SERESSA 2022

5th to 9th of December at CERN, Geneva

Introduction to G4SEE:

a toolkit for simulating radiation effects in electronics

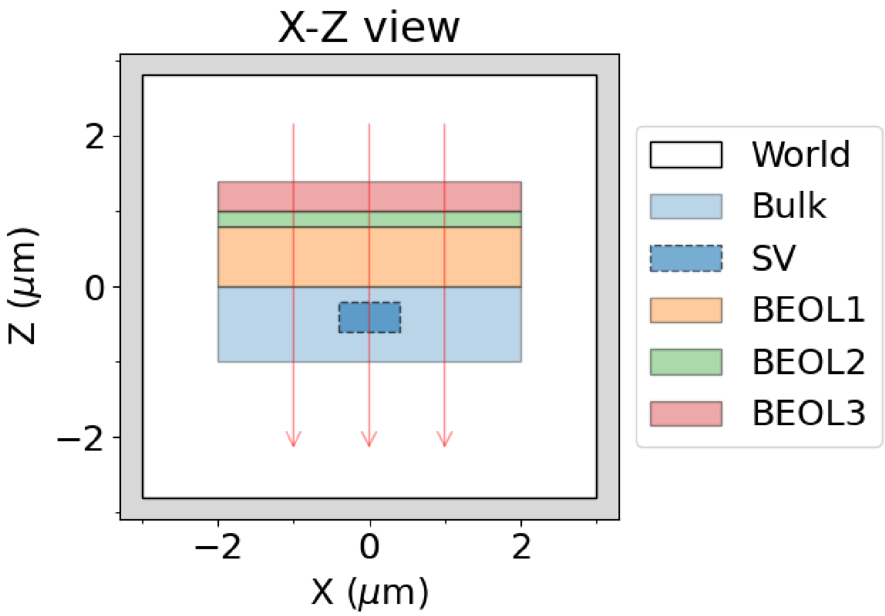
Dávid Lucsányi, CERN

Agenda

- Introduction to G4SEE toolkit:
  - An open-source toolkit
  - Capabilities
  - User support & community
- Hands-on tutorial #1:
  - The simulation case: Proton induced SEUs in ISSI SRAM
  - 1.1) General, Input/Output
  - 1.2) Geometry
  - 1.3) Primary particles, Scoring
  - 1.4) Scoring, SEE cross-section
- To be continued tomorrow...

Introduction – An open-source toolkit

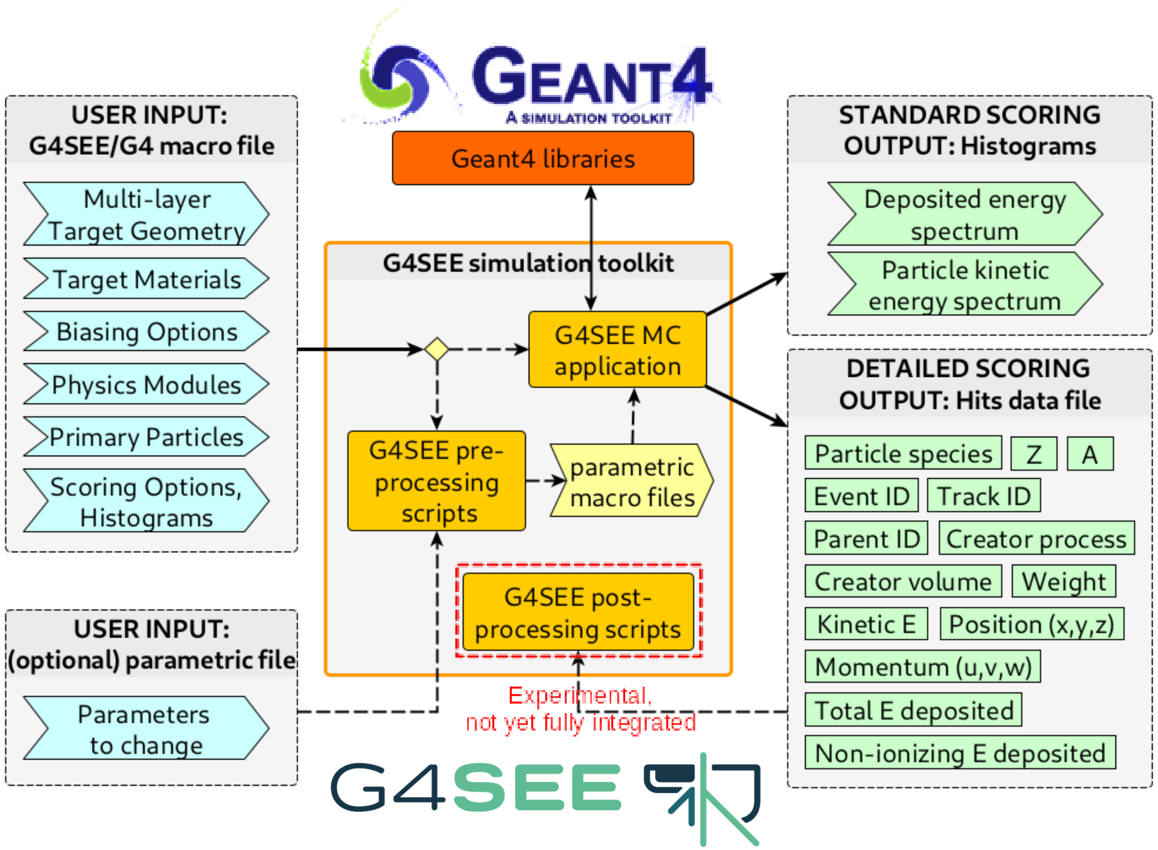
- G4SEE is a Geant4-based Monte Carlo Single Event Effect (SEE) simulation toolkit
- Free and open-source, available for the whole radiation effects community for a wide variety of use cases
- It is being developed in CERN Radiation To Electronics (R2E) project, but developers, contributors and beta testers outside CERN are also welcome!



