Noise Modelling

saison 2018-2019

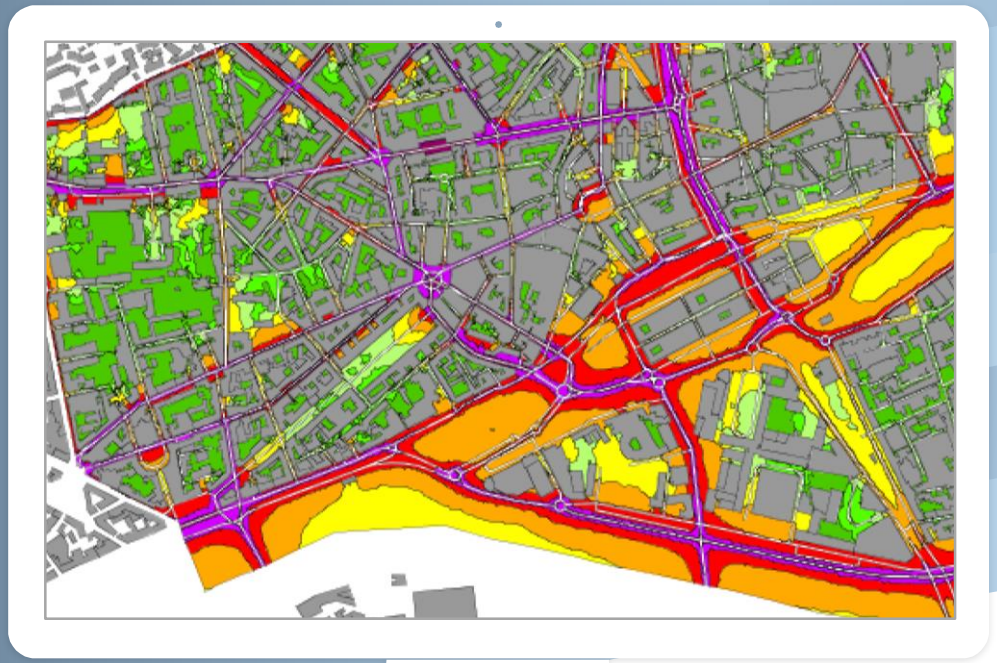
Pierre Aumond, JTAV 2019

JOURNEES TECHNIQUES ACOUSTIQUE ET VIBRATIONS

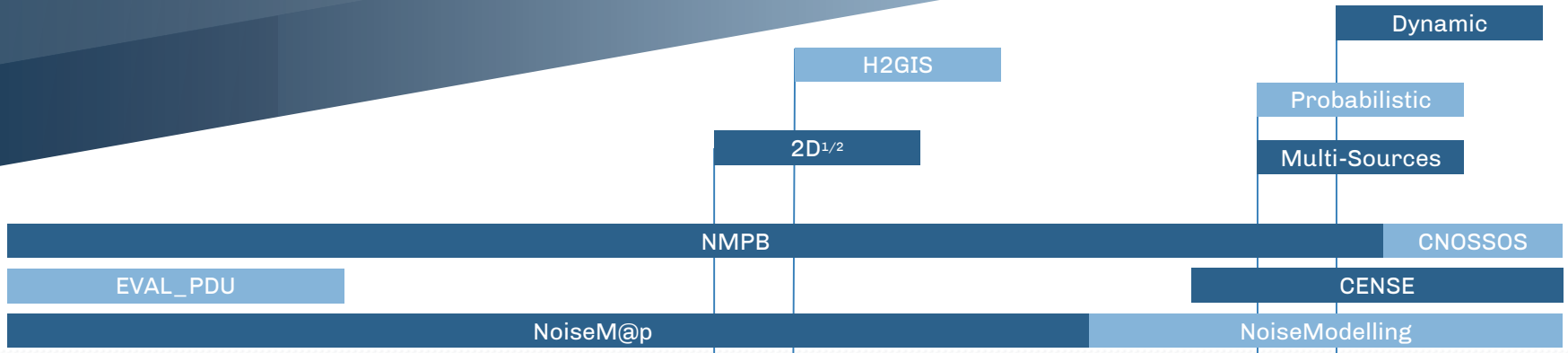
Noise Modelling

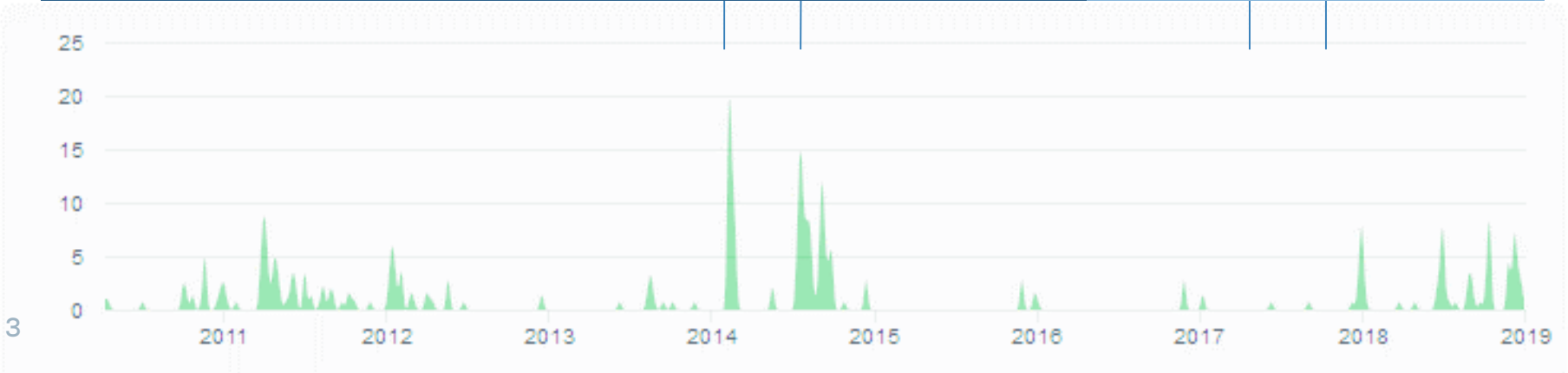
Outil gratuit & open-source
pour la cartographie du bruit

noise-planet.org

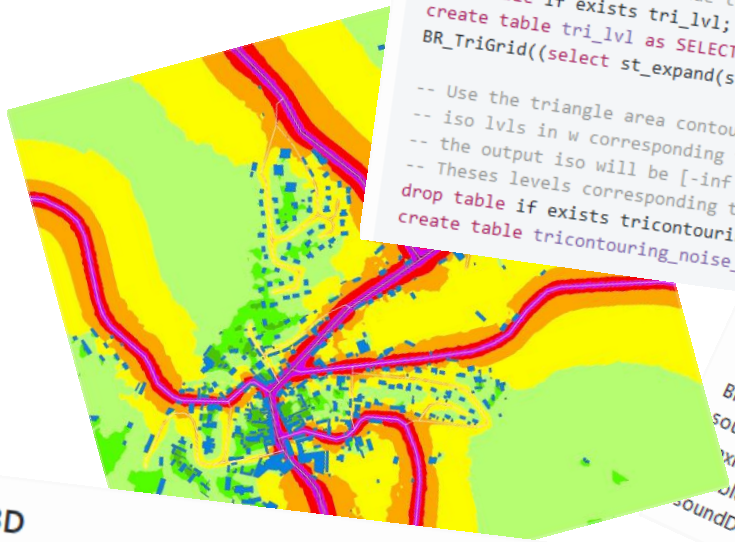


Depuis 2010…


Dynamic
H2GIS
Probabilistic
2D1/2
Multi-Sources
NMPB
CNOSSOS
EVAL_PDU
CENSE
NoiseM@p
NoiseModelling


25
20
15
10
5
0
2011
2012
2013
2014
2015
2016
2017
2018
2019

JTAV 2017



```
-- Sound propagation from sources through buildings
-- Compute only 750m inside the defined source input envelope
drop table if exists tri_lvl;
create table tri_lvl as SELECT * from
