



Outline

- Working with a terminal
- Demo
- Install exercises
- Structure of an exercise

Working in a terminal (linux, mac, win)

- See doc “linux command reference”
- “*Interpréteur de commandes*”

```
→ ~ cd simulation
→ simulation pwd
/Users/dsarrut/simulation
→ simulation ls
data      mac      output  readme.txt
→ simulation Gate mac/main.mac python main.py
```

Workflow

- A simulation = a macro file (or several)
- A macro file = a text file
- A macro = a command

Run a simulation

```

→ ~ cd simulation
→ simulation pwd
/Users/dsarrut/simulation
→ simulation ls
data      mac      output  readme.txt
→ simulation Gate mac/main.mac

```

python main.py

```

/control/execute mac/verbose.mac
/control/execute mac/world.mac
#/control/execute mac/visu.mac

/control/alias SPECT_RADIUS 16
/control/alias SPECT_LENGTH_CM 18

/control/execute mac/spect_head.mac
/control/execute mac/spect_collimator_megp.mac
/control/execute mac/spect_digitizer_In111.mac
/control/execute mac/spect_translation.mac

/control/alias CT_data/ct.mhd
/control/alias CT_ISOCENTER_X 2
/control/alias CT_ISOCENTER_Y 38
/control/alias CT_ISOCENTER_Z -19
/control/alias START_ANGLE 0

/control/execute mac/phantom_nested.mac

/control/execute mac/physics_list_em1.mac
/control/execute mac/physics_cuts_head.mac

/control/execute mac/output_stat.mac

```

_In111.mac

d.mhd

source pixels
8109

.1.mac

Workflow

- A simulation = a ~~macro~~ **python** file (or several)
- A ~~macro~~ file = a text file
- A ~~macro~~ = a command **function**

Run a simulation

```

→ ~ cd simulation
→ simulation pwd
/Users/dsarrut/simulation
→ simulation ls
data      mac      output  readme.txt
→ simulation Gate mac/main.mac

```

python main.py

```

more_1_starting_demo.py

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import opengate as gate
from pathlib import Path

# this first line is required at the beginning of all scripts
if __name__ == "__main__":
    """
    Create a simulation object. The class is 'gate.Simulation'.
    The single object that will contain all parameters of the simulation is called 'sim' here.
    """
    sim = gate.Simulation()

    """
    Main global options.
    The 'sim' object contains a structure called 'user_info' that gather all global options.
    For shorter coding, we call it 'ui'
    - the verbosity means : texts that are displayed during the simulation run (mostly for debug
    - 'visu', if ON, display a windows with a 3D view of the scene.
    - random_engine and random_seed control the pseudo random engine. We recommend MersenneTwister.
      A seed can be specified, e.g. 123456, for reproducible simulation. Or you can use 'auto', an
      will be generated.
    """
    sim.verbose_level = gate.logger.DEBUG
    sim.running_verbose_level = gate.logger.RUN
    sim.g4_verbose = False
    sim.g4_verbose_level = 1
    sim.visu = True
    sim.visu_type = "vrml"

```

Editors

Goal: read and edit the Python code

You can use the one you prefer. Examples:

- Spyder
- VS code
- Pycharm
- gedit
- ... many others !

Example with the first exercise

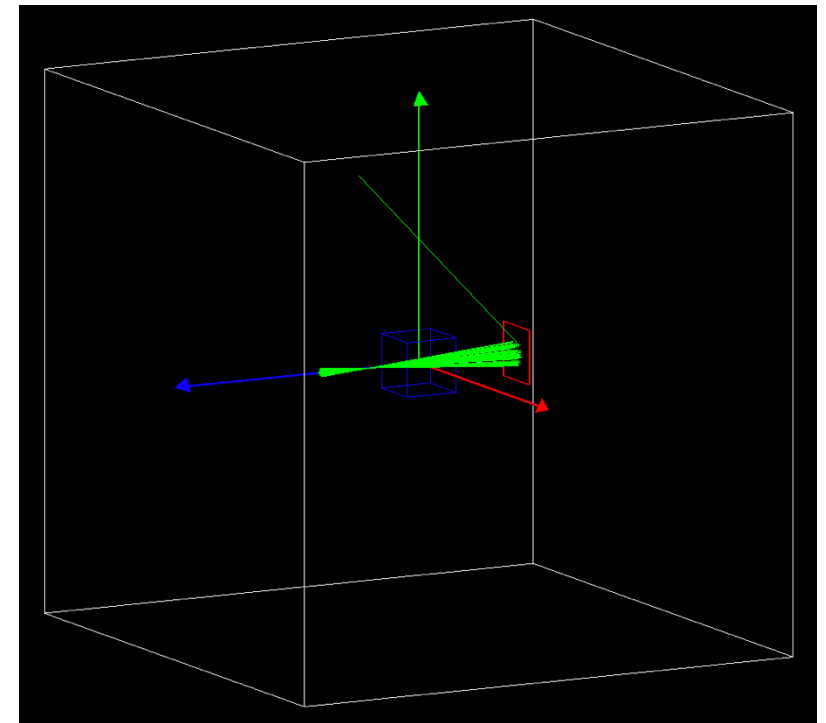
... live ...