Multi-layer, micro-metric geometry used in a G4SEE simulation to obtain energy deposition in Sensitive Volume (SV)

Introduction – Capabilities

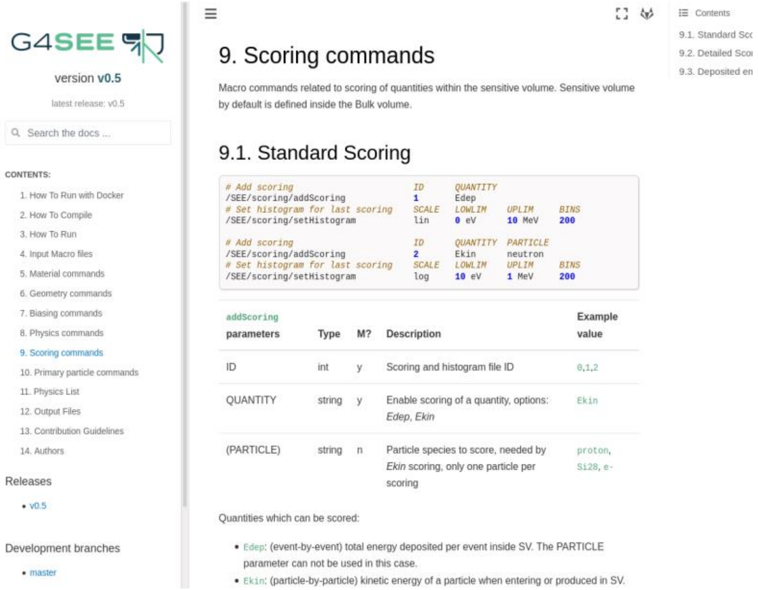
- Extracting low-level information relevant for SEEs, event-by-event and particle-by-particle, based on the needs of users
- Direct and indirect energy deposition scoring in a micro-metric, user-defined sensitive volume ⇒ SEE cross-section (or rate) estimation
- Primary motivation and use cases so far were neutron and proton induced SEEs (relevant at CERN environments)



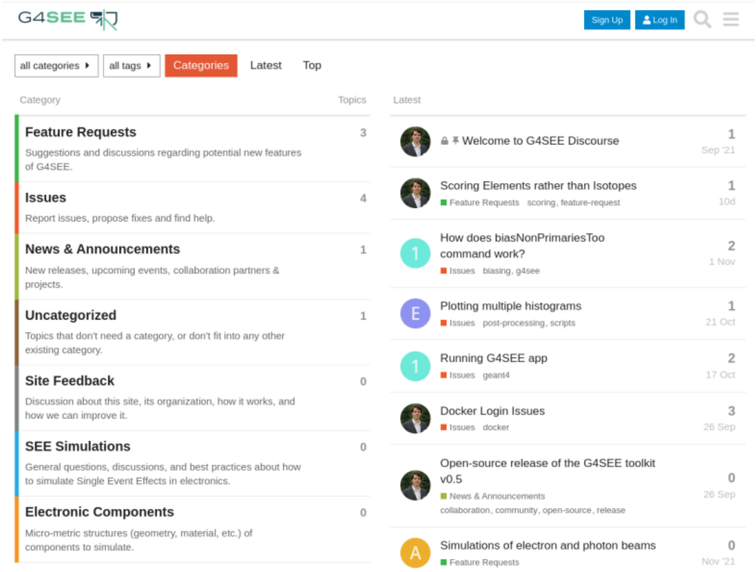
High-level architecture of the G4SSE toolkit with user inputs and outputs (ASCII files)

