



MUNICH model test case: a Paris suburb Le Perreux-sur-Marne

Lya Lugon Cornejo Von Marttens and Youngseob Kim

Centre d'Enseignement et de Recherche en Environnement Atmosphérique

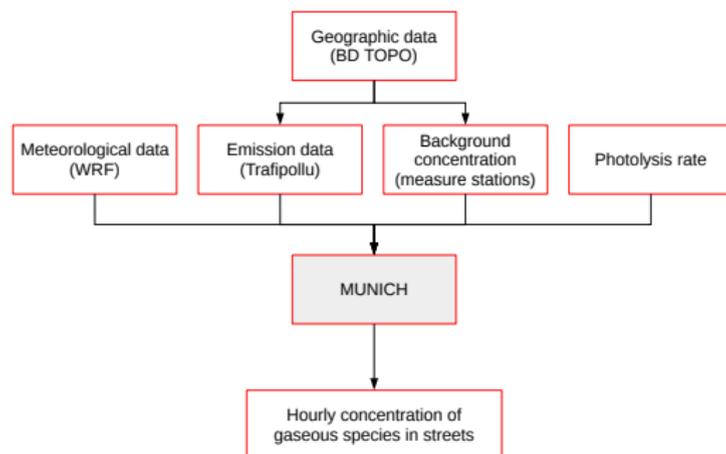
June 25, 2019

Plan

- 1 Introduction
- 2 Preprocessing
- 3 Processing
- 4 Post-Processing
- 5 Conclusions

Introduction

- Model of Urban Network of Intersecting Canyons and Highways (MUNICH)
- Street-canyon component
- Intersection component



Configuration applied

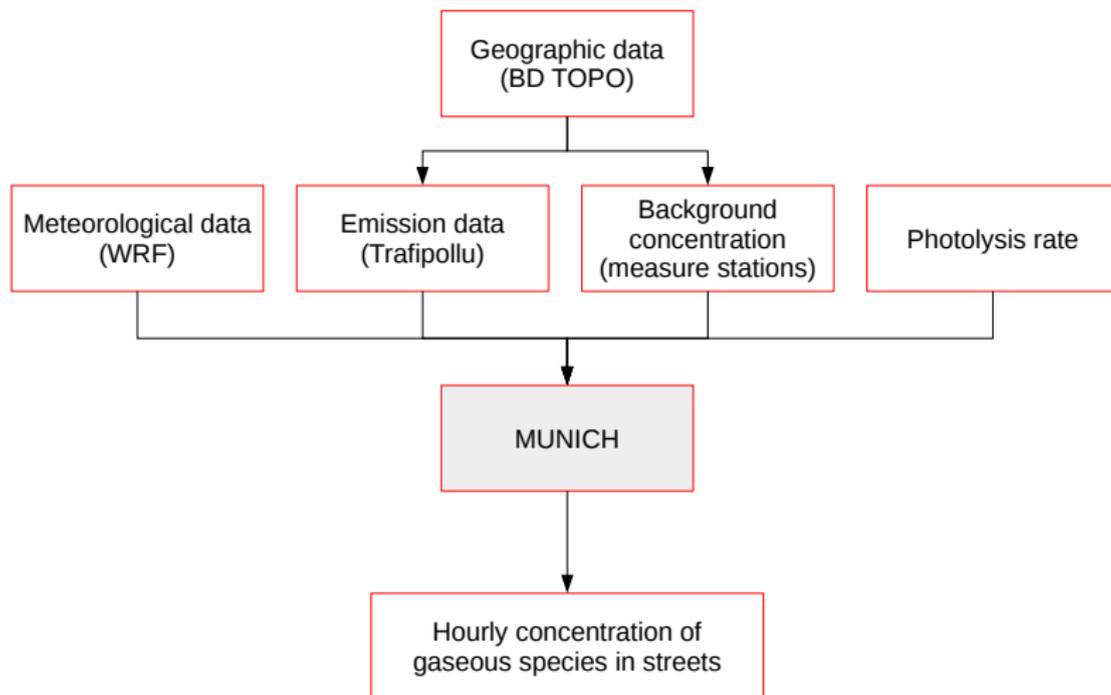
- Domain: Le Perreux-sur-Marne
- Simulated period: March 24th to March 30th
- Simulation of gas-phase species (including NO_x, and O₃)
- Chemical reactions: CB05 chemical kinetic mechanism (Yarwood et al., 2005)

Measured data - Trafipollu campaign

- ANR Project - Agence nationale de la recherche
- Multi-scale modeling of traffic pollution in an urban environment
- Main objective: Development of modeling tools to dynamically determine the location of pollutants generated by road traffic in an urban environment
- **Emission and concentration data at Boulevard Alsace-Lorraine**



- 1 Introduction
- 2 Preprocessing
- 3 Processing
- 4 Post-Processing
- 5 Conclusions



Geographical data

- Traffic lane widths and building heights → BD TOPO
- Street widths and building heights of the 15 major streets were explicitly estimated;
- For minor surfaces roads → width = 3m;
- Average street → width = 7.5m and building height = 6.9m (default);

IGN L'ESPACE PROFESSIONNEL

MON ESPACE »
MON PANIER »

GÉOSERVICES EXPERTISES DONNÉES SECTEURS D'ACTIVITÉS SUPPORT & CONTACTS DOSSIERS & ACTUALITÉS TÉMOIGNAGES

RECHERCHE

ACCUEIL / CATALOGUE

DONNÉES

Les cartes »
Les ortho-images »
Les bases de données en format vectoriel »

BD TOPO®

BD ADRESSES
ADRESSES PRESSION
BD FOUILLE
Lignes d'été

BD TOPO®

Le BD TOPO® est une description vectorielle 3D (incluant en objet) des éléments du territoire et de ses infrastructures, de précision métrique, exploitable à des échelles allant de 1 : 500 au 1 : 50 000.

