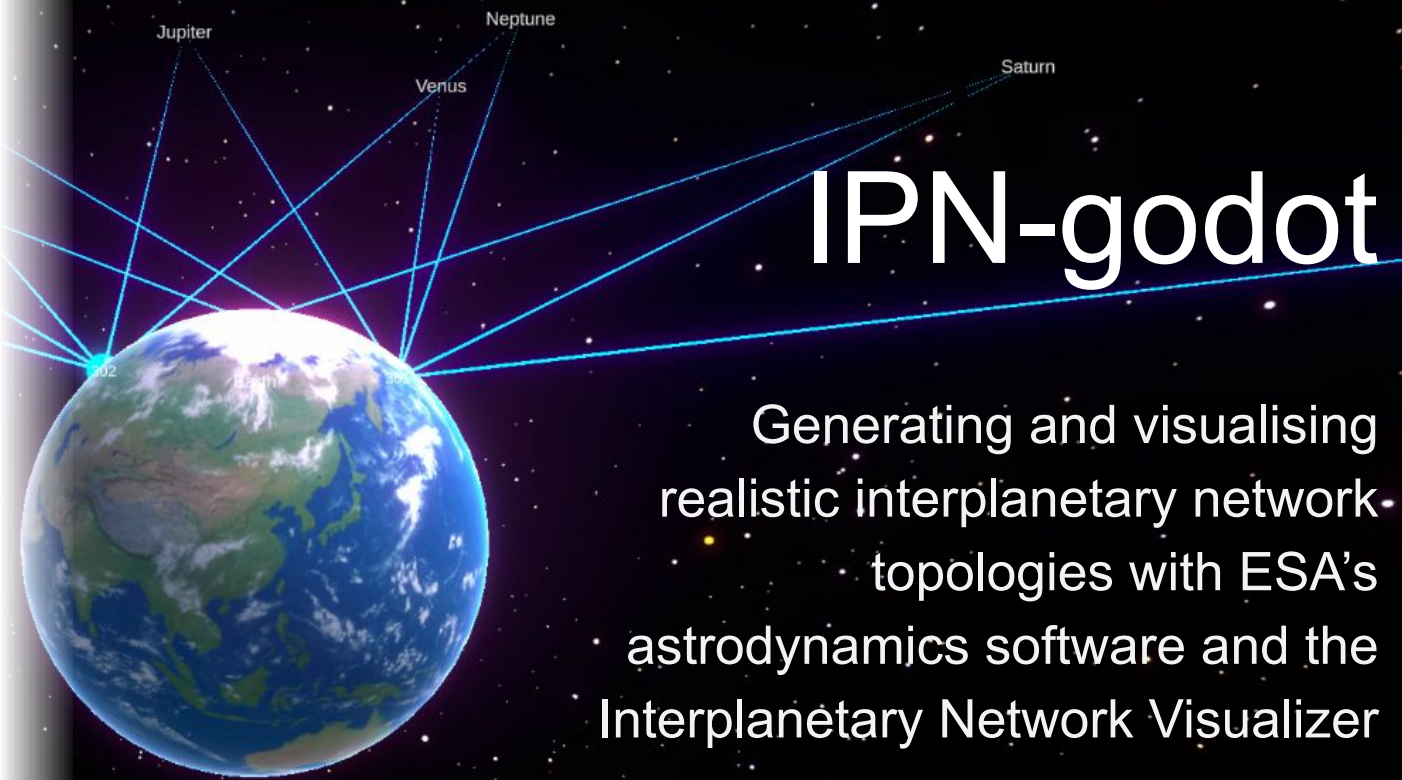
INSA
INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
LYON

Inria

NTN
DAYS



IPN-godot

Generating and visualising realistic interplanetary network topologies with ESA's astrodynamics software and the Interplanetary Network Visualizer

Lucas Poulhe, Alice Le Bihan, Juan Fraire, Pierre Francois

GODOT from ESA

- Astrodynamics library: General Orbit Determination and Optimization Toolkit
- Accessible via registration to anyone in an ESA member state
- C++ or Python
- A complete tool
  - Trajectory and orbit propagation
  - Geometric events
  - Light time correction
  - Multiple coordinate and time systems

What is IPN-godot ?