Introduction – User support & community

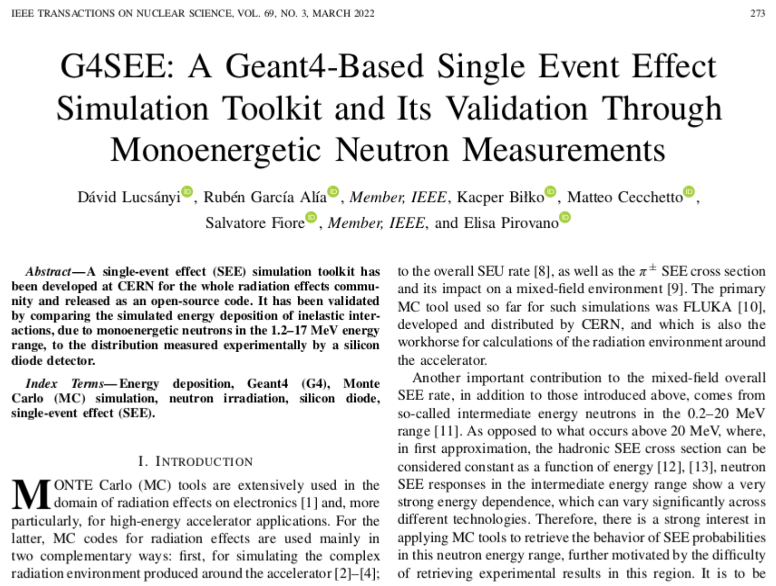
G4SEE documentation: g4see-docs.web.cern.ch



G4SEE community forum: g4see-forum.web.cern.ch



G4SEE paper in IEEE TNS: doi.org/10.1109/TNS.2022.3149989


IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 69, NO. 3, MARCH 2022 273

G4SEE: A Geant4-Based Single Event Effect Simulation Toolkit and Its Validation Through Monoenergetic Neutron Measurements

Dávid Lucsányi, Rubén García Alía, *Member, IEEE*, Kacper Biłko, Matteo Cecchetto, Salvatore Fiore, *Member, IEEE*, and Elisa Pirovano

***Abstract*—A single-event effect (SEE) simulation toolkit has been developed at CERN for the whole radiation effects community and released as an open-source code. It has been validated by comparing the simulated energy deposition of inelastic interactions, due to monoenergetic neutrons in the 1.2–17 MeV energy range, to the distribution measured experimentally by a silicon diode detector.**

***Index Terms*—Energy deposition, Geant4 (G4), Monte Carlo (MC) simulation, neutron irradiation, silicon diode, single-event effect (SEE).**

I. Introduction

MONTE Carlo (MC) tools are extensively used in the domain of radiation effects on electronics [1] and, more particularly, for high-energy accelerator applications. For the latter, MC codes for radiation effects are used mainly in two complementary ways: first, for simulating the complex radiation environment produced around the accelerator [2]–[4]; to the overall SEU rate [8], as well as the $\pi^{\pm}$ SEE cross section and its impact on a mixed-field environment [9]. The primary MC tool used so far for such simulations was FLUKA [10], developed and distributed by CERN, and which is also the workhorse for calculations of the radiation environment around the accelerator.

Another important contribution to the mixed-field overall SEE rate, in addition to those introduced above, comes from so-called intermediate energy neutrons in the 0.2–20 MeV range [11]. As opposed to what occurs above 20 MeV, where, in first approximation, the hadronic SEE cross section can be considered constant as a function of energy [12], [13], neutron SEE responses in the intermediate energy range show a very strong energy dependence, which can vary significantly across different technologies. Therefore, there is a strong interest in applying MC tools to retrieve the behavior of SEE probabilities in this neutron energy range, further motivated by the difficulty of retrieving experimental results in this region. It is to be

G4SEE website: g4see.web.cern.ch

Contact: g4see.toolkit@cern.ch

Introduction – Get the toolkit!

