Accelerating the production of noise maps via metamodeling : Application to the Rhine Avenue in Strasbourg

Internship

Carl SCHMITTBIEL, Gwenaël GUILLAUME, Barnabé MIESCH

2025-08-27

Outline

Basic Acoustics

Context

The measure of sound

Sound levels

- Atmospheric pressure ⟶ Pascals (Pa)
- Acoustic pressure ⟶ micropascals (µPa)
- Acoustic pressure difference << Atmospheric pressure
- To better match human perception ⟶ dB / A-weighted dB(A)

Decibels

- 0 dB(A) ⟶ hearing threshold (reference level)

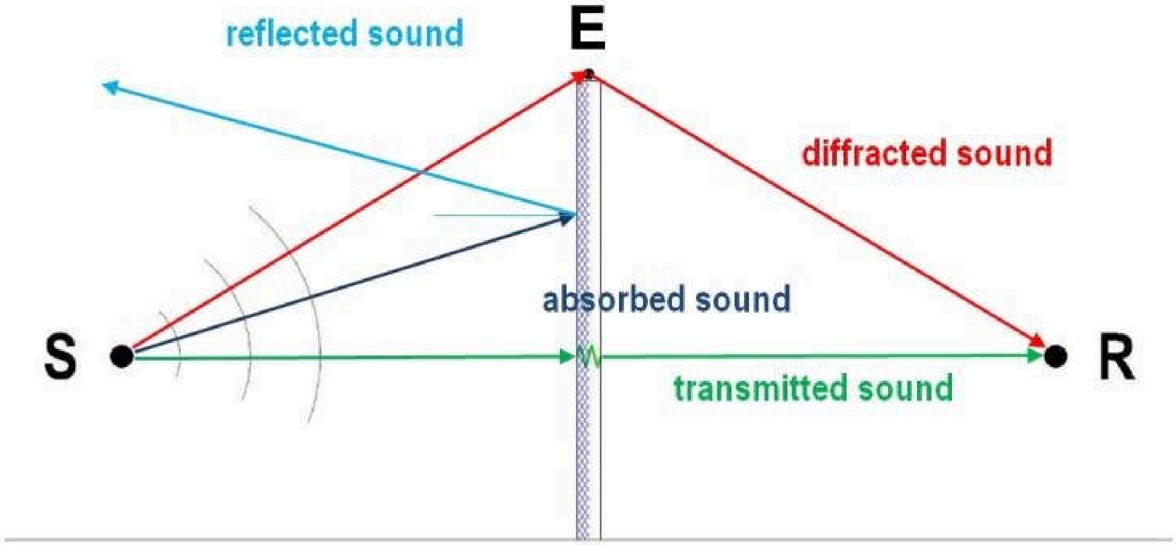
↳ 2e-5 Pa

- 60 dB(A) ⟶ normal conversation
- 80 dB(A) ⟶ busy traffic
- 120 dB(A) ⟶ pain threshold

- +10 dB ⟶ perceived as twice as loud

Noise propagation

- Sound waves reflect, diffract, refract
- Affected by terrain, buildings, vegetation, weather



Noise sources

- Road traffic
- Rail traffic
- Air traffic
- Industrial activities

Health impacts

- Hearing loss
- Sleep disturbance
- Cardiovascular issues
- Cognitive impairment in children
- Mental health effects
- Ecosystem disruption

WHO guidelines: < 53 dB(A) daytime, < 45 dB(A) nighttime

Standard EU method for noise mapping :
CNOSSOS-EU : Common noise assessment methods in Europe