Elle couvre de manière cohérente l'ensemble des entités géographiques et administratives du territoire national. Elle permet la visualisation, le positionnement, la simulation au service de

Caractéristiques
Gratuit pour... »
Couverture »

Géoservices

<http://professionnels.ign.fr/bdtopo>



Geographical data

- Street data are available in "*street.dat*" file;
- For each street we obtain:
 - Column 1: Street ID
 - Column 2: Initial intersection
 - Column 3: End intersection
 - Column 4: Street length
 - Column 5: Street width
 - Column 6: Street height

```
1;1;2;122.686160495;7.5;6.9
3;1;6;107.94798805;7.5;6.9
4;7;8;155.856467313;7.5;6.9
5;8;10;124.490042998;7.5;6.9
6;11;12;196.113095869;41.0;10.2
7;13;10;99.5930099431;7.5;6.9
8;1;13;169.563212564;7.5;6.9
9;17;18;156.567562811;7.5;6.9
10;17;8;145.452588007;7.5;6.9
11;1;22;198.442274833;7.5;6.9
12;22;24;189.386592243;7.5;6.9
14;27;28;129.283403254;7.5;6.9
15;29;27;28.243530365;7.5;6.9
17;33;34;88.3260073982;7.5;6.9
19;8;38;110.337649;7.5;6.9
```

Geographical data

- Intersection data are available in "*intersection.dat*" file;
- For each intersection we obtain:
 - Column 1: Intersection ID
 - Column 2: Longitude
 - Column 3: Latitude
 - Column 4: Number of streets that generate interception
 - Others columns: List of streets that generate interception



Meteorological data

- WRF (*Weather Research and Forecasting Model*) results were applied to calculate meteo data in each street and intersection;
- Grid resolution: $1.5 \times 1.5 \text{ km}^2$;
- Attenuation;
- Monin–Obukhov length;
- Planetary boundary layer height;
- Specific Humidity;
- Pressure;
- Temperature;
- Friction velocity;
- Wind direction;
- Wind speed;



Traffic model and emissions

- Hourly traffic emissions for NO, NO₂ and VOC
- COPERT4
 - Total vehicle number;
 - Hourly average speed for each street;
 - Optional fleet in circulation data: carburant, vehicles category (heavy or light), etc...
 - Four categories to calculate EF: NO_x and VOC's → Group 1
 - Pollutants for which a detailed methodology exists, based on specific emission factors and covering different traffic situations (i.e. urban, rural, highway) and engine conditions

Traffic model and emissions

- In Trafipollu project: HEAVEN model was applied based in COPERT4
- Two typical days were simulated: 25/03/2014 (working day) and 30/03/2014 (weekend day)
- Hourly emission data for several atmospheric pollutants, including: NO, NO₂, NMHC, CH₄, PM₁₀...
- For CB05 gas-phase chemical mechanism → NMHC speciation;



Loading Software. Please wait

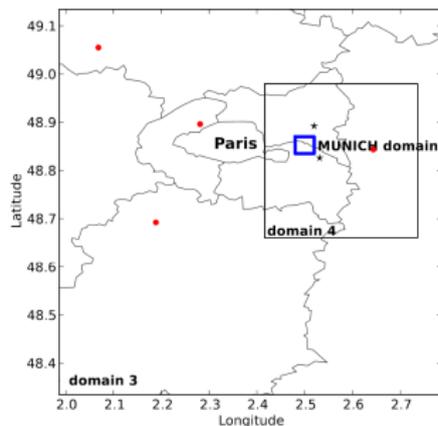


emisIA S.A. - COPERT4 2014



Boundary conditions

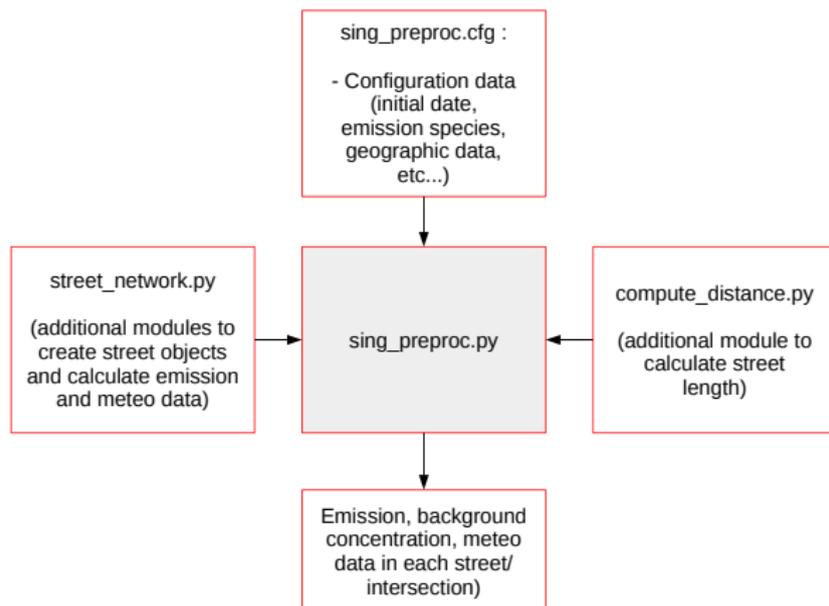
- Mean value \rightarrow Mass concentration measured in near air quality stations for O_3 , NO and NO_2 ;
- Distance of 5 to 7km from the area;



Photolysis Rate

- Photolytic reactions play a critical role in controlling the abundance of many atmospheric pollutants \rightarrow NO_2 ;
- FAST-J algorithm \rightarrow (Wild et al., 2000) - UC Irvine
 - Date min; Delta t, Ndays;
 - Time angle min, Delta time angle, Ntime angle;
 - Latitude min, Delta latitude, Nlatitude;
 - Altitudes;
 - Chemical species;