BR_TriGrid((select st_expand(st_envelope(st_accum(the_geom)), -750, - 750) the_geom fr

-- Use the triangle area contouring interpolation (split triangle covering level param
-- iso lvls in w corresponding to dB->'45,50,55,60,65,70,75,200'
-- the output iso will be [-inf to 45] -> 0 ]45 to 50] -> 1 etc..
-- Theses levels corresponding to the ranges specified in the standard NF S 31 130
drop table if exists tricontouring_noise_map;
create table tricontouring_noise_map AS SELECT * from ST_TriangleContouring('tri_lvl',
```

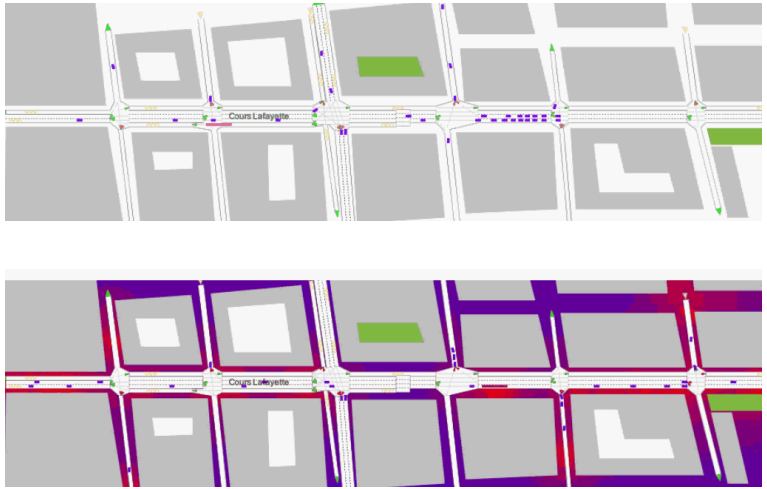
Table function.Sound propagation in 2 dimension. Return 6 columns. TRI_ID integer,THE_GEOM polygon,W_V1 double,W_V2 double,W_V3 double,CELL_ID integer.