Purpose

- Visualize spatial distribution of noise levels
- Inform urban planning

Problem

- Data collection campaigns are rare
- Models are computationally intensive

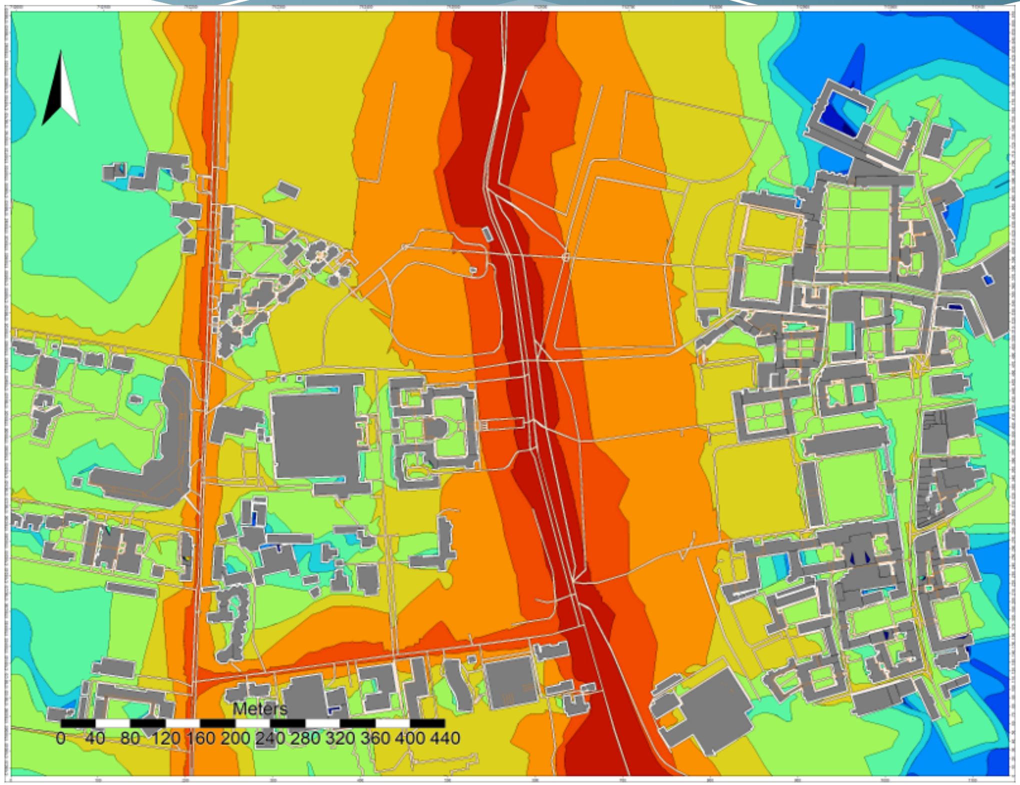


Figure 2: Example of a standard noise map

The Rhine avenue

Eurométropole de Strasbourg

Strasbourg.eu
eurométropole

Challenges

- Growing population & urbanization
- Increasing traffic & noise pollution
- Need for efficient noise monitoring & mapping

The Rhine avenue

- Location & traffic (40k vehicles/day)
- Measured >90 dB(A) near avenue daily

EMS

- 500,000 inhabitants
- 33 communes
- Diverse urban and natural environments

Figure 3: The Rhine avenue in Strasbourg (source: OpenStreetMap)

