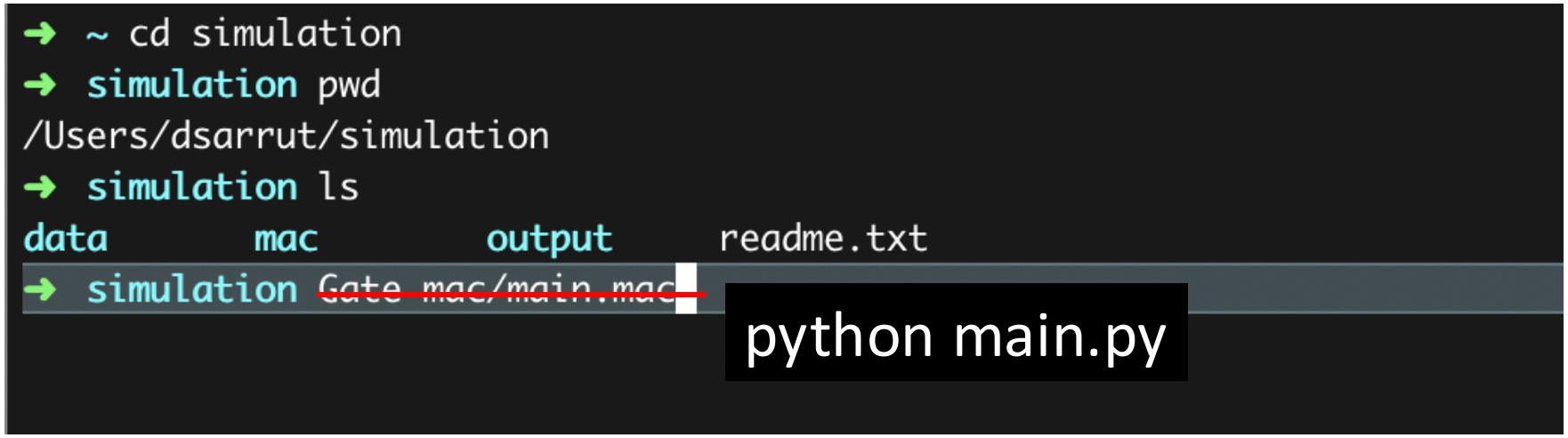
Outline

- Working with a terminal
- Demo
- Install exercices
- Structure of an exercice

Working in a terminal (linux, mac, win)

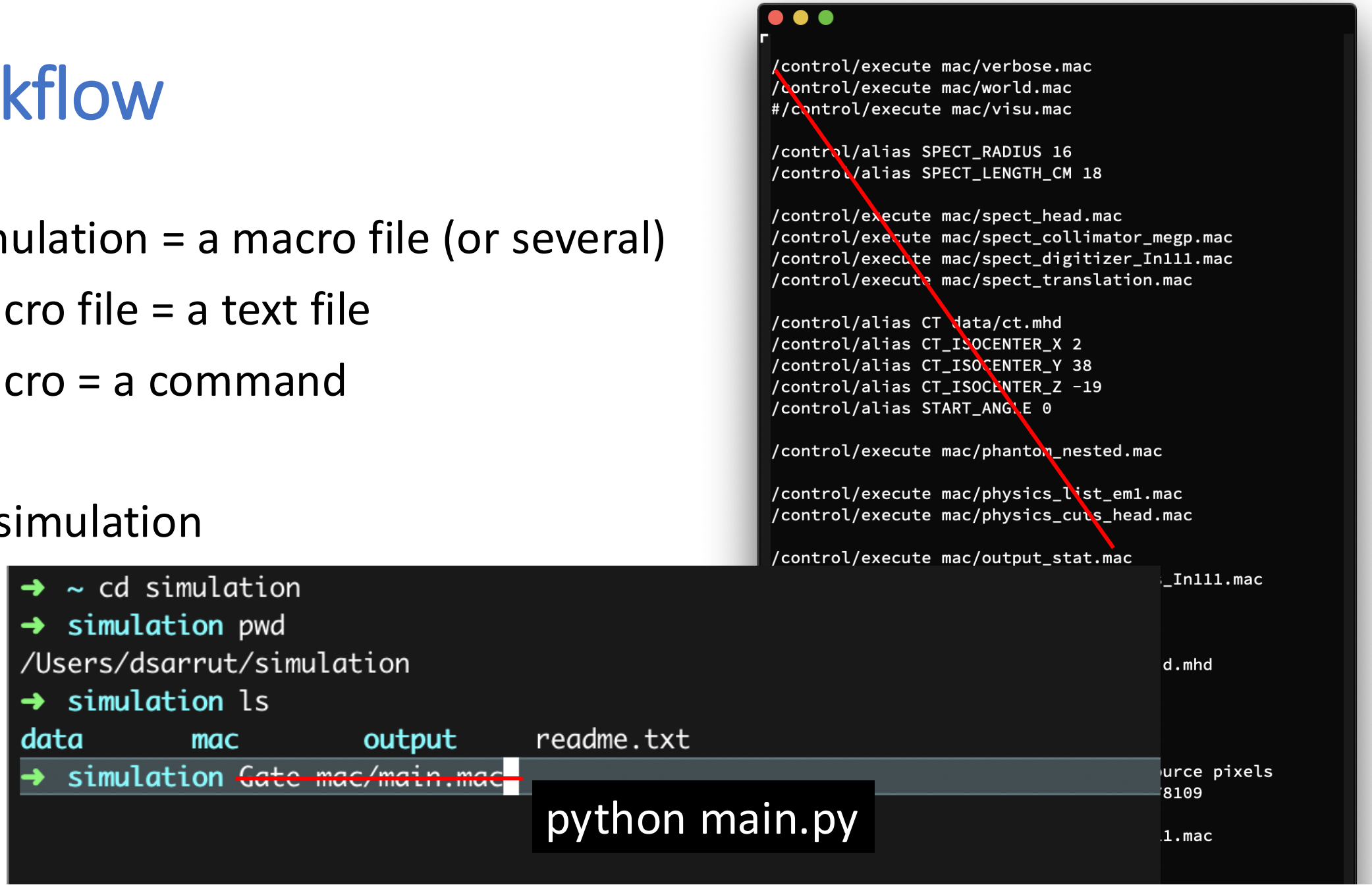
- See doc “linux command reference”
- “*Interpréteur de commandes*”



