$$

A and E depending on reaction,
R gas constant, T temperature

Part of kpp4palm/KPP

- Photolysis frequencies J are not calculated in **`chem_gasphase_mod.f90`**

From reaction scheme to PALM subroutines: kpp4palm and KPP

$$NO_2 \xrightarrow{J} NO + O_3$$
$$NO + O_3 \xrightarrow{k} NO_2,$$

with $k = A \exp\left(\frac{E}{RT}\right)$

Example for input to KPP preprocessor for photostationary equilibrium (phstatp)

```
{phstat.eqn
Current revision
----------------
 20180319 Photostationary O3-NO-NO2-equilibrium
}
#EQUATIONS

  { 1.}  NO2 + hv = NO + O3 :                    phot(j_no2)  ;
  { 2.}  NO + O3 = NO2 :                  arr2(1.8E-12_dp, 1370.0_dp, temp) ;
```

From reaction scheme to PALM subroutines: kpp4palm and KPP

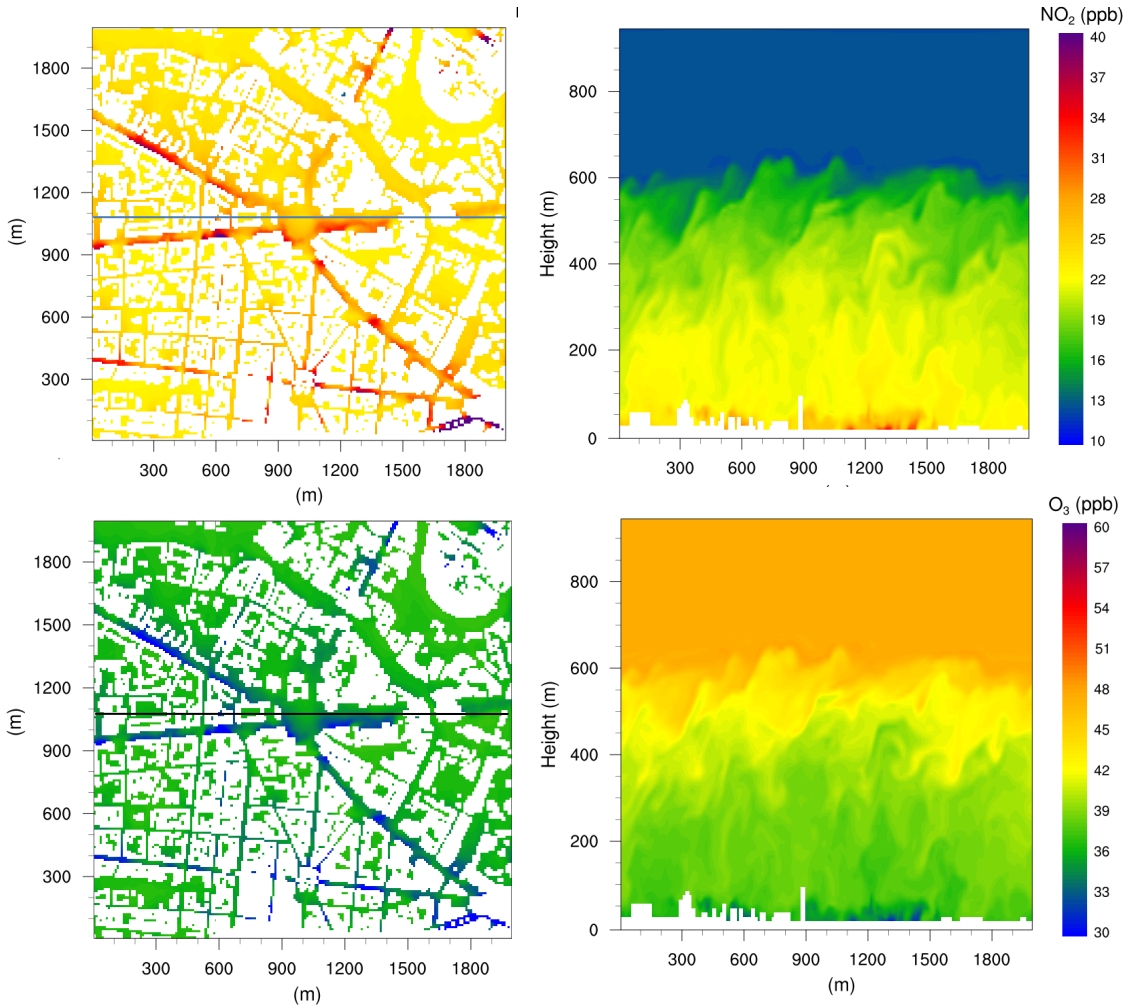
- For creating **`chem_gasphase_mod.f90`** for a new mechamism **`kpp4palm`** (linked to script in **`packages/chemistry/kpp4palm/scripts`**) must be run
- For applying the new mechanism, the newly created **`chem_gasphase_mod.f90`** must be copied into **`USER_CODE`**
- For instructions and background informations see **`README.md`** in **`packages/chemistry/kpp4palm/`**

Typical application for an urban area (Berlin, around Ernst-Reuter-Platz)