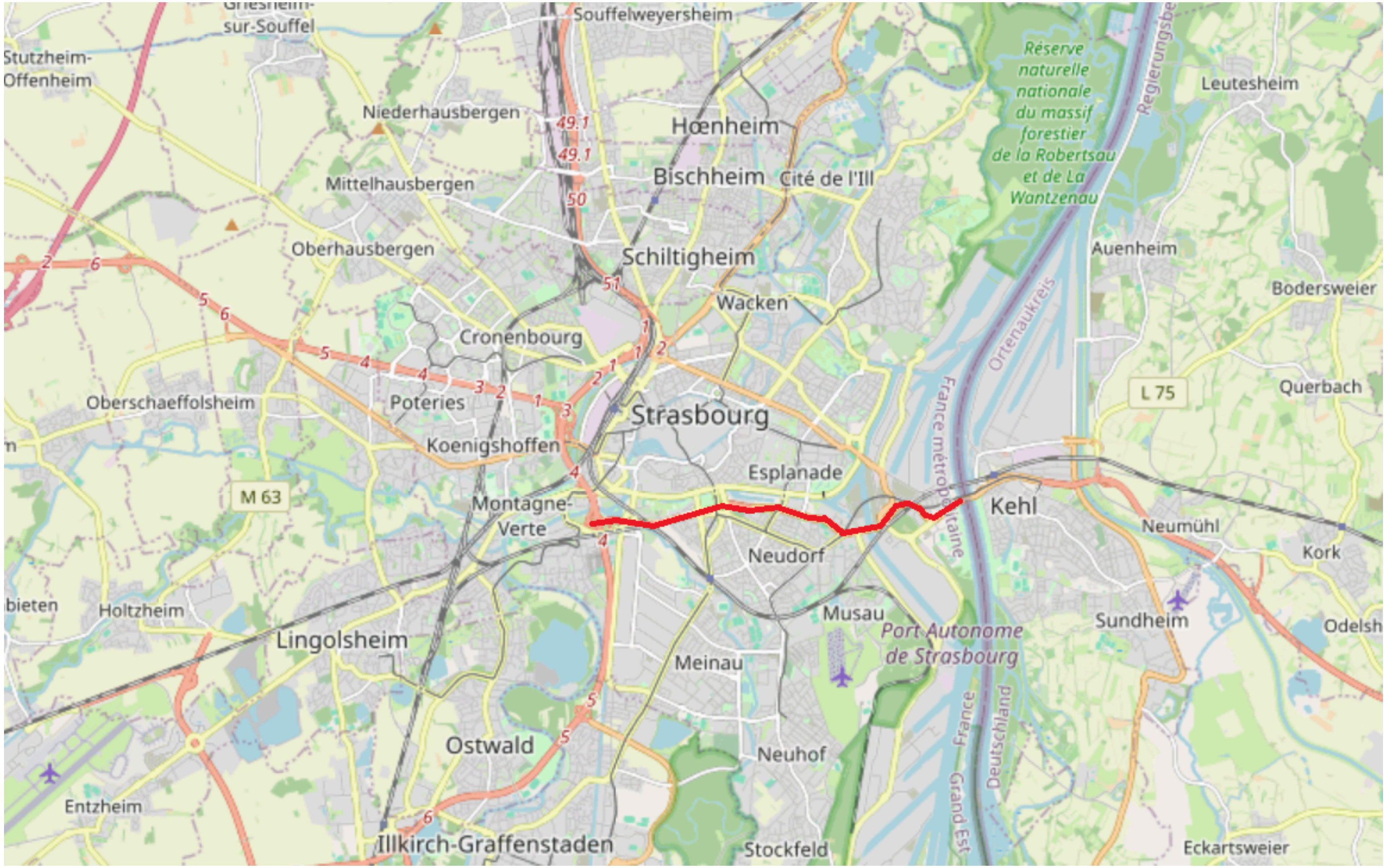


Figure 4: Rhine avenue location in EMS (source: OpenStreetMap)

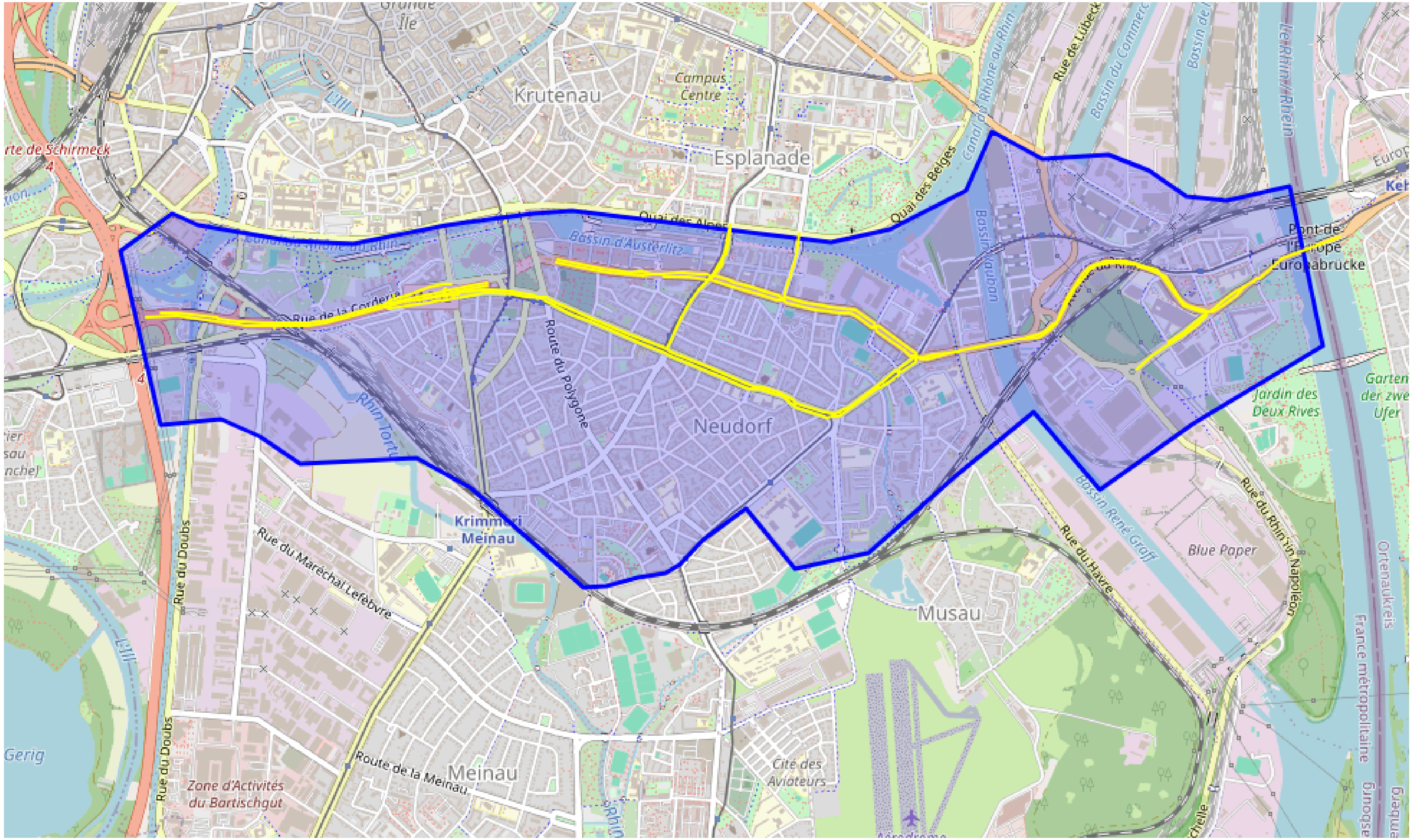
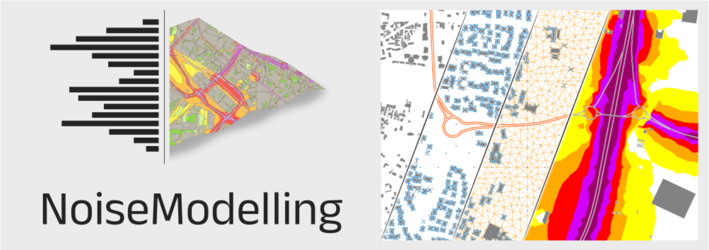


Figure 5: Selected study area around the Rhine avenue (source: OpenStreetMap)

Noise Model

NoiseModelling



What is it?

