technische universität
dortmund

LAMARR
INSTITUTE FOR MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

astroparticle physics

Radio Interferometer Simulations with **pyvisgen**

Physics Monthly | News from `radionets`

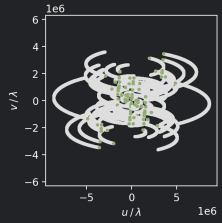
Anno Knierim
anno.knierim@tu-dortmund.de

December 15, 2025

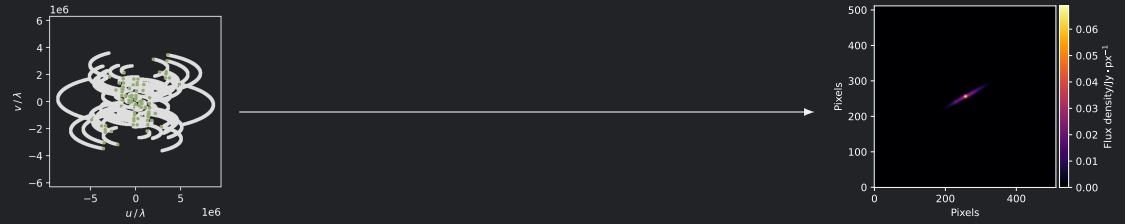
Astroparticle Physics | WG Rhode/Elsässer

Title Image: **C. Michel.** *VLBA – Mauna Kea Observatories*. 2018.

Goal: Clean measurements of radio interferometer experiments

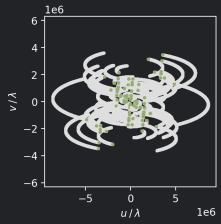


Goal: Clean measurements of radio interferometer experiments

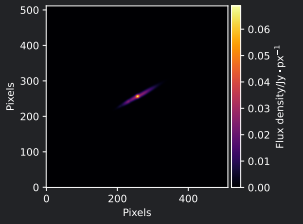


Goal: Clean measurements of radio interferometer experiments

Goal: Clean measurements of radio interferometer experiments

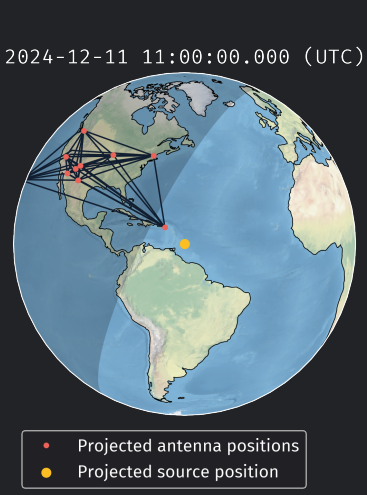


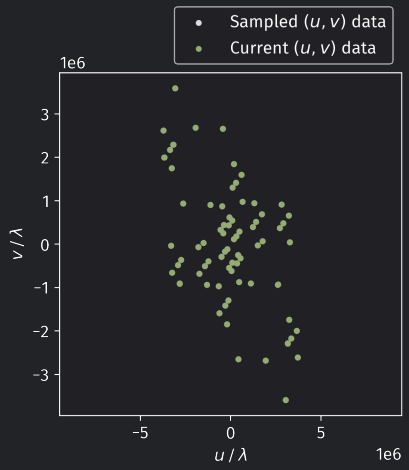
RADIONETS
Deep Learning-Based Imaging in Radio Interferometry

