

Injection Parameter Study with musrSim: Update

Siew Yan Hoh, Yusuke Takeuchi, Yuzhi Shang

5th April 2024

muEDM Collaboration Meeting



PAUL SCHERRER INSTITUT

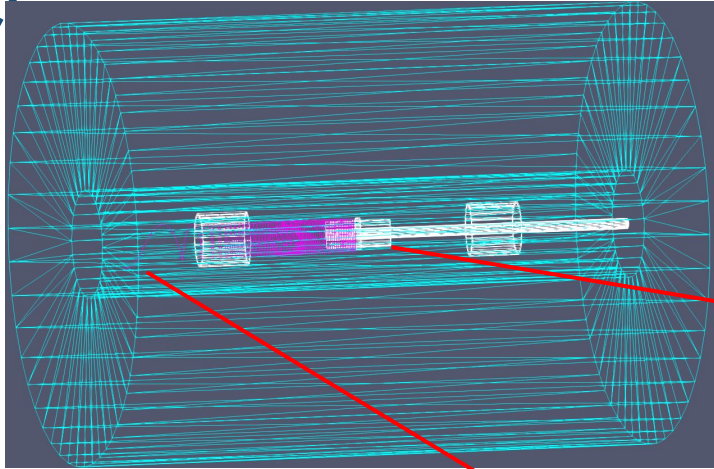


李政道研究所

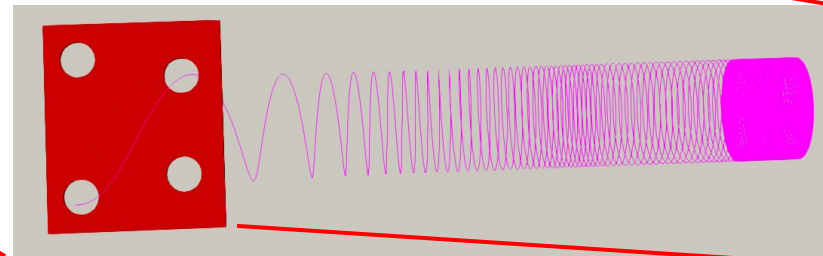
TSUNG-DAO LEE INSTITUTE



Overview of Group Responsibility

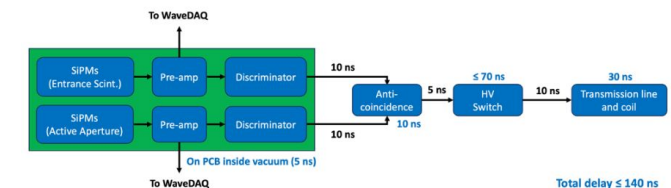
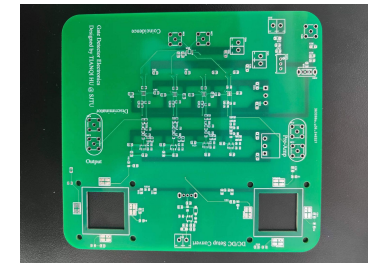


Beam Injection and storage study
(Siew Yan & Yuzhi)



Entrance detectors design and
detector response study
(Guan Ming)

Fast electronics for entrance trigger
and pulsed magnetic kicker
(Tianqi)

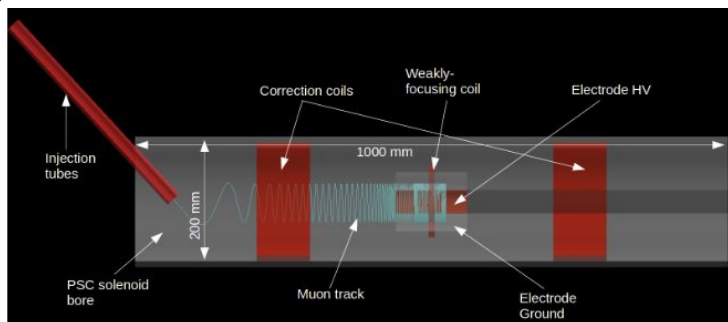


Introduction

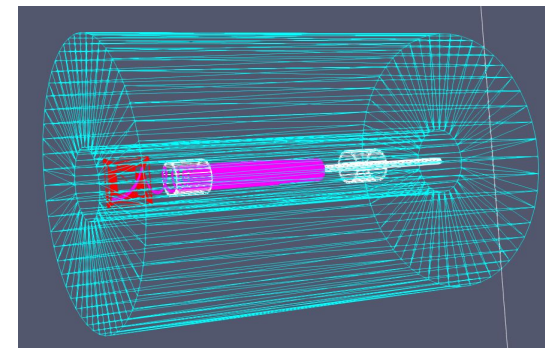


李政道研究所
TSUNG-DAO LEE INSTITUTE

- **G4BL** has been the bread and butter for muEDM experiment study (injection parameter and storage study).
- Establish alternative effort on study the injection parameters based on **musrSim**:
 - supports the entrance detector design effort,
 - Flexibility in implementing user plugin function,
 - provides independent simulation for cross-checking **G4BL**'s results.



