Model of
Urban
Network of
Intersecting
Canyons and
Highways

MUNICH 2.0
User's Guide

CEREA (ENPC – EDF R&D)
Youngseob Kim, Lya Lugon,
Karine Sartelet, Yelva Roustan,
Thibaud Sarica, Alice Maison

http://cerea.enpc.fr/munich
https://github.com/cerea-lab/munich

Last update: September 1, 2021

Contents

1 Introduction

The Model of Urban Network of Intersecting Canyons and Highways (MUNICH) is used to simulate subgrid concentrations in the urban canopy represented by the street network [Kim et al., 2018].

MUNICH consists of two main components:

- street-canyon component, which represents the atmospheric processes in the volume of the urban canopy.
- street-intersection component, which represents the processes in the volume of the intersection.

These components are designed to connect to the Polair3D model at roof level and are also interconnected [Kim et al., 2018].

In this new version, aerosol model SSH-aerosol `http://cerea.enpc.fr/ssh-aerosol` [Sartelet et al., 2020] is coupled to MUNICH to simulate the concentrations of primary and secondary aerosols.

MUNICH has been used for several modeling studies, for example, Lugon et al. [2020a], Lugon et al. [2020b], Wu et al. [2020], Gavidia-Calderón et al. [2021], Lugon et al. [2021].

2 Download

MUNICH can be downloaded at the homepage `http://cerea.enpc.fr/munich/` or the git repository `https://github.com/cerea-lab/munich`. MUNICH consists of its core part and sub modules including SSH-aerosol. Each sub module has the own git repository. To clone all repositories at same time, using a shell script at https://raw.githubusercontent.com/cerea-lab/munich/master/utils/munich-public-clone is recommended.

3 Requirements

MUNICH is designed to run under Unix or Linux-based systems. And it is based on three scientific languages: C++, Fortran 77/90 and Python. Tested C++ compilers are GNU gcc 4.x, 5.x, 7.x, 9.x. Corresponding Fortran compilers are tested: GNU gfortran 4.x, 5.x, 7.x, 9.x. For any compiling issue, please report to `munich-help@liste.enpc.fr`.

Python version 3.x is supported.

3.1 External Libraries and Python Modules

With regard to software requirements, below is a list of requirements

- the C++ library Blitz++ (`https://github.com/blitzpp/blitz` and see Section 3.1.1).
- Blas/Lapack (compiled libraries): any recent version. If you install Blas/Lapack using a package manager like apt or yum, you need to install liblapack-dev.
- NetCDF library may be required if you have precomputed coagulation repartition coefficients in SSH-aerosol model. You can download from the following site: `http://www.unidata.ucar.edu/downloads/netcdf`

- SWIG (`http://www.swig.org/index.php`)

- NumPy: any recent version. Make sure that your versions of NumPy and Matplotlib (see below) are compatible.

- Matplotlib: any recent version and corresponding pylab version (usually, pylab is included in Matplotlib package). It is recommended to install the corresponding version of Basemap in order to benefit from AtmoPy map-visualizations. Basemap is a toolkit available on Matplotlib website (`https://matplotlib.org/`), but usually not included in Matplotlib package.

- SciPy: any recent version.

3.1.1 Install Blitz++

If you install Blitz++ using a package manager like apt or yum, you need to install libblitz0-dev and libblitz0v5.

If you can not install using a package manager, you can install from source files. To do that, `cmake` is required to build Blitz++ from source files.

```
cd blitz
mkdir build
cd build
cmake ..
make lib
```

And edit CMakeCache.txt to indicate where blitz is installed.

```
CMAKE_INSTALL_PREFIX:PATH=YourHome/usr/local
```

```
make install
```

3.2 Build

MUNICH is supposed to be compiled with SCons. Please make sure that SCons has already been installed on your system `https://www.scons.org`. SCons is supposed to find the compilers and to properly determine all dependencies, on any platform.

Please set the following paths to the installed packages (Blitz++, Lapack/Blas, etc) in your environment before compiling.

```
export CPLUS\_INCLUDE\_PATH=YourHome/usr/local/include:$CPLUS\_INCLUDE\_PATH
export LD\_LIBRARY\_PATH=PATH\_TO\_LAPACK:$LD\_LIBRARY\_PATH
```

If Lapack/Blas is installed by the package manager, `PATH_TO_LAPACK` should be `/usr/lib`

```
cd processing/ssh-aerosol
compile
```

If you are not able to install Scons, you can execute standalone Scons package. Please download any recent scons-local package at `https://scons.org/pages/download.html` and compile with scons-local package as follows:

```
cd munich
mkdir scons-local
cd scons-local
tar xvf scons-local-4.0.1.tar.gz
cd ../processing/ssh-aerosol
python3 ../../scons-local/scons.py
```

If you experience anyway problems, please contact the MUNICH mailing list `munich-help@liste.enpc.fr`.

3.3 Parallel Computing

The chemistry part of MUNICH is parallelized with OpenMPI `https://www.open-mpi.org`. Versions 1.10, 2.1, 4.0 are tested. The street segments are partitioned for available cores.