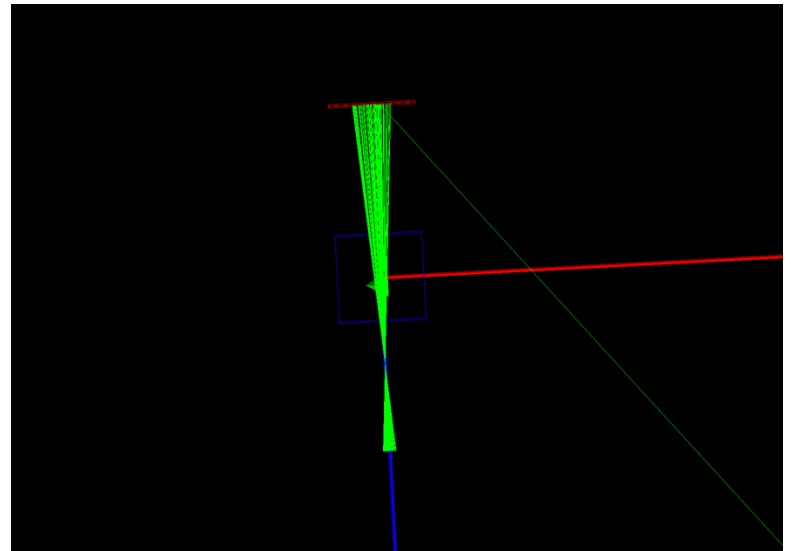
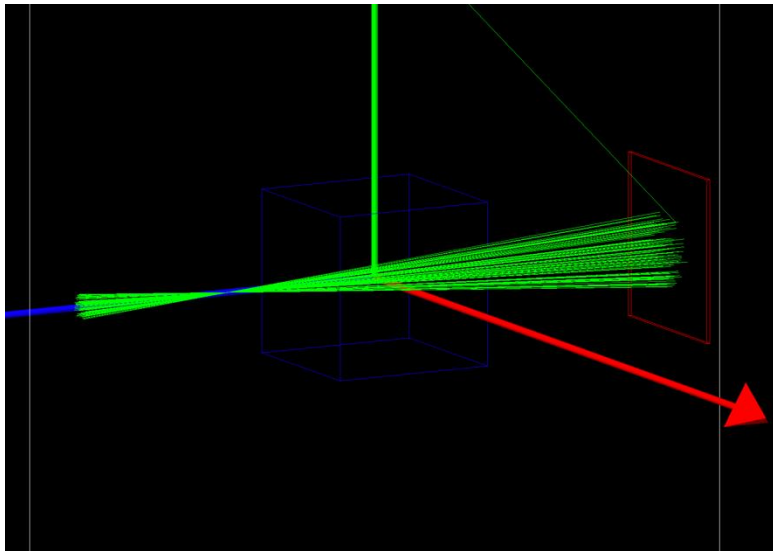
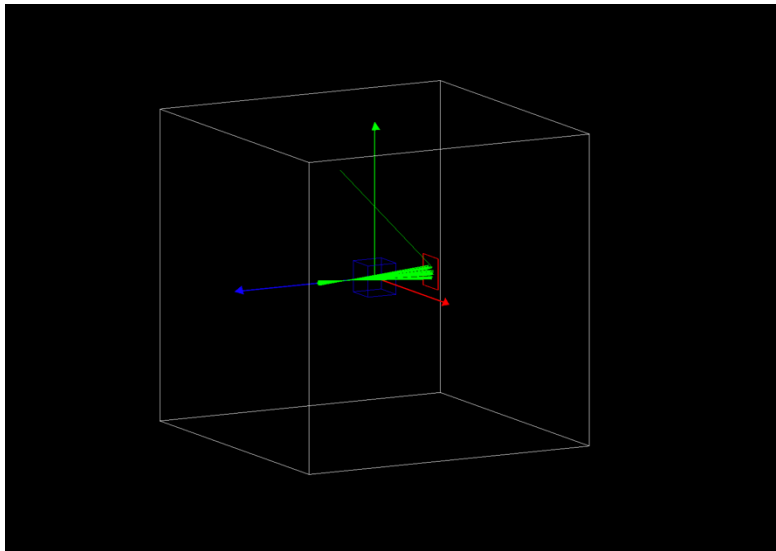