- **G4BL** is oriented to beamline design study.
- user friendly.
- limited flexibility and limited user plugin feature.



- **musrSim** is oriented to detector development study.
- Exist dedicated c++ class for detector response study.
- Flexible and user plugin friendly.

- Today update:
 - Mirror **G4BL**'s configuration to **musrSim**, to establish baseline performance.
 - Reproduce **G4BL**'s storage efficiency in **musrSim**.

Kicker Field Implementation

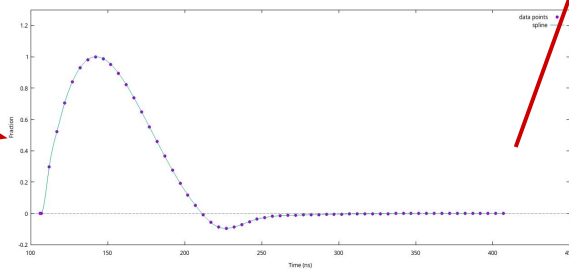
- Implementing kicker function in **musrSim**

time-dependent field txt

```
...
149 0.0 -357.0 0 0
150 0.0 -356.0 0 0
151 0.0 -355.0 0 -3.6105e-06
152 0.0 -354.0 0 -3.65143e-06
153 0.0 -353.0 0 -3.69302e-06
154 0.0 -352.0 0 -3.73514e-06
155 0.0 -351.0 0 -3.77789e-06
156 0.0 -350.0 0 -3.82125e-06
157 0.0 -349.0 0 -3.86519e-06
158 0.0 -348.0 0 -3.90985e-06
...
```

```
...
154 149.0 499.0 0 0
155 149.0 500.0 0 0
156 time
157 106 0
158 107 0.00129040377191317
159 112 0.297332825272224
160 117 0.523224775794070
161 122 0.705266468957932
162 127 0.839188193054098
163 132 0.930289041373811
...
```

Field tabulation
within musrSim



spline validation

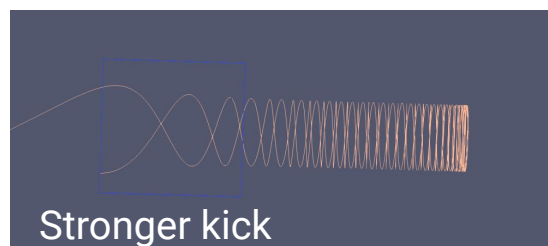
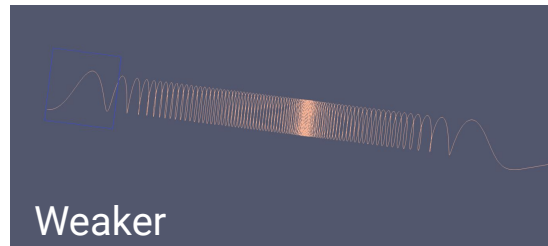
```
Time Structure detected
spline at 107.000000 ns is 0.001290 with derivative 0.012772
spline at 108.000000 ns is 0.029697 with derivative 0.042455
spline at 109.000000 ns is 0.083027 with derivative 0.062617
spline at 110.000000 ns is 0.151757 with derivative 0.073257
spline at 111.000000 ns is 0.226366 with derivative 0.074375
spline at 112.000000 ns is 0.297333 with derivative 0.065971
```

musrTabulatedElementField.cc

```
double frc=1.;

if (!pulse.isEmpty()) {
    //std::cout<<"pulse is not empty"<<std::endl;
    if (point[3] <= pulse.get_x_min() )
        frc=0.;
    else
        frc = pulse(point[3]);
}

// Interpolate between the neighbouring points
double Bfield_R =
    xField2D[xindex ][zindex ] * (1-xlocal) * (1-zlocal) +
    xField2D[xindex ][zindex+1] * (1-xlocal) *  zlocal  +
    xField2D[xindex+1][zindex ] *  xlocal  * (1-zlocal) +
    xField2D[xindex+1][zindex+1] *  xlocal  *  zlocal  ;
B[0] = (x>0) ? frc * Bfield_R * (local.x() /x) : 0.;
B[1] = (x>0) ? frc * Bfield_R * (local.y() /x) : 0.;
B[2] =
    zField2D[xindex ][zindex ] * (1-xlocal) * (1-zlocal) +
    zField2D[xindex ][zindex+1] * (1-xlocal) *  zlocal  +
    zField2D[xindex+1][zindex ] *  xlocal  * (1-zlocal) +
    zField2D[xindex+1][zindex+1] *  xlocal  *  zlocal  ;
B[2] = frc * B[2];
```