BR_TriGrid(VARCHAR buildingsTable, VARCHAR sourcesTable,VARCHAR sourcesTableSoundFieldName, VARCHAR groundTypeTable, double maximumPropagationDistance, double maximumWallSeekingDistance, double roadsWidt le receiversDensification, double maximumAreaOfTriangle, int soundReflecti oundDiffractionOrder, double wallAlpha)

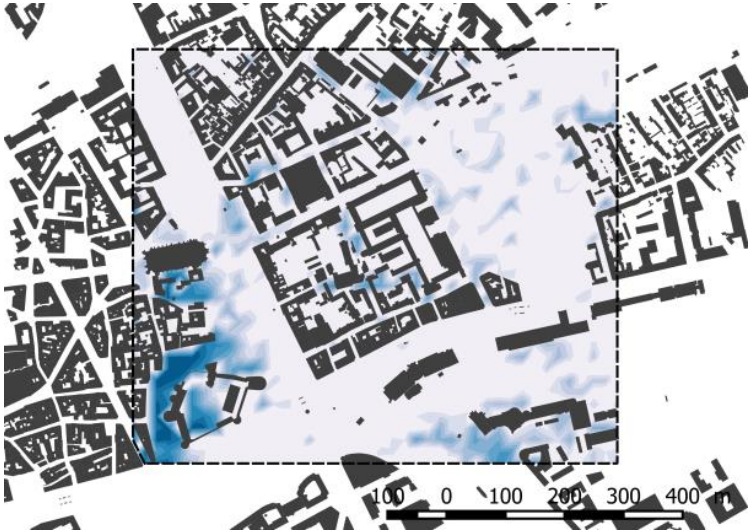
BR_TriGrid3D

Table function.Sound propagation in 3 dimension. Return 6 columns. TRI_ID integer,THE_GEOM polygon,W_V1 double,W_V2 double,W_V3 double,CELL_ID integer.

JTAV 2018

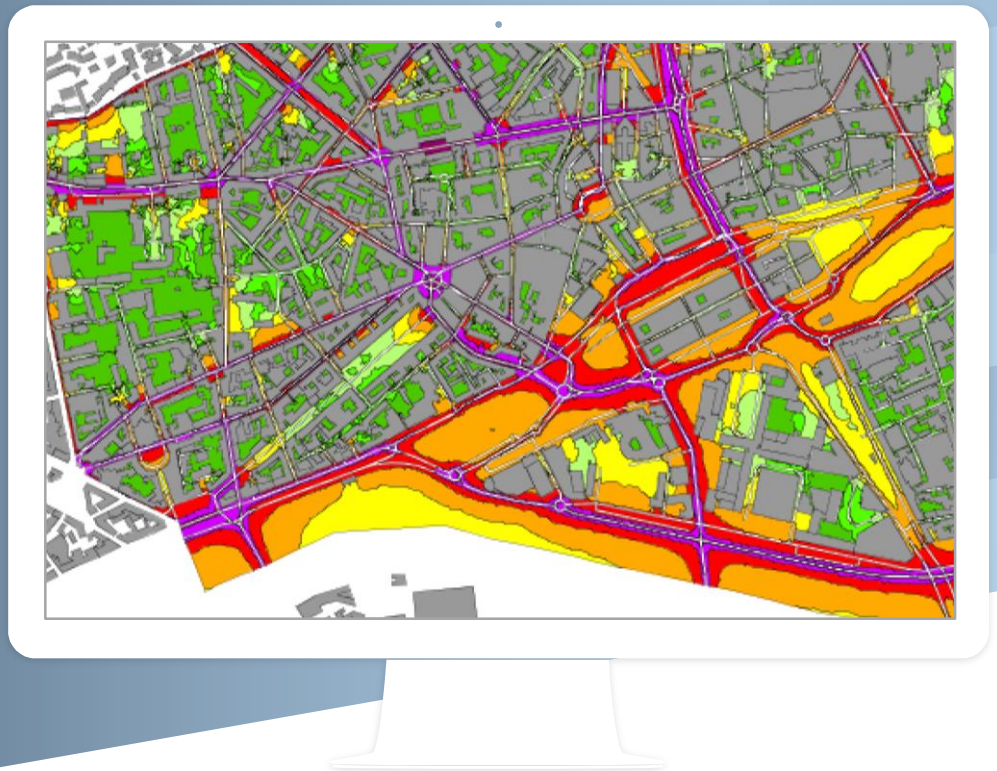


Dynamique



Multi-sources

Et en 2018-2019 ?