Preprocessing code



Preprocessing code - sing_preproc.py

- 1 Read current date and associate to emission file (weekend or weekday);
- 2 *read_traffic_data*: Reads total number of streets and emission of selected chemical species;
- 3 *get_street_geog*: Get street geographical informations: length, width, building height;
- 4 Merging streets:
 - Automatic merging → "street-merging-lookup-table.txt"
 - Manual merging → "street-merging.txt"
- 5 Merging nodes:
 - Removing the same nodes;
 - Removing the nearest nodes (Distance inferior to 10m);

Preprocessing code - sing_preproc.py

- ⑥ *get_meteo_data*:
 - Reads meteorological data for each street and intersection;
 - Compute wind direction
 - Compute heights;
 - Compute attenuation;
 - Compute LMO;
- ⑦ *get_background_concentration*: Measurements data for NO, NO₂ and O₃;
- ⑧ *write_output*: ASCII files → Intersection and street data;
- ⑨ Output meteo, emission and background files in binary format;

Preprocessing code - sing_preproc.cfg

```
[input]
t_min: 20140324_00
Delta_t: 1 # in_hr
Nt: 168 # for 7 days until 30/03/2014

emission_dir_weekday: input/data-newref/20140325-TrafiPollu-REF/ # Tuesday
emission_dir_weekend: input/data-newref/20140330-TrafiPollu-REF/ # Sunday
emission_species: CH4 NMHC CO NOx

input_dir: input/

geog_info: <input_dir>/street-geog-info.dat

background_concentration: <input_dir>/airparif_background_concentration.dat

meteo_dir: <input_dir>/meteo/
wrfout_prefix: wrfout_d04

[output]
Output_dir: output/
```

Preprocessing code

```
python sing_preproc.py sing_preproc.cfg
```

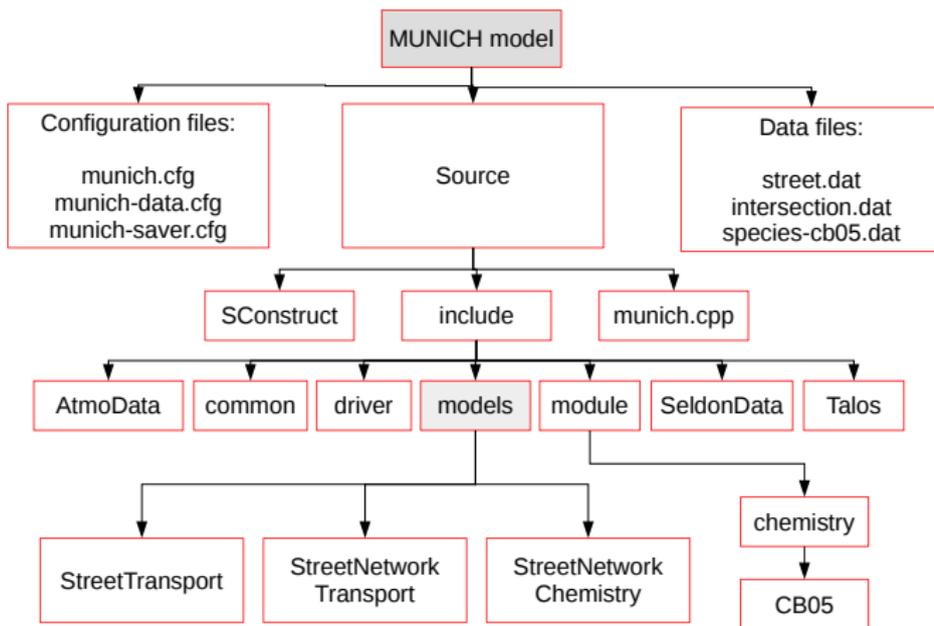
Preprocessing code - speciation_aggregation.py

Script used to organic speciation from NMHC emission data →
CB05 chemical kinetic mechanism ;

- 1 COVNM.dat
- 2 aggregation_cb05-siream.dat

- 1 Introduction
- 2 Preprocessing
- 3 Processing**
- 4 Post-Processing
- 5 Conclusions

General model view



General model view - munich.cpp

```
|
| ///////////////
| // INCLUDES //
|
| #define SELDONDATA_DEBUG_LEVEL_4
|
| #include "AtmoData.hxx"
| #include "StreetDriver.cxx"
| #include "BaseOutputSaver.cxx"
| #include "StreetNetworkChemistry.cxx"
| #include "PhotochemistrySin6.cxx"
|
| using namespace Polyphemus;
|
| // INCLUDES //
| ///////////////
|
| int main(int argc, char** argv)
| {
|     TRY;
|
|     if (argc != 2)
|     {
|         string mesg = "Usage:\n";
|         mesg += string(" ") + argv[0] + " [configuration file]";
|         cout << mesg << endl;
|         return 1;
|     }
|
|     typedef double real;
|     typedef StreetNetworkChemistry<real, Photochemistry<real> > ClassModel;
|
|     StreetDriver<real, ClassModel, BaseOutputSaver<real, ClassModel> >
|     Driver(argv[1]);
|
|     Driver.Run();
|
|     END;
|
|     return 0;
| }
|
```

General model view - StreetDriver.cxx

```

    //! Performs the simulation.
    //! Initializes the model and the output saver, and then performs the loop
    //! over all meteorological conditions with calls to the model (over all time
    //! steps) and to the output saver.
    */
    template<class T, class ClassModel, class ClassOutputSaver>
    void StreetDriver<T, ClassModel, ClassOutputSaver>::Run()
    {
        string line;

        /*** Initializations ***/

        Model.ReadConfiguration();
        Model.Init();
        Model.InitData();
        OutputSaver.Init(Model);
        Model.InitOutputSaver();

        for (int i = 0; i < Model.GetNt(); i++)
        {
            if (option_display["show_iterations"])
                cout << "Performing iteration #" << i << endl;

            if (option_display["show_date"])
                cout << "Current date: "
                    << Model.GetCurrentDate().GetDate("%y-%m-%d %h:%i") << endl;

            Model.InitStep();
            OutputSaver.InitStep(Model);

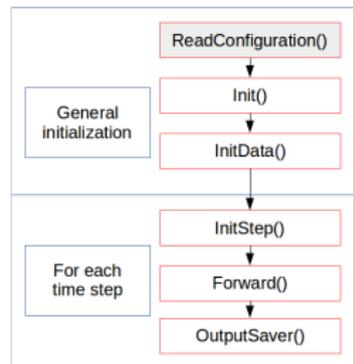
            Model.Forward();

            OutputSaver.Save(Model);
            Model.OutputSaver();
        }
    }
} // namespace Polyphemus.