- Software to compute noise levels in outdoor environments
  - Use of sound rays propagation
- Based on standard CNOSSOS-EU sound level calculation method
- Accounts for terrain, buildings, weather, traffic density

Applications

- Urban planning
- Environmental impact assessments
- Noise mapping for large areas

NoiseModelling

Inputs

- Terrain Elevation Map
- Roads (OSM)
- Buildings (OSM)
- Traffic data (% of heavyweights, speed limits)

- Maximum distance of noise propagation: 800m
- Grid resolution: 4m
- Number od reflections to simulate: 2, 3
- Diffraction on edges
- Atmospheric conditions (temperature, humidity, wind)

processing steps

- Data preprocessing,
- 3D mapping of environment,
- 2D Delaunay triangulation ot generate mesh of receivers,
- Backwards ray tracing of sound rays
- Sound level calculation at receiver points,
- Post-processing and visualization
  - Heatmaps, contour maps

⟶ Total computation time: **1h30** on a supercomputer for a $8km^2$ area

Figure 6: Sound propagation via ray tracing (source: NoiseModelling docs)

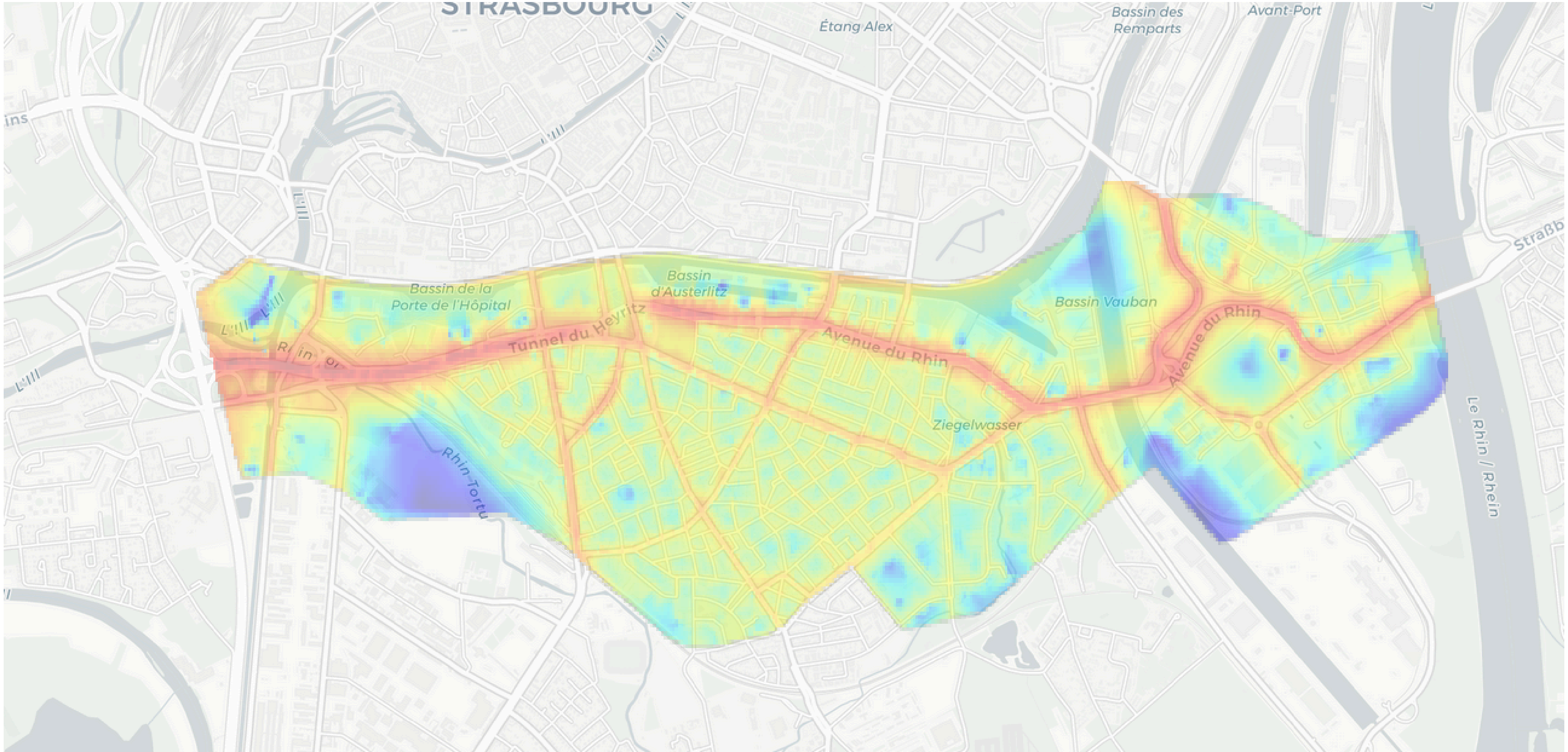


Figure 7: Heatmap of the sound levels computed by NoiseModelling