By default, MUNICH is compiled with OpenMPI. If you would like to compile without OpenMPI, a compiling option can be added.

```
compile --mpi=no
or
compile -m=no
```

4 Programs

4.1 Main program: `ssh-aerosol`

If compiling is successful, `munich-ssh` program is generated. It runs with the main configuration file `munich.cfg`.

```
cd processing/ssh-aerosol
munich-ssh munich.cfg
```

Three configuration files need to be modified: `munich.cfg`, `munich-data.cfg` and `munich-saver.cfg`.

To run a test simulation with provided input data at `http://cerea.enpc.fr/munich/`, please edit `Directory` field in `munich-data.cfg`.

```
Directory: Path_to_input_data
```

Setting of the configuration files is explained in detail in Section 5.1.

4.2 Legacy program without aerosol model: `photochemistry`

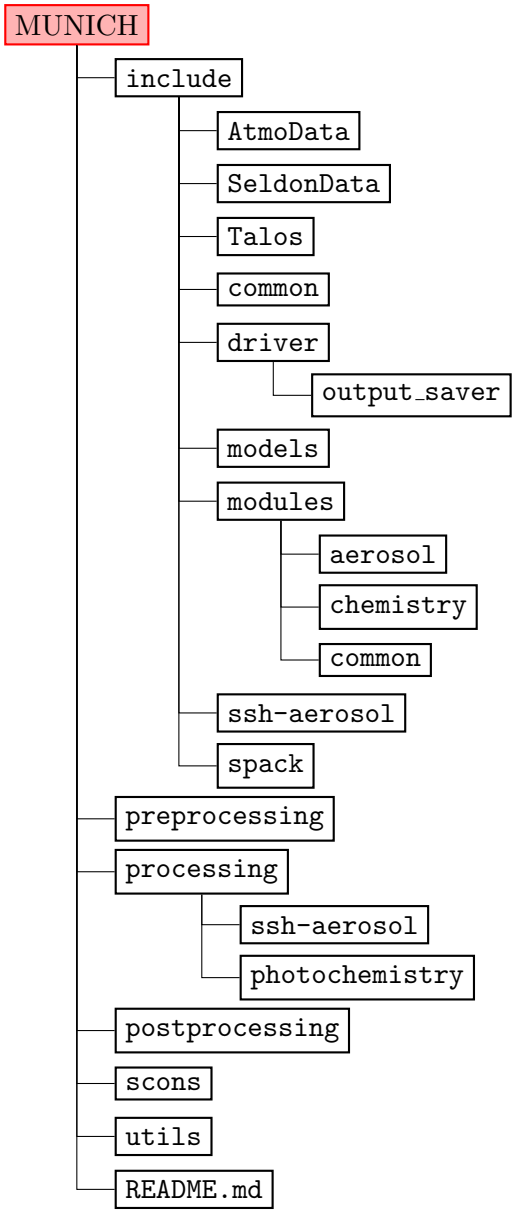
A legacy program `photochemistry` is also available. Only gas-phase chemistry is taken into account in this program without coupling to SSH-aerosol.

```
cd processing/photochemistry
scons mpi=yes
munich munich.cfg
```

Setting of the configuration files is explained in detail in Section 5.2.

5 Source code

The following diagram shows the tree structure of the source code.



The folder `include` is where all source code files are stored.

`AtmoData` is a tool for data processing in atmospheric sciences.

`SeldonData`: C++ library to perform data processing (interpolations, input/output operations);

`Talos`: C++ library to manage configuration files, dates and string processing;

`common`: mostly functions used to parse and manage the arguments of preprocessing programs;

`driver` contains `StreetDriver` and subroutines for output saver.

`models` contains models including three models `StreetNetworkAerosol`, `StreetNetworkChemistry` and `StreetNetworkTransport` which are used by the driver `StreetDriver`

`modules`

- `aerosol`: contains functions which allow to communicate with SSH-aerosol model.
- `chemistry`: chemical kinetic mechanisms. These routines are generated from a list of reactions and species specified in the folder `spack`.
- `common`: base modules and parallelism module

`ssh-aerosol` contains SSH-aerosol model.

`spack` contains the gas-phase chemical model generator. It is only used for the legacy program `photochemistry`.

The folder `processing` contains two subfolders `ssh-aerosol` and `photochemistry`.

For the modeling of gas and aerosol species, the program `munich-ssh.cpp` in its subfolder `ssh-aerosol` can be used. This program uses functions in `aerosol` module for communication with SSH-aerosol model. SSH-aerosol model (`http://cerea.enpc.fr/ssh-aerosol`) version 1.2.2 is coupled to MUNICH to simulate primary and secondary aerosol concentration in the streets. The coupling between MUNICH and SSH-aerosol is done using the API (Application Programming Interface) module of SSH-aerosol. The API allows to use all functions of SSH-aerosol in MUNICH and communicate arrays between the models. The model options of SSH-aerosol are configured in `namelist.ssh`. SSH-aerosol is described in User Manual of the model (`http://cerea.enpc.fr/ssh-aerosol/user_manual.pdf`).

The second program `munich.cpp` in its subfolder `photochemistry` is provided for the modeling of only gas-phase species. The second module `chemistry` is used for this program. It implements five chemical mechanisms: RACM, RACM2, CB05, MELCHIOR2, Leighton ozone photostationary state relation [Kim et al., 2009]. It uses a second-order Rosenbrock method for time integration. Computations are performed by Fortran routines (automatically generated by the chemical preprocessor SPACK, see SPACK Guide `https://www.cerea-lab.fr/dossiers/racine/articles/guide-0.pdf`) and a C++ program is used as a frame to launch all these calculations.