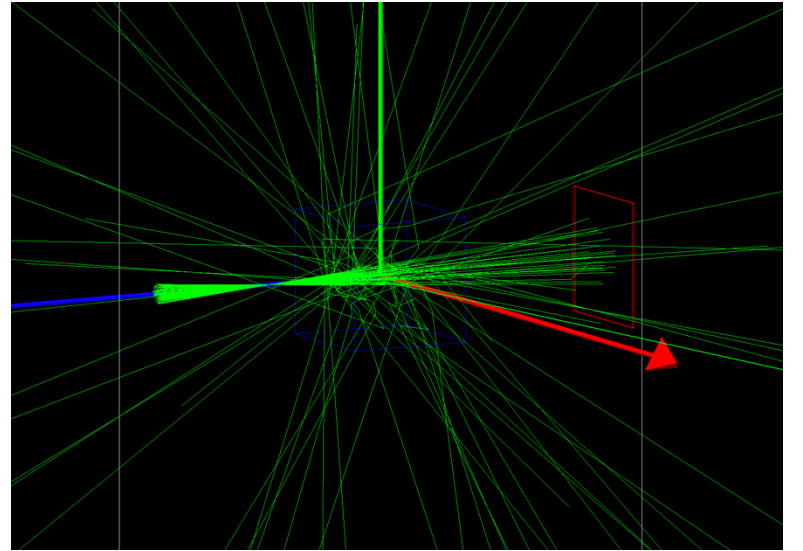
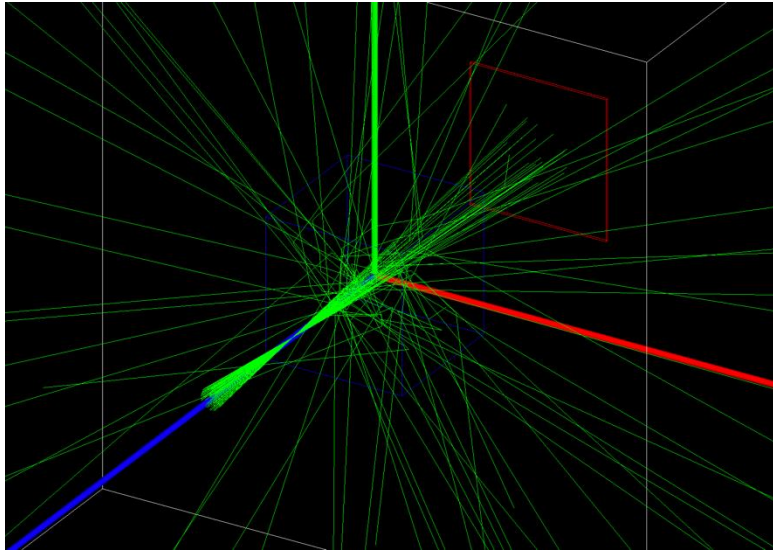
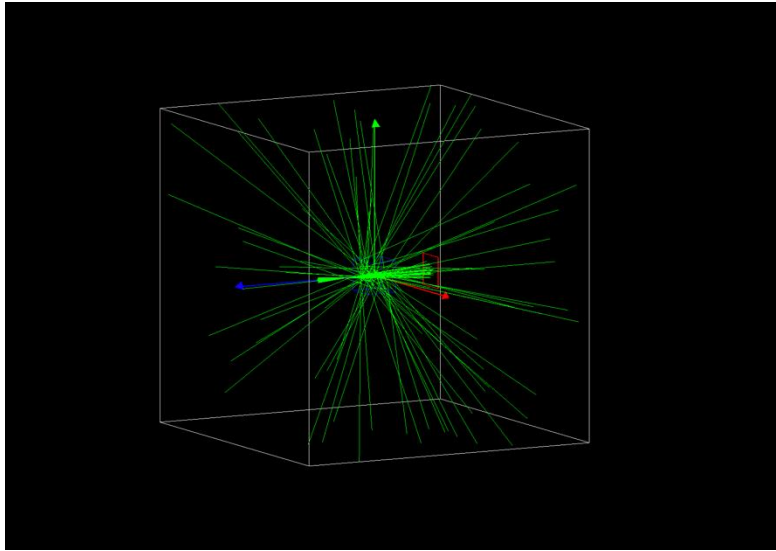
Example1