PALM simulation with chemistry for 30.July 2017 with

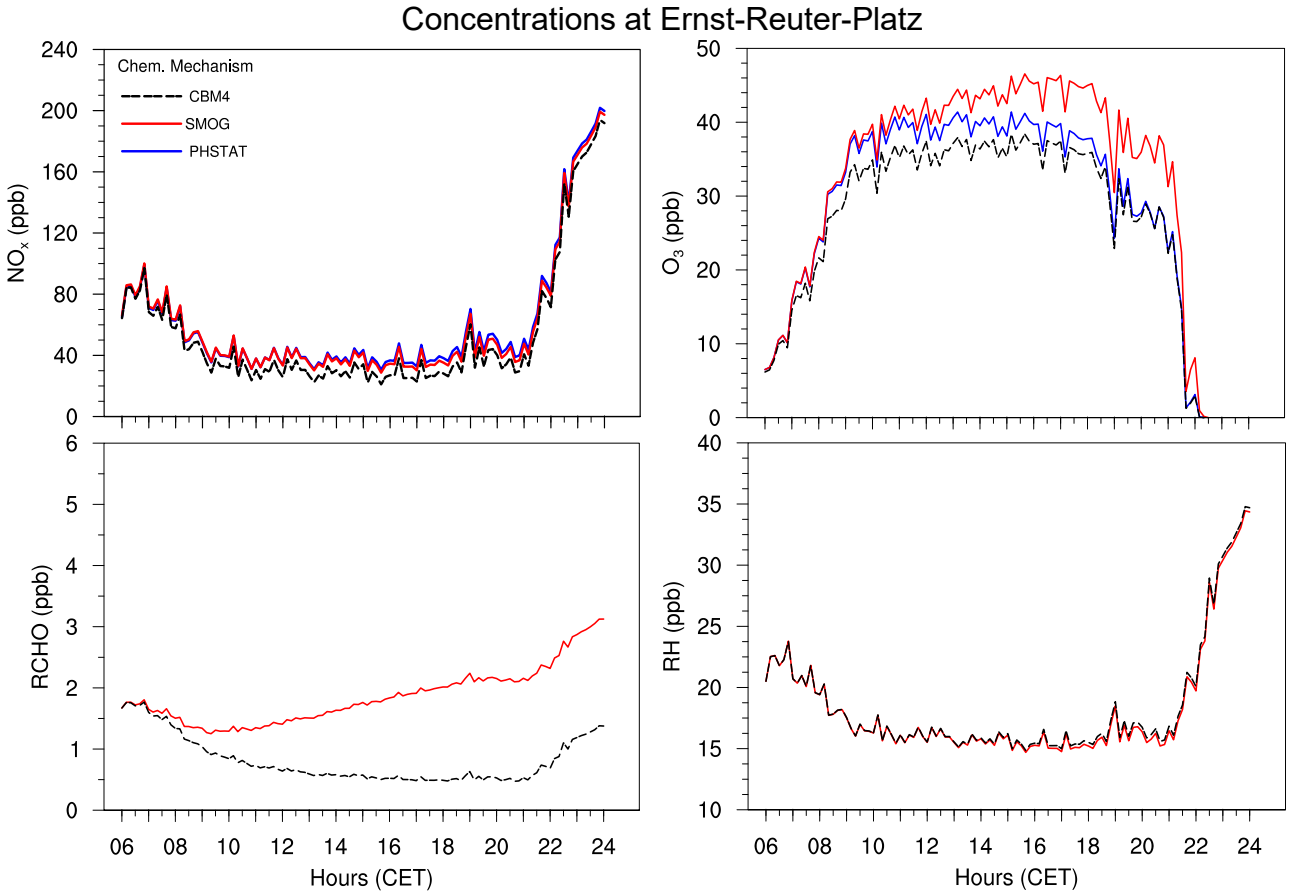
- cyclic boundary conditions for meteorology and for chemistry variables
- initial meteorolgy from COSMO
- 2 km x 2 km x 3.2 km
- 10 m grid width
- mechanism phstat
- parameterized emissions

Concentrations of ozone and NO_2 at 10 h CEST. Cross section along horizontal line.



Applications with different chemical mechanisms

- **cbm4**: Carbon Bond Mechanism (32 compounds, 81 reactions)
- **smog**: Photochemical smog mechanism (13 compounds, 12 reactions)
- **phstat**: Photostationary state (3 compounds, 2 reactions)



Results depend strongly on emissions. Here: Too high formaldehyde sources for smog?

Applications with different chemical mechanisms

- For many urban applications results for O_3 and NO_x will be quite similar to those with phstat. Consider whether phstat will do for your case study.
- Large mechanisms with many compounds and reactions are extremely time consuming. This is not only due to the computation of the chemical reactions. The transport of these many compounds needs even more computation time.
- Using small mechanisms with strongly simplified VOC chemistry like the ‚simple' mechanism is always a risk, they may run out or build up too much of some important compound.

„I do not like any of the available mechanisms, I want to build a new one."

Applications with different chemical mechanisms

Creating a chemical mechanism

- Sometimes this is useful. You can create any number of passive tracers.
- ‚Real' chemistry mechanisms are quite tricky - in particular the small ones (but there are many well tested larger mechanisms developed for mesoscale models)

Example: Extend an available mechanism by a passive tracer named PM10
Let's extend ‚smog' and name the new mechanism ‚smogp'

1) Enter **`packages/chemistry/kpp4palm/mechanisms`**
2) Create directory **`def_smogp`**
3) Create files **`smogp.eqn`**, **`smogp.spc`** and **`chem_gasphase_mod.kpp`**
   (in our case we can copy **`chem_gasphase_mod.kpp`**, **`smog.spc`** and **`smog.eqn`** from **`def_smog`** to **`def_smogp`**, rename and modify them.)
   Also copy **`UserRateLaws.f90`**.

Applications with different chemical mechanisms

Example: Creating a new chemical mechanism based on smog (cont.):

```
{smogp.eqn}
#EQUATIONS
{Photochemical smog with photolysis plus passive tracer PM10}
{ 1.}  NO2 + hv = NO + O :                    phot(j_no2)   ;
{ 2.}  O + O2 = O3       :                    arr2(3.2e-11_dp,  -70.0_dp, temp) ;
{ 3.}  NO + O3 = NO2 + O2 :                   arr2(1.8E-12_dp, 1370.0_dp, temp) ;
{ 4.}  RH + OH = RO2 + H2O :                  arr2(2.E-11_dp, 500.0_dp, temp) ;
{ 5.}  RCHO + OH = RCOO2 + H2O :              arr2(7.0E-12_dp, -250.0_dp, temp) ;
{ 6.}  RCHO + hv = RO2 + HO2 + CO :           phot(j_rcho)     ;
{ 7.}  HO2 + NO = NO2 + OH :                  arr2(3.7E-12_dp, -240.0_dp, temp) ;
{ 8.}  RO2 + NO = NO2 + RCHO + HO2 :          arr2(4.2E-12_dp, -180.0_dp, temp);
{ 9.}  RCOO2 + NO = NO2 + RO2 + CO2 :         arr2(5.4E-12_dp, -250.0_dp, temp) ;
{10.}  OH + NO2 = HNO3 :                      arr2(1.0E-12_dp, -713.0_dp, temp) ;
{11.}  RCOO2 + NO2 = RCOO2NO2 :               arr2(1.2e-11_dp, 0.0_dp, temp) ;
{12.}  RCOO2NO2 = RCOO2 + NO2 :               arr2(9.4E+16_dp, 14000.0_dp, temp) ;
{13.}  PM10 = PM10 :                          1.0_dp   ;
```

```
{smogp.spc}
...
    RCOO2NO2= ignore ;
    PM10    = ignore ;
```

Applications with different chemical mechanisms

Example: Creating a new chemical mechanism based on smog (cont.)

```
//chem_gasphase_mod.kpp
...
#include      smogp.spc
#include      smogp.eqn
...
```

4. Caution: Before proceeding with the next steps, read the instructions in **`packages/chemistry/kpp4palm/docs/content/Guide/`**
5. Run **`kpp4palm`** (link from **`bin`** to **`packages/chemistry/kpp4palm/scripts/kpp4palm.sh`** in order to create fortran code, e.g.:
   `kpp4palm -o output_dir -m smogp`
6. Note that running **`kpp4palm`** requires an output directory (-o option) where to write **`chem_gasphase_mod.f90`** (absolute path required).

Applications with different chemical mechanisms

Example: Creating a new chemical mechanism based on smog (cont.)

