Karlsruhe Institute of Technology

MOSAIK

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

PALM-4U Chemistry Module

Renate Forkel, Basit Khan, Matthias Mauder (KIT, IMK-IFU)
Sabine Banzhaf, Emmanuele Russo (FUB, TRUMF)

Renate Forkel, KIT, IMK-IFU, Kreuzeckbahnstr. 19, 82467 Garmisch-Partenkirchen
renate.forkel@kit.edu

Contents

- Background info: Prognostic equation for concentrations of chemical compounds
- Background info: Reaction rates
- From gas phase reaction scheme to code: KPP
- KPP implementation in PALM4U
- KPP input files
- Running PALM4U with chemistry
  - Modifications to 3dpar
  - Chemistry namelist chemistry_par
- Anthropogenic emissions
  - Emission options
  - Namelist example for parameterized first guess emission
- Todo list
- Concluding remarks

Gas phase chemistry background info: Basic equations

- **Prognostic equation for concentrations of chemical compounds**

$$\frac{dc_i}{dt} = -\frac{\partial}{\partial x}\left(\overline{u''c_i''}\right) - \frac{\partial}{\partial y}\left(\overline{v''c_i''}\right) - \frac{\partial}{\partial z}\left(\overline{w''c_i''}\right) + \left(\frac{dc_i}{dt}\right)_{chem} + S_i$$

i : index of chemical compound ($i=1,\ldots,n$)

c_i : concentration of chemical compound i

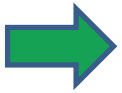
$\left(\frac{dc_i}{dt}\right)_{chem}$: change of c_i due to chemical reactions

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

Gas phase chemistry background info: Basic equations

- **Reaction scheme**

$$RH + OH \rightarrow R + H_2O$$
$$R + O_2 + M \rightarrow RO_2 + M$$
$$RO_2 + NO \rightarrow RO + NO_2$$
$$NO_2 + h\nu \xrightarrow{O_2} NO + O_3$$
$$RO + O_2 \rightarrow R'CHO + HO_2$$
$$HO_2 + NO \rightarrow OH + NO_2$$



Prognostic equations for NO_2, NO, OH, HO_2, RH etc.
(RH represents an Alkane)

- **Description of chemical conversion:**
  System of differential equations

$$\left(\frac{dc_i}{dt}\right)_{chem} = PROD_i - LOSS_i = P_i(\mathbf{c}_{j\neq i}) - g_i(\mathbf{c}_{j\neq i}) \cdot c_i$$

  Huge differences in speed of reactions (and thus g_i)
  $\Rightarrow$ stiff system of differential equations

Background info: Production and loss, reaction rates

- **1. order reactions**

  A → C+D Rate constant k

  $$\frac{d[C]}{dt} = \frac{d[D]}{dt} = -\frac{d[A]}{dt} = k[A]$$

  e.g. photolysis $\frac{d[A]}{dt} = -J_A[A]$

- **2. order reactions**

  A + B → C(+D)

  $$\frac{d[C]}{dt} = \frac{d[D]}{dt} = -\frac{d[A]}{dt} = -\frac{d[B]}{dt} = k[A][B]$$

- **3. order reactions**

  A + B +M→ C(+D)+M M: usually inert or abundant, e.g. nitrogen

  $$\frac{d[C]}{dt} = \frac{d[D]}{dt} = -\frac{d[A]}{dt} = -\frac{d[B]}{dt} = k'[A][B] \qquad k' = k[M] = k\frac{p^m}{RT}$$

Background info: Rate constants

- **Arrhenius equations**

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

A and E depending on reaction

- **Photolysis frequencies**

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

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

Besides computation of *J* with a radiation model also simplified parameterization may be used, e.g.
http://mcm.leeds.ac.uk/MCMv3.1/parameters/photolysis.htt or
http://www.emep.int/ozone/lagr_pho/photolysis_rates.html