```

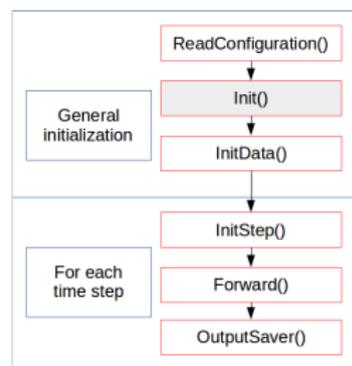
ReadConfiguration()

- Initial date;
- Simulation time;
- Species list → CB05: 56 gaseous species;
- Transfert parameterization: "Sirane" or "Schulte";
- Mean wind speed parameterization: "Sirane or Lemonsu";
- With horizontal fluctuation;
- Minimum street wind speed → 0.1m/s;
- Inlet files: meteo, emission, background;
- Chemistry and photolysis option;



Init()

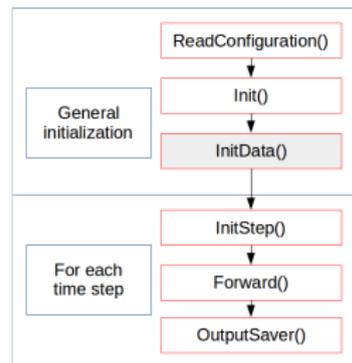
- Create street objects;
- Create intersection objects;
- Compute the street angle which is defined as the angle between the street intersections (begin intersection to end intersection);
- Init chemistry module;



InitData()

- Input data initialization:

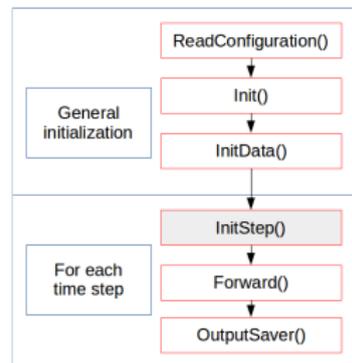
- Read meteo files;
- Read atmospheric emissions for each emitted specie;
- Read concentrations for each background specie;



InitStep()

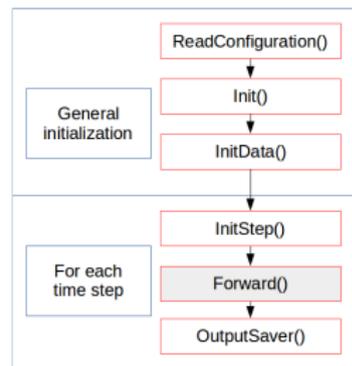
- Method called at each time step to initialize the model:

- Read meteo data for each street and intersections;
- Read emission data in each street;
- Read background data in each street;



Forward()

- Transport section:
 - Compute street mean velocity;
 - Compute the standard deviation of the vertical wind speed at a roof;
 - Compute the transfer velocity at roof level;
 - Compute the horizontal fluctuation of the wind direction;
 - Compute inflow and outflow in each street;



Forward()

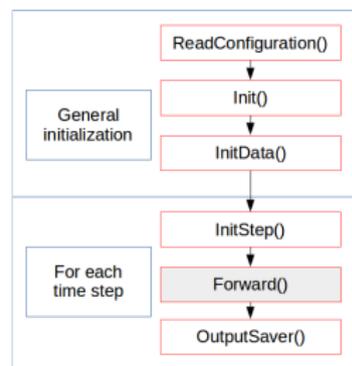
- Transport section:
 - Compute street concentration → Stationary state assumption;

$$Q_{emis} + Q_{inflow} + Q_{chem} = Q_{vert} + Q_{outflow} + Q_{dep}$$

$$C = \frac{Q_{emis} + Q_{inflow} + (transfer_{velocity} \times W \times L) \times C_{background}}{Q_{outflow} + transfer_{velocity} \times W \times L}$$

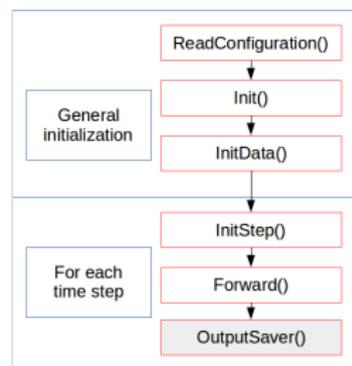
Forward()

- Chemistry section:
 - Calls the chemistry module to calculate the gain or loss of each chemical specie;
 - CB05 gas-phase chemical mechanism (Yarwood et al., 2005)
 - 52 species
 - 155 reactions (23 photolysis);



OutputSaver()

- Option of write the results in the output files in the text format (O_3 , NO and NO_2);
- Street's hourly concentration in binary files for each one of the 52 species;



Configuration files - munich.cfg

```
[display]

Show_iterations: yes
Show_date: yes

[domain]

# Domain where species concentrations are computed.
Date_min = 2014-03-24-00      Delta_t = 3600.0  Nt = 168

# File containing the species data.
Species: species-cb05.dat

[options]

# With chemistry module (yes or no)
With_chemistry: yes

# Which chemistry model (RACM, RACM2 or CB05, Leighton)?
Option_chemistry: CB05

# With adaptive time stepping for gaseous chemistry?
With_adaptive_time_step_for_gas_chemistry: yes
Adaptive_time_step_tolerance: 0.001
# Minimum time step that can be used.
Min_adaptive_time_step: 10.0
# Maximum time step that can be used.
Max_adaptive_time_step: 600.0

With_photolysis: yes

# Photolysis tabulation option compute from tabulations (1) or read from binary
# files (2).
Photolysis_option: 2
```