- A tool for astrodynamics computations of interplanetary network scenarios
  - Precise positions of celestial object
  - Visibility intervals between network nodes
- Python program and Jupyter Notebook
  - Built upon [1], expanded and generalised to customisable scenarios
- Powered by ESA GODOT
- Export scenarios compatible with IPN-v (InterPlanetary Network Visualizer)
  - Dynamic visualisation of scenarios
- Available at https://gitlab.space-codev.org/godot/community-software/ipn-g

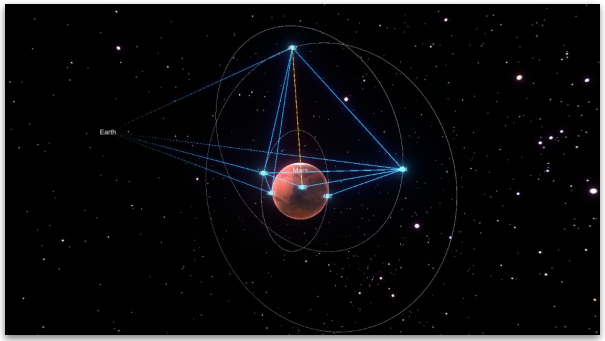
[1]

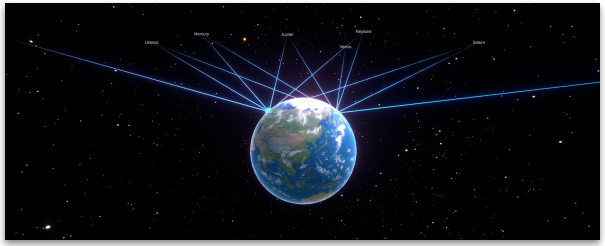
[1] C. Malnati and F. Flentge. “Reference Scenarios for Evaluation of Disruption Tolerant Network Technologies”. SpaceOps 2025. Montreal, Canada, May 2025.

IPN-v (the InterPlanetary Networks visualizer)

Open source 3D visualization for space networks [2]
https://gitlab.inria.fr/jfraire/ipn-v

- Visualize an IPN scenario in its context
  - See the planets, nodes and links move and evolve in real time
- Move around a planet or a node in a 3D environment
- Visualize theoretical contact plans→understand the scenario, find potential weaknesses
- Scenarios generated using IPN-godot





[2] A. Le Bihan, J. A. Fraire, P. Francois, F. Flentge. “IPN-V: The Interplanetary Network Visualizer”. 2024 IEEE 10th International Conference on Space Mission Challenges for Information Technology (SMC-IT), Jul 2024, Mountain View, California, United States.

Workflow

Initialisation
IPN scenario input

User inputs
- ➔ Available **orbiters**
- ➔ Available **ground stations**
- ➔ Optional additional **celestial bodies**

IPN-godot

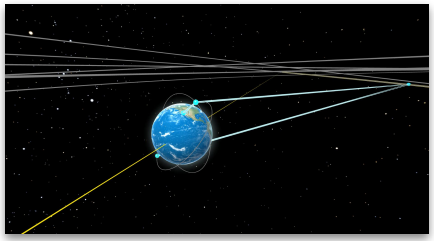
User chooses
- ➔ **Start** and **end** time
- ➔ **Granularity**
- ➔ Which celestial bodies and network nodes to include
- ➔ Special rules for contacts