Information about species and reactions is given below.

Chemical mechanisms	no of species	no of reactions
`RACM`	72	237 including 23 photolysis
`RACM2`	113	349 including 34 photolysis
`CB05`	52	155 including 23 photolysis
`Leighton`	4	3 including 1 photolysis
`MELCHIOR2`	48	112 including 21 photolysis

Users can choose a program depending on the purpose of modeling.

Some useful tools are added to generate input data and to visualize output data in the folder `preprocessing` and `postprocessing`, respectively. `scons` folder contains the files necessary for compiling the program.

5.1 Configuration for `ssh-aerosol`

The main program `munich-ssh` in the folder `processing/ssh-aerosol` is configured with three configuration files (`munich.cfg`, `munich-data.cfg` and `munich-saver.cfg`) and data files in-

cluding `intersection.dat`, `street.dat` and `species-cb05.dat`. The main configuration file `munich.cfg` provides the paths to all other configuration files and data files.

5.1.1 Main Configuration File: `munich.cfg`

The main configuration file `munich.cfg` gives information on the options of the simulation. There are many sections, e.g., `[display], [domain]`, in the file.

The section `[display]` is configured with StreetDriver in the folder `driver`

	`[display]`
`Show_iterations`	If activated, each iteration is displayed on screen.
`Show_date`	If activated, the starting date of each iteration is displayed on screen in format `YYYY-MM-DD HH:II` (For notations, please see Section 9).
`Show_configuration`	If activated, some major options are displayed on screen.

The sections `[domain]`, `[data]` and `[output]` are necessary to set general information about the simulation.

	`[domain]`
`Date_min`	Starting date in any legal format (see Section 9). The date can therefore include seconds.
`Delta_t`	Time step in seconds.
`Nt`	Number of iterations of the simulation (integer).
`Bin_bounds`	List of bounds of aerosol size bin (in μm).
`Species`	Path to the file that defines involved species and their chemical properties.
	`[data]`
`Data_description`	Path to the configuration file that describes input data.
	`[output]`
`Configuration_file`	Path to the configuration for the output saver.

The section `[options]` describes simulation parameters. For some parameters, default value is used when they are not configured by user.

	`[options]`
`With_initial_condition`	If initial concentrations of gas-phase species are taken into account.
`With_initial_condition_aerosol`	If initial mass concentrations of aerosol are taken into account.
`With_initial_condition_number_aerosol`	If initial number concentrations of aerosol are taken into account.
`With_deposition`	Is dry deposition taken into account?
`Compute_dep_SVOC`	If set to yes, dry deposition velocities for SVOC are computed.
`Config_dep_SVOC`	File containing the input data for SVOC deposition (`svoc-dep.dat`).
`With_deposition_aerosol`	Is dry deposition taken into account for aerosol species?

Particles_dry_velocity_option	Parameterization used to compute dry deposition velocities for particles (venkatran, zhang, giardina, muyshondt).
Brownian_diffusion_resistence_option	Parameterization used to compute diffusion resistence for giardina option (paw, chamberlain).
With_scavenging	Is scavenging taken into account?
With_scavenging_aerosol	Is there scavenging for aerosol species?
With_resuspension	Is resuspension taken into account?
Resuspension	File containing the input data for resuspension (resuspension.dat).
With_drainage_aerosol	Is drainage taken into account?
Max_rain	Precipitation that indicates a complete drainage in streets (recommended value: 0.254 in mm/h).
Min_water_drainage	recommended value: 0.5 in mm.
Collect_dry_flux	Are the dry deposition fluxes collected in order to postprocess them if dry deposition is taken into account?
Collect_wet_flux	Are the wet deposition fluxes collected in order to postprocess them if wet deposition is taken into account?
Collect_dry_flux_aerosol	Are the dry deposition fluxes are collected in order to postprocess them if dry deposition is taken into account?
Collect_wet_flux_aerosol	Are the wet deposition fluxes are collected in order to postprocess them if wet deposition is taken into account?
With_chemistry	Is chemistry taken into account?
With_number_concentration	If yes, number concentration is calculated. If no, number concentration is estimated from mass.
With_bg_number_concentration_data	If yes, the model reads the background number concentration, if no, these concentration is calculated using aerosol mass and size data.
Redistribution_method	Which redistribution method is used (moving-diameter, siream, siream-euler-coupled, or siream-moving-diameter)?
aqueous_module	Which aqueous module is used (No, simple or VSRM)?
With_fixed_density	Is aerosol density fixed in all cells and all bins?
Fixed_aerosol_density	Fixed aerosol density in $\mathrm{kg\,m^{-3}}$ used in the model.
Wet_diameter_option	Which method is used to compute wet diameter: gerber, chemistry or none. gerber uses Gerber formulation. chemistry uses the same wet diameter computed in chemical module. none uses dry diameter.

The section [street] describes some additional parameters which are associated with street-scale modeling.

[street]	
With_stationary_hypothesis	Stationary hypothesis or not
sub_delta_t_min	minimum time step for numerical solver when stationary hypothesis is not used.
Numerical_method_parameterization	numerical solver: ETR or Rosenbrock.