- Check resulting chem_gasphase_mod.f90 for smogp

```
MODULE chem_gasphase_mod

!   Mechanism: smogp
!
...
 CHARACTER(len=15), PARAMETER, DIMENSION(17):: spc_names =  (/ &
     'PM10            ','HNO3            ','CO              `,&
...
  CHARACTER(len=100), PARAMETER, DIMENSION(13):: eqn_names =  (/ &
     '         NO2 --> O + NO                                  `,&
     '        PM10 --> PM10
` /)
... etc etc etc ....
```

There is a large number of changes! Take a look with a diff tool, e.g.

```
kdiff3 def_smog/chem_gasphase_mod.f90 def_smogp/chem_gasphase_mod.f90
```

Nested chemistry run

- Adapt namelist for nested run as for meteo nested run

  e.g. for *_p3d

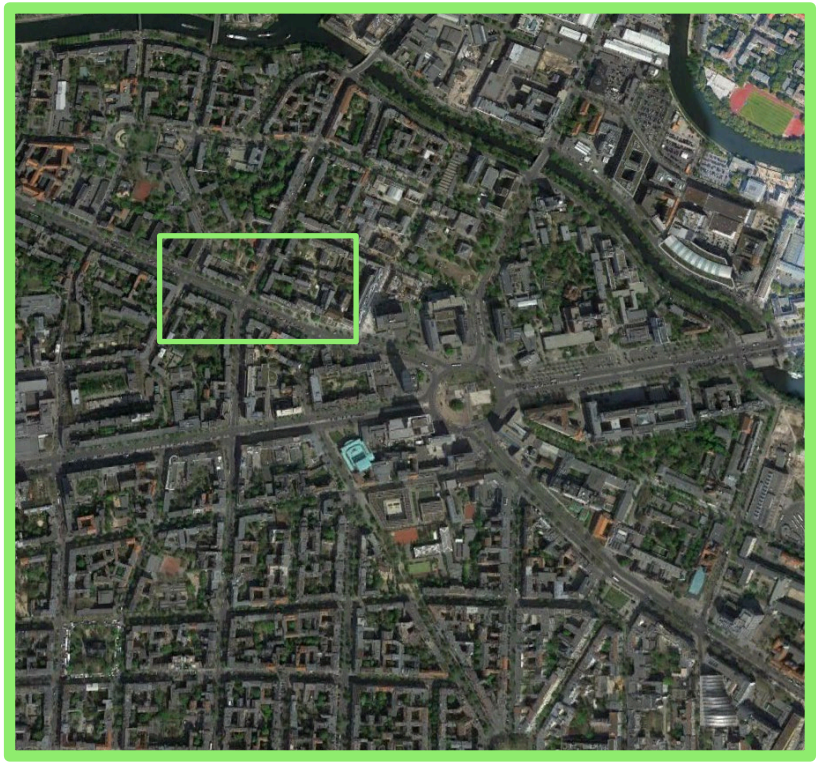
```
&nesting_parameters
    domain_layouts = 'parent',  1, -1,  16,    0.0,   0.0,
                     'child',   2,  1,  16,  280.0, 770.0,
```

  - ➢ No additional namelist entry needed for chemistry
  - ➢ Add *_p3d_N02
  - ➢ Add *_static, *_static_N02 and evtl. *_chemistry, *_chemistry_N02

Nested chemistry run

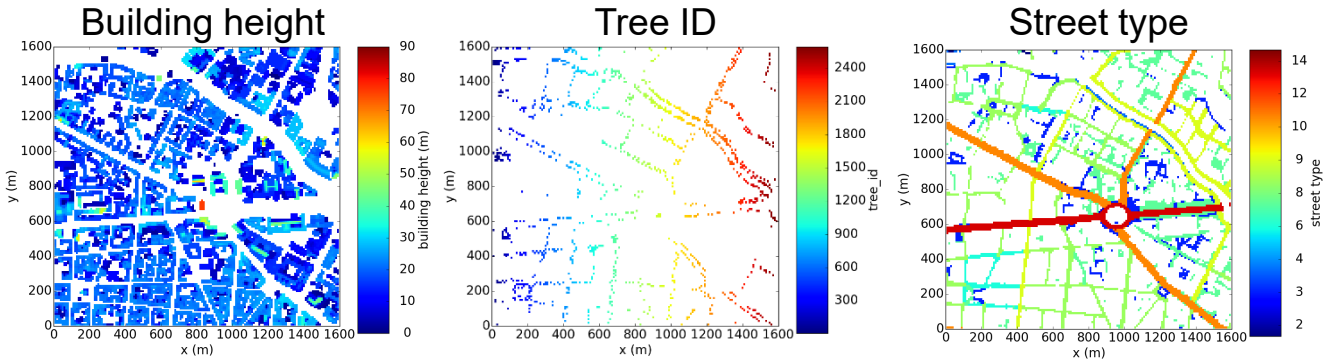
- Case study for a City quarter of Berlin, Germany, for the end of July 2017
  → IOP (Intensive Observation Period)

- Ernst-Reuter-Platz: Large roundabout with some high buildings and heavy car traffic
  - Nested run:
    1.6 km x 1.6 km x 3.8 km, 10 m resolution
    480 m x 240 m x 312 m, 1m resolution
- Chemistry options „photo-stationary state”
- Initialization with meteorological data from COSMO-DE model
- Cyclic boundary conditions for chemistry and meteorology for the parent domain