Configuration files - munich.cfg

```
[street]

# Parameterization to compute turbulent transfert velocity: "Sirane" or "Schulte".
Transfert_parameterization: Schulte

# Parameterization to compute mean wind speed within the street-canyon: "Sirane" or "Lemonsu".
Mean_wind_speed_parameterization: Lemonsu

# If the horizontal fluctuation is taken into account.
With_horizontal_fluctuation: yes # yes or no

# File containing the input data for intersections.
Intersection: intersection.dat

# File containing the input data for streets.
Street: street.dat

# Minimum wind speed within the streets.
Minimum_Street_Wind_Speed: 0.1 # in m/s

# If meteo data and background concentrations are available for each street.
With_local_data: yes

[data]

# File describing the input data.
Data_description: munich-data.cfg

[output]

# File describing which concentrations are saved.
Configuration_file: munich-saver.cfg
```

Configuration files - munich-data.cfg

```
Directory: /net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/data-munich/

[emission]
Date_min: 20140316_00
Delta_t = 3600.
Nt = 2568

Fields: ALD2 ALDX API CH4 CO ETH ETHA ETOH FORM IOLE ISOP LIM MEOH OLE PAR TERP TOL XYL NO NO2
Filename: <Directory>/emission/&f.bin

[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
Filename: <Directory>/meteo/&f.bin

[background_concentration]
Date_min: 20140316_00
Delta_t = 3600.
Nt = 2568

Fields: O3 NO NO2
Filename: /net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/data-munich-background/background/&f.bin
```

Configuration files - munich-data.cfg

```
[photolysis_rates]
Date_min: 2014-01-01_12
# In days:
Delta_t = 1
Ndays = 365
# In hours:
Time_angle_min = 0.
# In hours:
Delta_time_angle: 1.
Ntime_angle = 9
# In degrees:
Latitude_min = 0.
# In degrees:
Delta_latitude = 10.
Nlatitude = 10
# In meters:
Altitudes: 0. 1000. 2000. 3000. 4000. 5000. 10000. 15000. 20000.
Fields: NO2 O3O3P O3O1D NO3NO2 NO3NO HONO H2O2 HN04 HN03 N2O5 ORGNIT HOP MHP HCHOrad HCHOmol ALD PAN C2CHO MGly OPEN ISPD PANX PACD
Filename: /net/libre/yomi/kimy/StreetInGrid/Trafipollu/data_photolysis/&f.bin
```

Configuration files - munich-saver.cfg

```
[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: 1 # 1 for all steps.
Averaged: no
# Save initial concentrations in case concentrations are not averaged?
Initial_concentration: no

# Choices: street
Type: street

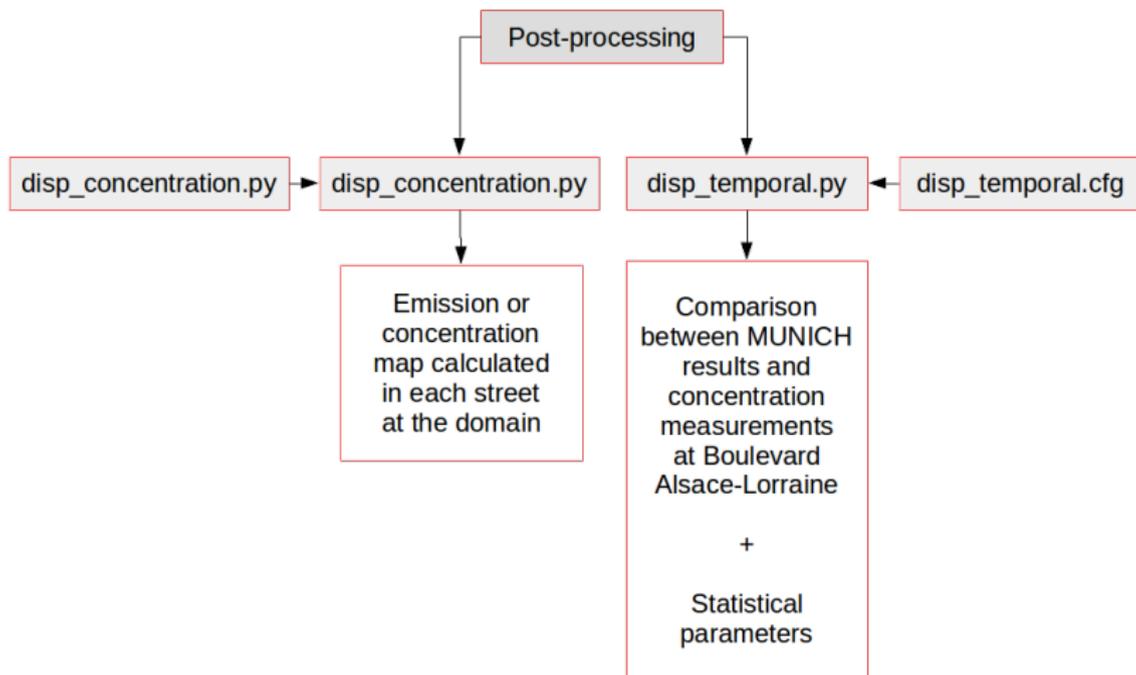
Levels: 0

Output_dir: /net/libre/halong/kimy/munich/test-case/results/
Output_file: <Output_dir>/&f.bin
Text_file: yes # yes
```

Processing...

- Compilation: compile
- munich munich.cfg

- 1 Introduction
- 2 Preprocessing
- 3 Processing
- 4 Post-Processing**
- 5 Conclusions



Post-processing: `disp_concentration.py`

This script calculates the hot emissions of gasoline passenger car for each arc during after-work rush hours.