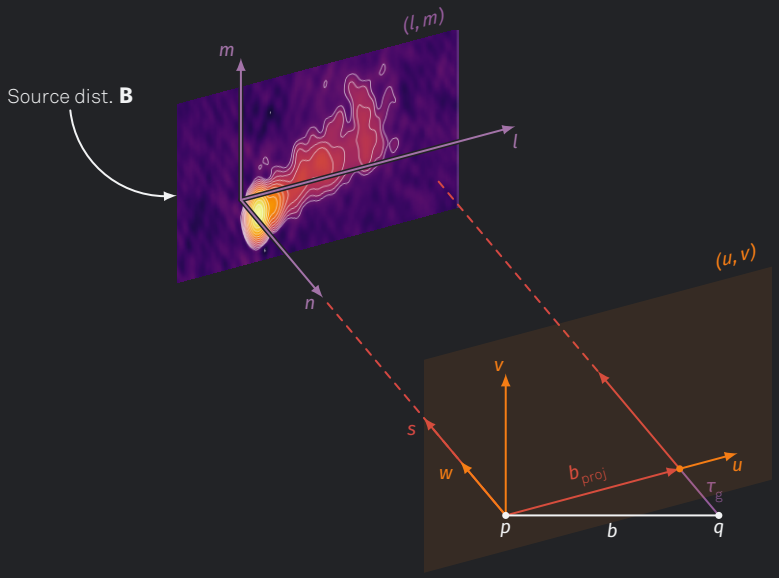


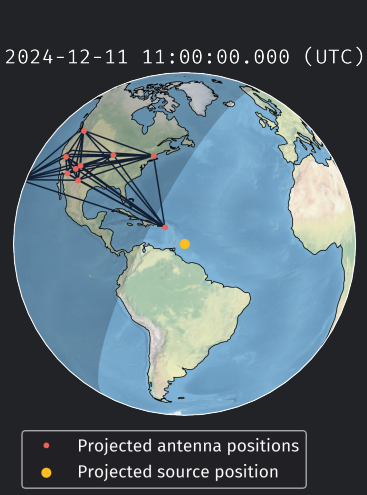
PYVISGEN
Visibility Simulations in Python

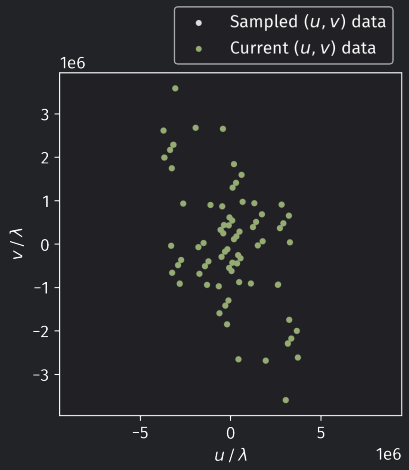
Modern Radio Astronomy

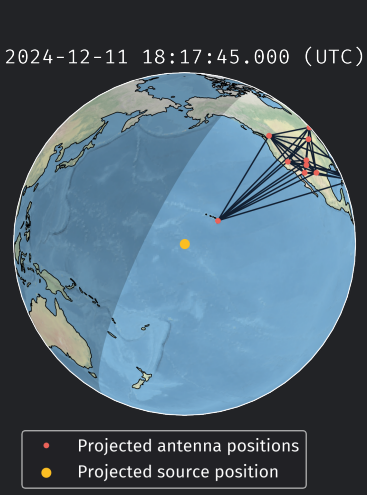

2024-12-11 11:00:00.000 (UTC)
Projected antenna positions
Projected source position

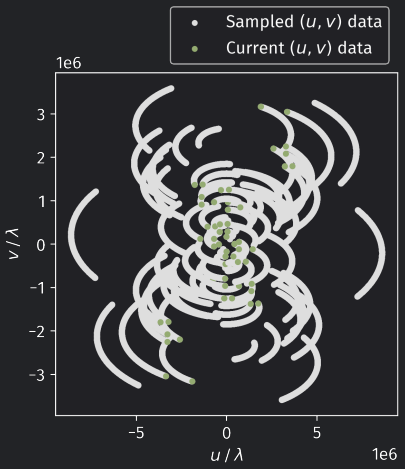

Sampled (u, v) data
Current (u, v) data
1e6
3
2
1
0
−1
−2
−3
v / λ
−5
0
5
u / λ
1e6


Source dist. **B**
(l, m)
m
l
n
(u, v)
v
s
w
b proj
u
T g
p
b
q

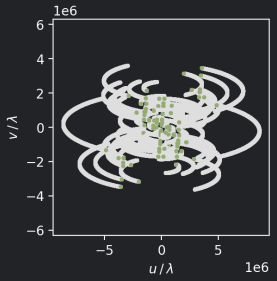

2024-12-11 11:00:00.000 (UTC)
Projected antenna positions
Projected source position


Sampled (u, v) data
Current (u, v) data
1e6
3
2
1
0
−1
−2
−3
v / λ
−5
0
5
u / λ
1e6