From gas phase reaction rates to code: KPP

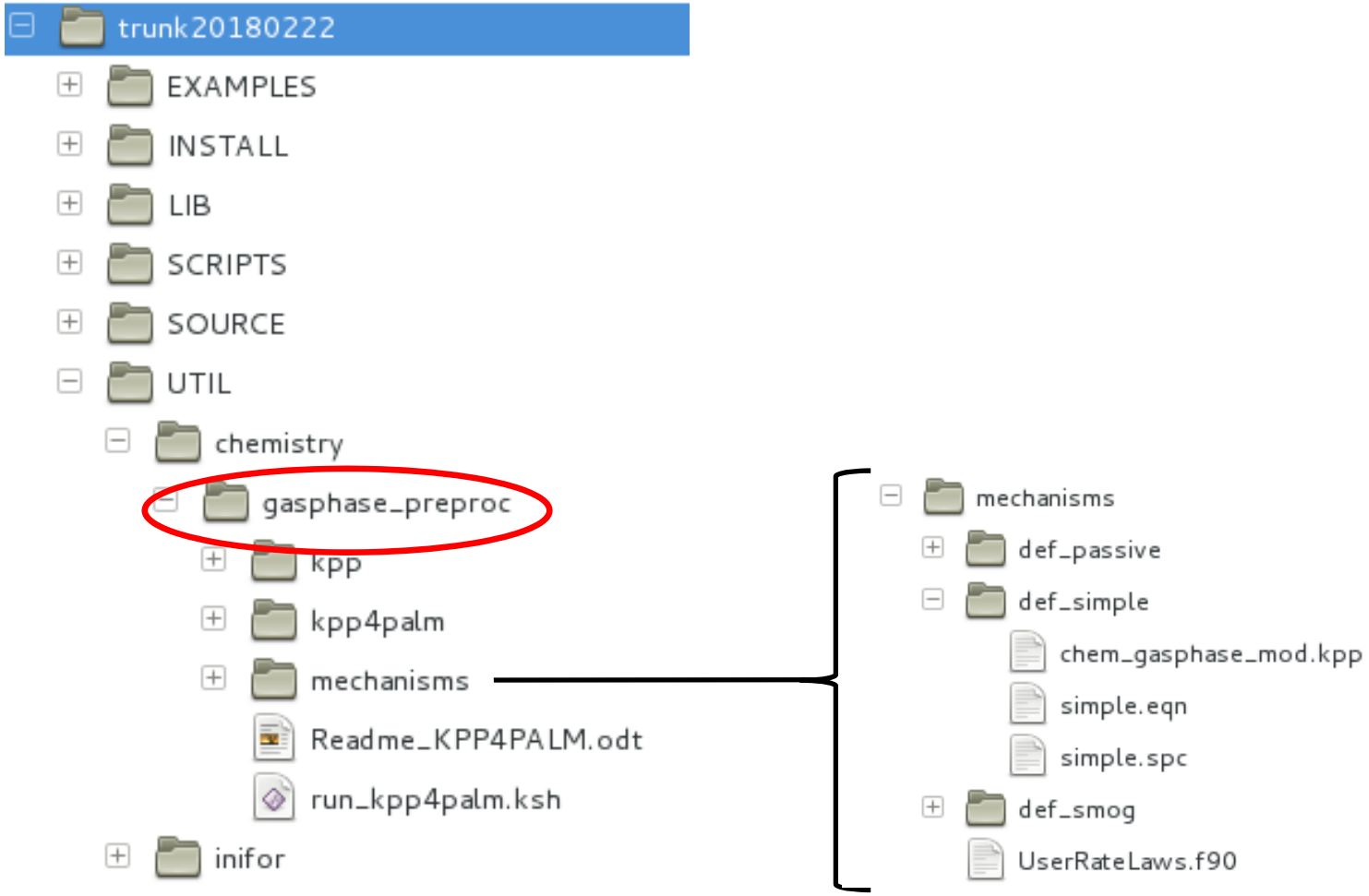
Stiff system of differential equations

- **Software tool KPP (Kinetic PreProcessor)**
  http://people.cs.vt.edu/~asandu/Software/Kpp
  Atmos. Chem. Phys., 6, 187–195, 2006,
  https://www.atmos-chem-phys.net/6/187/2006/
  - KPP translates a specification of the chemical mechanism into Fortran77, Fortran90, C, or Matlab code
  - KPP includes a comprehensive suite of stiff numerical integrators.

An alternative option would be removing fast reactions, e.g. by QSSA (Quasi steady state approximation): Error prone and not flexible ⇒ not applied

KPP implementation in PALM4U

- **Location of chemistry preprocessor within PALM4U**



KPP implementation in PALM4U

- **Directory kpp**

  Unchanged KPP from download site

  ➢ Note: For running KPP, the flex library is required!