- See lecture preparation slides on SERESSA Indico page
- Options to run G4SEE toolkit on your computer:

  A) Cloning CERN GitLab repos & building from source *(not recommended now)*

  — OR —

  B) Using Docker ***(recommended, see lecture preparation slides)***

  - After Docker installation, pull the latest, **G4SEE v0.5.1** Docker image
  - Run a Docker container based on G4SEE v0.5.1 image (with shared folder)

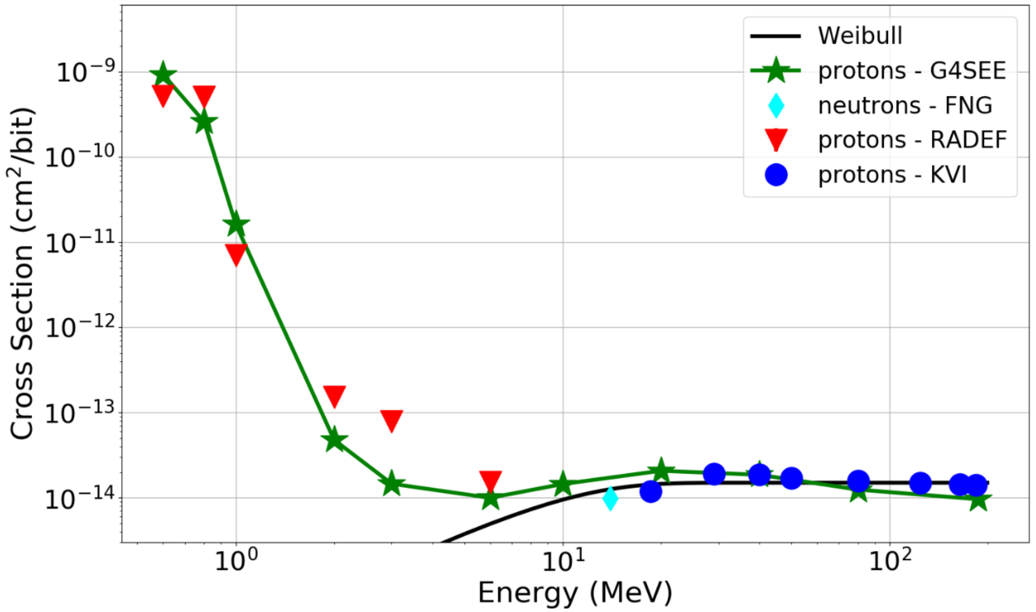
CLI commands to run on host computer to start G4SEE Docker container

```
$ docker pull gitlab-registry.cern.ch/g4see/g4see:v0.5.1_G4-11.0.3
$ export DISPLAY=:0.0
$ xhost +local:docker
$ docker run -it -h g4see -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix \
      -v /host/path/to/shared_folder:/home \
      gitlab-registry.cern.ch/g4see/g4see:v0.5.1_G4-11.0.3
root@g4see:/home# cp –r $G4SEE_BUILD/examples/tutorial_1 /home
```

Once inside Docker container, copy tutorials to /home folder!

Tutorial #1 – ISSI SRAM & protons [1]

- DUT: 65-nm ISSI SRAM memory
- Effects: Single Event Upsets
- $Q_{critical}$ = 0.96 fC (≈ 21.6 keV)
- Facilities: RADEF, PARTREC
- Primaries: beam of protons
- Physics: direct ionization
- Scoring: standard ($E_{kinetic}$ and $E_{deposited}$)
- [Tutorial files in GitLab]



Cross-section of proton induced SEUs in 65nm ISSI SRAM as function of proton beam energy [1]

Reference (open-access)

[1] Andrea Coronetti *et al.*, "Proton direct ionization upsets at tens of MeV", in *IEEE TNS*, 2022, (link)

Tutorial 1.1 – General, Input/Output

- To run G4SEE, a G4-style macro (UI) file is needed as input, with general G4 and G4SEE-specific UI commands
- Open the *tutorial_1-1.mac* input macro file:

Optional commands

```
#################
### General G4 commands

/run/printProgress                 1
/run/numberOfThreads               4
/control/cout/ignoreThreadsExcept  0

/random/setSeeds                   1234, 4321

/tracking/verbose                  1
```

General Geant4 (G4) macro commands can be also used

Multi-threading related commands

Seeds for random number generator (if you want to reproduce exact same results multiple times)

Increasing verbosity (from 0 to 1), use only for few events when testing your macro!