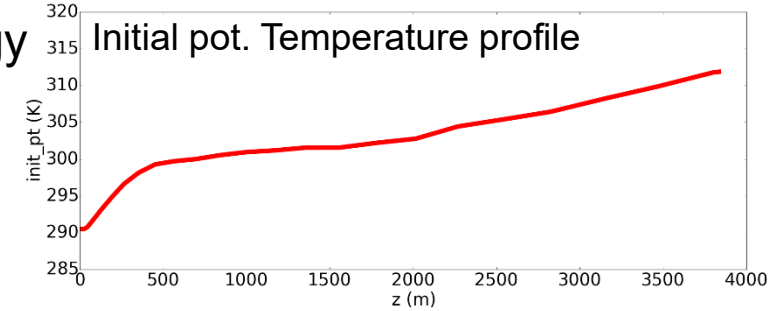


Nested chemistry run

- Static data

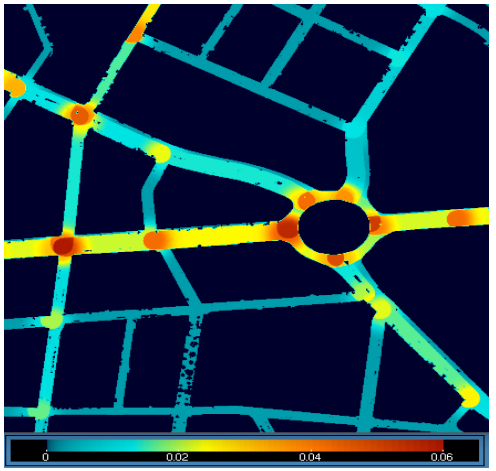


- Initial meteorology COSMO-DE meteorological data

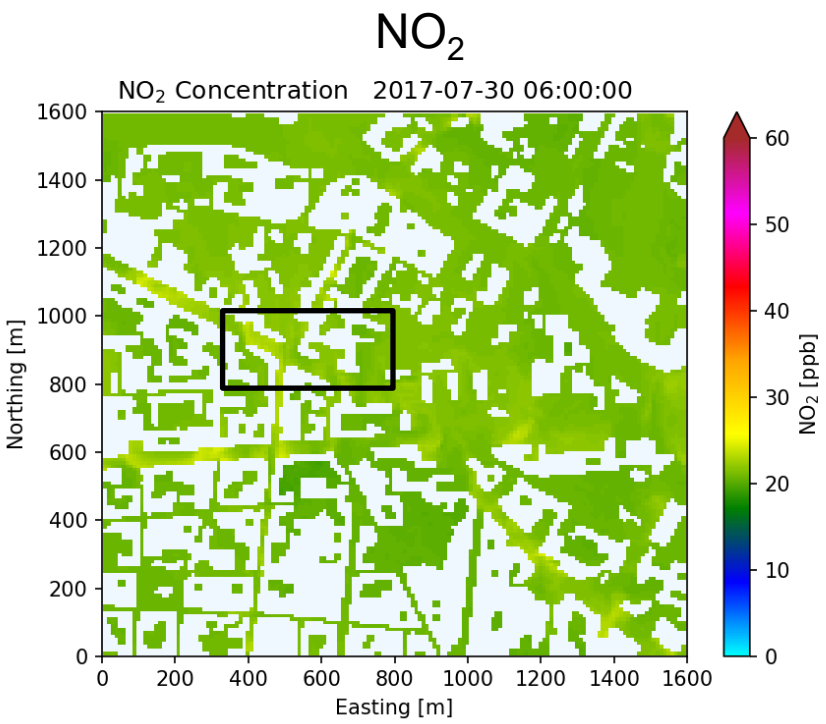


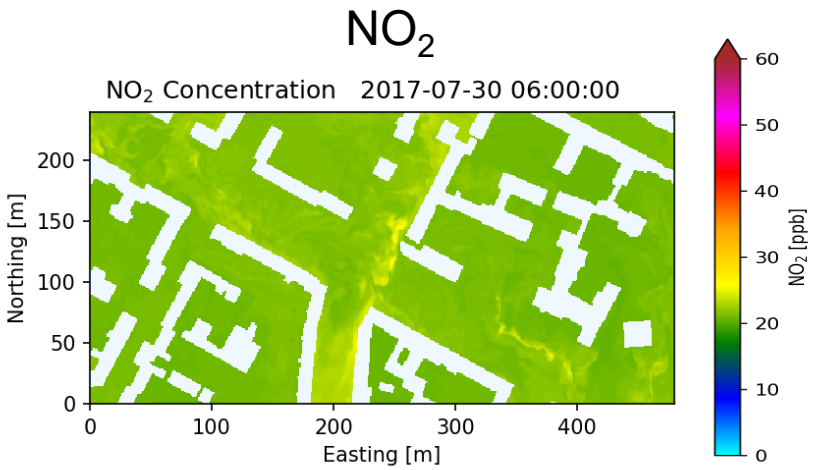
- Emission data
  - Data based on calculations applying the IMMIS^{em} emission model (Version 7.0; IVU Umwelt, 2018)
  - using HBEFA 3.3 emission factors (INFRAS, 2017)
  - using real-time traffic counts and temperature time series.
  - ➢ Gridded hourly real-time traffic emission data (LOD 2)

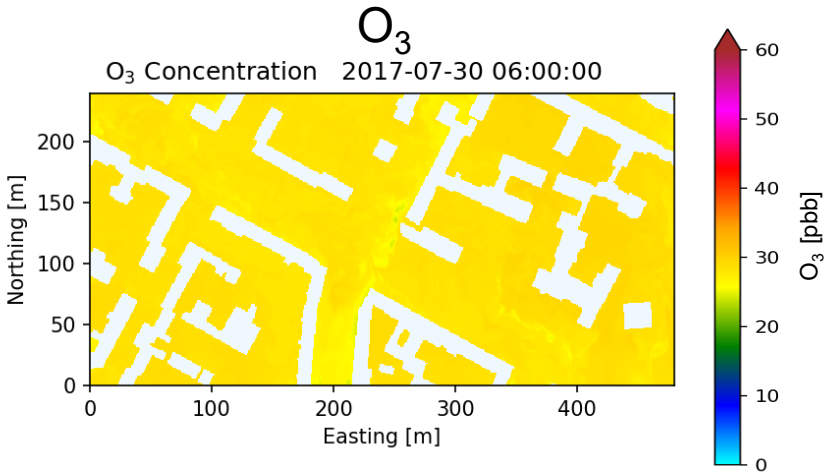
NO (in g/m^2/h)