- **Directory kpp4palm**

  Code that adapts KPP output to PALM4U (by Klaus Ketelsen)

- **Directory mechanisms**
  - Subdirectories containing specifications files for each mechanism (3 for each mechanism)
  - Currently just 3 mechanisms available: passive, simple, and smog (will be extended)
- **File Readme_KPP4PALM.odt** (additional explanations)
- **File run_kpp4palm.ksh**

  Script that starts KPP and creates chem_gasphase_mod.f90
  Application**: run_kpp4palm.ksh –m *mech***
  with *mech* standing for passive, simple or smog

KPP input files in mechanisms/def_*mech*

- **Contents of def_*mech***

  1. **General specifications: chem_gasphase_mod.kpp**
     - Specification of mechanism name, target language, solver
     - Link to photolysis model

  2. **List of chemical compounds: *mech*.spc**

```
{simple.spc}
#include atoms
#DEFVAR
    NO          = N + O ;       {nitric oxide}
    NO2         = N + 2O ;      {nitrogen dioxide}
    NO3         = N + 3O ;      {nitrogen trioxide}
    RH      = ignore ;
    RO2     = ignore ;
    RCHO    = ignore ;
…
#DEFFIX
    H2O     = H + 2O ;      {water}
    H2      = 2H ;          {molecular hydrogen}
    O2      = 2O ;          {molecular oxygen}
    N2      = 2N ;          {molecular nitrogen}
    CH4     = C + 4H ;      {methane}
    CO2     = C + 2O ;      {carbon dioxide}
```

Variable compounds

File may contain more compounds than required

Compounds that are considered as constant with time or are computed somewhere else (e.g. H2O)

KPP input files

- **Contents of def_*mech***

3) List of reactions: *mech*.eqn

```
{smog.eqn}
#EQUATIONS
{Photochemical Smog with photolysis}

{ 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 = RCO2 + 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.}  RCO2 + 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.}  RCO2 + NO2 = RCO2NO2 :               arr2(1.2e-11_dp, 0.0_dp, temp) ;
{12.}  RCO2NO2 = RCO2 + NO2 :               arr2(9.4E+16_dp, 14000.0_dp, temp) ;
```

`arr2`: Code for Arrhenius equations is created on the basis of file **UserRateLaws.f90** (more rates can be added there)

Running PALM4U with chemistry

- A prepared chem_gasephase.f90 is included in the SOURCE directory (currently prepared for passive)
- Calculation of gas phase chemistry is automatically activated if a **chemistry_par** namelist is available in the namelist file (_p3d).
- **Modifications to namelist 3dpar**
  Output for chemistry variable must be set, e.g. for ‘passive’

```
data_output = 'u',  'u_av', 'pt', 'pt_av',
              'kc_PM10', 'kc_PM25',
```

or for ‘simple’

```
data_output = 'u',  'u_av', 'pt', 'pt_av',
              'kc_NO', 'kc_NO2', 'kc_O3', 'kc_RCHO',
              'kc_RH',
```

Be careful to add only names of compounds that occur in the mechanism you are using!

Chemistry namelist chemistry_par

Example chemistry_par for 'passive'

```
&chemistry_par

   chem_gasphase_on           = .F.,    ! .F. = no chemical reactions
   call_chem_at_all_substeps  = .FALSE.,
   photolysis_scheme          = 'simple',

   surface_csflux_name        = 'PM10','PM25',
   surface_csflux             =  2.0,  0.01,

   cs_name                    = 'PM10','PM25',
   cs_surface                 =  15.0,  1.0,

   cs_profile(1,:)           =  15.0, 15.0, 10.0, 10.0,  5.0,  3.0,
   cs_heights(1,:)           =  0.0,  10.0, 12.0, 20.0, 25.0, 30.0,

   bc_cs_b                    = 'neumann',
   bc_cs_t                    = 'neumann',

   decycle_chem_lr           = .T.
   decycle_chem_ns           = .F.
/ ! end of chemistry_par namelist
```

For compounds with no surface_csflus_name speciefied, zero fluxes are assumed

cs_name refers to both, cs_surface and cs_profile
Zero value for species which are not mentioned