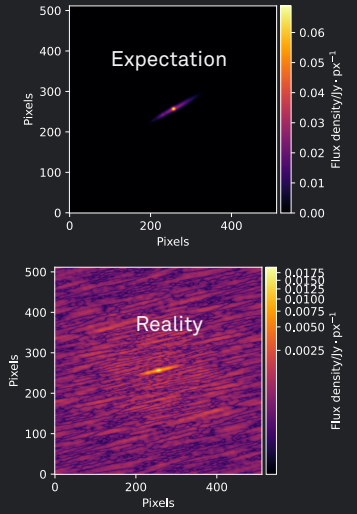

2024-12-11 18:17:45.000 (UTC)
Projected antenna positions
Projected source position


Sampled (u, v) data
Current (u, v) data
1e6
3
2
1
0
−1
−2
−3
v / λ
−5
0
5
u / λ
1e6

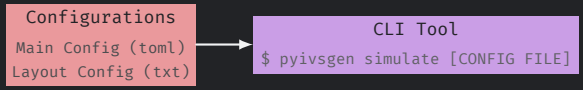
Problem



Fourier Transform



pyvisgen



- CLI tool loads main configuration file
- `Pydantic`-based configuration system
- Configuration passed to `Observation` class

PYVISGEN
Visibility Simulations in Python

Configurations
Main Config (toml)
Layout Config (txt)

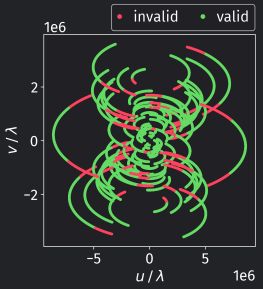
→

CLI Tool
`$ pyivsgen simulate [CONFIG FILE]`

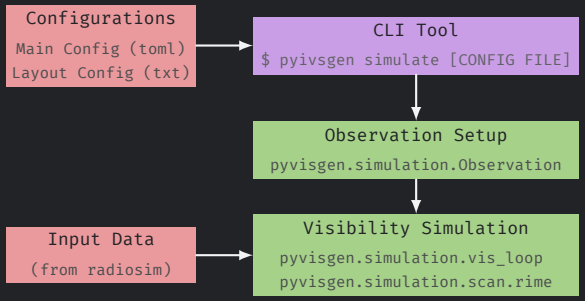
↓

Observation Setup
`pyvisgen.simulation.Observation`

- Observation parameters
- Array configuration
- Output: Valid baselines dataclass object



PYVISGEN
Visibility Simulations in Python



- Compute complex visibilities for baselines
- Based on *RIME*
- Simulates direction dependend and independend effects
- Adds noise simulation
- Output: Visibility dataclass object

PYVISGEN
Visibility Simulations in Python

Radio Inteferometer Measurement Equation (RIME)

$$\mathbf{V}_{pq}(l,m) = \sum_s \mathbf{E}_{sp}(l,m)\,\mathbf{K}_{sp}(l,m)\,\mathbf{B}_s(l,m)\,\mathbf{K}^H_{sq}(l,m)\,\mathbf{E}^H_{sq}(l,m)$$

Source Distribution:

$$\mathbf{B}(l,m)$$

Phase Delay:

$$\mathbf{K}(l,m) = \exp\left(-2\pi i\left(u_p l + v_p m + w_p(n-1)\right)\right)$$

Antenna Beam:

$$\mathbf{E}(l,m) = \mathrm{jinc}\left(\frac{2\pi}{\lambda_{\mathrm{obs}}} d \cdot \theta_{lm}\right)$$

$$\mathrm{jinc}(x) = \frac{J_1(x)}{x}$$

Radio Inteferometer Measurement Equation (RIME)

$$\mathbf{V}_{pq}(l,m) = \sum_s \mathbf{E}_{sp}(l,m)\,\mathbf{K}_{sp}(l,m)\,\mathbf{B}_s(l,m)\,\mathbf{K}^H_{sq}(l,m)\,\mathbf{E}^H_{sq}(l,m)$$