Transfer_parameterization	Parameterization to compute turbulent transfer velocity: "Sirane" or "Schulte"
Mean_wind_speed_parameterization	Parameterization to compute mean wind speed within the street-canyon: "Sirane" or "Exponential"
Building_height_wind_speed_parameterization	Parameterization to compute wind speed at the roof level: "Sirane" or "Macdonald"
Zref	Reference altitude for wind speed (m)
With_horizontal_fluctuation	If the horizontal fluctuation is taken into account.
Deposition_wind_profile	Wind profile option for dry deposition: masson or macdonald.
Intersection	File containing the input data for intersections.
Street	File containing the input data for streets.
Minimum_Street_Wind_Speed	Minimum wind speed within the streets (in m/s).
Building_density	Building density for dry deposition: 0.2 for sparse suburban area, 0.4 for suburban area, 0.6 for downtown area.
With_local_data	If meteo data and background concentrations are available for each street.

5.1.2 Data Description File: `munich-data.cfg`

This configuration file describes input data files (binary files). It is divided into sections which describe binary files for input data. A section roughly looks like this:

```
Directory: data-munich/

[meteo]

Date_min: 20140316_00
Delta_t = 3600.
Nt = 2568

Fields: WindDirection WindSpeed PBLH UST LMO WindDirectionInter
        WindSpeedInter PBLHInter USTInter LMOInter Attenuation
        SpecificHumidity SurfacePressure SurfaceTemperature Rain
Filename: <Directory>/meteo/&f.bin
```

It is assumed that all binary files start at the same date, and this date is `Date_min` (see dates formats in Section 9). The time step is `Delta_t`, in seconds.

Then a list of fields is provided after `Fields`. These are fields that the model needs, and their names are determined by the model. Below, all fields required by the model (depending on its options) are listed. A generic path (full file name) is then provided (entry `Filename`). In this path, the shortcut '`&f`' refers to a field name.

Note that:

1. entries `Fields`, `Filename` and additional paths *must* be at the *end of the section.*
2. *at least one element* (possibly not a required field) must be provided to `Fields` and at least one element (possibly not a path) to `Filename`; for instance:

   ```
   Fields: ---
   Filename: --- # means no generic path.
   ```

but:

```
Fields:    # Illegal: one element required.
Filename:     # Illegal: one element required.
```

In most sections, `Fields` is used to specify all chemical species involved in the process, e.g.:

```
[emission]

Date_min: 20140316_00
Delta_t = 3600.
Nt = 2568

Fields: ALDX API CH4 ETH ETHA ETOH FORM IOLE ISOP LIM MEOH OLE PAR TERP TOL XYL NO NO2
Filename: <Directory>/emission/&f.bin

ALD2 ALD2.bin
CO    0.002
```

If a few fields are not stored in a file with a generic path, their specific paths can be provided after the entry `Filename`. This is the case for `ALD2` in the above example.

Notice that `CO` is not associated with a path but with a numerical value. This is a feature: a binary file may be replaced with a numerical value. In this case, the field (in the example, CO emission) is set to a constant value (in every location and at every time step). This works with any field, including meteorological fields (section `[meteo]`).

In `munich-data.cfg`, several sections are required. In the following table, all possible sections are listed, with their entries.

Section	Entries	Comments
`[initial_condition]`	`Fields`, `Filename`	
`[initial_condition_aerosol]`	`Fields`, `Filename`	
`[emission]`	`Date_min`, `Delta_t`, `Nt`, `Fields`, `Filename`	
`[meteo]`	`Date_min`, `Delta_t`, `Nt`, `Fields`, `Filename`	Required fields are: `WindDirection`, `WindSpeed`, `PBLH`, `UST`, `LMO`, `WindDirectionInter`, `WindSpeedInter`, `PBLHInter`, `USTInter`, and `LMOInter`. `Attenuation`, `SpecificHumidity`, `SurfacePressure`, `SurfaceTemperature`, `Rain` with option `with_chemistry`.
`[background_concentration]`	`Date_min`, `Delta_t`, `Nt`, `Fields`, `Filename`	
`[emission_aer]`	`Date_min`, `Delta_t`, `Nt`, `Fields`, `Filename`	
`[bg_concentration_aer]`	`Date_min`, `Delta_t`, `Nt Fields`, `Filename`, `Format`	default value for `Format` is Internal
`[bg_number_concentration]`	`Fields`, `Filename`	`Date_min`, `Delta_t`, `Format` of `bg_concentration_aer` section are used.

[deposition]	Fields	
[scavenging]	Fields	
[deposition_velocity_aerosol]	Fields	List of size sections
[scavenging_aerosol]	Fields	List of size sections
[ground]	Roughness_file, LUC_file, Urban_index_zhang	Urban_index_zhang = 10 for GLC2000 database
[traffic]	Date_min, Delta_t, Nt, Fields, Filename	Required fields are: RoadTraffic_2R, RoadTraffic_HDV, RoadTraffic_PC, RoadTraffic_LCV

The *units* of concentration input data (e.g., initial condition, boundary condition etc.) should be given as $\mu g/m^3$.