- Simple source of 140 keV photon
- Through a cubic waterbox
- Fake detector at the opposite, counts photon

Play with

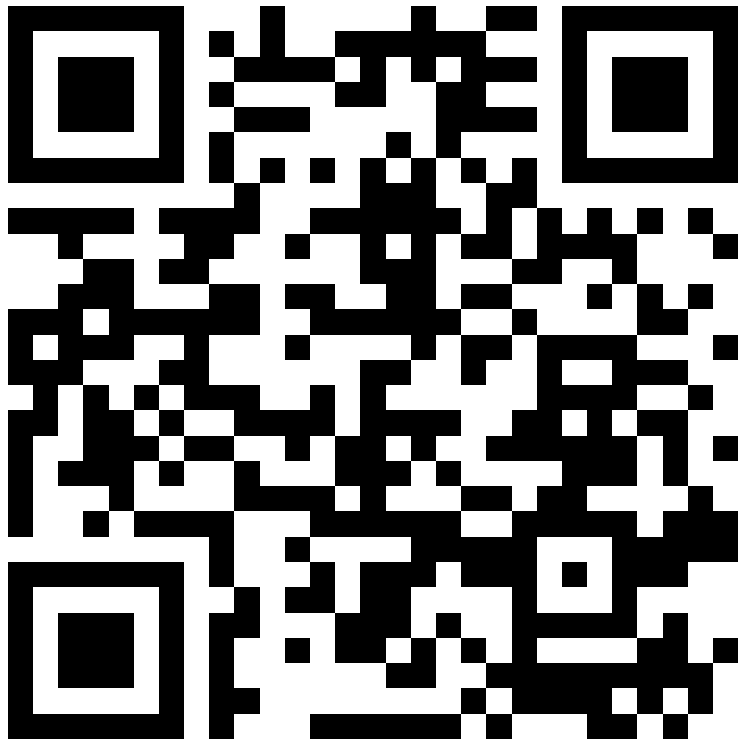
- Box material Water/Lung/Aluminium
- Photon energy
- Beam size
- ...





Do it yourself !

https://gitlab.in2p3.fr/davidsarrut/gate_exercices_2



SARRUT David / gate_exercices_2

G **gate_exercices_2**

Star 0 Fork 0

25 Commits 1 Branch 0 Tags 5.1 MiB Project Storage

change to jupyter notebook
SARRUT David authored 2 minutes ago

fdb1b288

main gate_exercices_2 +

Add README Add LICENSE Add CHANGELOG Add CONTRIBUTING + Enable Auto DevOps Add Add Wiki Configure Integrations

History Find file Edit **Code**

Clone with SSH

git@gitlab.in2p3.fr:davidsarrut

Clone with HTTPS

https://gitlab.in2p3.fr/davidsa

Open in your IDE

Visual Studio Code (SSH)
Visual Studio Code (HTTPS)
IntelliJ IDEA (SSH)
IntelliJ IDEA (HTTPS)

Download source code

zip
tar.gz
tar.bz2
tar

Name	Last commit	
1_starting_demo	change to jupyter notebook	
2_dosimetry	change to jupyter notebook	
3_linac	change to jupyter notebook	
4_radio_pharmaceutical_therapy	change to jupyter notebook	
6_spect	change to jupyter notebook	
7_pet	change to jupyter notebook	
9_cbct	consolidate	
.gitignore	consolidate	
download_data.py	data dl	
exercices_helpers.py	update output path and notebooks	2 weeks ago
test_all.py	consolidate	6 days ago