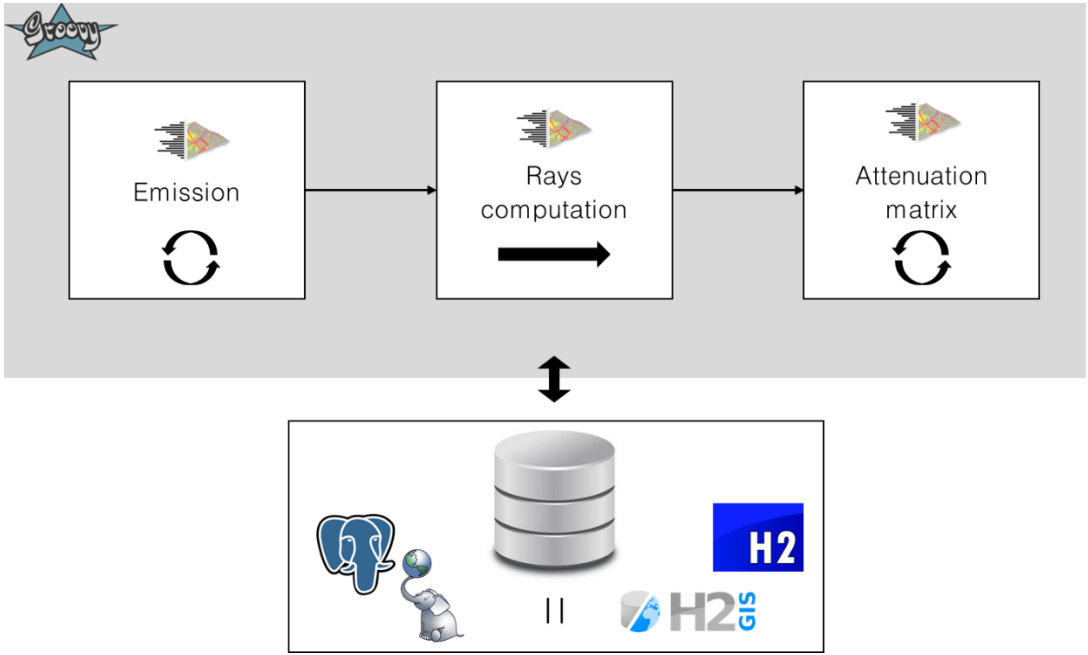


Chaîne de calcul “classique “

Chaîne de calcul “dynamique“

Chaîne de calcul 2019

Lancer pleins de calculs !



Chaîne de calcul 2019

Propagation

3

CNOSSOS – Directive 2015
NMPB

A faire:
- *Rayleigh criterium*

Recherche de chemins

Reflexions
Diffractions

2

A faire:
- Rayleigh criterium
- Backward Rays
- *Diffraction over ground > 15°*
- « retro-diffraction »
.

Emission

1

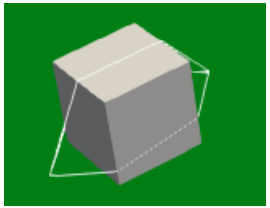
CNOSSOS – Road traffic
NMPB – Road Traffic & Tramway

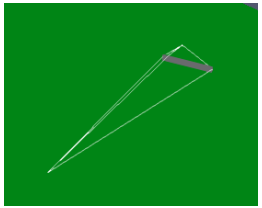
A faire:
- *Autres sources*

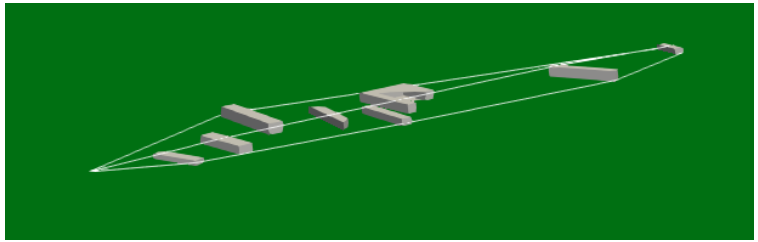
Tests “Académiques”

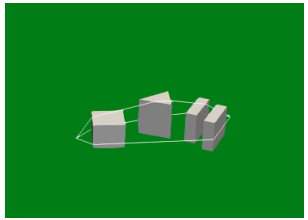
Pour :

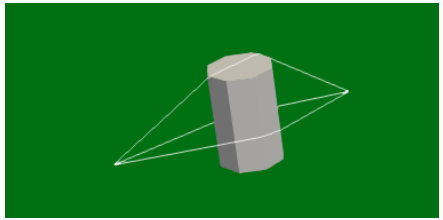
- L’émission
- La recherche de chemins
- La propagation