Processing time…

IPN scenario output folder in IPN-v format

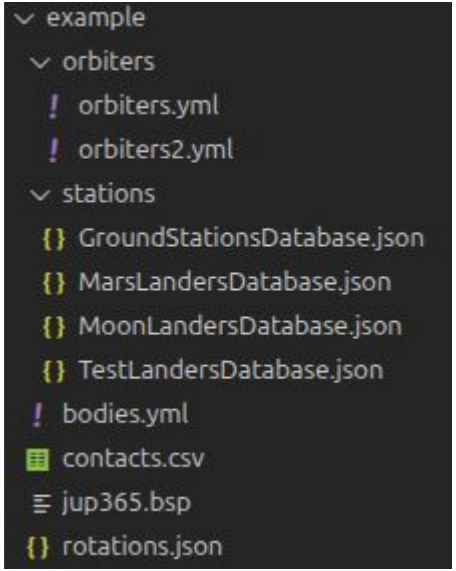
IPN-godot outputs a scenario folder
- ➔ **Position** and **rotation** for all celestial bodies
- ➔ **Position** for all nodes
- ➔ **Contact plan** in IPN-v and ION format

IPN-v



Initialisation phase

- File formats
  - Static files
    → universe.yml and ephemeris files
  - Scenario specific files
    → landers, orbiters, bodies, rotations
- Settings
  - Start and end dates, time step
  - Choose planets, moons and nodes
  - Contact configuration
- Options
  - Contact plan
    - Special rules
    - Manual configuration → contacts.csv
  - Simplified LoS model, or realistic apparent positions model



Input files used by IPN-godot for an example scenario with two orbiters and four landers.

IPN-g : Generate IPN-v Files using Godot

For documentation and tutorial, refer to README.md. For any additional information, don't hesitate to contact us!

Imports

```
[ ]
from data.static.src.simulation import *
```

Simulation Configuration

Scenario folder to use

```
[ ]
scenario = "example" #folder containing all your files in ./data/custom/
```

Load the universe

```
[ ]
uni, uni_cfg = load_universe(scenario, logs=False)
```

Simulation Parameters

```
[ ]
start_date = "2025-01-01T00:00:00 TT"
end_date = ""
analysis_days = 1
step = 100 #time step in seconds
min_elevation_deg = 10.0 # deg
```

Celestial Bodies to use in the simulation

```
planets = ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune", "Pluto"]
moons = ["Moon", "Callisto"]
```

[ ]

Network nodes to use in the simulation

Available nodes

```
print('The available ground stations are {}'.format(', '.join(all_ground_stations(uni))))
print('The available orbiters are {}'.format(', '.join(all_orbiters(uni_cfg))))
```

[ ]

Nodes you want to use

```
landers = ["Madrid_63", "TEST_LANDER_URANUS", "TEST_LANDER_NEPTUNE"] # List of landers/ground stations to use in the simulation
orbiters = ["Earth_orbiter", "Mars_orbiter", "Venus_orbiter", "Sun_orbiter"] # List of orbiters to use in the simulation
print(f"Landers/Ground stations used: {', '.join(landers)}\nOrbiters used: {', '.join(orbiters)}")
gateways = {}
```

[ ]

Special rules for networks nodes

```
rules = {"localNode": []}
print(f"Nodes that will only communicate with nodes on the same planet: {', '.join(rules['localNode'])}")
```

[ ]

Run the calculations

Options

See the associated section in README.md for more details

```
ion = False # Set to True to also export the contact plan in ion format
owlt = True # Set to True to use apparent visibility, False to use line of sight only
logs = True # Set to True to print debug information
contact_step = 0 # time step in seconds for the contact plan, can be left out or set to 0 to use the same step as the simulation
# /!\ Reducing the contact step will greatly increase time of computation and the program may look frozen, but it is still working
# /!\ It may eventually crash if your device does not have enough memory
# /!\ If the contact step is too large, it may miss some contacts and you may experience inconsistencies in IPN-v (Links established despite being through the planets)
contact_direction = "unidirectional" # "unidirectional" will use one direction only, "directional" will use both directions and "manual" will look for predefined contact pairs
```

[ ] Python

Start the script

Generate configuration files in ./output/'scenario'

```
print(export_files(start_date, end_date, analysis_days, step,
                   min_elevation_deg,
                   planets, moons,
                   landers, orbiters,
                   uni, uni_cfg,
                   gateways, scenario,
                   ion=ion, owlt=owlt, logs=logs,
                   rules=rules,
                   contact_step=contact_step, contact_direction=contact_direction))
```