- 1 Pre-setting figures for articles → size;
- 2 Display streets map → "node-trafipollu-eff.txt"
- 3 Input visualisation data → "emission-trafipollu-eff.txt"
- 4 If we analyze NO_x data:

$$emis_{NO_x} = emis_{NO_2} + \frac{46}{30}emis_{NO}$$

- 5 Plot data

Post-processing: disp_concentration.cfg

```
[input]
Species: N02

Directory: /net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/munich/reference/results/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/munich/noxcorr/results/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/results_sing/street/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/sing/test/results/street/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/results_sing/street/

Nx = 1
Ny = 1

# Be careful with Nz and Nt, they can differ from the values
# given in general.cfg.

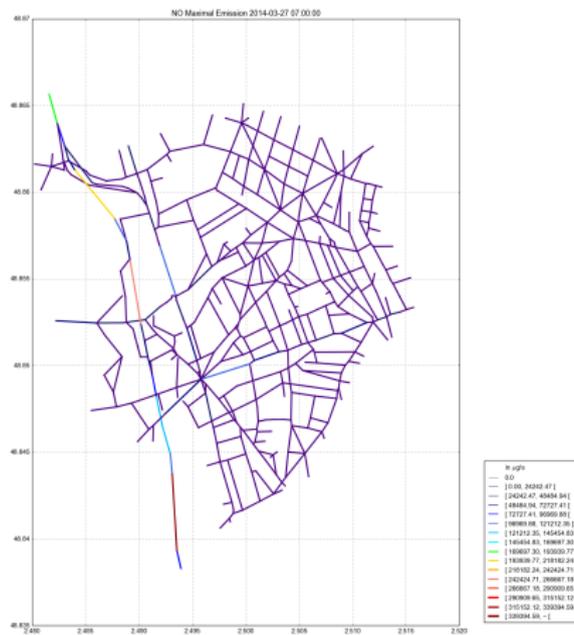
Nz = 577
Nt = 2016 # 336 # for all time step
```

Post-processing: disp_concentration.cfg

```
python disp_concentration.py disp_concentration.cfg
```

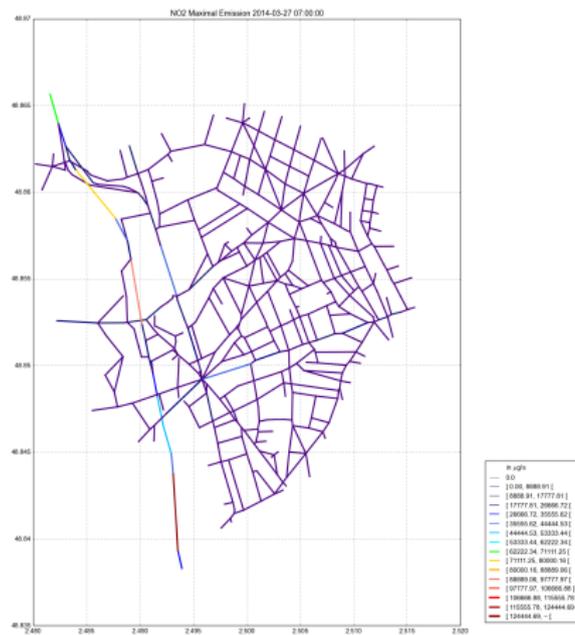
Post-processing: NO emissions

March 27th - 7h00 am: Max emission = 363,637.06 $\mu\text{g}/\text{s}$



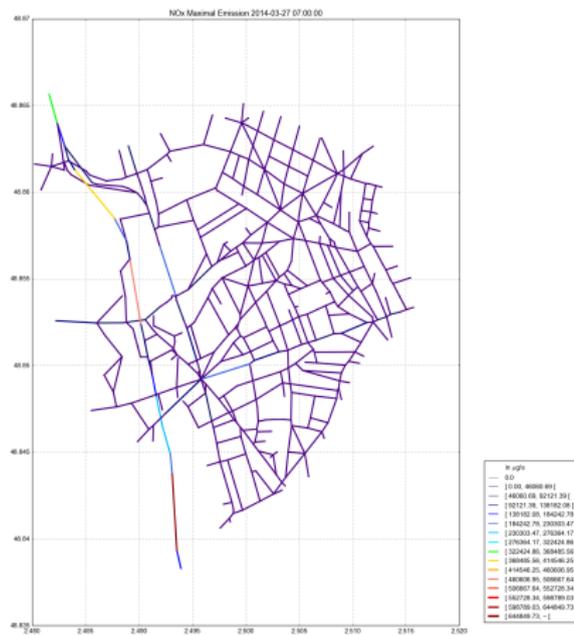
Post-processing: NO₂ emissions

March 27th - 7h00 am: Max emission = 133,333.59 $\mu\text{g}/\text{s}$



Post-processing: NO_x emissions

March 27th - 7h00 am: Max emission = 690,910.42 $\mu\text{g}/\text{s}$



Post-processing: disp_concentration.cfg

```
[input]
Species: NO2

Directory: /net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/munich/reference/results/
#net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/munich/noxcorr/results/
#net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/results_sing/street/
#net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/sing/test/results/street/
#net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/results_sing/street/

Nx = 1
Ny = 1

# Be careful with Nz and Nt, they can differ from the values
# given in general.cfg.

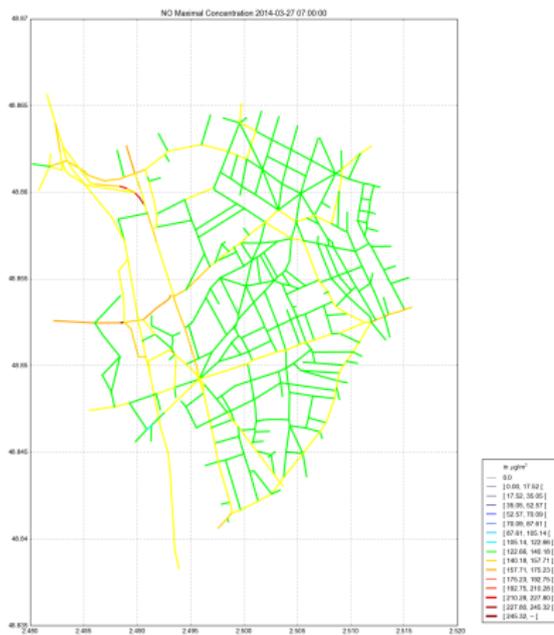
Nz = 577
Nt = 2016 # 336 # for all time step
```