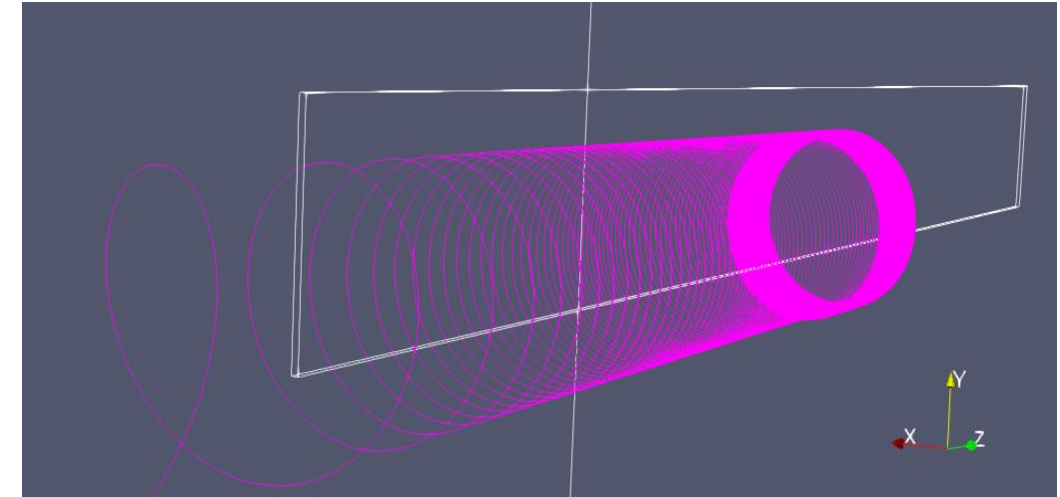
Thanks GM!

- This implementation:
 - detects time structure in the field map text file,
 - consumes time data into a spline object,
 - evaluates the fraction for given interpolated field components, at a given time in the simulation.

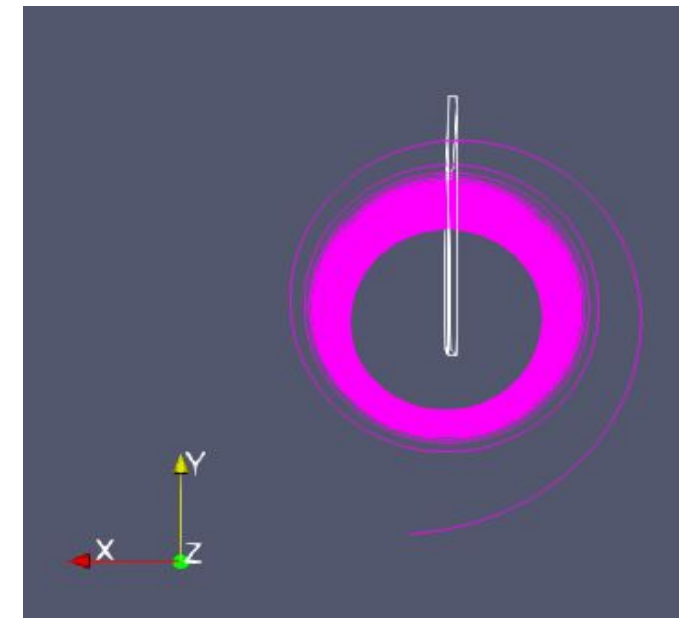
musrSim and G4BL Comparison

- Using one good event, taken from injection phase space for the exercise.
- Using injection parameter:

Parameter	Description	Value
θ	Injection angle	-45.022°
ϕ	Transverse angle	9.244°
R_{inj}	Injection radius	45.561 mm
Z	Longitudinal Injection Coordinate	-443.836 mm