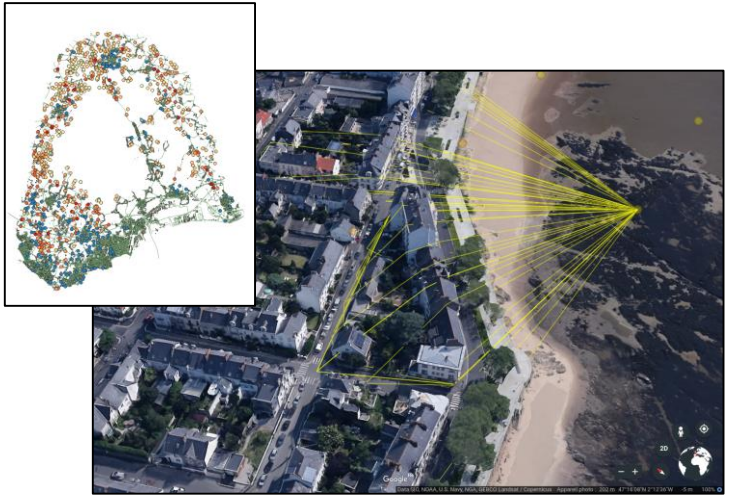




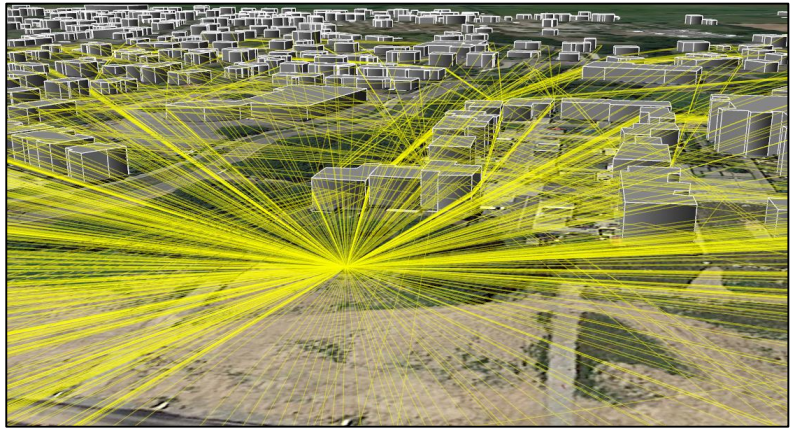


Tests de robustesse

Cas “test” réalistes

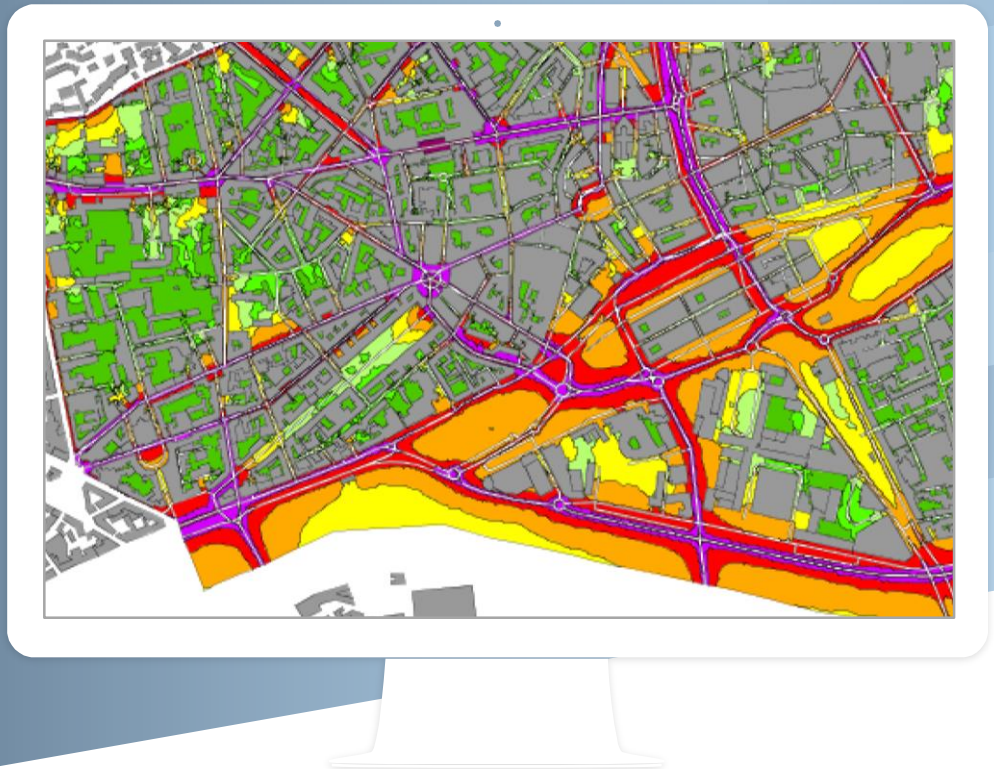


Marais de la Brière



Lorient, France – CENSE Project

Interface ?