Exercices

```
$ git clone https://gitlab.in2p3.fr/davidsarrut/gate_exercices_2.git
```

```
$ cd gate_exercices_2
```

```
$ pip install -r requirements.txt
```

```
$ ./download_data.py
```

URLs

- Source code

https://gitlab.in2p3.fr/davidsarrut/gate_exercices_2

- Data

<https://www.creatis.insa-lyon.fr/~dsarrut/dqprm/gate-exercices/2025>

- GATE documentation

<https://opengate-python.readthedocs.io>

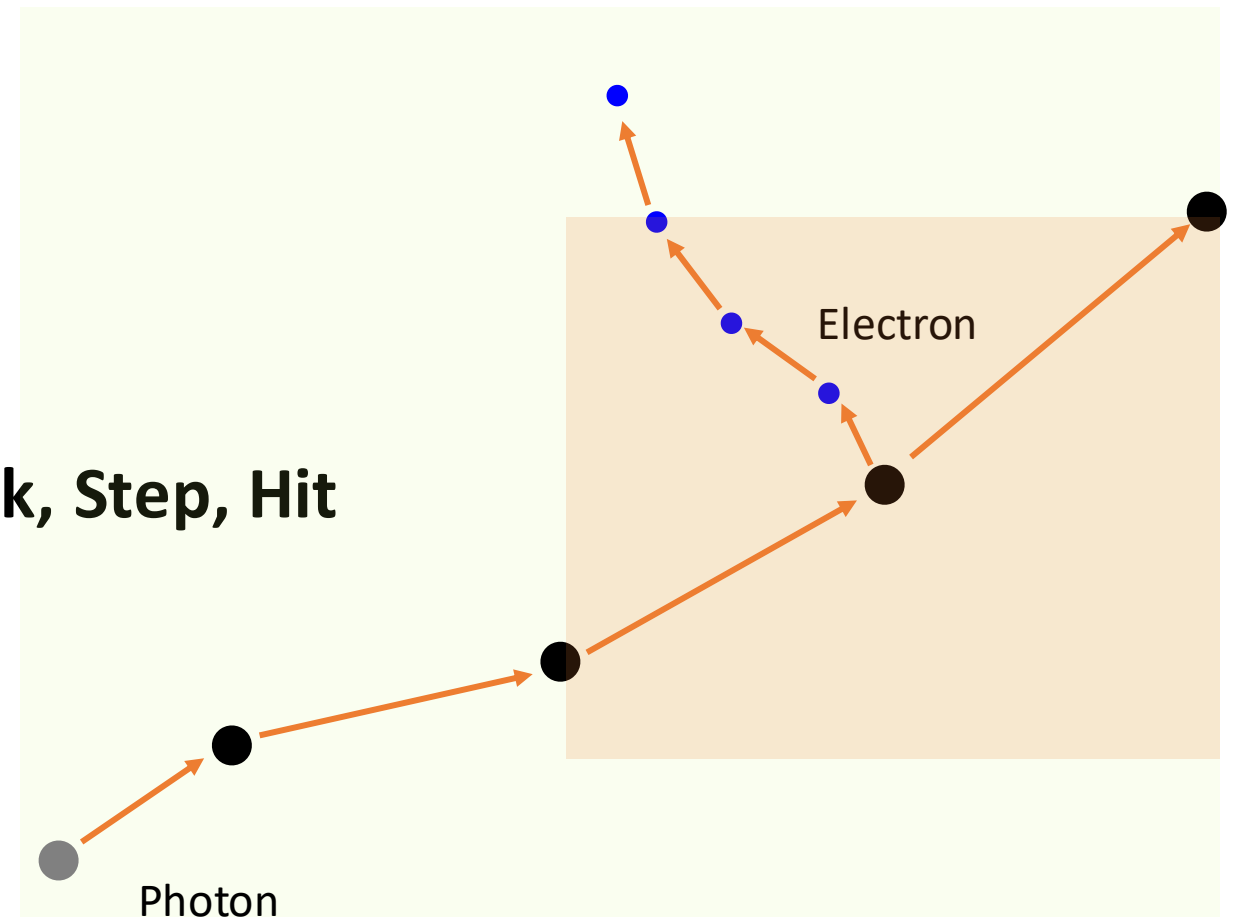
- GATE tests

See *opengate_info*

Structure of a simulation

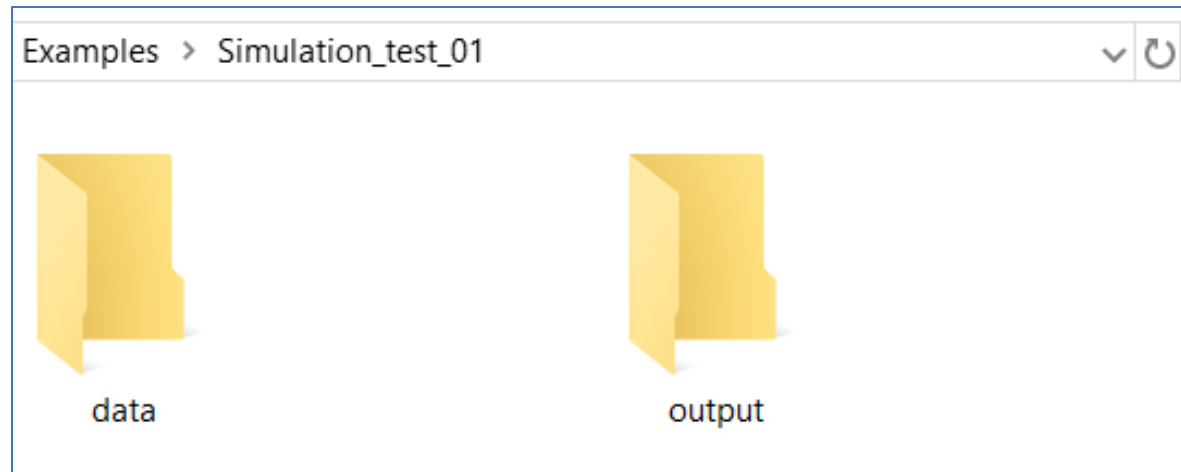
1. Geometry
2. Physics
3. Source
4. Observables (actors)

Vocabulary (G4, Gate): **Event, Track, Step, Hit**



Workflow

- Usage: one simulation = one folder
- Two subfolders
 - `data/` contains data used by simulation (images, material, etc)
 - `output/` where the output will be



Data analysis

- Output of a simulation: data ... now the work begins !
- Data analysis, interpretation
- Need tools
 - Python, matlab, gnuplot, excel, etc ...

- Modules:  python™



Simulation analysis

- We use Jupyter notebook
 - Start the notebook from the terminal: `jupyter notebook`
 - Open web browser, open a XXXX.ipynb file
 - Execute all cell with “shift+Enter”