5.1.3 Output Description File: `munich-saver.cfg`

Some parameters must be provided to save output results. The file type of output files is binary as input data.

```
Output_dir: results/

[save]

# Put "all" to output all species.
Species: all

Date_beg: -1 # Put -1 to start from the simulation initial date.
Date_end: -1 # Put -1 to end at the simulation final date.
Interval_length: 36 # 1 for all steps.
Averaged: yes
# Save initial concentrations in case concentrations are not averaged?
Initial_concentration: no

# Choices: streeet
Type: street

Output_file: <Output_dir>/&f.bin

Text_file: no
```

[save]	
Species	Chemical species to be saved. If it is set to all, concentrations for all species are saved.
Date_beg	The date from which the concentrations are saved. If concentrations are averaged, the first step at which concentrations are actually saved if not Date_beg, but Date_beg plus the number of steps over which concentrations are averaged. If the value - 1 is supplied, Date_beg is set at the start of the simulation.
Date_end	The last date at which concentrations may be saved. If the value - 1 is supplied, Date_end is set at the end of the simulation.

Interval_length	The number of steps between saves.
Type	see the below table for the list of available types
Output_file	The full path of output files, in which &f will be replaced by the name of the chemical species. Note that the directory in which the files are written must exist before the simulation is started.
Text_file	output files in a text type are provided for an easy and fast file checking

Note that Species, Date_beg, Date_end, Interval_length must appear before Type. After Type, put additional options relevant for the chosen output saver. All possible Type are listed:

* List of output saver Type

For gas-phase species
- street: concentration of gas-phase species
- street_dry_deposition_vel: dry deposition velocity

For aerosol species
- street_aer: concentration of aerosol species
- street_dep_mass_aer: aerosol mass deposition
- street_resusp_aer: aerosol resuspension rate
- street_washoff_aer: aerosol wash-off rate
- street_surf_dry_dep_aer: aerosol dry deposition rate

If multiple [save] sections are required when several output type are used.

5.2 Configuration for photochemistry

The program munich in the folder processing/photochemistry is configured with three configuration files (munich.cfg, munich-data.cfg and munich-saver.cfg) and data files including intersection.dat, street.dat and species-cb05.dat. The main configuration file munich.cfg provides the paths to all other configuration files and data files.

Many options are same to those of ssh-aerosol program. Here only differences between the programs are explained.

5.2.1 Main Configuration File: munich.cfg

[options]	
Option_chemistry	provides the list of available chemical kinetic mechanisms. CB05, Leighton, MELCHIOR2, RACM and RACM2
With_adaptive_time_step_for_gas_chemistry	With adaptive time stepping for gaseous chemistry?
Adaptive_time_step_tolerance	Tolerance for the adaptive time step.
Min_adaptive_time_step	Minimum for the adaptive time step.
Max_adaptive_time_step	Maximum for the adaptive time step.
With_photolysis	Should photolysis occur?
Photolysis_tabulation_option	Read only if With_tabulated_photolysis. If 1, the tabulation generated by SPACK is used. If 2, the binary files obtained by preprocessing tools (JProc or FastJ) are used.

5.2.2 Data Description File: `munich-data.cfg`

A section `[photolysis_rates]` may be required (if the chemical mechanism includes photolysis reactions). Depending on the chosing option for photolysis rates, different fields are read. If photolysis rates are tabulated, they depend on days, time angle, latitude and altitude. During the time integration, they are linearly interpolated in all cells. The following fields describe the tabulation parameters that must be filled.

Section	Entries	Comments
`[photolysis_rates]`	`Ndays`	Number of steps.
	`Time_angle_min`	Starting time angle in hours.
	`Delta_time_angle`	Time angle step in hours.
	`Ntime_angle`	Number of time angles.
	`Latitude_min`	First latitude in degrees.
	`Delta_latitude`	Step along latitude in degrees.
	`Nlatitude`	Number of latitude steps.
	`Altitudes`	List of altitudes in meters at which photolysis rates are provided.
	`Date_min`	Starting dates of photolysis rates.
	`Delta_t`	Time step (in days if tabulated photolysis rates).
	`Fields, Filename`	Photolysis reaction names and the paths to the files in which photolysis rates are stored.

5.3 Input data files

5.3.1 `intersection.dat` and `street.dat`

- `intersection.dat`: intersection id, longitude and latitude of the intersection, number of streets whice are connected to the intersection followed by the connected street id.

```
#id;lon;lat;number_of_streets;1st_street_id;2nd_street_id;...
1;2.49961040621;48.8639959388;6;1;605;852;3;8;11;
2;2.49977706824;48.8650938211;1;1;
6;2.49810836399;48.8642684425;1;3;
7;2.50328738802;48.8643530022;1;4;
8;2.50263219496;48.8630192684;4;4;5;10;19;
10;2.50111432732;48.8635255708;4;5;7;664;851;
11;2.48158743003;48.8656369541;1;6;
12;2.48236756109;48.8639495836;3;6;749;710;
13;2.50067262662;48.8626783551;4;7;8;23;676;
```

- `street.dat`: street id, two intersection id which connect the street, street length, width and averaged building height, street typologies. `typo` is set to 1 for highway and 0 for the others. It is used to compute the resuspension rate.

```
#id;begin_inter;end_inter;length;width;height;typo
1;1;2;122.686160495;7.5;6.9;0
3;1;6;107.94798805;7.5;6.9;0
4;7;8;155.856467313;7.5;6.9;0
5;8;10;124.490042998;7.5;6.9;0
```

```
6;11;12;196.113095869;41.0;10.2;0
7;13;10;99.5930099431;7.5;6.9;0
8;1;13;169.563212564;7.5;6.9;0
```

5.3.2 Input data files: `species-cb05.dat`

This file indicates information about the used chemical kinetic mechanism including the species list.