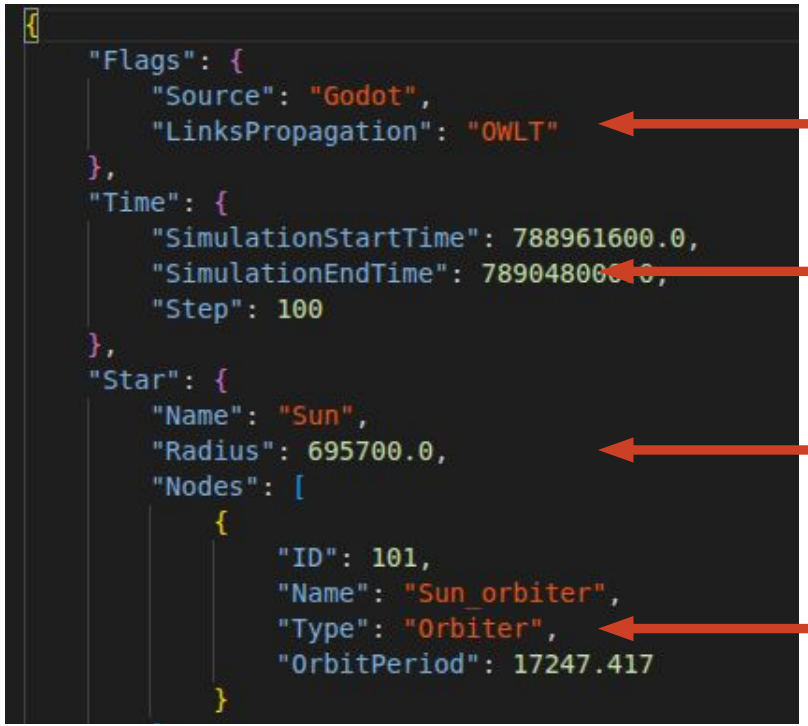
[ ] Python

IPN-godot features

- Precise **positions and rotations**
- User specific **network nodes**
  - Ground stations / landers
  - Orbiters
- Visibility intervals → Contact plan
- Export in IPN-v scenario format
- Support any additional celestial body
  - Positions using extra ephemeris data
  - Rotations using the rotation parameters to define a rotating frame