- Warning: *lab* or *notebook* ?
 - `jupyter lab`: need `%matplotlib widget` at the end of the script
 - `jupyter notebook`: need `%matplotlib notebook` at the end of the script

Notebook

```
$ pip install ipyml  
$ pip install jupyterlab_widgets
```

Warning: *lab* or *notebook* ?

`jupyter lab` needs `%matplotlib widget` at the end of the script

`jupyter notebook` needs `%matplotlib notebook` at the end of the script

Exercices

Each folder is composed of:

- A set of simulation (python scripts)
- A folder with data for the simulation
- A readme file, with exercices
- A notebook for the analysis

Where are the tests ?

```
> opengate_info
Python version  3.11.
Platform       darwin
Site package    /Users/dsarrut/src/envs/opengate_3_11/lib/python3.11/site-packages
Geant4 version  geant4-11-01-patch-01 [MT]
Geant4 MT       True
Geant4 GDML     False
Geant4 date     (10-February-2023)
Geant4 data     /Users/dsarrut/src/gate/opengate/core/opengate_core/geant4_data
ITK version     5.2.1
GATE version    10.0b7
GATE folder     /Users/dsarrut/src/gate/opengate/opengate
GATE git sha    d8e170268cc8603a4a8517b98d464371d7de368c
GATE date       2024-01-26 17:57:
                /Users/dsarrut/src/gate/opengate/opengate/tests/src
```