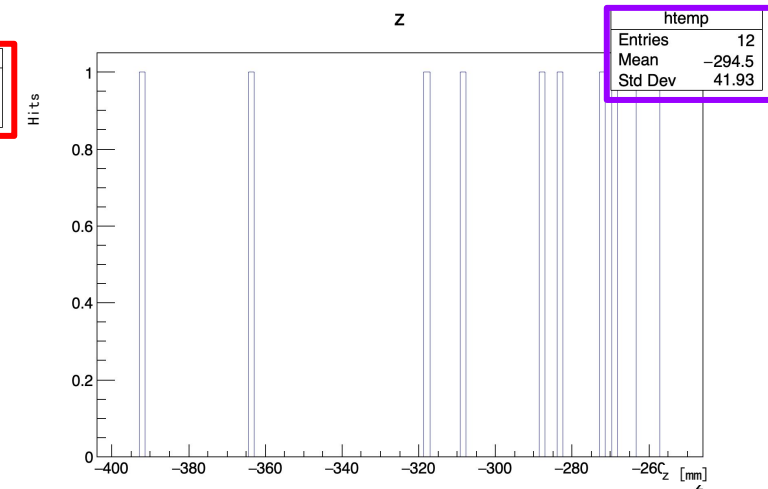
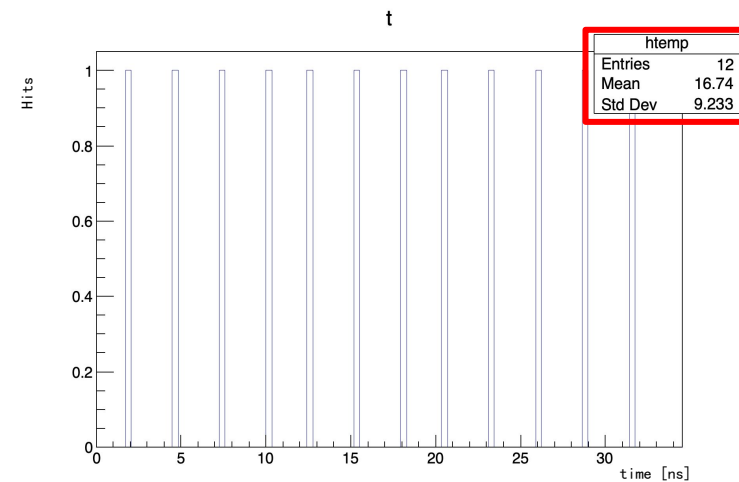
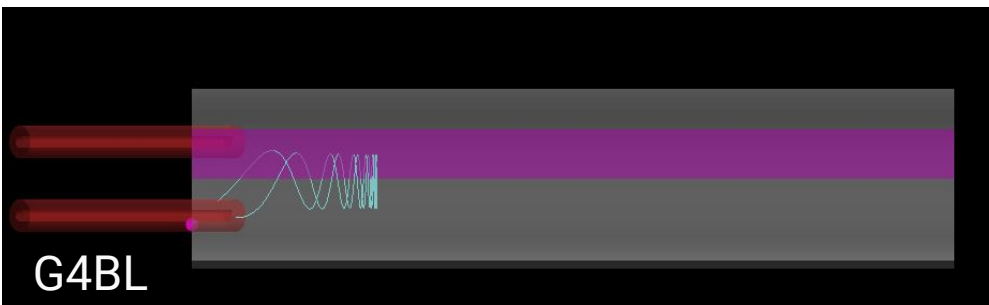
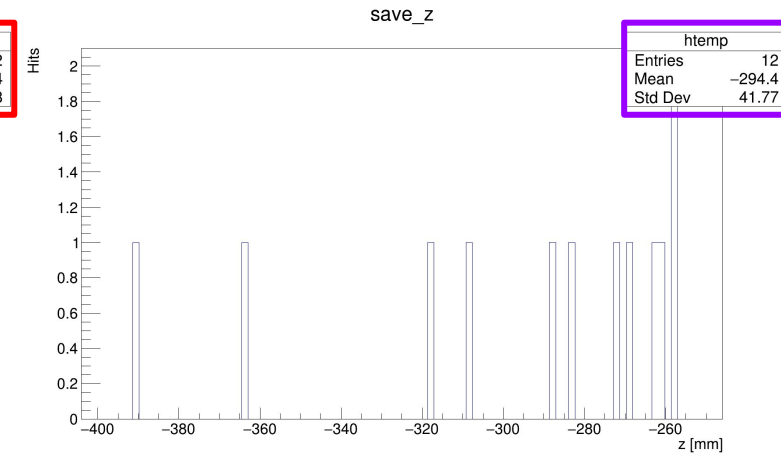
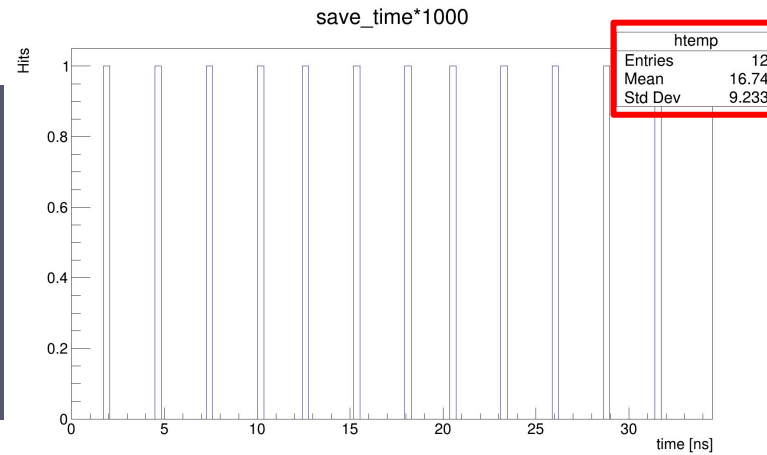


- Setup single virtual plane, on both **musrSim** and **G4BL**:
 - Spanning the full range of solenoid's length.
 - Ensure muon trajectory “curl” perpendicularly into the virtual plane: one hit per muon's entrance.
- Study the hit's information along the virtual plane between **musrSim** and **G4BL**.