Source Distribution:

$$\mathbf{B}(l,m)$$

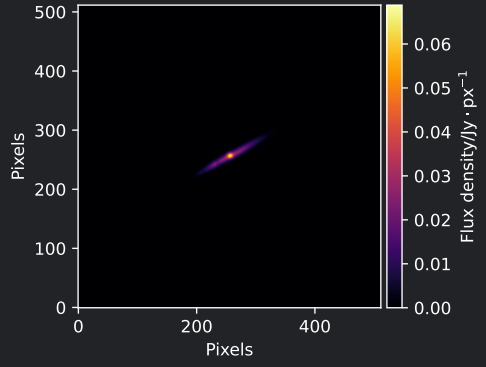
Phase Delay:

$$\mathbf{K}(l,m) = \exp\left(-2\pi i\left(u_p l + v_p m + w_p(n-1)\right)\right)$$

Antenna Beam:

$$\mathbf{E}(l,m) = \mathrm{jinc}\left(\frac{2\pi}{\lambda_{\mathrm{obs}}} d \cdot \theta_{lm}\right)$$

$$\mathrm{jinc}(x) = \frac{J_1(x)}{x}$$



Radio Inteferometer Measurement Equation (RIME)

$$\mathbf{V}_{pq}(l,m) = \sum_s \mathbf{E}_{sp}(l,m)\, \mathbf{K}_{sp}(l,m)\, \mathbf{B}_s(l,m)\, \mathbf{K}^H_{sq}(l,m)\, \mathbf{E}^H_{sq}(l,m)$$

Source Distribution:

$$\mathbf{B}(l,m)$$

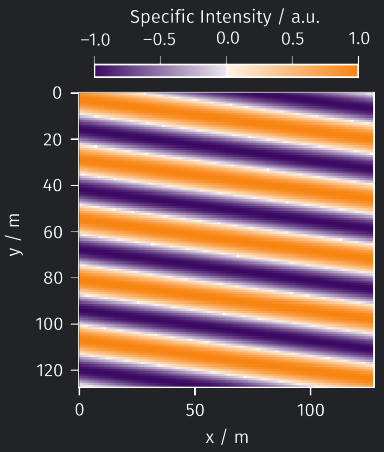
Phase Delay:

$$\mathbf{K}(l,m) = \exp\left(-2\pi i \left(u_p l + v_p m + w_p (n-1)\right)\right)$$

Antenna Beam:

$$\mathbf{E}(l,m) = \mathrm{jinc}\left(\frac{2\pi}{\lambda_{\mathrm{obs}}} d \cdot \theta_{lm}\right)$$

$$\mathrm{jinc}(x) = \frac{J_1(x)}{x}$$



Radio Inteferometer Measurement Equation (RIME)

$$\mathbf{V}_{pq}(l,m) = \sum_s \mathbf{E}_{sp}(l,m)\,\mathbf{K}_{sp}(l,m)\,\mathbf{B}_s(l,m)\,\mathbf{K}^H_{sq}(l,m)\,\mathbf{E}^H_{sq}(l,m)$$

Source Distribution:

$$\mathbf{B}(l,m)$$

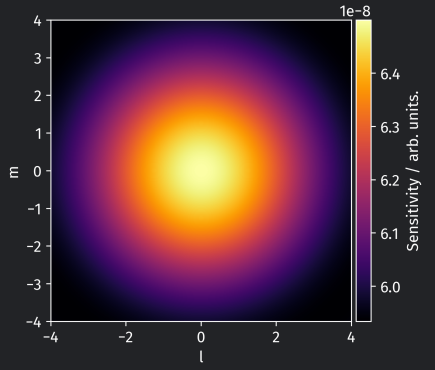
Phase Delay:

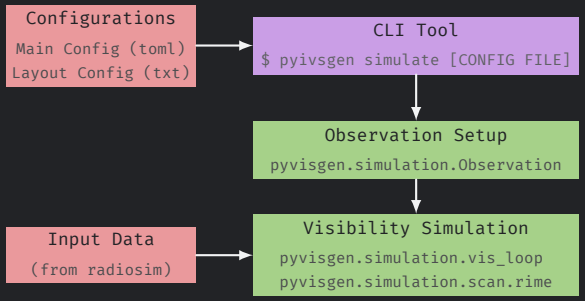
$$\mathbf{K}(l,m) = \exp\left(-2\pi i\left(u_p l + v_p m + w_p(n-1)\right)\right)$$

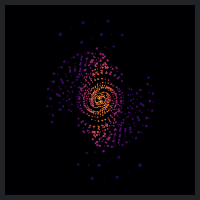
Antenna Beam:

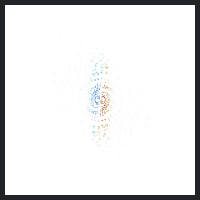
$$\mathbf{E}(l,m) = \mathrm{jinc}\left(\frac{2\pi}{\lambda_{\mathrm{obs}}} d \cdot \theta_{lm}\right)$$

$$\mathrm{jinc}(x) = \frac{J_1(x)}{x}$$



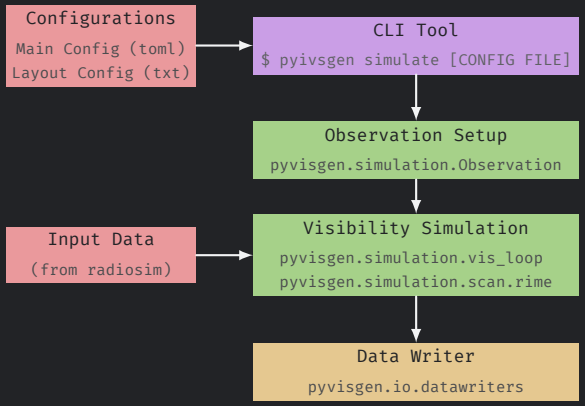






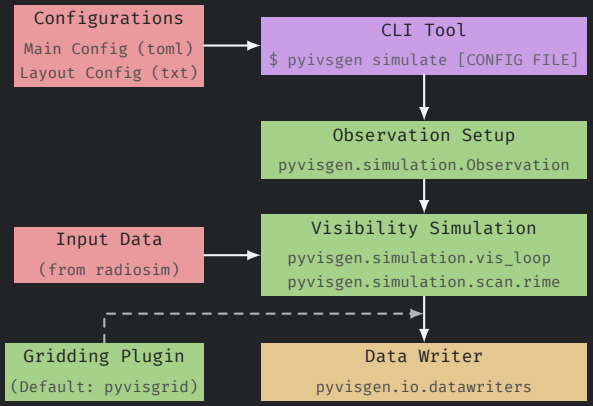
- Compute complex visibilities for baselines
- Based on *RIME*
- Simulates direction dependend and independend effects
- Adds noise simulation
- Output: Visibility dataclass object

PYVISGEN
Visibility Simulations in Python



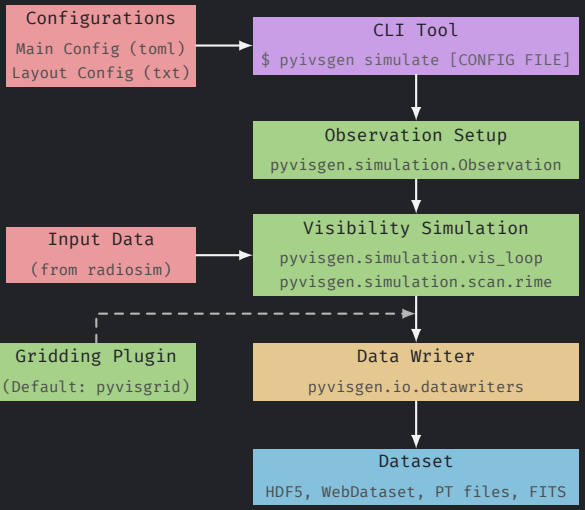
- Format conversion
- Metadata
- Dataset storage

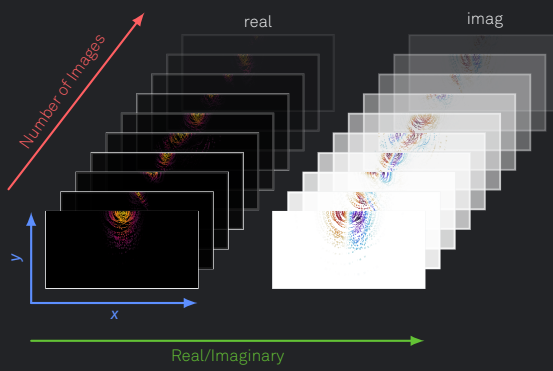
PYVISGEN
Visibility Simulations in Python



- Gridding converts non-uniformly sampled data into a regular grid for FFT
- Plugin support for user-written gridders
- Default gridder: `pyvisgrid.core.Gridder`