Metamodeling

Why metamodeling?

- NoiseModelling is accurate but slow
- EMS $\longrightarrow$ need for tools that can evaluate many scenarios quickly

Ideally:

user changes one parameter $\longrightarrow$ new noise map within under a second

Workflow:

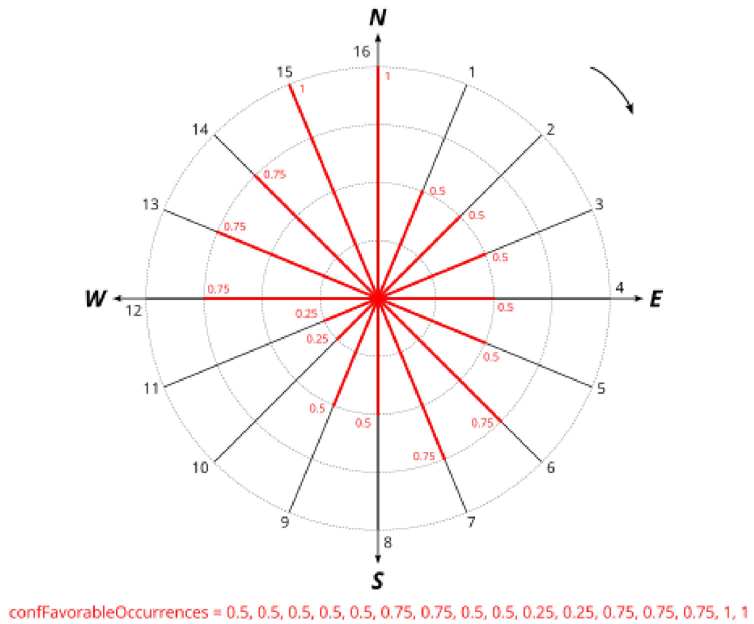
1. Train on a small amount of executions from NoiseModelling
2. Use Primary component analysis (PCA) to reduce the output space size
3. Use gausian processes (Kriging) to estimate a transform from inputs to reduced output space
4. Estimate maps
5. Analyse results

Input parameters

Parameters

- Humidity $H \in [60\%, 100\%]$
- Temperature $T \in [-10°C, 35°C]$
- Wind speed $W \in [0 \text{ m/s}, 14 \text{ m/s}]$
- Wind direction $D \in [0°, 360°]$

Constitute our inupt space $\mathcal{I} = (H, T, W, D) \in \mathbb{R}^4$



LHS Sampling

- Sample is uniformly
  distributed across the input space
- More efficient than simple random sampling
- Each interval of every input dimension is sampled exactly once

Definition

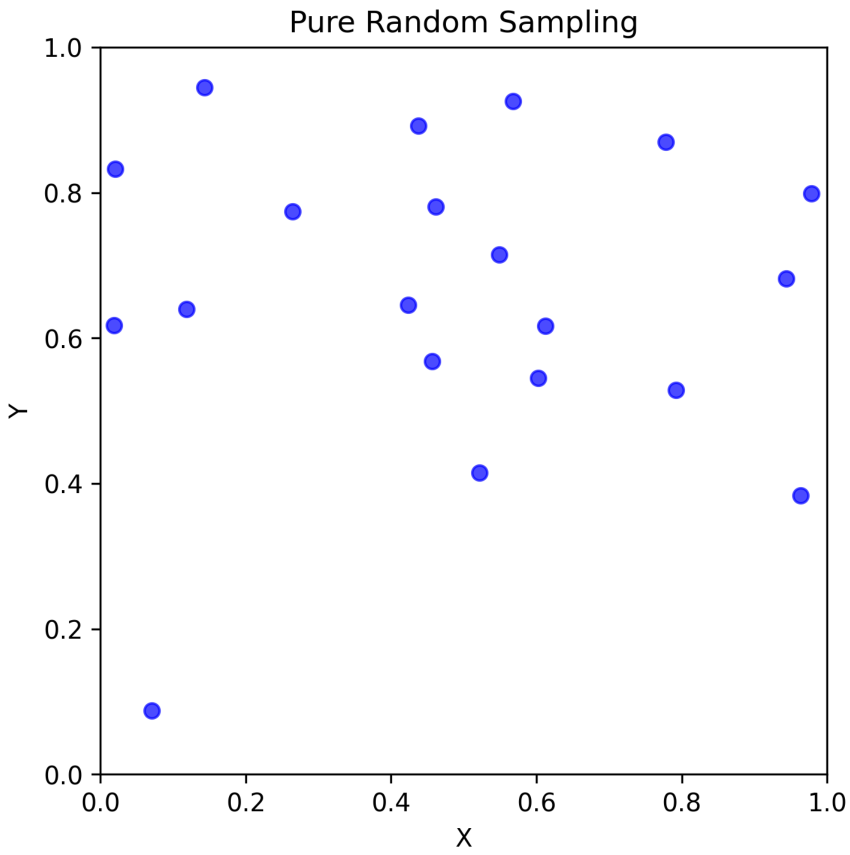
For $i = 1, ..., N$ samples and $j = 1, ..., d$ dimensions:

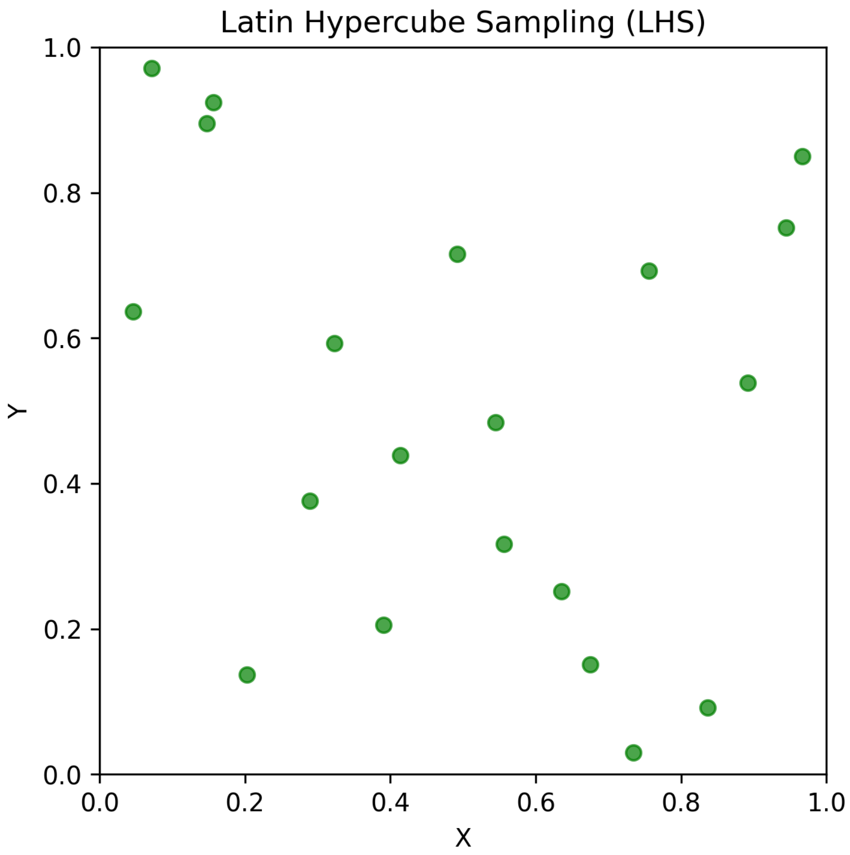
$$x_{ij} = \mathcal{U}\left(a + (j-1)\left(\frac{b-a}{N}\right), a + j\left(\frac{b-a}{N}\right)\right)$$

where:

- $\mathcal{U}(a, b)$ = uniform random variable in interval $[a, b]$
- N = number of samples
- d = number of input dimensions
- a, b = bounds of input domain

Comparison of Sampling Techniques





Problem

- Outputs: $y_{i \in [0,N]} \in \mathbb{R}^d$, with $d \approx 26000$ receivers
- Direct learning in this space is intractable

PCA: Principal Component Analysis

Center data:

$$Y = Y_c - \bar{Y}, \quad Y \in \mathbb{R}^{N \times d}$$

Covariance:

$$\Sigma = \frac{1}{N} Y^T \cdot Y \in \mathbb{R}^{d \times d}$$

Dimensionality reduction

Eigen decomposition:

$$\Sigma\nu_j = \lambda_j \nu_j, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$$

Projection (keep first k components, with $k \ll d$):

$$z_i = \Psi^T (y_i - \bar{y}), \quad \Psi = [\nu_1, \ldots, \nu_k] \in \mathbb{R}^{d \times k}$$

in our case, $k = 96$ and $d = 26000$

Reconstruction:

$$\hat{y}_i = \bar{y} + \Psi z_i$$

Regression Models

Gaussian Processes / Kriging

Each latent coordinate:

$$z_{i(x)} \sim \mathcal{GP}(0, K(x, x'))$$

- **interpolates** from input to reduced PCA space
- GP trained on each PCA component
- Captures nonlinear input-output mapping