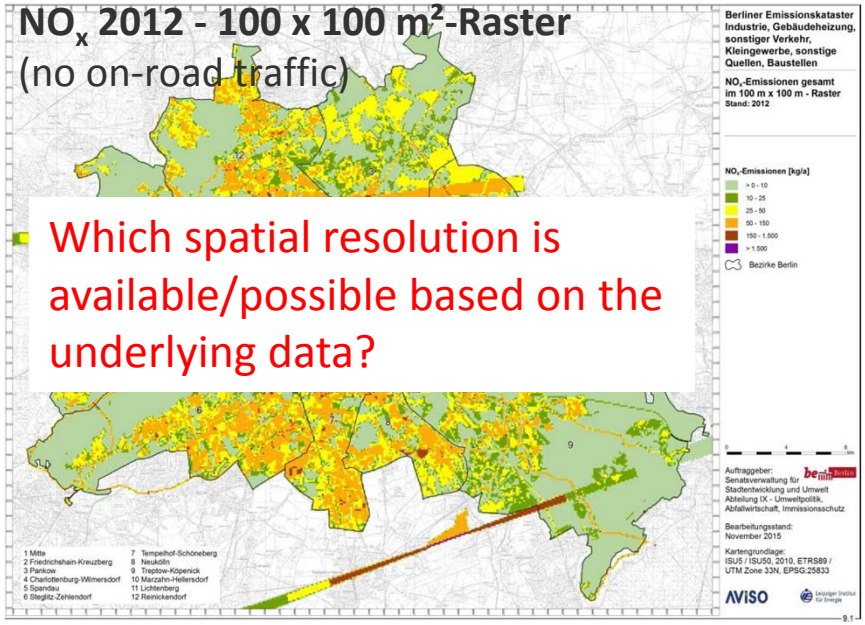
Chemistry namelist chemistry_par

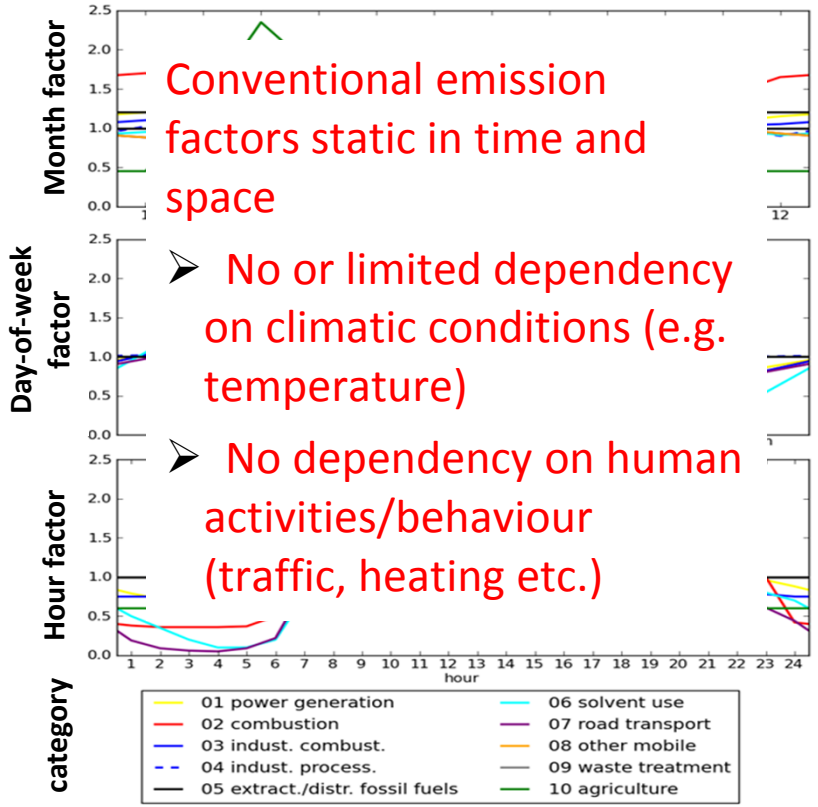
- **Example chemistry_par for ‚simple'**

```
&chemistry_par

   chem_gasphase_on           = .T.,    ! .F. = no chemical reactions
   call_chem_at_all_substeps  = .FALSE.,
   photolysis_scheme          = 'simple',

   surface_csflux_name        = 'NO', 'NO2', ,RH',
   surface_csflux             = 0.05, 0.005, 0.05,

   cs_name                    = 'O3', 'NO2', 'RCHO',
   cs_surface                 = 15.0,   1.0,    1.0,

   cs_profile(1,:)           =  15.0, 20.0, 30.0, 40.0, 50.0, 50.0,
   cs_heights(1,:)           =  0.0,  10.0, 12.0, 20.0, 25.0, 30.0,
   cs_profile(3,:)           =  15.0, 15.0, 10.0, 5.0,  3.0,   1.0,
   cs_heights(3,:)           =  0.0,   5.0, 15.0, 20.0, 25.0, 30.0,

   bc_cs_b                    = 'neumann',
   bc_cs_t                    = 'neumann',

   decycle_chem_lr           = .T.
   decycle_chem_ns           = .F.
/ ! end of chemistry_par namelist
```