Nested chemistry run

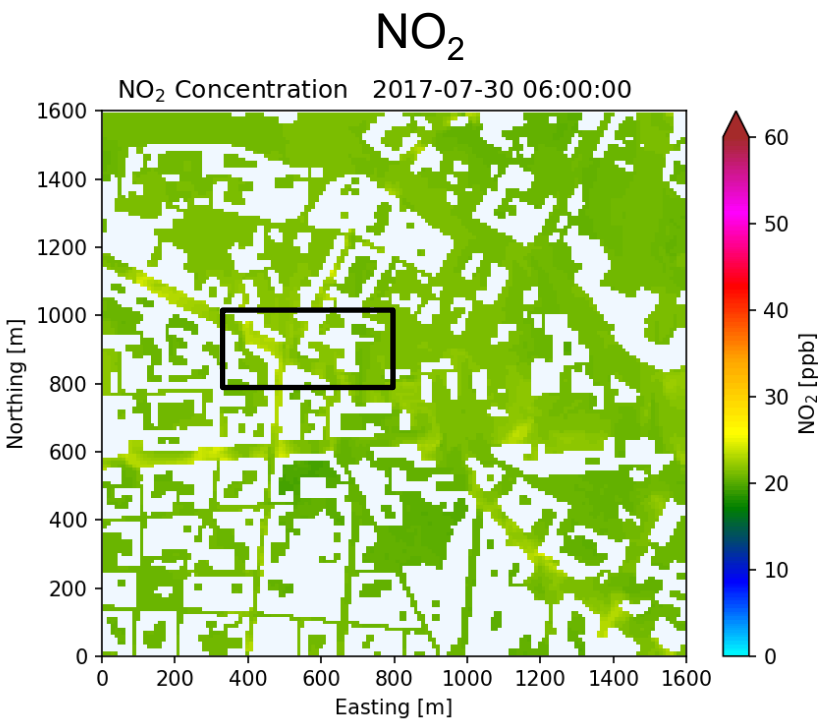


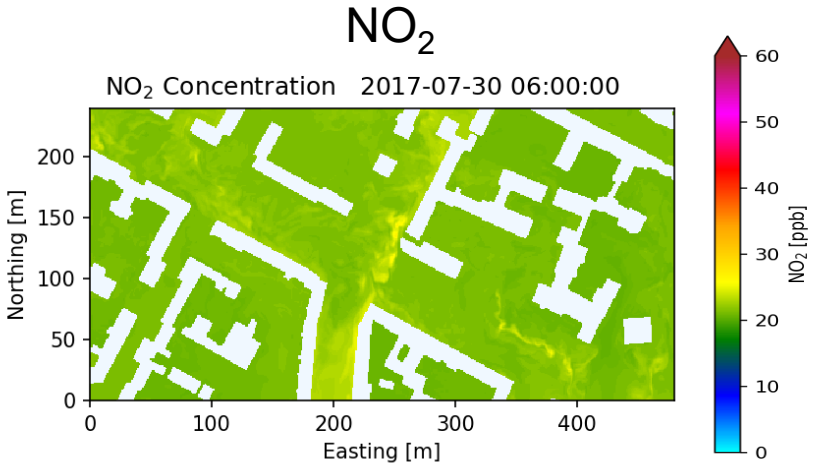


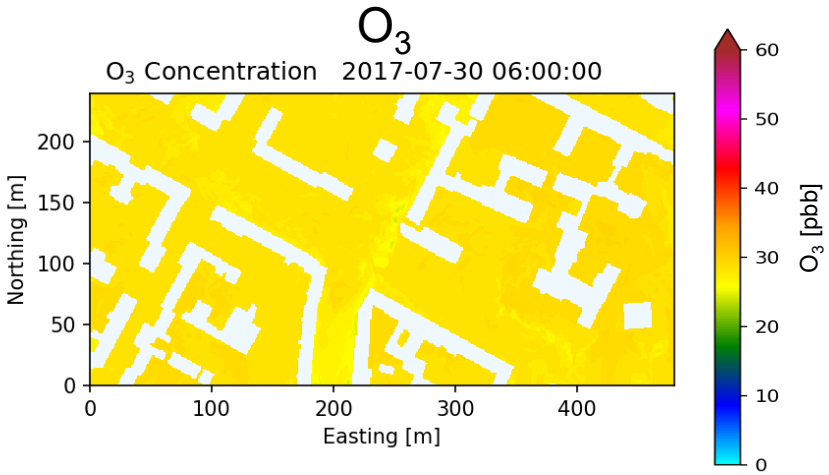


- Sunday, July 30th
- 6 – 8 UTC
- 18h run
- Photostationary state

Nested chemistry run

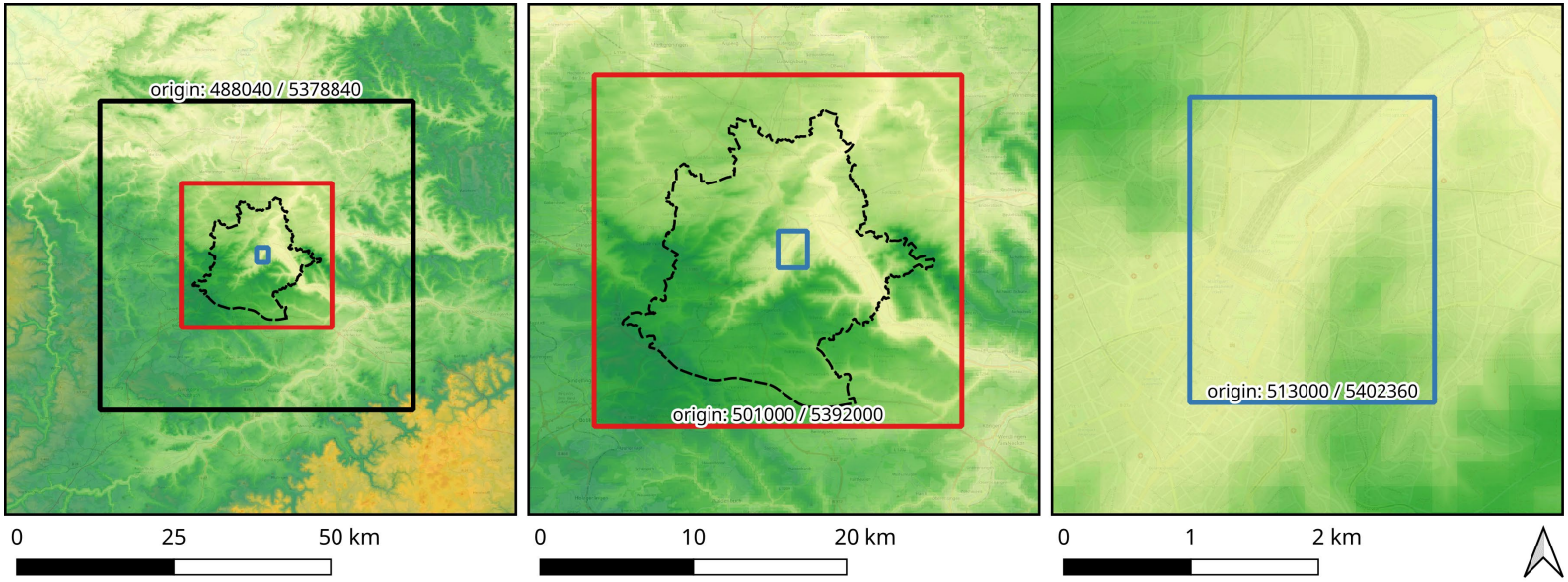






- Sunday, July 30th
- 6 – 8 UTC
- 18h run
- Photostationary state

Validation run for Stuttgart – 08.-09.07.2018



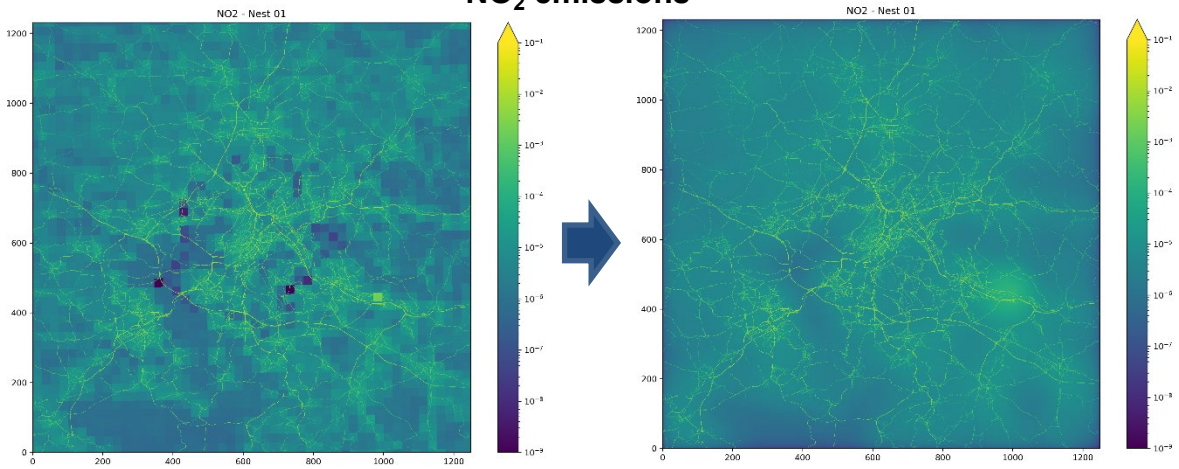
Nest-ID	x (m)	y (m)	dx (m)	nPE	nxpe	nype
1	49920	49280	40	858	39	22
2	24000	22960	10	3075	75	41
3	1920	2400	2	750	30	25

Validation run for Stuttgart – 08.-09.07.2018

Emission input (via chemistry driver → LOD2)