Output

Configuration file



Flags are used to specify source software and links propagation type

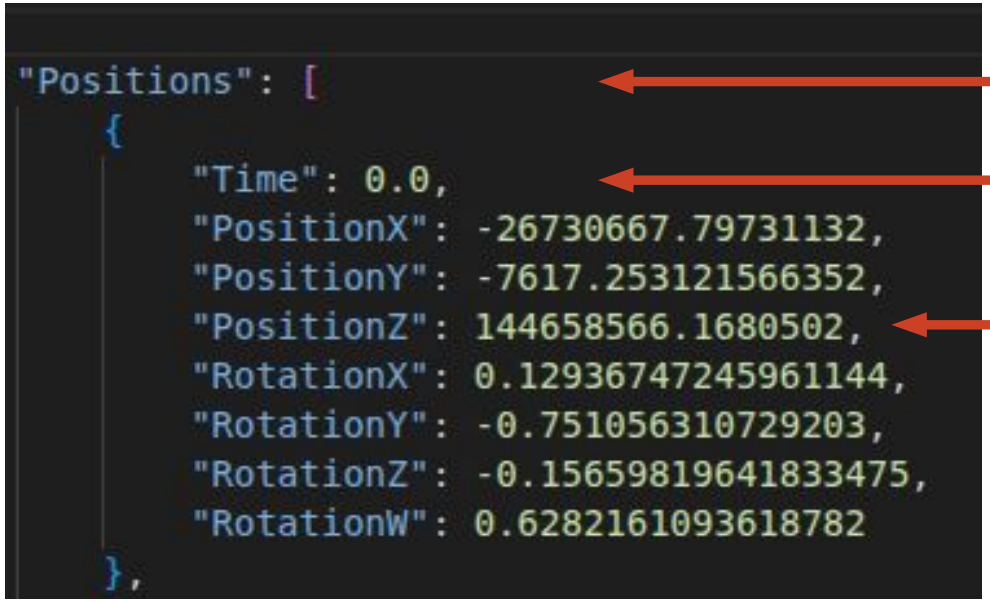
Time parameters of the scenario

Celestial object description

Network node description

Output

Positions file: 1 per planet/moon/node



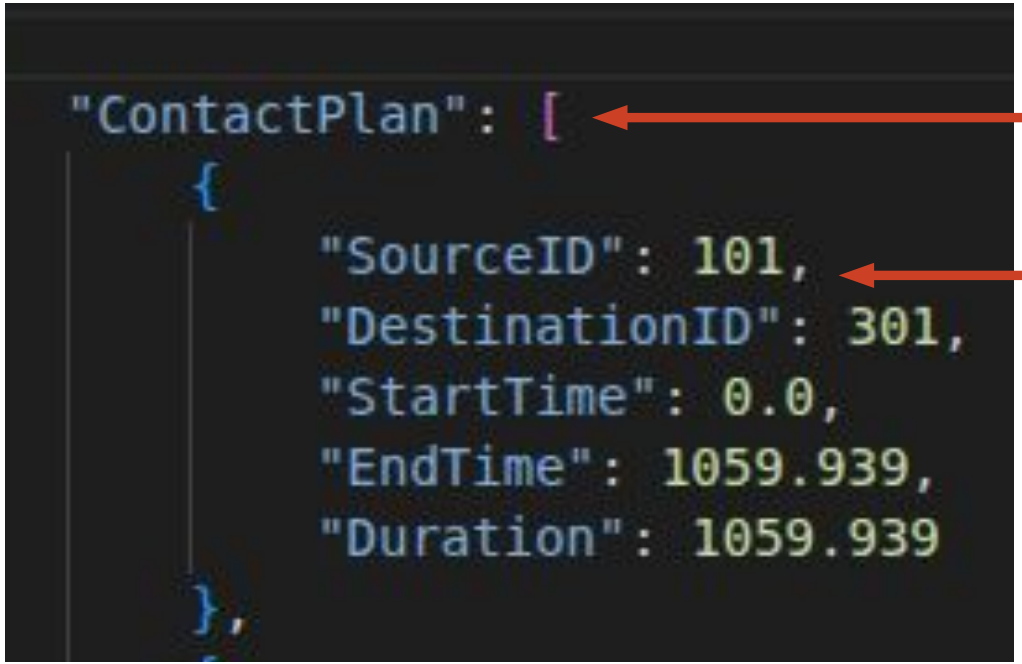
List of positions

Timestamp

Position and rotation of the body at that timestamp

Output

Contact Plan file



Validation and future works

- Validation with ESA
  - Multiple exchanges with DTN experts
  - Relevant results with their data
    - Helped with troubleshooting
- Future of the tool
  - New functionalities
    - Deep space probes
    - Color grading of links → based on latency or any other parameter
    - Physical/logical contact plan separation
  - Containerized version

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Inria

godot

esa

NTN DAYS

Questions ?

lucas.poulhe@insa-lyon.fr
alice.le-bihan@insa-lyon.fr
juan.fraire@inria.fr
pierre.francois@insa-lyon.fr

Bibliography

[1] C. Malnati and F. Flentge. “Reference Scenarios for Evaluation of Disruption Tolerant Network Technologies”. In: *SpaceOps 2025*. Montreal, Canada, May 2025.

[2] A. Le Bihan, J. A., Francois, P., & Flentge, F. (2024). IPN-V : The Interplanetary Network Visualizer; https://hal.science/hal-04996148v1/file/IPN-V_The_Interplanetary_Network_Visualizer.pdf

IPN-g gitlab on space-codev, https://gitlab.space-codev.org/godot/community-software/ipn-g

IPN-v gitlab, https://gitlab.inria.fr/jfraire/ipn-v

IPN-v demo, https://ipnv.net/

Contact us: lucas.poulhe@insa-lyon.fr, alice.le-bihan@insa-lyon.fr