Interface utilisateur

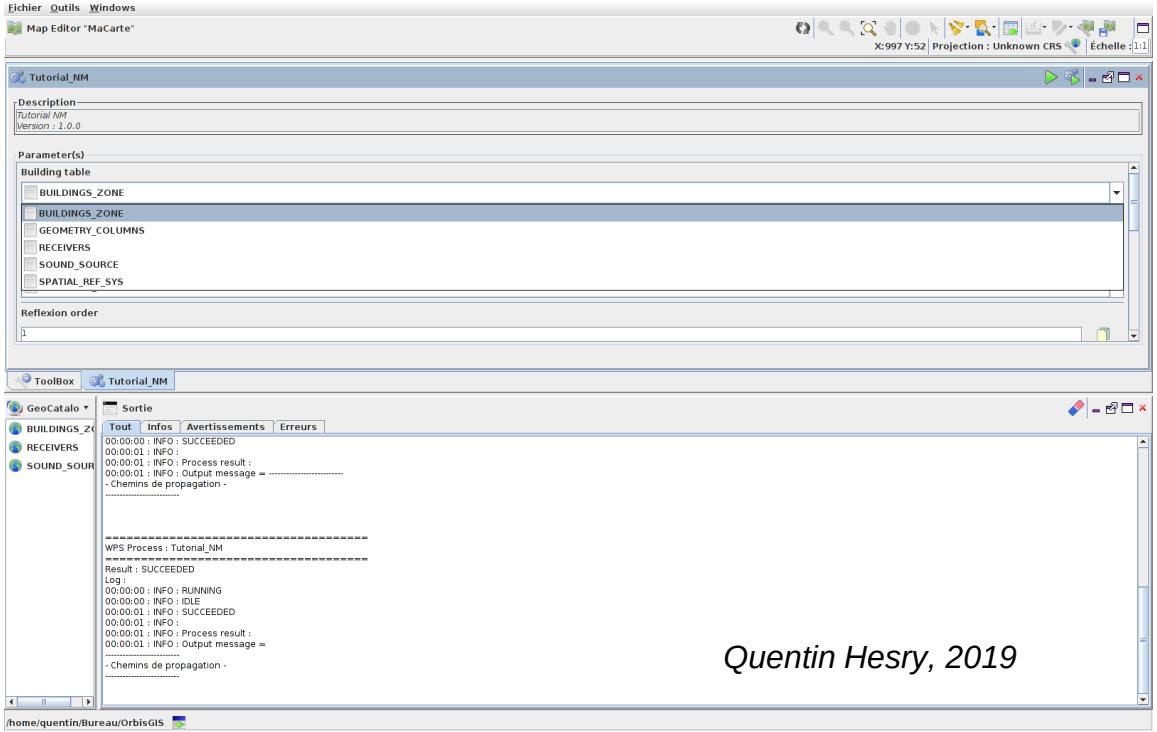
Mode utilisateur avancée:

```
//-----------------------------------------------------------------
// ---------- ICI On calcul les rayons entre sources et recepteurs (c est pour ça r=0, on le fait qu'une fois)
//-----------------------------------------------------------------

PointNoiseMap pointNoiseMap = new PointNoiseMap( buildingsTableName: "BUILDINGS_ZONE",  sourcesTableName: "ROADS_SRC_ZONE",  receiverTableName: "RECEIVERS2")
pointNoiseMap.setComputeHorizontalDiffraction(compute_horizontal_diffraction)
pointNoiseMap.setComputeVerticalDiffraction(compute_vertical_diffraction)
pointNoiseMap.setSoundReflectionOrder(reflexion_order)
pointNoiseMap.setHeightField("HAUTEUR")
pointNoiseMap.setDemTable("DEM_LITE2")
pointNoiseMap.setMaximumPropagationDistance(max_src_dist)
pointNoiseMap.setMaximumReflectionDistance(max_ref_dist)
pointNoiseMap.setWallAbsorption(wall_alpha)
pointNoiseMap.setSoilTableName("LAND_USE_ZONE_CAPTEUR2")
pointNoiseMap.setThreadCount(n_thread)
JDBCComputeRaysOut jdbcComputeRaysOut = new JDBCComputeRaysOut()
pointNoiseMap.initialize(connection, new EmptyProgressVisitor())
pointNoiseMap.setComputeRaysOutFactory(jdbcComputeRaysOut)

Set<Long> receivers_ = new HashSet<>()
for (int i = 0; i < pointNoiseMap.getGridDim(); i++) {
    for (int j = 0; j < pointNoiseMap.getGridDim(); j++) {
        IComputeRaysOut out = pointNoiseMap.evaluateCell(connection, i, j, new EmptyProgressVisitor(), receivers_)

        if (out instanceof ComputeRaysOut) {
            allLevels.addAll(((ComputeRaysOut) out).getVerticesSoundLevel())
            propaMap2.addAll(((ComputeRaysOut) out).getPropagationPaths())
        }
    }
}

jdbcComputeRaysOut.closeKML()
```

Interface utilisateur

Mode Pointer-et-cliquer :