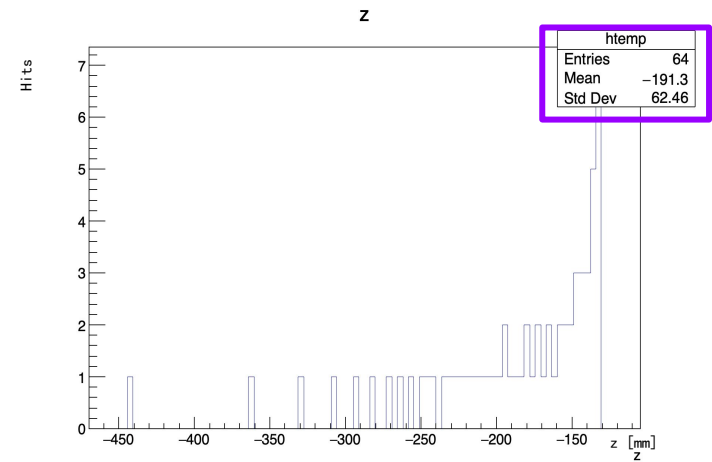
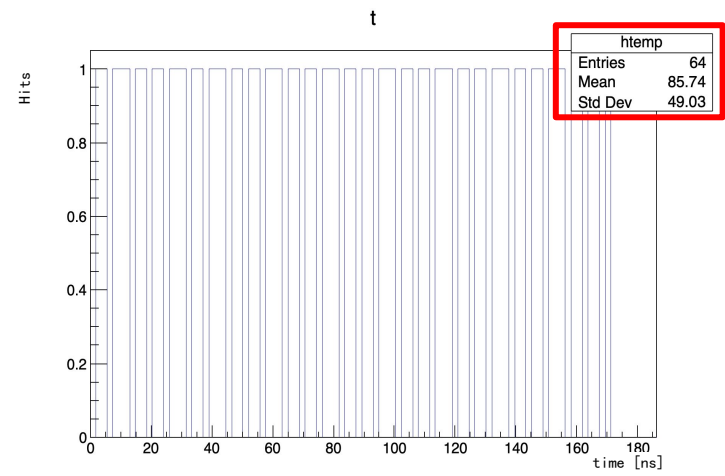
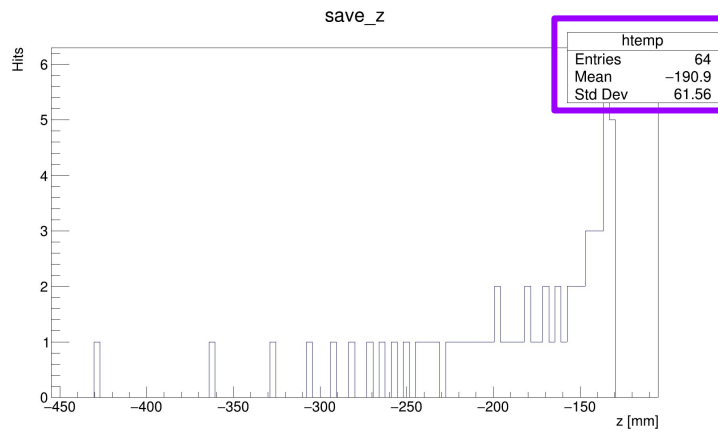
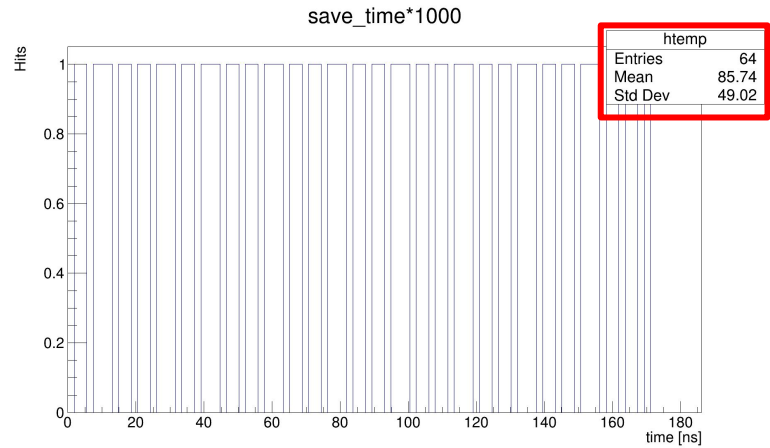
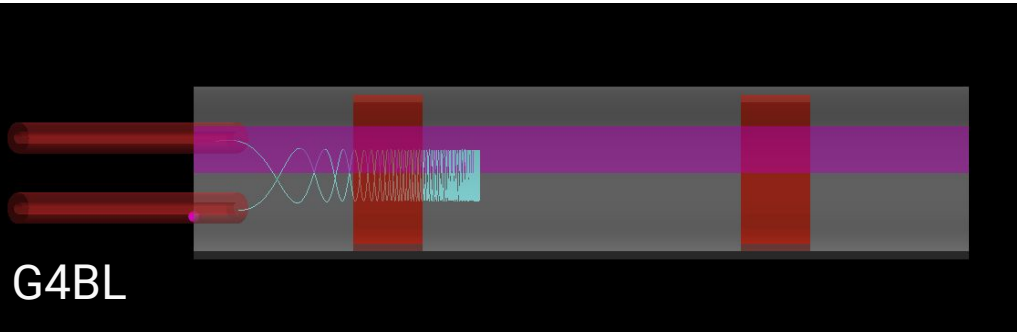
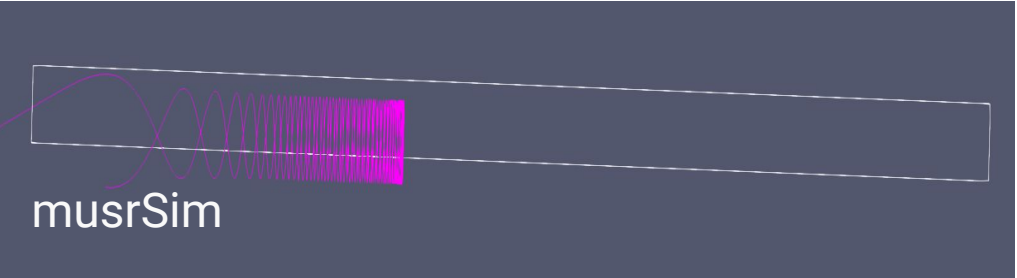
Step-by-Step Comparison