Matérn kernel

Instead of standard RBF kernel, we use a Matérn kernel:

$$K(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} \cdot \|x - x'\|}{\lambda} \right)^{\nu} k_{\nu} \left(\frac{\sqrt{2\nu} \cdot \|x - x'\|}{\lambda} \right)$$

- ν: smoothness
- λ: length scales of variations

Advantages:

- More flexible than RBF (Radial basis functions)
- Can model rougher functions (particularly useful for our case)
- Better interpretability via ν parameter

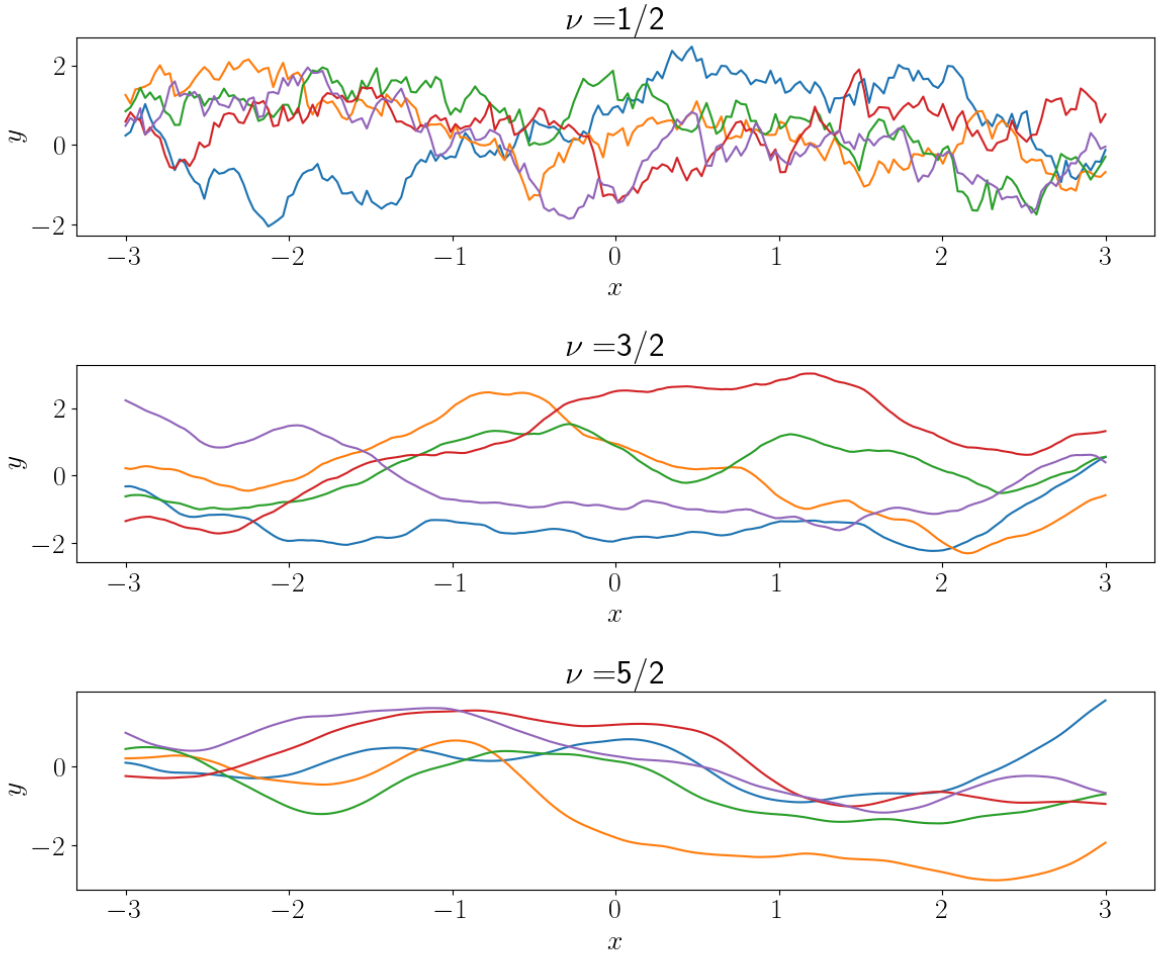


Figure 8: Draws from a GP with Matérn kernel for different nu values

Results

Results

Accuracy

- PCA keeps >99% variance with $k \geq 96$
- GP adds nonlinear accuracy, Matérn kernel strictly outperforms RBF kernels.