Please do not make a change in this `[species]` section. The order of the chemical species in this section is determined by SPACK during compiling. If you need to change chemical kinetic mechanism, e.g., add species and/or reactions, please see SPACK Guide (`https://www.cerea-lab.fr/dossiers/racine/articles/guide-0.pdf`).

When the program `ssh-aerosol` is used, the information in this file should match that in the folder `species-list/` of `SSH-aerosol` model.

Section	Comments
`[species]`	list of gas-phase species
`[aerosol_species]`	list of aerosol species
`[molecular_weight]`	molecular weight of gas-phase species in g/mol
`[molecular_weight_aer]`	molecular weight of aerosol speceis in g/mol
`[mass_density_aer]`	mass density in g/cm^3
`[gas_species_cloud_interact]`	list of gas species interacting with cloud phase, the order is imposed by VSRM.
`[diffusivity]`	gas-phase diffusivities: cm^2 / s.
`[henry]`	Henry constant: mol / L / atm. It must be effective Henry constant except HONO, HNO3, HCL, NH3 and SO2.
`[dissolution_heat]`	Heat of dissolution: kcal / mol at 298K.
`[reactivity]`	Chemical reactivity.
`[alpha]`	Alpha scaling factor for deposition.
`[beta]`	Beta scaling factor for deposition.
`[drainage_efficiency]`	Drainage efficency.
`[Rm]`	Mesophyll resistance.

```
[species]

HC8      NH3      HCL      O1D      ETOH
CH4      ETHA     TOL      XYL      SO2
SULF     HUM      BiBmP    BiBlP    POAlP
POAmP    POAhP    SOAlP    SOAmP    SOAhP
RppO2    RpppO2   BENZ     PHEN     CAT
ACIDMAL  CRESp    MCAT     DHMB     SYR
GUAI     NAPH     MNAPH    RelvocO2         Dimer
Monomer  BiA3D    TO2      N2O5             BiMGA
...


[aerosol_species]

# Water must be the last species.
PMD    PBC     PNA    PSO4    PNH4    PNO3
```

```
PHCL   PBiA2D  PBiA1D PBiAOD  PAGLY   PAMGLY
PBiMT  PBiPER  PBiDER PBiMGA  PAnBlP  PAnBmP  PBiBlP PBiBmP
PBiNGA PBiNIT3 PBiNIT PAnClP
PSOAlP PSOAmP  PSOAhP PPOAlP  PPOAmP  PPOAhP
PMonomer       PDimer PBiA3D  PACIDMAL  PDHMB
PPAHlN         PPAHhN PPSYR   PGHDPerox PH2O
...
```

For the program `photochemistry`, an additional section `photolysis_reaction_index` is necessary. It indicates the photolysis reactions numbers in the used chemical kinetic mechanism. The number begins from 0 which corresponds to the first reaction.

```
[photolysis_reaction_index]

NO2     0   O3O3P   7   O3O1D   8   NO3NO2 13
NO3NO   14  HONO    23  H2O2    34  HNO4   49
HNO3    50  N2O5    51  ORGNIT  60  HOP    62
MHP     69  HCHOrad 72  HCHOmol 73  ALD    84
PAN     88  PACD    94  C2CHO   99 PANX   103
OPEN    133 MGLY    138 ISPD  146
```

6 preprocessing

A python script `sing_preproc.py` can be used and run with a configuration file `sing_preproc.cfg`.

```
cd preprocessing
python sing_preproc.py sing_preproc.cfg
```

The sections `[input]`, `[option]`, `[output]`, `[domain]` and `[background]` are necessary.

[input]	
t_min	Initial date of the simulation.
Delta_t	Time step in hours.
Nt	Number of time step.
emission_dir_weekday	Directory where emission data for weekday are stored.
emission_dir_weekend	Directory where emission data for weekend are stored.
weekday_file_prefix	Prefix in file names for weekday emission data as the format of ¡prefix¿ + ¡hour in two digits¿.
weekend_file_prefix	Prefix in file names for weekend emission data as the format of ¡prefix¿ + ¡hour in two digits¿.
epsg_code	Map projection type used for emission data.
country_code	Country code to consider holidayss.
is_local_hour	Are emission data given as local hour.
time_zone	Time zone of local hour.
emission_species	Species for emission: `CH4` (methane), `NMHC` (non-methane hydro-carbon), `CO` (carbon monoxide), `NOx` (nitrogen oxide). For the species `NMHC`, emission data need to be calculated for model species. For this version, informations for CB05 chemical kinetic mechanism are included `aggregation_cb05-siream.dat`.

geog_info	File describing the geographic data.
background_concentration	File describing the background concentration.
meteo_dir	Directory where meteorological data are stored. Only WRF data can be used.
wrfout_prefix	Prefix in file names of WRF output.
speciation_dir	Input files for speciation.
	[option]
is_street_merged	Using these options is recommended only if traffic emission data are estimated for the vehicle flow for each traffic direction of a two-way street.
is_street_manually_merged	Streets are merged as indicated in street-merging.txt.
is_near_node_merged	Two nodes are merged when their distance is less than a given value.
is_node_manually_merged	Nodes are merged as indicated in intersection-merging.txt.
meca	cb05 or melchoir2 for speciation.
is_voc_speciated	If VOC is speciated.
is_nox_speciated	If NOx is speciated.
is_isvoc_speciated	If ISVOC is speciated.
is_pm10_speciated	If PM_{10} is speciated.
Nsize_sections	Number of aerosol size sections for PM_{10} speciation.
Size_dist_ec_om_emis	Redistribution factor for EC and OM.
Size_dist_dust_emis	Redistribution factor for dust.
om_redist	couvidat or may for OM redistribution.
	[output]
Output_dir	Directory where output data will be saved.