Fields included: PSC field



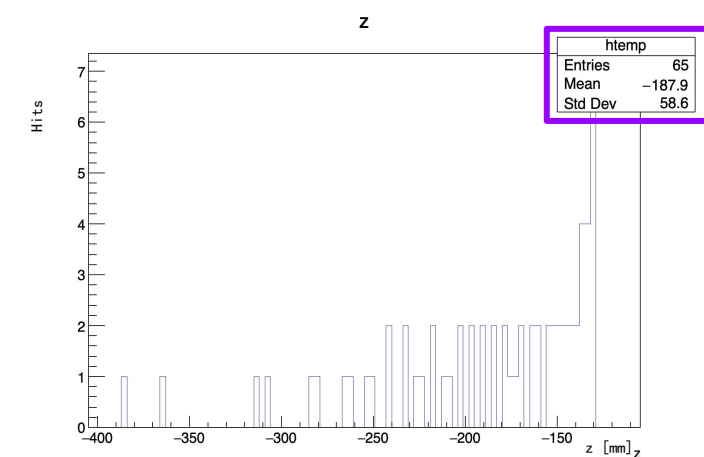
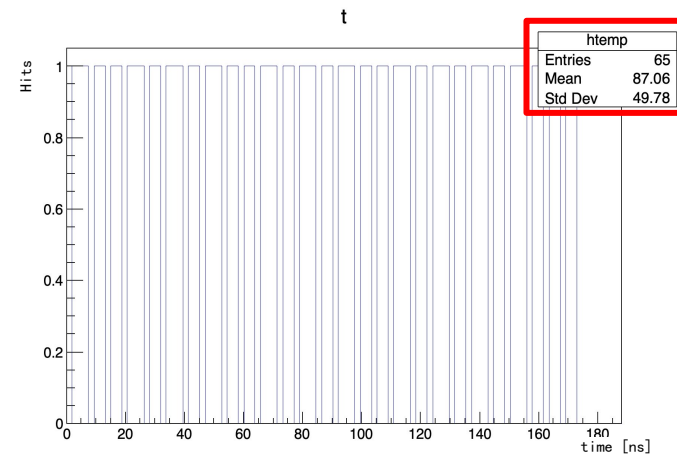
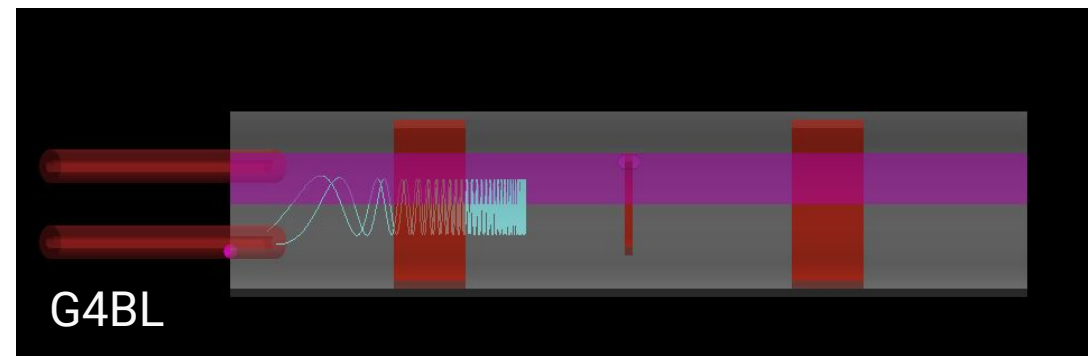
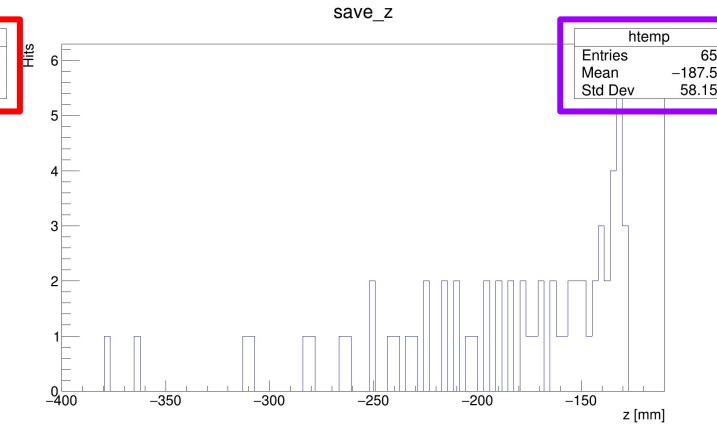
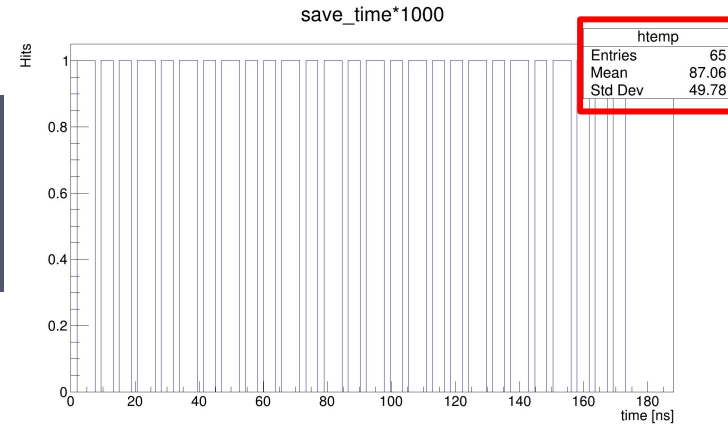
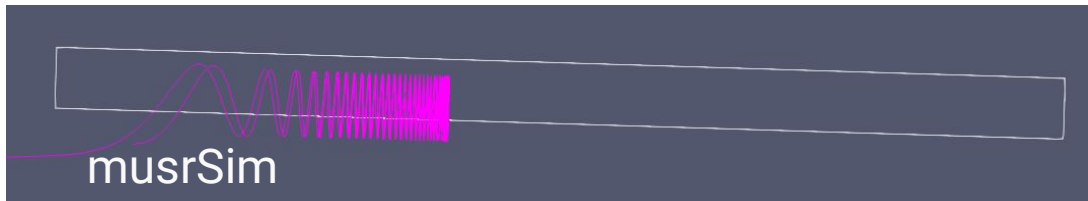
Step-by-Step Comparison