Mandatory commands

```
#################
### Geometry

# BULK COMMAND         MATERIAL        WIDTH unit   THICK unit   BIAS
/SEE/geometry/Bulk     VACUUM          10 um        10 um        false
# SV COMMAND           POSITION unit   WIDTH unit   THICK unit   BIAS
/SEE/geometry/SV       0 0 -4.5 um     1 1 um       1 um         false

#################
### Initialize
/run/initialize

#################
### Run
/run/beamOn                        100
```

G4SEE-specific commands ('/SEE/...') to define geometry (see Tutorial 1.2)

These are macro (UI) commands with multiple parameters in a single line
One command = one line
Comments start with # character

Initialization of a G4 simulation

Last command: start N=100 primary particles (default: "geantino" virtual particles)

Tutorial files in GitLab

Tutorial 1.1 – General, Input/Output

- Run a simulation with this macro file, then check the output files of G4SEE:

Commands in G4SEE Docker container (CLI)

```
:/home# cd tutorial_1/ && mkdir output_1 && cd output_1/
:/home/tutorial_1/output_1# g4see ../tutorial_1-1.mac > stdout.log
:/home/tutorial_1/output_1# ls
```

* Optional redirection to file

```
root@g4see:/home/tutorial_1/output_1# ls
g4see.out  n_event_t0.out  n_event_t1.out  n_event_t2.out  n_event_t3.out  stdout.log
root@g4see:/home/tutorial_1/output_1#
```

- **G4/G4SEE stdout** (redirected to **stdout.log**):
  - Material information
  - List of all particles and their processes
  - List of all physics models enabled
  - Production thresholds (a.k.a. Range cuts)
  - [optional] Steps of each particle track (a.k.a. G4Track info)
- **g4see.out** – useful info about simulation run, like G4 and G4SEE versions, running time, random seeds, number of primary particles, macro file
- **n_event_t<j>.out** – number of events run by a specific thread/job
- No scoring was added yet, so no actual data has been scored and written to file (see Tutorial 1.3)

Tutorial files in GitLab

Tutorial 1.2 – Geometry

- Let's define a simplified geometry for a target SRAM cell (Bulk, SV and BEOL volumes)!
- Open the *tutorial_1-2.mac* input macro file:

Silicon is defined as the material of Bulk vol.

Dimensions (width and thickness) of a volume

Enable/Disable biasing for a given volume

Name of given Back End Of Line (BEOL) layer

```
##################
### Geometry

# BULK COMMAND                          MATERIAL            WIDTH unit     THICK unit    BIAS     
/SEE/geometry/Bulk                      G4_Si               6 um           1 um          false    
# SV COMMAND                            POSITION unit       WIDTH unit     THICK unit    BIAS     
/SEE/geometry/SV                        0 0 0 nm            310 310 nm     310 nm        false    
# BEOL COMMAND                          MATERIAL            WIDTH unit     THICK unit    BIAS     NAME
/SEE/geometry/BEOL/addLayer             G4_SILICON_DIOXIDE  6 um           6 um          false    Oxide
```

(X,Y,Z) position of a SV (scoring) volume inside the Bulk (relative to Bulk cube's top center point)

SV can by asymmetric in size along X and Y axes

Any predefined material from G4 material database can be used here
Users can also define custom materials, like elements, compounds and mixtures

One can add arbitrary number of BEOL layers on top of each other, but only a single Bulk and SV for now (NEW FEATURE SOON: arbitrary number of SVs, and nested SVs!)