Anthropogenic emissions

- Derived generally by:
  Emission = *AD* x *EF* (*AD*: Activity Data; *EF*: Emission Factor)
- Conventionally available on rather coarse resolution and on annual basis
  - ➢ **Disaggregation needed in space and time!**





Freie Universität Berlin
TrUmF
Renate Forkel, KIT, IMK-IFU Garmisch-Partenkirchen
renate.forkel@kit.edu
Karlsruhe Institute of Technology
KIT-Campus Alpin
IMK-IFU: Atmospheric Environmental Research

Anthropogenic emissions

Emission data input to PALM-4U

- To adapt to the end-user, three levels of detail will be available:
  - **DEFAULT**: The user has to provide gridded annual emissions per species and category → **spatial disaggregation by end-user**
    - Temporal disaggregation from annual emission values is provided for different categories by means of conventional emission factors. The latter factors are static in time and space
    - Species disaggregation for NOx, SOx, PM and VOCs is provided
  - **PRE-PROCESSED:** The end-user has to provide **spatially and temporally disaggregated emissions for each emitted species** in the format required by the model for each emission timestep
  - **PARAMETERIZED:** The user must provide street type data, emissions are parameterized on the basis of surface_csflux and road specific factors

Anthropogenic emissions

Emission data input to PALM-4U

- A variable that allows to switch between the three options is introduced in the model **namelist**
- Data is handed to PALM-4U via chemistry driver NetCDF file

Current status

- DEFAULT: Under development
- PRE-PROCESSED: Implemented, but not yet available in trunk
- PARAMETERIZED: Available

- Subcontracts for spatially and temporally disaggregated emissions from traffic, industry and households for IOP periods for Berlin

Anthropogenic emissions

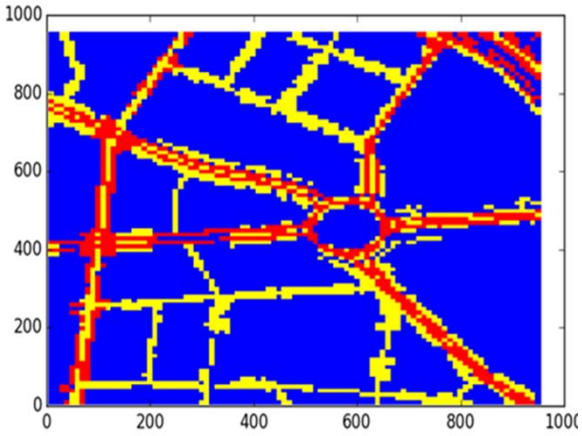
- **Set-up for first guess emissions on roads**
  - Surface emissions on roads based on the „street_type“ which has to be defined in the static_driver file
  - Discrimination between main and side roads can be performed
  - Emission flux for individual/each species and an emission factor for main and side roads must be defined in **namelist**
  - Main and side roads id read from static driver must be defined in **namelist**

  - Following **chemistry_par** example for current model version running with ‘passive’ mechanism
    → Two tracers PM10 and PM2.5
  - Tags for the road network as defined in OpenStreetMap

Anthropogenic emissions

- **Example chemistry_par with first guess emissions on roads**

```
&chemistry_par

   chem_gasphase_on           = .F.,
   call_chem_at_all_substeps  = .FALSE.,
   photolysis_scheme          = 'simple',

   surface_csflux_name        = 'PM10','PM25',
   surface_csflux             =  7.5,  3.75,

   cs_name                    = 'PM10','PM25',
   cs_surface                 =  13.0,  1.0,
   cs_profile(1,:)            =  15.0, 15.0, 10.0, 10.0,  5.0,  3.0,
   cs_heights(1,:)            =  0.0,  10.0, 12.0, 20.0, 25.0, 30.0,

   bc_cs_b                    = 'neumann',
   bc_cs_t                    = 'neumann',

   decycle_chem_lr            = .T.
   decycle_chem_ns            = .F.

   emiss_factor_main = 1.667,
   emiss_factor_side = 0.334,
   main_street_id = 11,
   side_street_id =  8,
   max_street_id = 19,
/ ! end of chemistry_par namelist
```