PYVISGEN
Visibility Simulations in Python





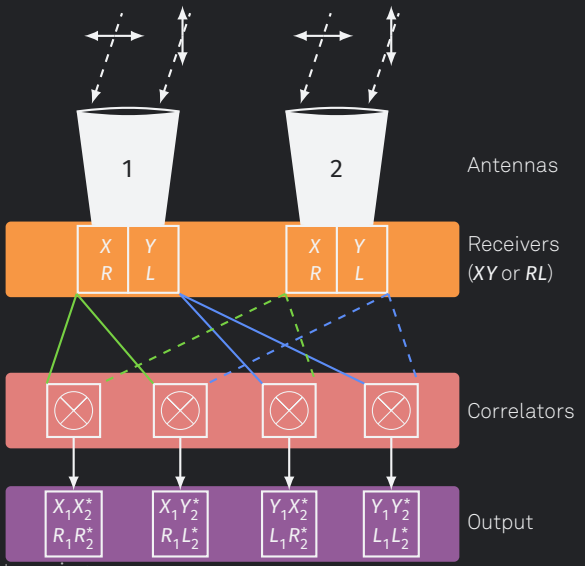
PYVISGEN
Visibility Simulations in Python

Recent Additions to **pyvisgen**

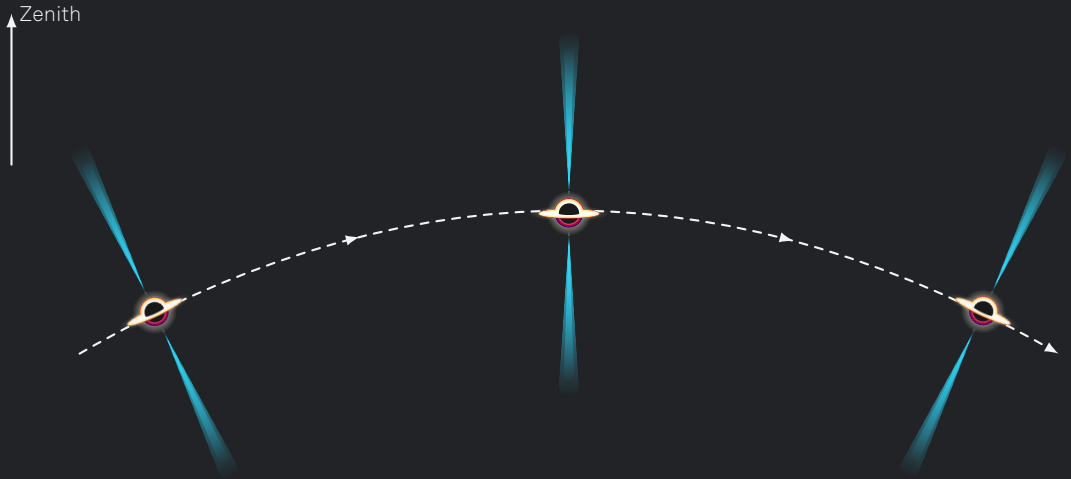
Polarization

$$\mathbf{B}_{+} = \begin{pmatrix} V_{XX^*} & V_{XY^*} \\ V_{YX^*} & V_{YY^*} \end{pmatrix} = \begin{pmatrix} I + Q & U + iV \\ U - iV & I - Q \end{pmatrix}$$

$$\mathbf{B}_{\odot} = \begin{pmatrix} V_{RR^*} & V_{RL^*} \\ V_{LR^*} & V_{LL^*} \end{pmatrix} = \begin{pmatrix} I + V & Q + iU \\ Q - iU & I - V \end{pmatrix}$$



Parallactic Angle Rotation



Further Additions

- CUDA only: CUDA implementation of Flatiron Institute Nonuniform Fast Fourier Transform (cuFINUFFT)
  - Replacement for traditional FFT
  - Multifold speed increase
- Simple carbon emission tracking through `CodeCarbon`

Conclusion

Conclusion

- `pyvisgen` allows us to simulate observations for any array configuration
- Modular architecture based on RIME allows more effects to be added
- (Near) future prospects: Add calibration, add ionosphere effects
- Ultimate goal: Clean real data with `radionets` models trained on `pyvisgen` simulation data

`github.com/radionets-project/pyvisgen`