Sections [domain] and [background] are not mandatory for MUNICH simulation.

[domain] is used to generate emission data for grid cells of Polair3d simulation from traffic emission. These data are necessary for Street-in-Grid simulation.

	[domain]
Date_min_polair	Initial date of Polair3d simulation.
Delta_t_polair	Time resolution in hour.
Nt_polair	Number of time step.
x_min, Delta_x, Nx	Grid information in x.
y_min, Delta_y, Ny	Grid information in y.

[background] is used to generate background concentration from Chimere simulation results.

	[background]
chimere_bg	False if background comes from another source.
chimout_dir	Directory of Chimere output.
chimout_lab	
melch2molmass_file	Molar mass of species.
Species	List of species.

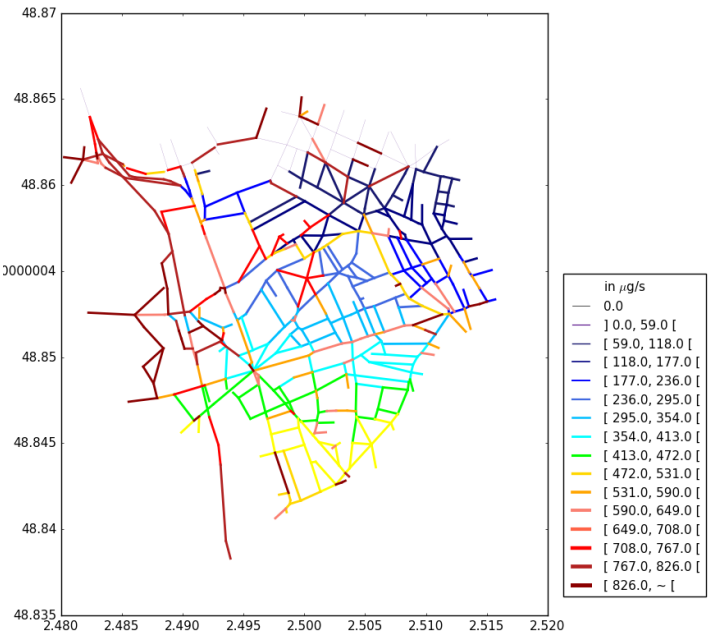


Figure 1: Emission map obtained by `display_emission.py`.

Please check that the unit of emissions input data is /*micro*g/km/h. The unit is converted to /*micro*g/s.

In `Output_dir`, the following data are generated. All files are binary except those in the directory `textfile`.

`background` - background concentration in the street canyons.

`emission` - traffic emission in the street canyons.

`meteo` - meteorological data in the street canyons and the street intersections.

`grid_emission` - Averaged emission data for a given grid cells (not necessary in MUNICH).

`textfile` - emission data and geographic data (`street.dat` and `intersection.dat` are used in `munich.cfg`).

A python script `display_emission.py` is provided to visualize emission data which are computed for a given domain.

```
cd preprocesing/utils
python display_emission.py
```

7 Postprocessing

`disp_concentrations.py` is provided to draw simulation results on the map.

```
cd postprocesing
python disp_concentrations.py disp_concentrations.cfg
```

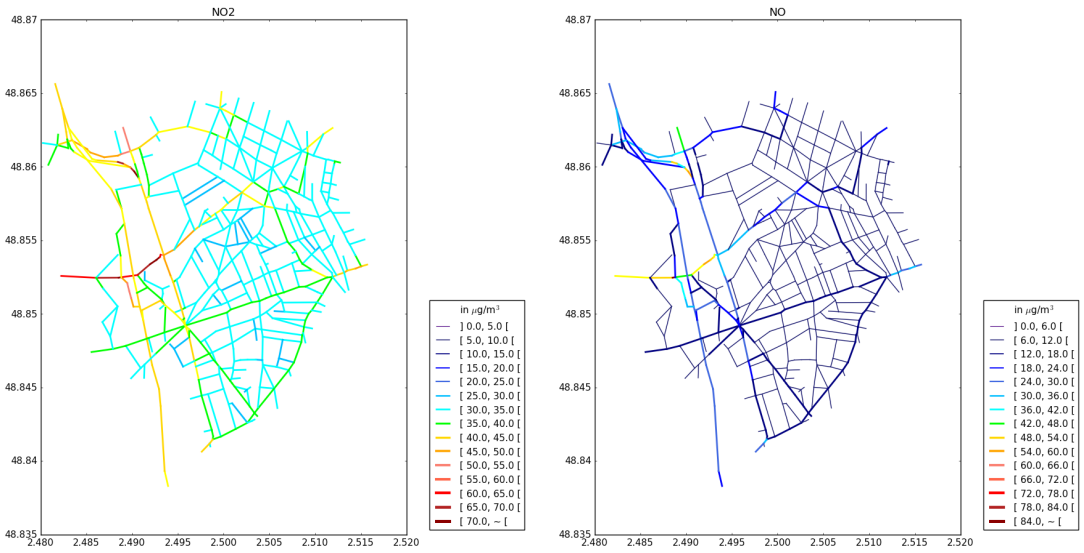


Figure 2: NO and NO_2 concentrations map obtained by `disp_concentration.py`.