Side streets: Street ID 8 – 10

Main streets: Street ID 11 – 19

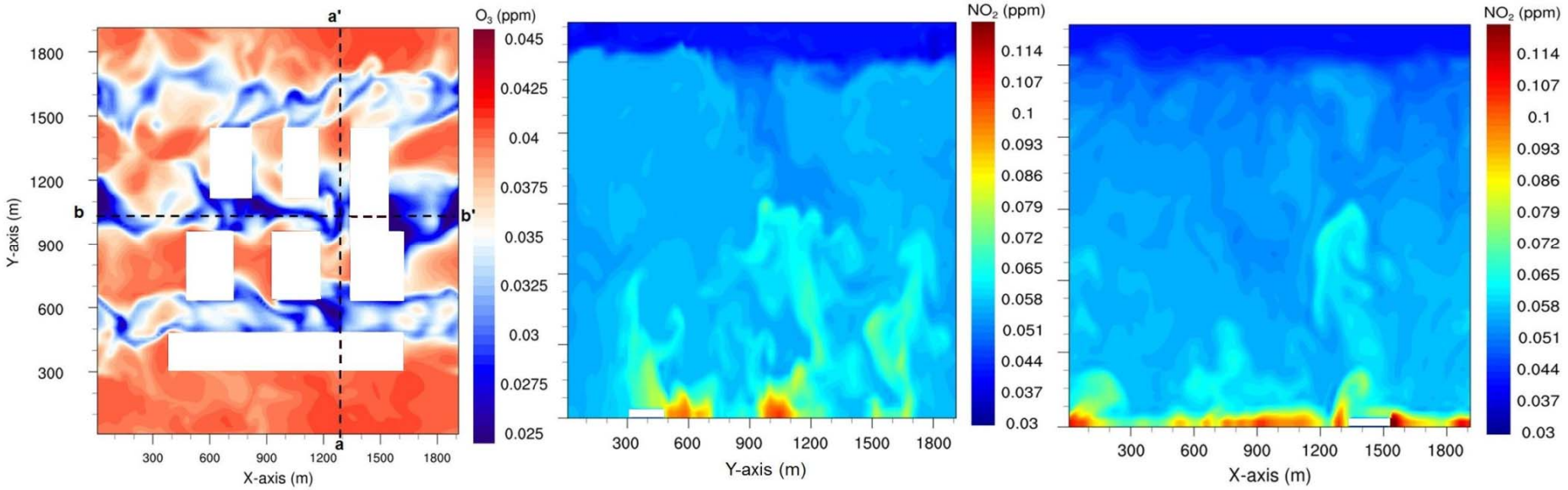
Todo list (only related to development)

- **To be finalized within the next months**
  - Improve decycling
  - Add more chemical mechanisms and
    more photolysis frequencies for simple parameterization
  - Add PRE-PROCESSED emissions routines to trunk
- **- within the next half year**
  - Finalize DEFAULT emissions module
  - Include dry deposition
  - Include simple BVOC emission module
- **- until the end of this year or early next year**
  - Include shading effects to parameterized photolysis
  - Complex photolysis model? Eventually postpone?
  - Simple efficient aerosol module

Renate Forkel, KIT, IMK-IFU Garmisch-Partenkirchen
renate.forkel@kit.edu

Concluding remarks

- **Problems**
  - Still dozens of small construction sites
  - Realistic time dependent boundary conditions
  - So far no work with RANS mode
  - Emission data sets: No project partner with full workpackage, only subcontracts (subcontracts can only do limited work)
- **Bright spots**
  - Contribution of KPP to PALM subroutine conversion by Klaus Ketelsen
  - Sectional aerosol module SALSA contributed by Mona Kurppa, University of Helsinki



Thank you for your attention!

Renate Forkel, KIT, IMK-IFU Garmisch-Partenkirchen
renate.forkel@kit.edu