Workflow

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

Run a simulation



Workflow

python

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

Run a simulation

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"
:
```

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

python main.py

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 exercice

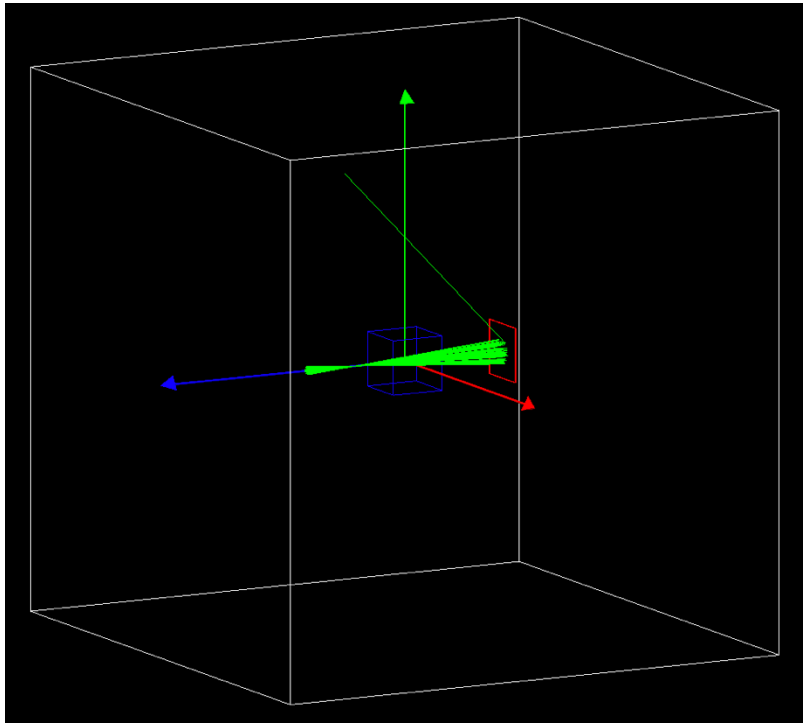
… live …

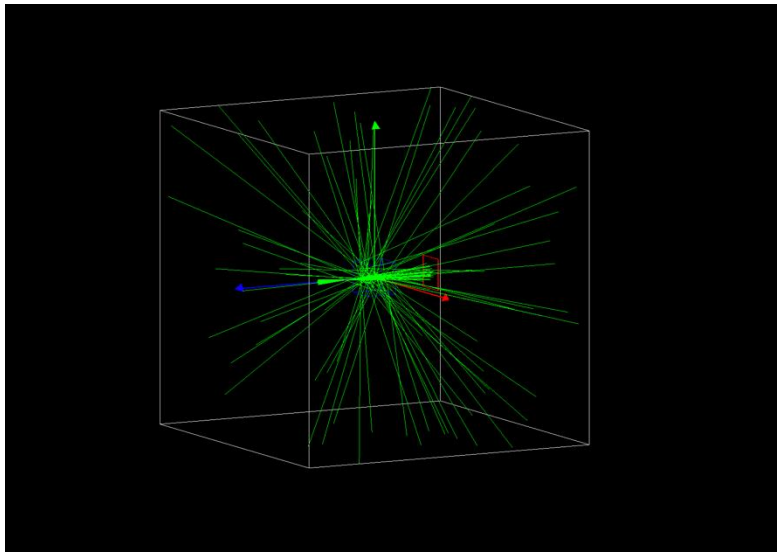
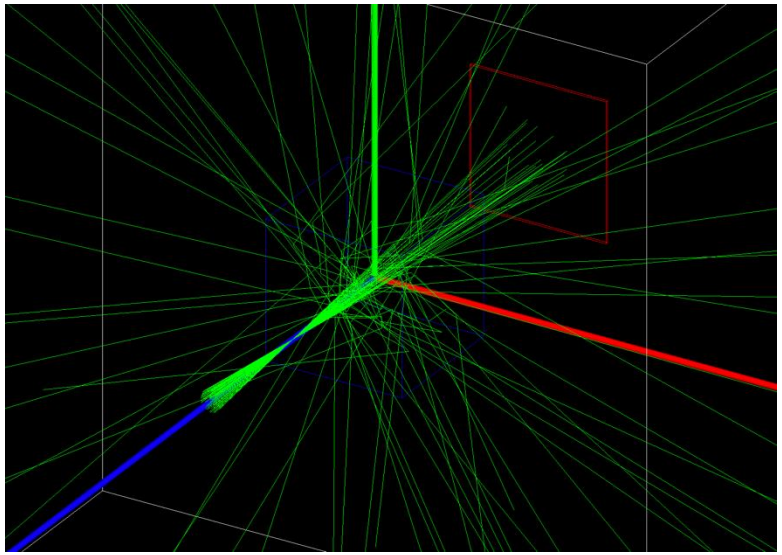
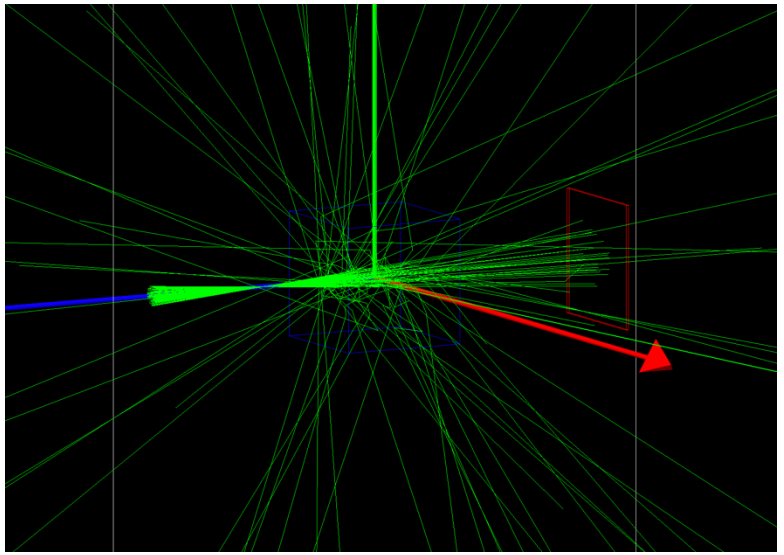
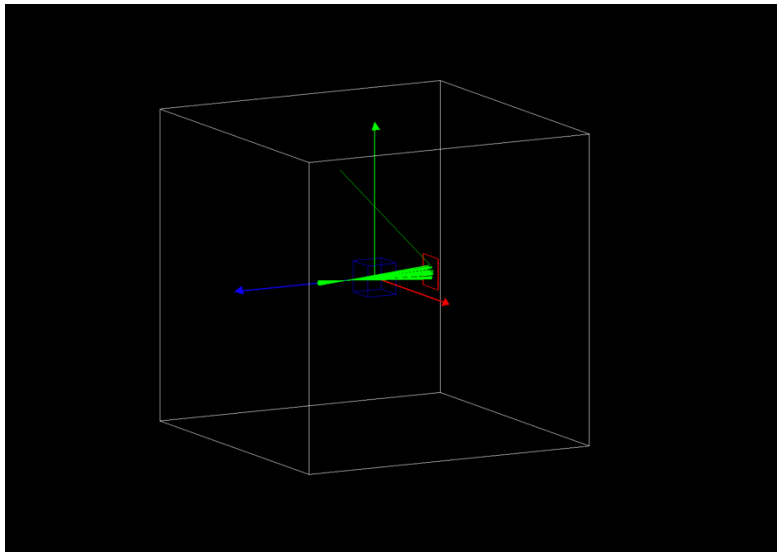