En utilisant une interface WPS

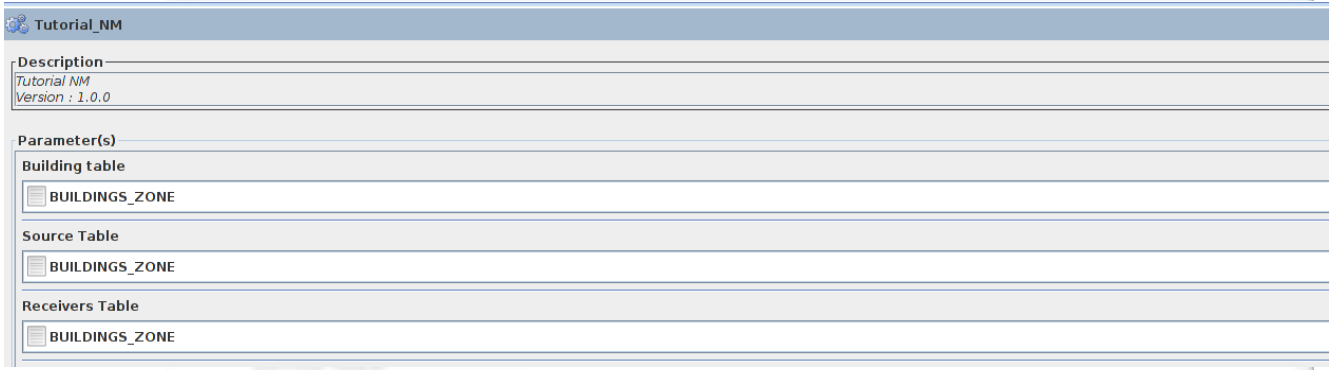


Quentin Hesry, 2019

Interface utilisateur

Mode Pointer-et-cliquer :

En utilisant une interface WPS

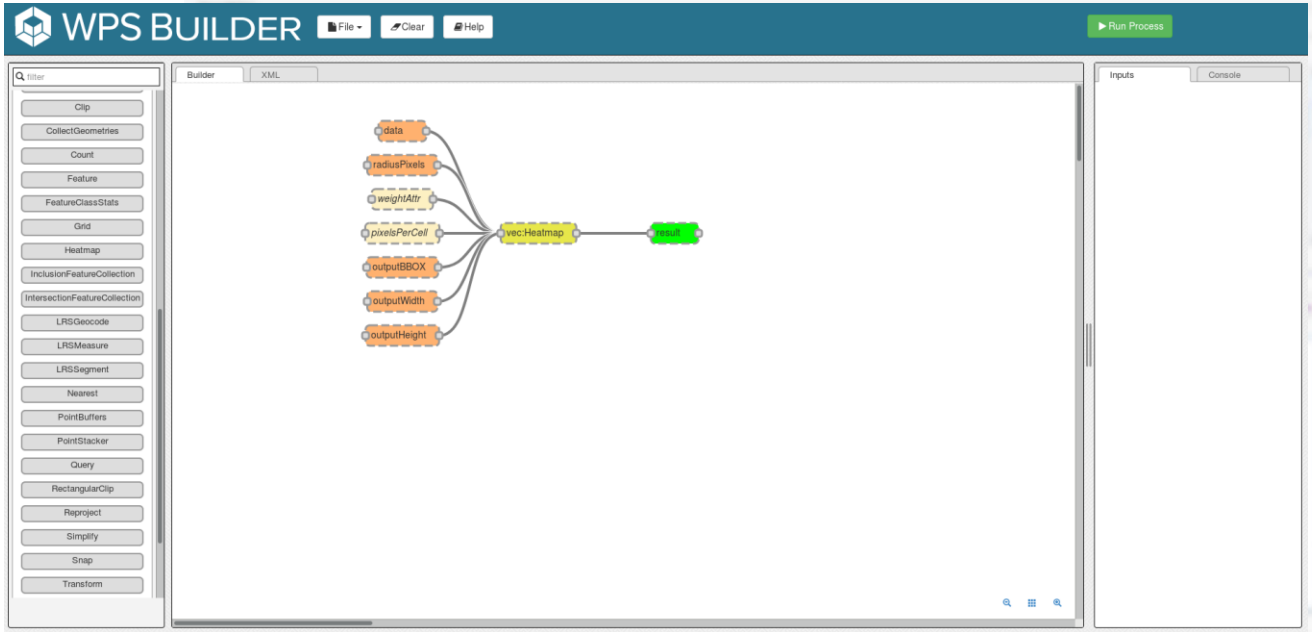


Interface utilisateur

Mode Pointer-et-cliquer :

En utilisant une interface WPS

(bientôt serveur)