- Uni Stuttgart, IER (Dorothea Schmid, Alexander Altstadt)
- ➢ Basis: Emissions from EU-Project ICARUS[3] on 1x1 km[2] for 2015
- ➢ Temporal disaggregation using time profile from EDGAR emission database
- ➢ Sectors: Industrial comb., stationary comb., product and solvent use, rail roads, agriculture
- MATSim emissions
- ➢ Traffic sector

NO_2 emissions

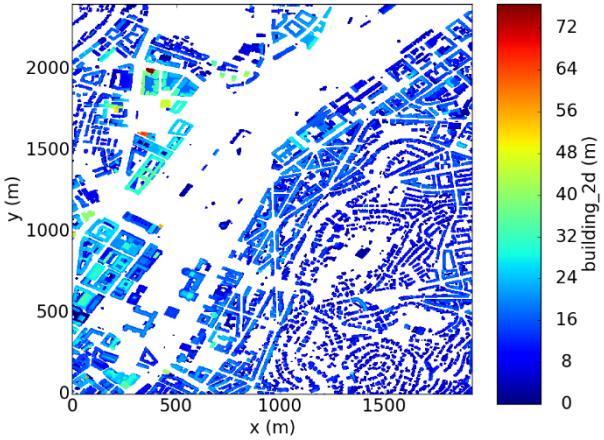


- **Chemical boundaries**
- ➢ WRF_Chem simulation
- ➢ Adapted inifor to include chemical boundaries in dynamic driver

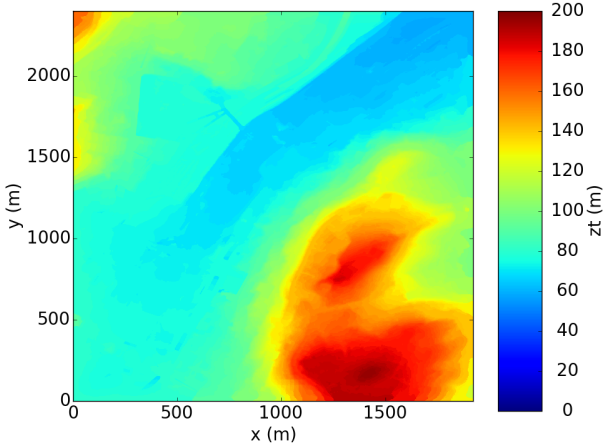
Validation run for Stuttgart – 08.-09.07.2018

- Nest 3

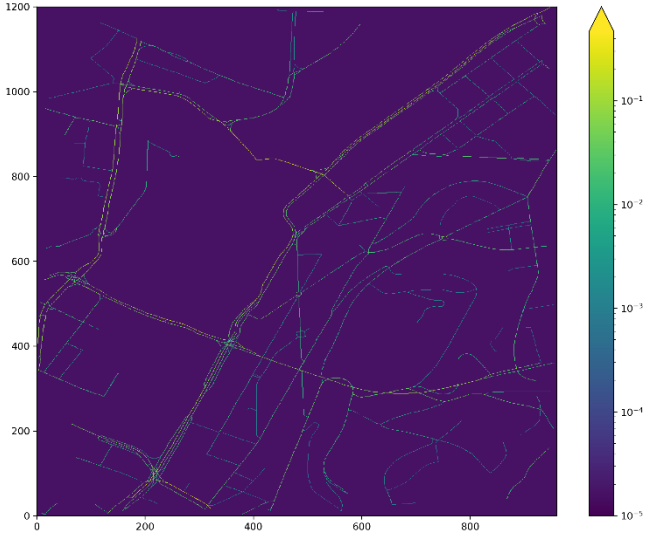
Building height



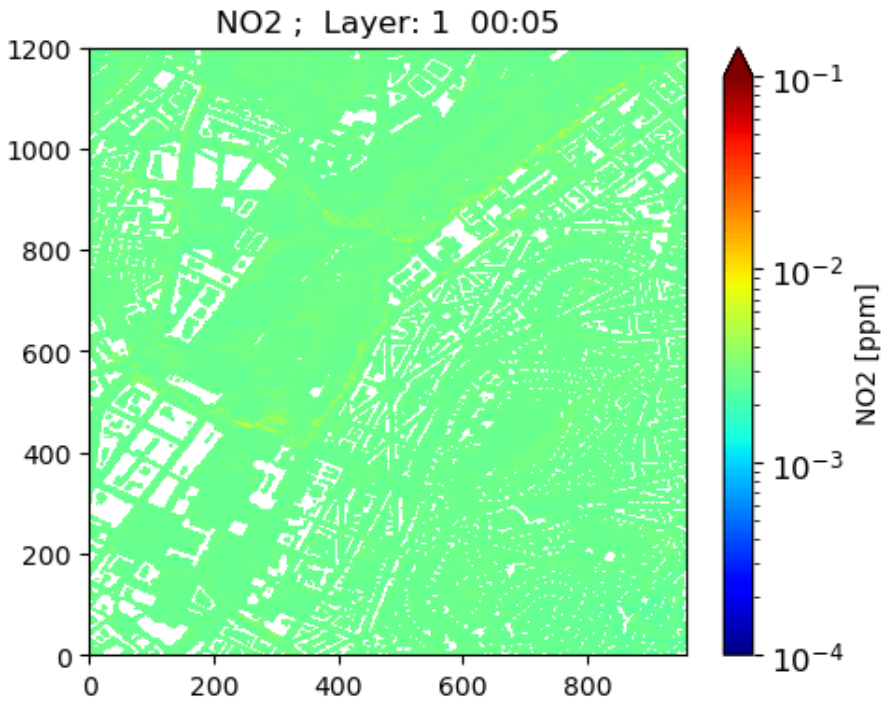
Orography

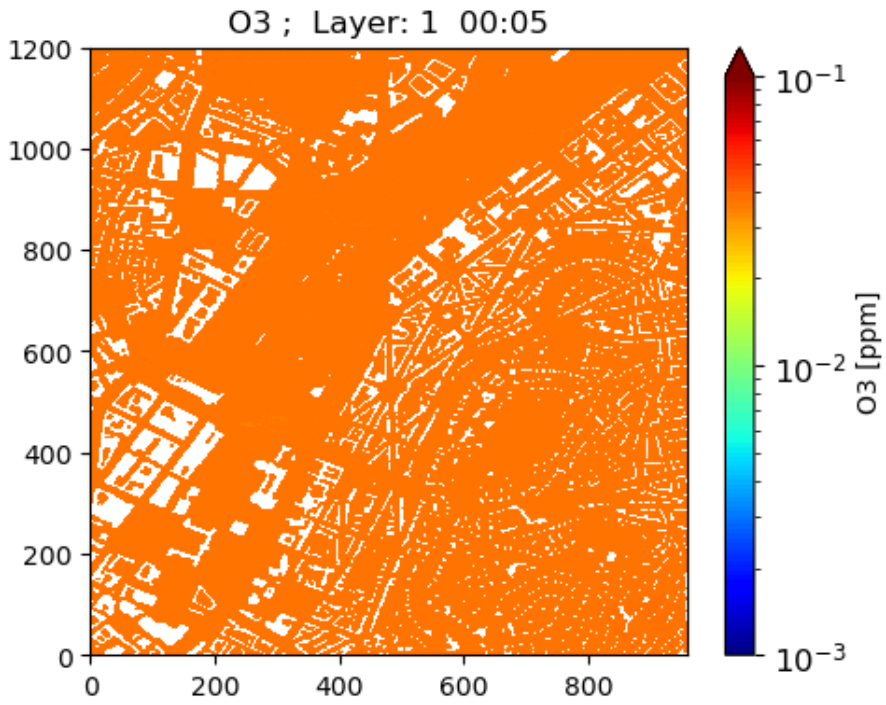


Emissions (NO_2 in $g/m^2/h$)