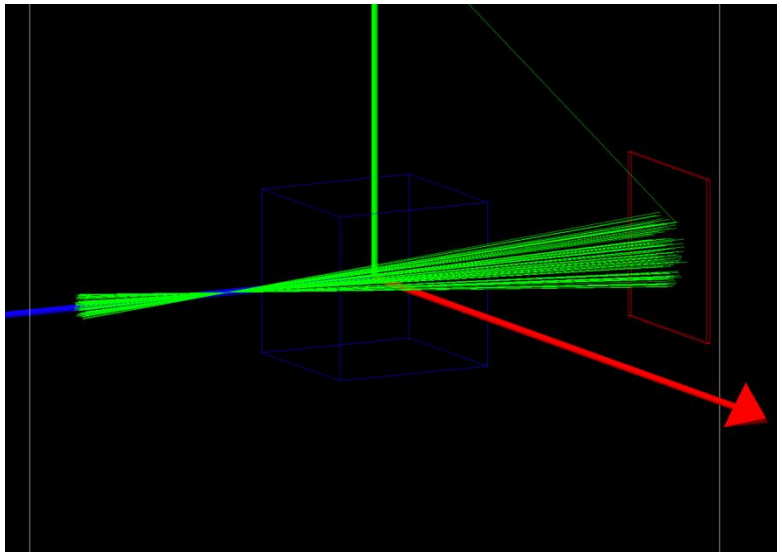
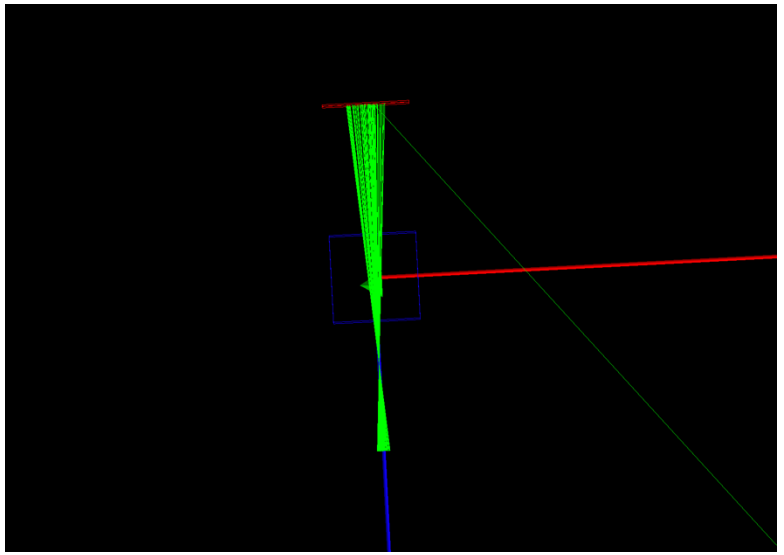
Example1

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

Play with

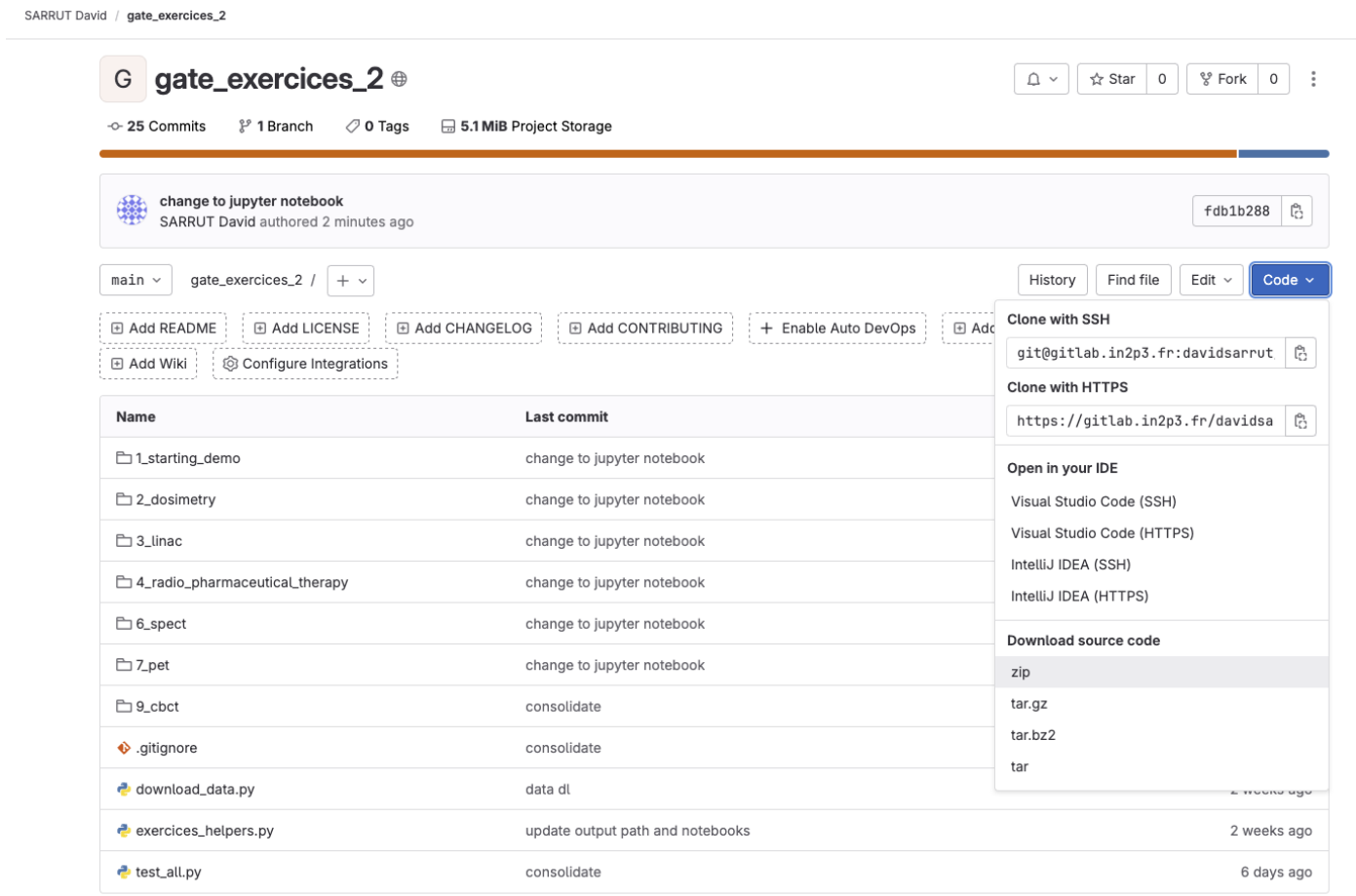
- Box material Water/Lung/Aluminium
- Photon energy
- Beam size