Model	RMSE	Max AE	Prediction time
linear regression	7e-2 dB(A)	2.2 dB(A)	70 ms
PCA GPR (RBF)	4e-2 dB(A)	1e-1dB(A)	170 ms
PCA GPR (Matérn)	8e-3 dB(A)	1.2e-1dB(A)	171 ms

Error metrics

- Rooted Mean Squared Error:

$$RMSE = \sqrt{\sum_{i=0}^{N} \sum_{j=0}^{d} \left(Y_{pred,ij} - Y_{ij}\right)^2}$$

- Maximum Absolute Error:

$$MaxAE = \max_{i \in [0,N], j \in [0,d]} \left|Y_{pred,ij} - Y_{ij}\right|$$

Necessary size of the train set

- For 96 PCA components, 90 samples are sufficient to reach a RMSE < 0.01 dB(A), for a 4-parameter input space
- No more noticeable improvement past 100 samples

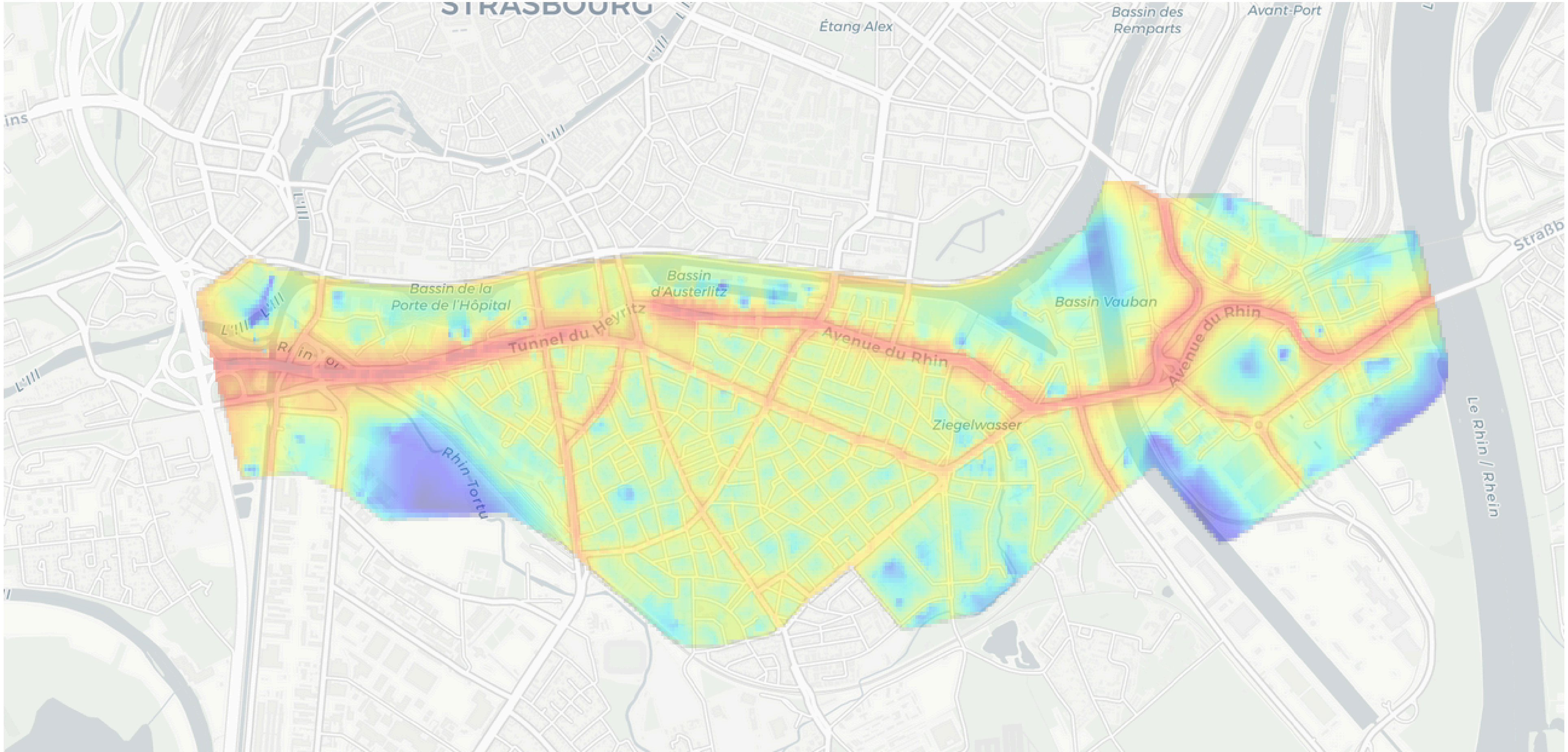


Figure 9: Heatmap of the sound levels estimated by the metamodel

Figure 10: Heatmap of the sound levels estimated by the metamodel

Conclusion

Conclusion

Summary

- Developed a metamodel for traffic noise along the Rhine avenue
- Reduced simulation time from 1h30 to $<$ 200 ms
- PCA reduced output space while preserving >99% variance
- Gaussian Processes with Matérn kernel gave highest accuracy

Implications

- Enables sensitivity analysis
- Opens the way for **dynamic / real-time noise maps** for urban planning

Perspectives

- Extend to more noise sources (rail, industry, trams, etc...)
- Integration with current measurement campaigns
- Deployment in EMS noise monitoring workflows

Thank you for your attention !