`emission.txt` and `node.txt` are used. You need to copy them from `preprocessing/output/textfile` if they are not found in the directory.

8 Markups

In order to avoid duplications in a configuration file, Polyphemus features a markup management. A markup is denoted with surrounding '<' and '>', e.g. <path>. A markup is automatically replaced with its value whenever it is found. Its value should be provided somewhere in the configuration file with a proper field; for instance, <path> refers to the field `path`. Here is a complete example:

```
Root: /home/user
Input_directory: <Root>/input/
Output_directory: <Root>/output/
```

means:

```
Input_directory: /home/user/input/
Output_directory: /home/user/output/
```

The markup can be used before its value is defined:

```
Input_directory: <Root>/input/
Output_directory: <Root>/output/
Root: /home/user # After calls to <Root>. This is legal.
```

Any field may be used as a markup. The user may define any new markup (that is a new field). Moreover, several markup substitutions can be performed in a single value, and nested markups are properly handled:

```
Home: /home/user
Root: <Home>/Polyphemus/work
```

```
Number = 7
Input_directory: <Root>/input-<Number>/
```

is accepted and means:

```
Input_directory: /home/user/Polyphemus/work/input-7/
```

Notice that markups may also replace numbers and may be based on preexisting fields:

```
x_min = 12.5    Delta_x = 0.5    Nx = 100
y_min = <x_min> Delta_y = 1.     Ny = <Nx>
```

9 Dates

Date formats are:

```
YYYY                # Year.
YYYY-MM             # With the month.
YYYY-MM-DD          # With the day.
YYYY-MM-DD_HH       # With the hour.
YYYY-MM-DD_HH-II    # With the minute.
YYYY-MM-DD_HH-II-SS # With the second.
```

Months range from `01` to `12`. Days range from `01` to `31`. Hours range from `00` to `23`. Minutes and seconds range from `00` to `59`.

If the month is not specified (format `YYYY`), then the month is set to `01` (January). If the day is not specified (formats `YYYY` and `YYYY-MM`), it is set to `01` (first day of the month). If the hour, the minute or the second is not specified, it is set to zero (`00`).

Hyphens and underscores may be replaced with any character that is neither a delimiter nor a cipher. They can also be removed. Examples:

```
19960413
1996-04-13_20h30
1996/04/13@2030
```

Recommandation – Use hyphens around the month and around minutes. Use an underscore between the day and the hour (`YYYY-MM-DD_HH-II-SS`).

10 References

References

Gavidia-Calderón, M. E., Ibarra-Espinosa, S., Kim, Y., Zhang, Y., and Andrade, M. D. F. (2021). Simulation of o_3 and no_x in so paulo street urban canyons with vein (v0.2.2) and munich (v1.0). *Geoscientific Model Development*, 14(6):3251–3268.

Kim, Y., Sartelet, K., and Seigneur, C. (2009). Comparison of two gas-phase chemical kinetic mechanisms of ozone formation over Europe. *J. Atmos. Chem.*, 62(2):89–119.

Kim, Y., Wu, Y., Seigneur, C., and Roustan, Y. (2018). Multi-scale modeling of urban air pollution: development and application of a Street-in-Grid model (v1.0) by coupling MUNICH (v1.0) and Polair3D (v1.8.1). *Geosci. Model Dev.*, 11:611–629.

Lugon, L., Sartelet, K., Kim, Y., Vigneron, J., and Chrétien, O. (2020a). Nonstationary modeling of no_2, no and no_x in paris using the street-in-grid model: coupling local and regional scales with a two-way dynamic approach. *Atmospheric Chemistry and Physics*, 20(13):7717–7740.

Lugon, L., Sartelet, K., Kim, Y., Vigneron, J., and Chrtien, O. (2021). Simulation of primary and secondary particles in the streets of paris using munich. *Faraday Discuss.*, 226:432–456.

Lugon, L., Vigneron, J., Debert, C., Chrétien, O., and Sartelet, K. (2020b). Black carbon modelling in urban areas: investigating the influence of resuspension and non-exhaust emissions in streets using the street-in-grid (sing) model. *Geoscientific Model Development Discussions*, 2020:1–27.

Sartelet, K., Couvidat, F., Wang, Z., Flageul, C., and Kim, Y. (2020). Ssh-aerosol v1.1: A modular box model to simulate the evolution of primary and secondary aerosols. *Atmosphere*, 11(5).

Wu, L., Chang, M., Wang, X., Hang, J., Zhang, J., Wu, L., and Shao, M. (2020). Development of the real-time on-road emission (roe v1.0) model for street-scale air quality modeling based on dynamic traffic big data. *Geoscientific Model Development*, 13(1):23–40.