- …



Do it yourself !

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



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

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

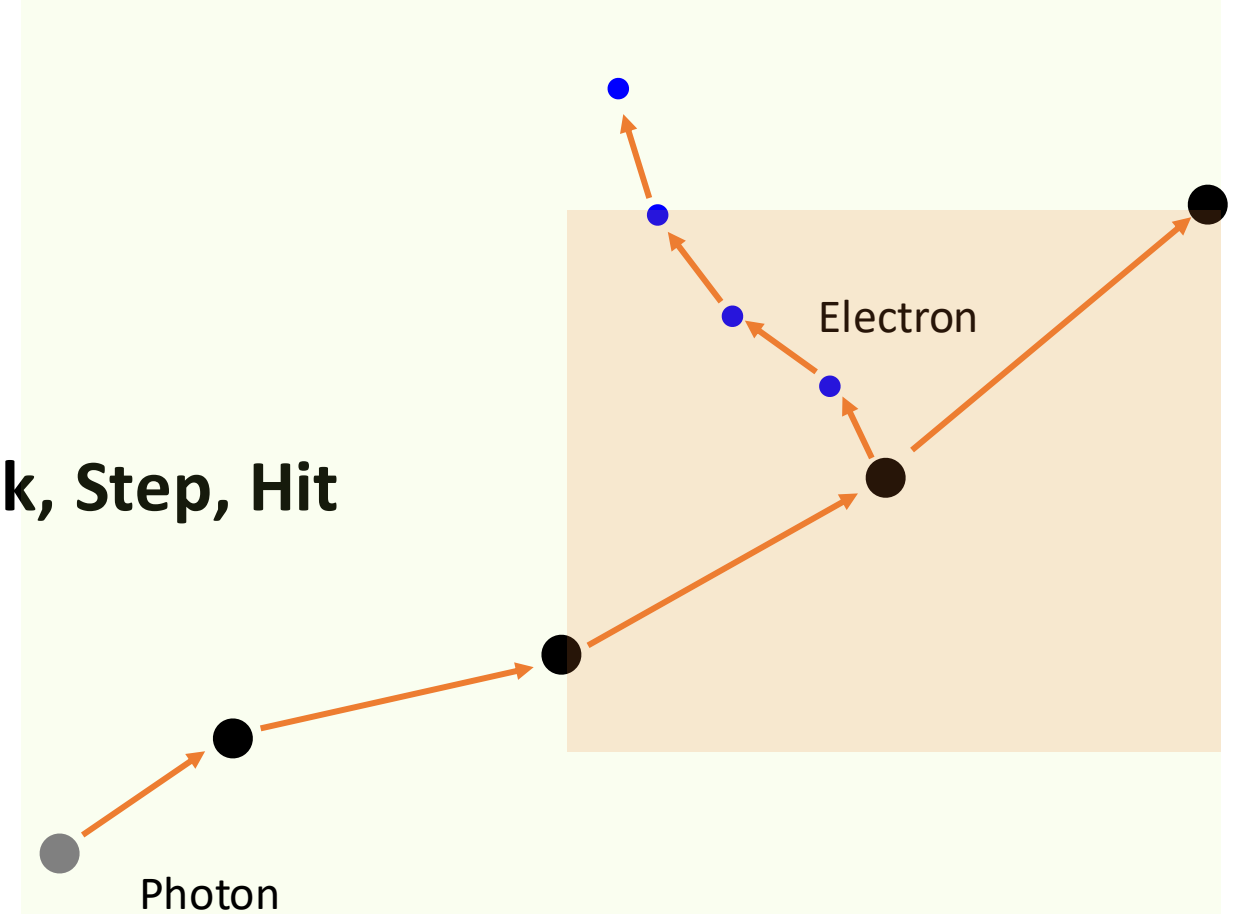
- GATE tests

  See *opengate_info*

Structure of a simulation