Fields included: PSC + Split coils



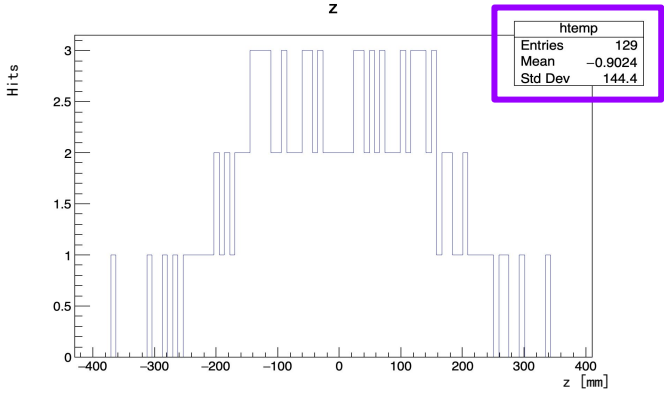
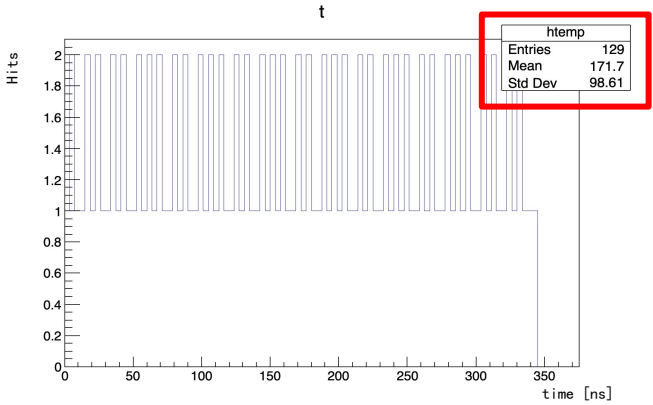
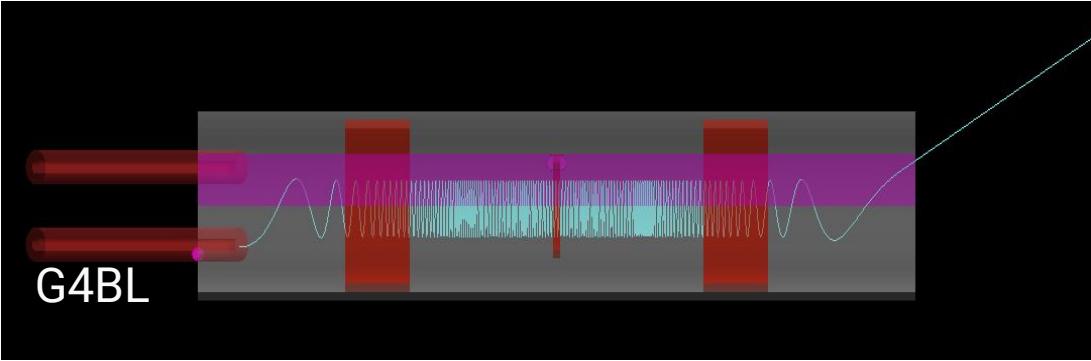
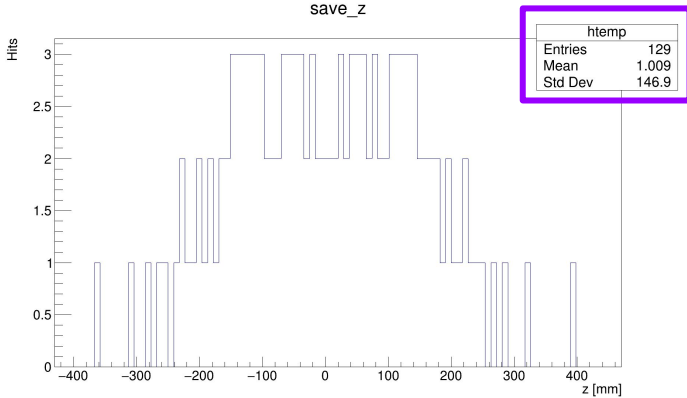
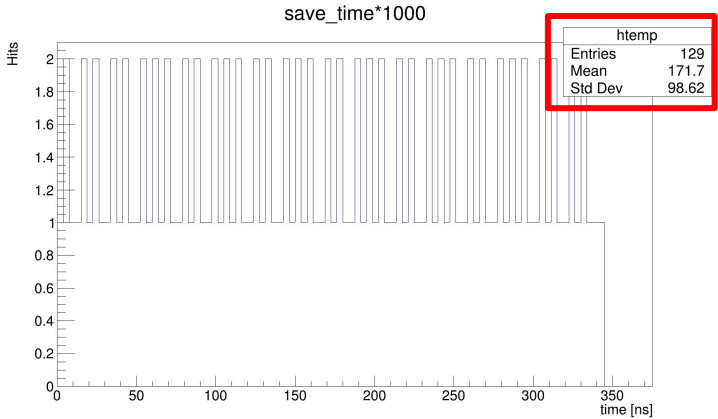
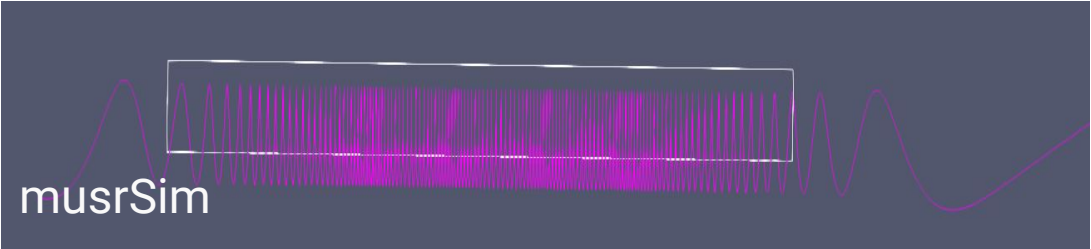
Step-by-Step Comparison