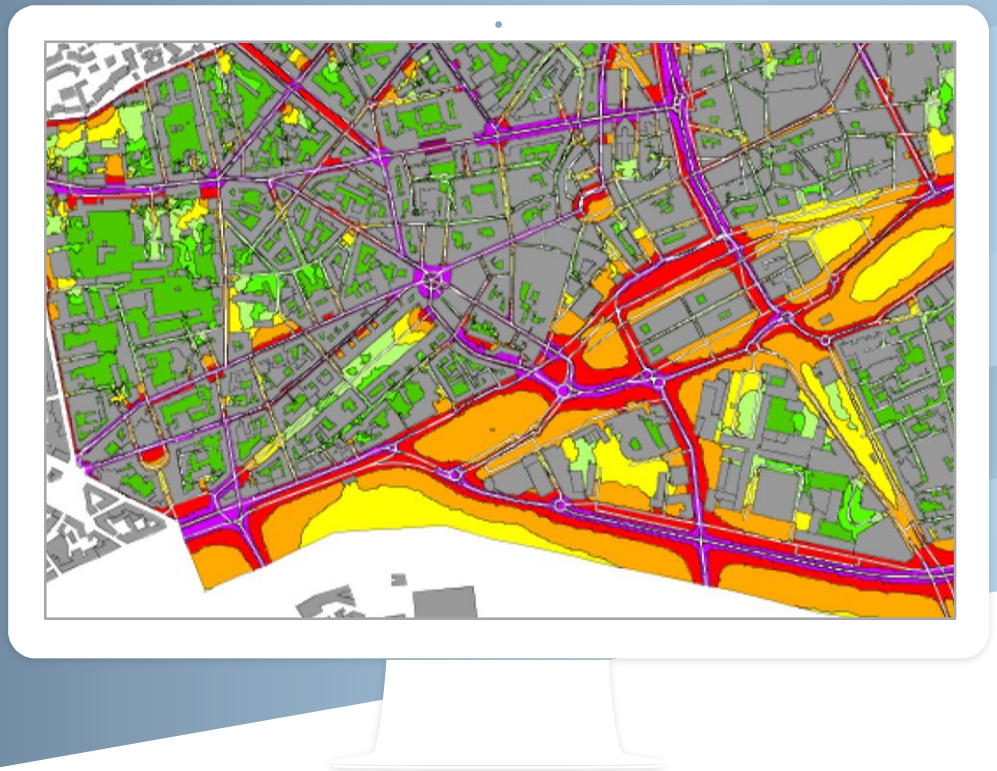


Interface utilisateur

Nouveau format de sortie:

Compatible Google Earth

Applications ?



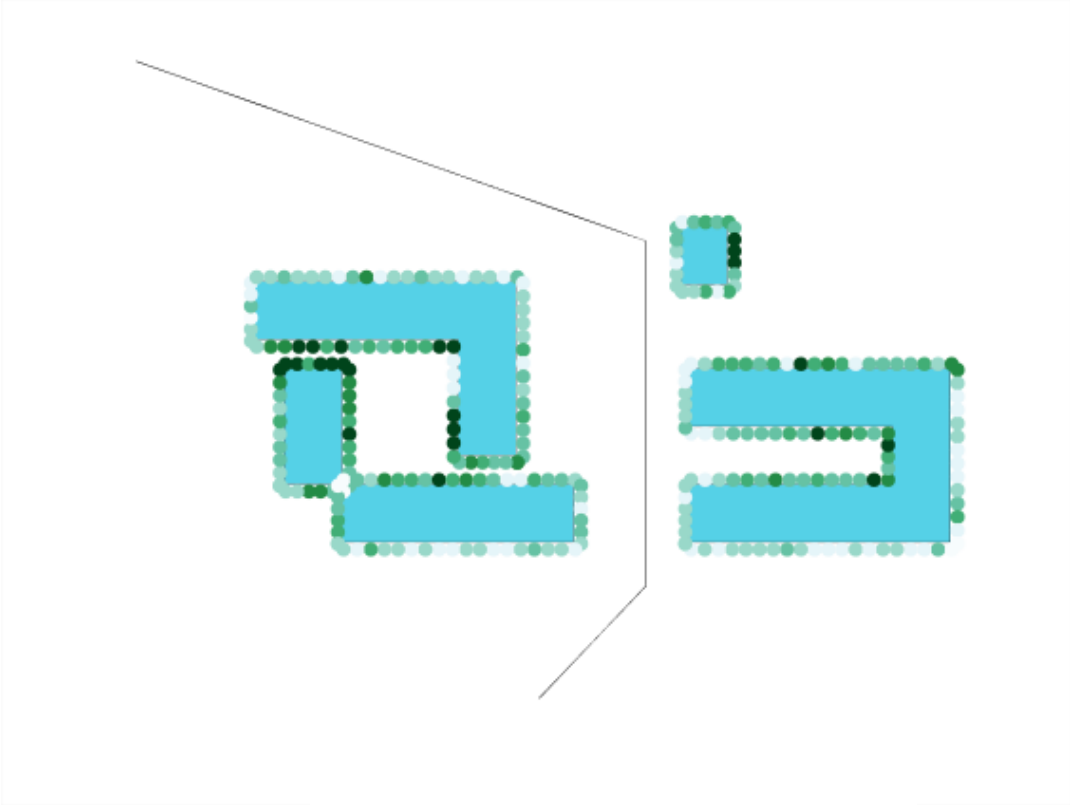
Applications 2018-2019

Exposition dynamique

Guillermo Quintero, 2018

Applications 2018-2019

Analyse de sensibilité



Sensibilité à la diffraction horizontale

Applications 2018-2019

Méta-modélisation

Antoine Lesieur (A suivre !)

Pierre Aumond - JTAV 2019
pierre.aumond@ifsttar.fr

Merci !

Join the community !

http://noise-planet.org

GitHub

“NoiseModelling: An Open Source GIS Based Tool to Produce Environmental Noise Maps”, E. Bocher, G. Guillaume, J. Picaut , G. Petit, N. Fortin, 2019