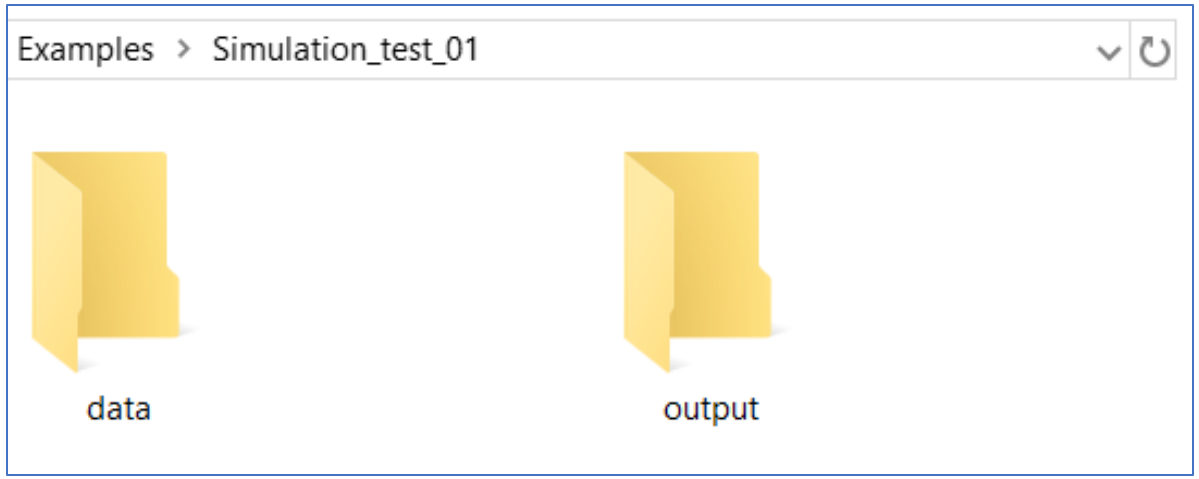
1. Geometry
2. Physics
3. Source
4. Observables (actors)

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:

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 ipympl
$ 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**