Post-processing: disp_concentration.cfg

```
python disp_concentration.py disp_concentration.cfg
```

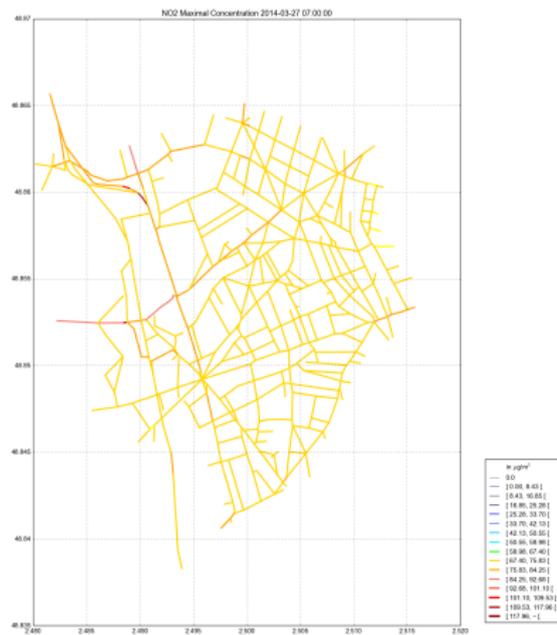
Post-processing: NO concentration

March 27th - 7h00 am: Max concentration = 262.84 $\mu\text{g}/\text{m}^3$



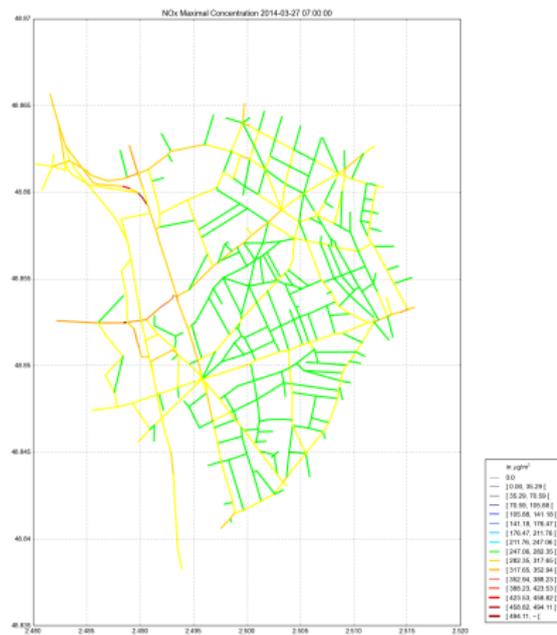
Post-processing: NO₂ concentration

March 27th - 7h00 am: Max concentration = 126.38 $\mu\text{g}/\text{m}^3$



Post-processing: NO_x concentration

March 27th - 7h00 am: Max concentration = $529.40 \mu\text{g}/\text{m}^3$



Post-processing: `disp_temporal.py`

This script calculates the hot emissions of gasoline passenger car for each arc during after-work rush hours.

- 1 Reads pollutant concentration in each street in Boulevard Alsace Lorraine;
- 2 Calculates the average concentration;
- 3 Reads measurement data;
- 4 Plot simulated and measured concentrations
- 5 Calculates statistical parameters → `statistics.py`

Post-processing: disp_temporal.cfg

```
[input]
Species: N02

Directory: /net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/munich/reference/results/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing-voc/munich/noxcorr/results/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/results_sing/street/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/sing/test/results/street/
#/net/libre/yomi/kimy/StreetInGrid/Trafipollu/sing/leperreux/results_sing/street/

Nx = 1
Ny = 1

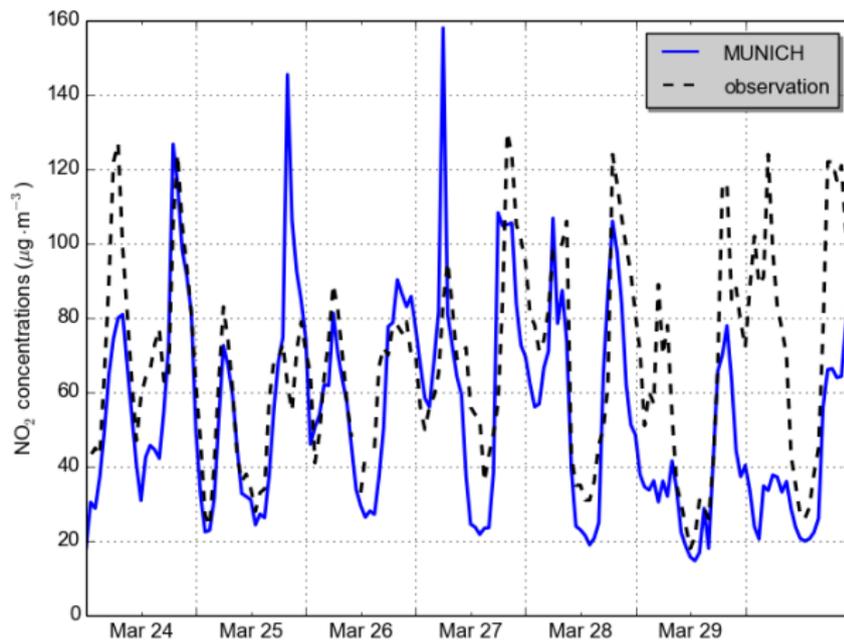
# Be careful with Nz and Nt, they can differ from the values
# given in general.cfg.

Nz = 577
Nt = 2016 # 336 # for all time step
```