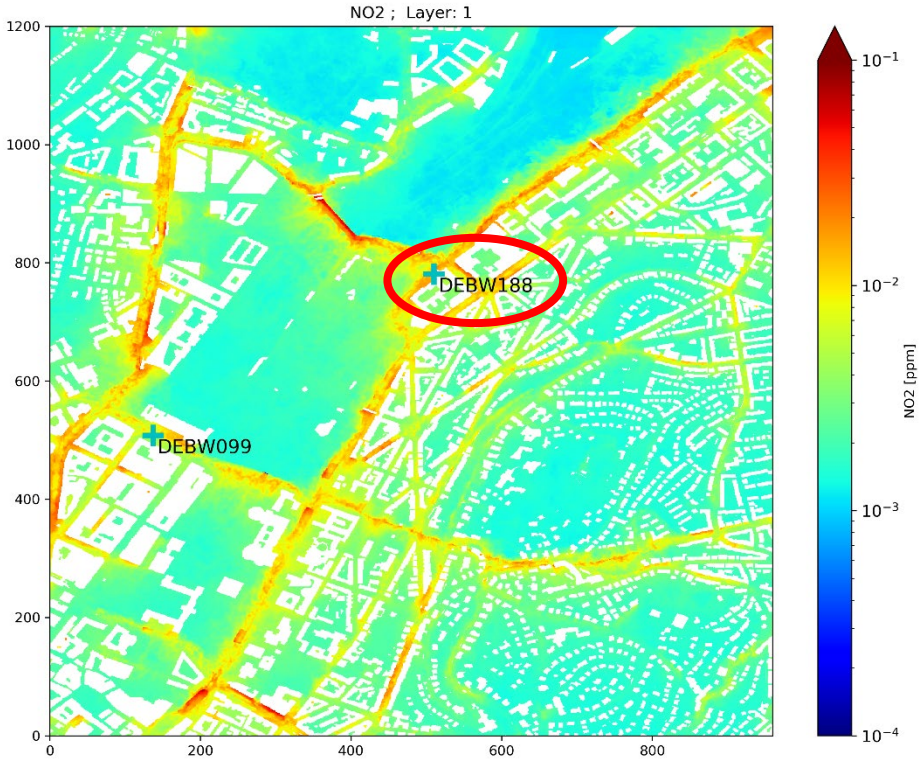


Validation run for Stuttgart – 08.-09.07.2018



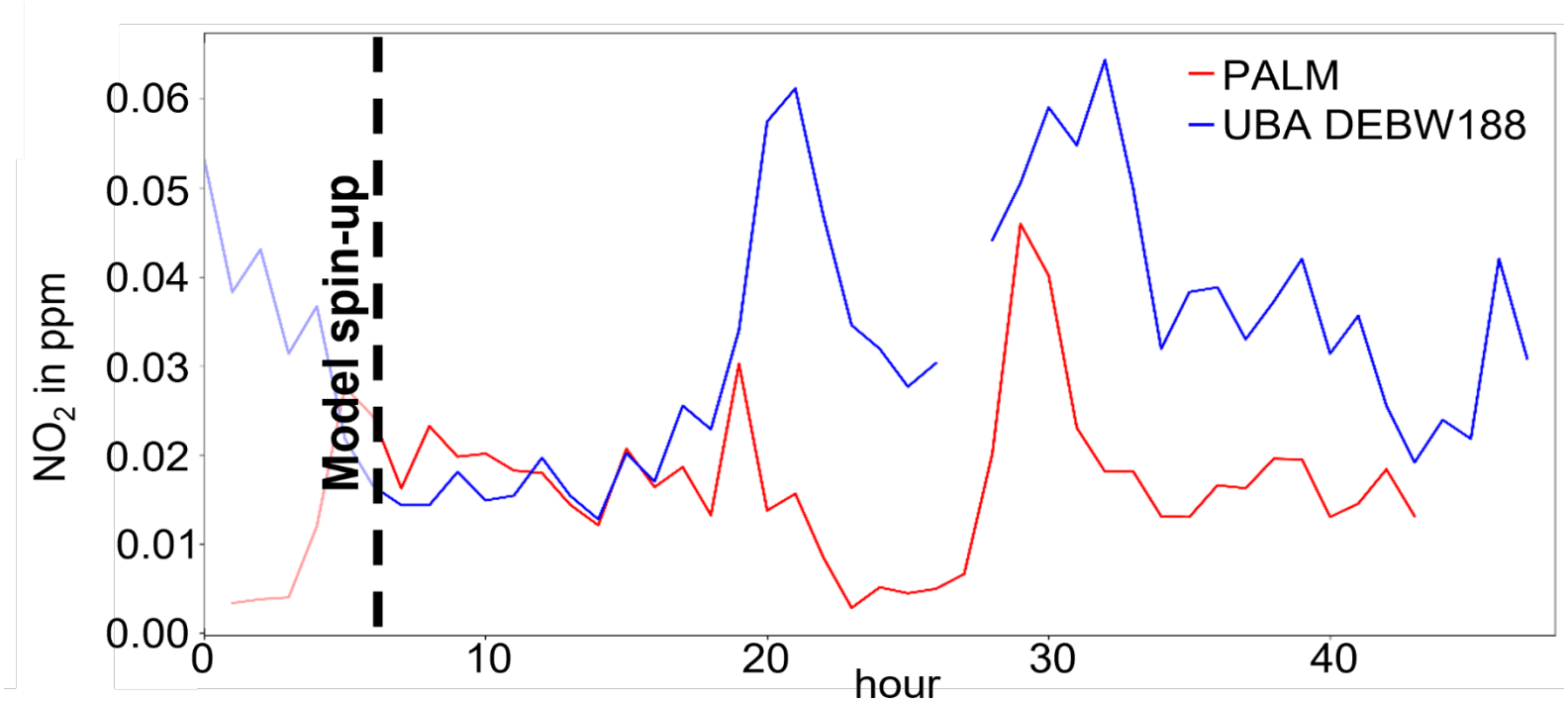


Validation run for Stuttgart – 08.-09.07.2018 - Comparison to observations

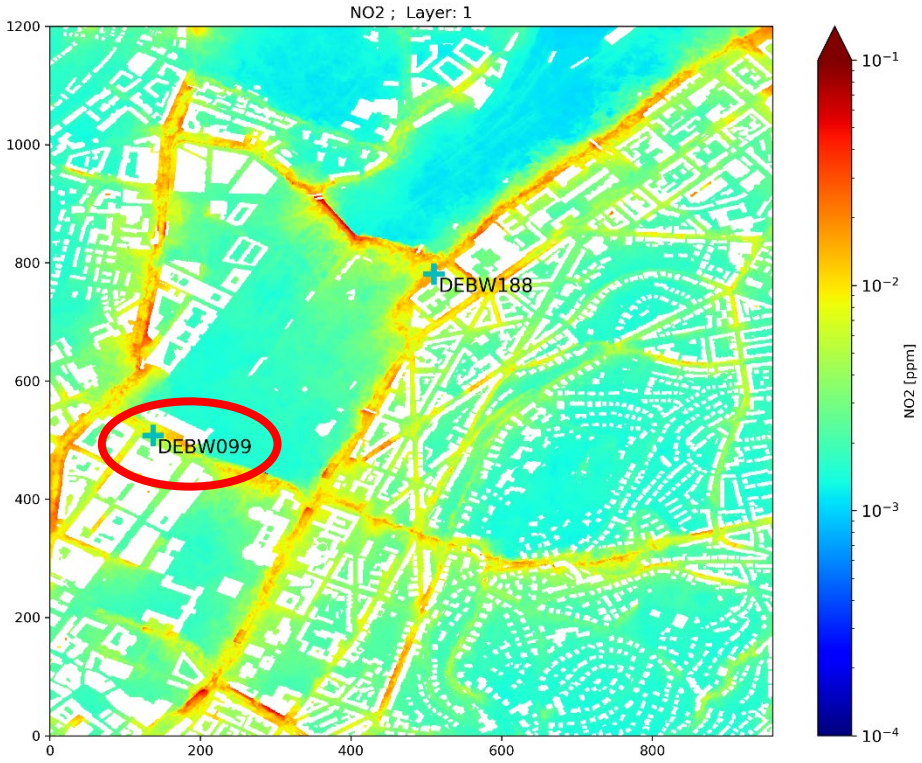


- **DEBW188 – Stuttgart Neckartor**
- DEBW099 – Stuttgart Anulf-Klett-Platz

Validation run for Stuttgart – 08.-09.07.2018 - Comparison to observations

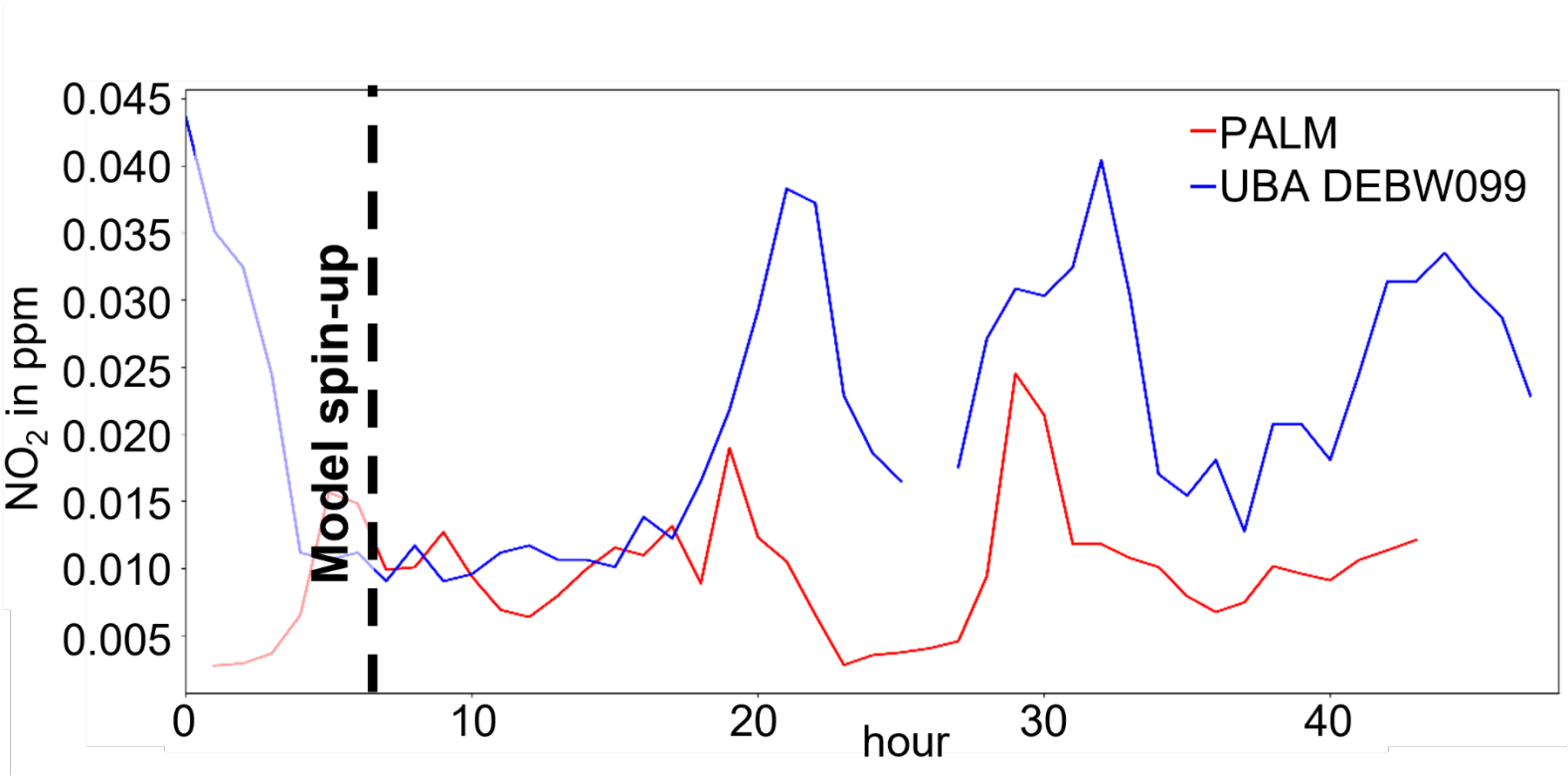


Validation run for Stuttgart – 08.-09.07.2018 - Comparison to observations



- DEBW188 – Stuttgart Neckartor
- **DEBW099 – Stuttgart Anulf-Klett-Platz**

Validation run for Stuttgart – 08.-09.07.2018 - Comparison to observations



Chemistry model

Summary

- Turbulence and building resolving LES model including chemistry
- Applicable for the urban environment on high resolution and for scales way beyond street canyon, up to city-scale
- Vector version available

- PALM-4U chemistry module still extendable!

Chemistry model

Outlook

Possible further developments and evaluation tasks

- Connection to multi agent module → pollutant exposure
- Chemical boundaries
- Secondary Organic Aerosols
- Number of included chemical mechanisms can be increased
- Additional photolysis model based on radiation model
- Particle resuspension
- Further urban quarter and city-scale applications, comparison to measurements

MOSAIK

[UC]² Urban Climate Under Change
Stadtklima im Wandel

Thank you for your attention!

reforkel@gmail.com
sabine.banzhaf@met.fu-berlin.de
basit.khan@kit.edu
edward.chan@rifs-potsdam.de
matthias.mauder@tu-dresden.de