Tutorial 1.2 – Geometry

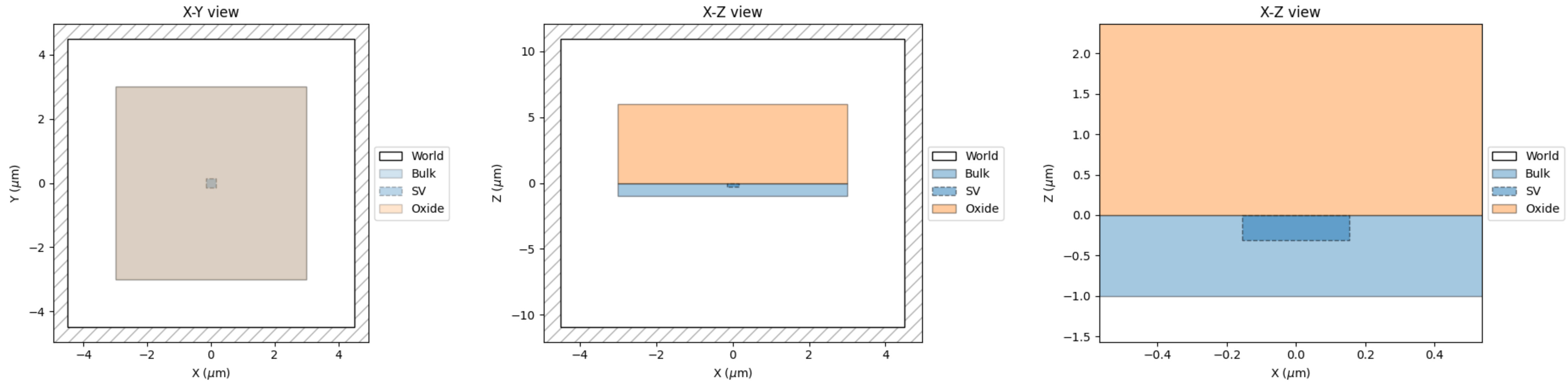
- Run a simulation with this macro file, and check changes in stdout (materials and particle tracks)!

Commands in G4SEE Docker container (CLI)

```
:/home# cd tutorial_1/ && mkdir output_2 && cd output_2/
:/home/tutorial_1/output_2# g4see ../tutorial_1-2.mac > stdout.log
:/home/tutorial_1/output_2# less stdout.log
:/home/tutorial_1/output_2# python $G4SEE_BUILD/scripts/g4see.py view ../tutorial_1-2.mac
```

* press q to exit 'less'

- Visualize geometry using G4SEE 'view' script:



Top (X-Y), side (X-Z) and zoomed-in (X-Z) view of simulated ISSI SRAM memory cell, proton beam will have –Z direction

Tutorial 1.3 – Primary particles, Scoring

G4SEE

- We can also define a simple monoenergetic proton beam (or any arbitrary particle source distribution)
- Add scoring commands to record and write to file binned kinetic energy values of protons
- Open the *tutorial_1-3.mac* input macro file:

```
#################
### Primaries

/gps/particle                     proton
/gps/ene/mono                     1 MeV
/gps/ene/sigma                    25 keV
/gps/direction                    0 0 -1
/gps/pos/type                     Plane
/gps/pos/shape                    Rectangle
/gps/pos/centre                   0 0 10 um
/gps/pos/halfx                    0.5 um
/gps/pos/halfy                    0.5 um

#################
### Scoring (standard)

/SEE/scoring/addScoring           0       Ekin    proton
/SEE/scoring/setHistogram         lin     0 MeV   2 MeV    200
```

Any G4 General Particle Source (GPS) command can be used in a G4SEE macro file, there is no limit in source definition

Particle species

Mean and std. deviation of energy

Particle direction (monodirectional, -Z direction)

Shape and type of source (point/surface/volume)

Position of source's center

Half dimensions of source plane
(now it's only 1×1 μm^2 plane to save time)

Scoring definitions

Kinetic energy scoring for protons (particle-by-particle scoring)
file name id: 0 --> 'Ekin_0_hist_tN.out'

Binning of histogram for previously defined scoring
(linear binning from 0 to 2 MeV in 200 bins, other option is log binning)

Energy and angular distributions can be user-defined functions or read directly from ascii file

Two types of scoring mechanisms implemented:
- Standard: Ekin or Edep;
- Detailed (see later)

Tutorial 1.3 – Primary particles, Scoring

- Run a simulation with a proton beam, while scoring energy of each proton:

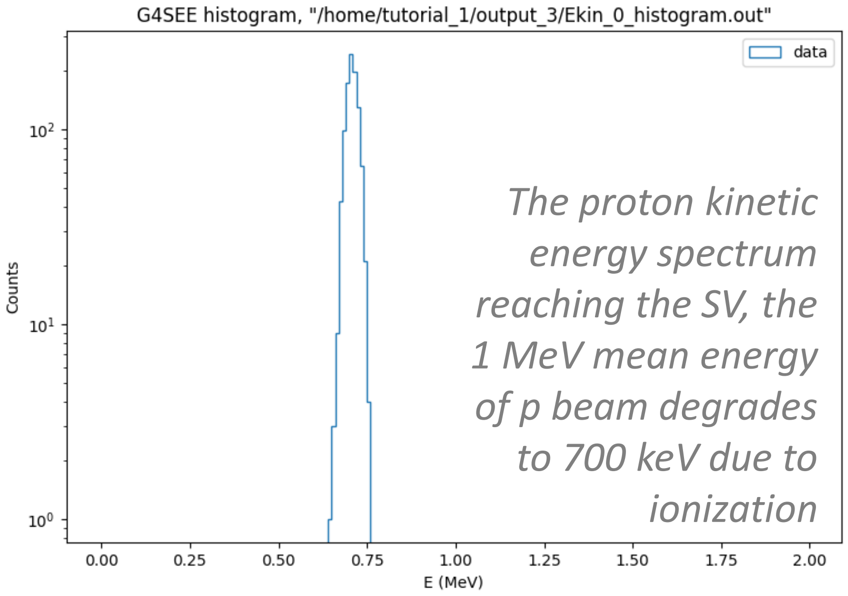
Commands in G4SEE Docker container (CLI)

```
:/home# cd tutorial_1/ && mkdir output_3 && cd output_3/
:/home/tutorial_1/output_3# g4see ../tutorial_1-3.mac > stdout.log
:/home/tutorial_1/output_3# mergeHistograms .
:/home/tutorial_1/output_3# ls
:/home/tutorial_1/output_3# python $G4SEE_BUILD/scripts/g4see.py plot Ekin_0_histogram.out
```

- Merge histogram files of the multiple threads into a single histogram!
- Only the kinetic energy of protons entering (or produced) in the SV is scored
- Plot histogram using G4SEE 'plot' script

Merged histogram of scored proton kinetic energy values, during merging std. deviation of each bin is also calculated

Ekin	Counts	StdDev(corr)
underflow	0	0
overflow	0	0
0.000e+00	0	0
...	...	...
6.300e-01	0	0
6.400e-01	0	0
6.500e-01	1.000e+00	5.000e-01
6.600e-01	6.000e+00	1.000e+00
6.700e-01	4.100e+01	5.315e+00
6.800e-01	8.600e+01	4.359e+00
6.900e-01	1.710e+02	4.500e+00
7.000e-01	2.450e+02	1.147e+01
7.100e-01	1.960e+02	6.325e+00
7.200e-01	1.190e+02	2.986e+00
7.300e-01	6.300e+01	1.258e+00
7.400e-01	1.900e+01	2.872e+00
7.500e-01	2.000e+00	5.774e-01
7.600e-01	0	0
7.700e-01	0	0



The proton kinetic energy spectrum reaching the SV, the 1 MeV mean energy of p beam degrades to 700 keV due to ionization

Tutorial 1.4 – Scoring, SEE cross-section

- Let's add energy deposition scoring as well, similarly to kinetic energy scoring!
- Open the *tutorial_1-4.mac* input macro file:

```
/SEE/scoring/addScoring            0      Edep
/SEE/scoring/setHistogram          log    100 eV   1 MeV     300

/SEE/scoring/dumpHistogramsAfter   10000

#################
### Run
/run/beamOn                        200000
```

Deposited energy scoring per event for all particles (event-by-event scoring)
file name id: 0 --> 'Edep_0_hist_tN.out'

Logarithmic binning with 300 bins

After how many simulated events we want to dump histograms to file

By increasing the primary particle number, we can get better statistics

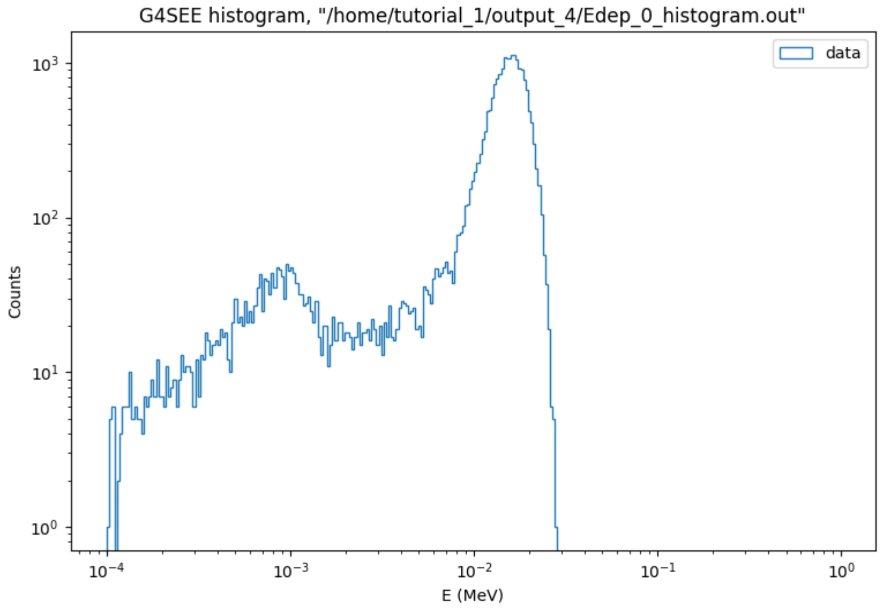
- Run the bit longer simulation (1-2 min), then merge histograms and create Edep histogram plot:

Commands in G4SEE Docker container (CLI)

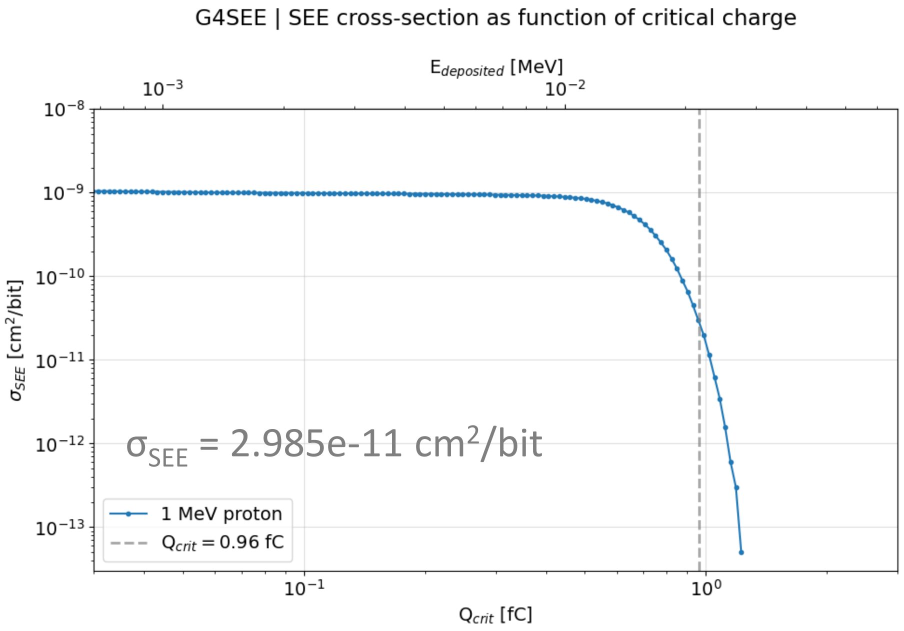
```
:/home# cd tutorial_1/ && mkdir output_4 && cd output_4/
:/home/tutorial_1/output_4# g4see ../tutorial_1-4.mac > stdout.log
:/home/tutorial_1/output_4# mergeHistograms .
:/home/tutorial_1/output_4# python $G4SEE_BUILD/scripts/g4see.py plot Edep_0_histogram.out
:/home/tutorial_1/output_4# python $G4SEE_BUILD/scripts/g4see.py see-xs ../config_1-4.yaml
```

Tutorial 1.4 – Scoring, SEE cross-section

- (new feature) By running the 'see-xs' script, one can obtain the SEE cross-sections in function of critical charge based on the inverse cumulative sum of energy deposition distribution --> SEE cross-section (and rate) estimation if we know $Q_{critical}$
- (new feature) The script 'see-xs' needs a YAML config file with input parameters and plotting options



Event-by-event energy deposition distribution, contribution of δ-electrons produced outside SV at lower E_{dep} values



SEE cross-section estimation at 1 MeV proton beam energy (assuming Q_{crit} = 0.96 fC) based on MC simulated E_{dep} histogram

```yaml
# ######################################## #
# YAML config file for Tutorial 1.4        #
# ######################################## #

see-xs:

  # Input parameters
  conversion_factor:      0.022469    # MeV/fC
  critical_charge:        0.96        # fC

  # List of Edep histograms
  data:
    - path:               Edep_0_histogram.out
      label:              '1 MeV proton'
      primary_number:     2e+5
      beam_area:          1e-8     # cm2 (1x1 um2)

  # XS plot config
  xs_plot:
    xlim:                 [3e-2, 3e+0]
    ylim:                 [None, 1e-8]
    figsize:              [10, 7]
```

YAML config file

To be continued tomorrow...

Thank you for your attention!

Questions?

If you have further questions, need more help, or you are interested in contributing to the G4SEE toolkit, please let us know!

g4see.toolkit@cern.ch