Post-processing: disp_temporal.cfg

```
python disp_temporal.py disp_temporal.cfg
```

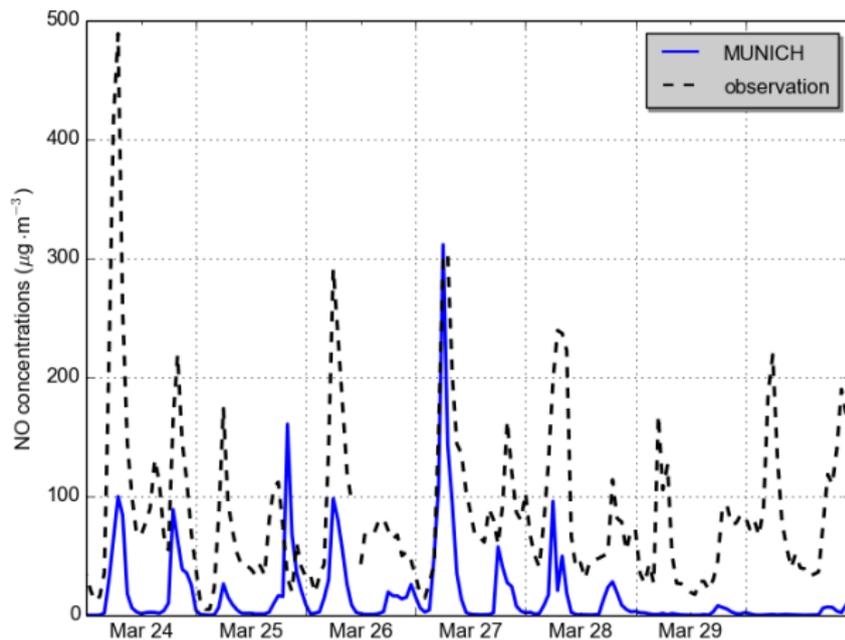
NO₂ concentrations



NO₂ concentrations - Statistical indicators

- **sim mean: 54.53**
- **obs mean: 68.33**
- RMSE: 26.93
- MFB: -0.26
- MFE: 0.33
- NMB: -0.20
- NME: 0.29
- MNGE: 0.28
- MNB: -0.18
- FB: -0.22
- VG: 1.22
- MG: 0.76
- FAC2: 0.90
- **correlation: 0.64**

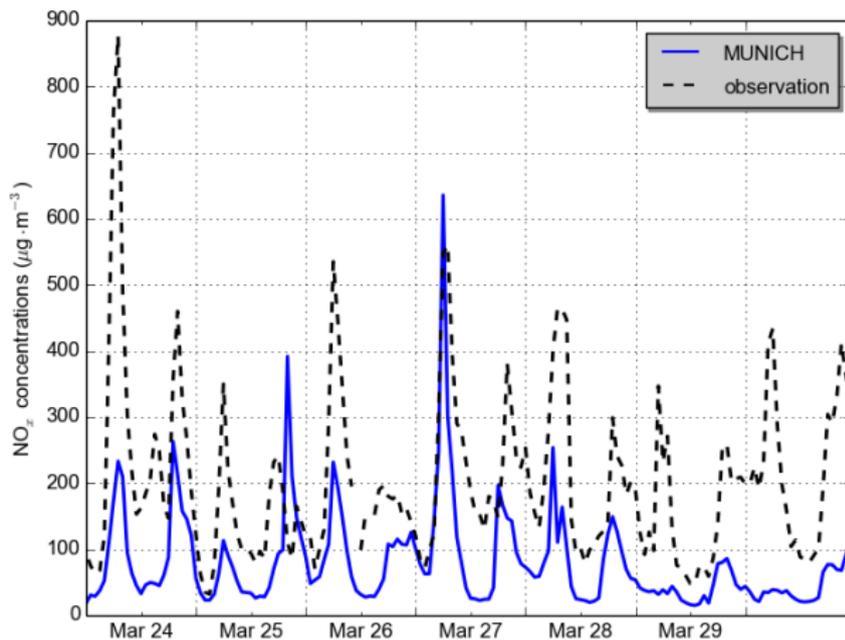
NO concentrations



NO concentrations - Statistical indicators

- **sim mean: 16.91**
- **obs mean: 88.75**
- RMSE: 93.98
- MFB: -1.57
- MFE: 1.60
- NMB: -0.81
- NME: 0.83
- MNGE: 0.89
- MNB: -0.81
- FB: -1.35
- VG: 46907.78
- MG: 0.05
- FAC2: 0.04
- **correlation: 0.58**

NO_x concentrations



NO_x concentrations - Statistical indicators

- **sim mean: 80.47**
- **obs mean: 204.44**
- RMSE: 164.06
- MFB: -0.91
- MFE: 0.94
- NMB: -0.60
- NME: 0.64
- MNGE: 0.63
- MNB: -0.57
- FB: -0.87
- VG: 4.23
- MG: 0.35
- FAC2: 0.22
- **correlation: 0.58**

- 1 Introduction
- 2 Preprocessing
- 3 Processing
- 4 Post-Processing
- 5 Conclusions**

Conclusions

- A street-network model MUNICH was applied to simulate NO_x concentrations in a suburb Paris area;
- The simulated period was 24/03/2014 to 30/03/2014;
- All steps of preprocessing, processing and post-processing were presented and applied;
- We observed a reasonable NO_2 estimation (mean negative bias of 20.2%);
- We have a considerable underestimation of NO and consequently NO_3 prediction;
- Great importance of background data;

Thank you for your attention !!