Fields included: PSC + Split coils + weak



Step-by-Step Comparison

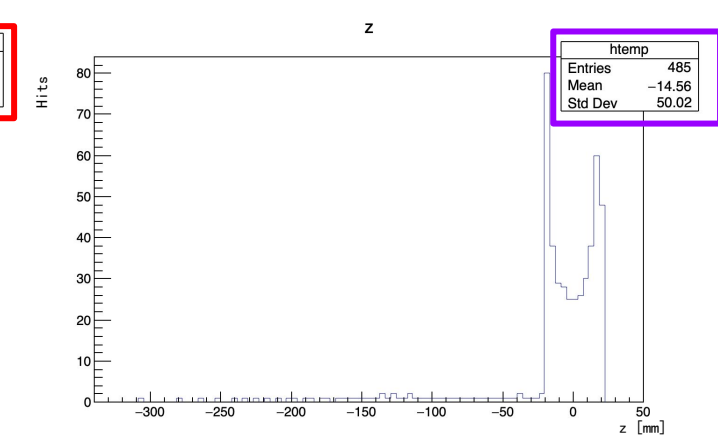
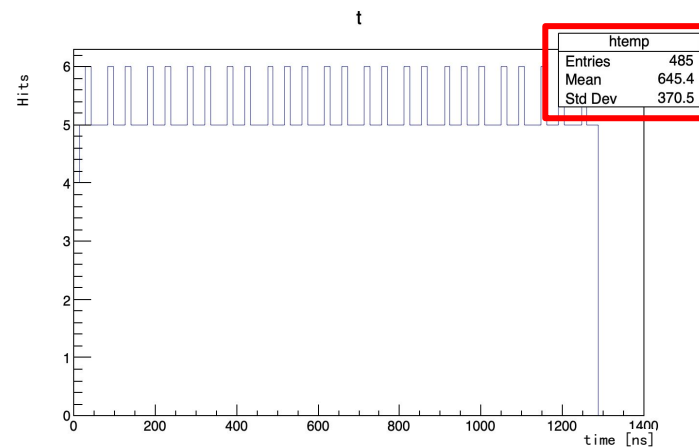
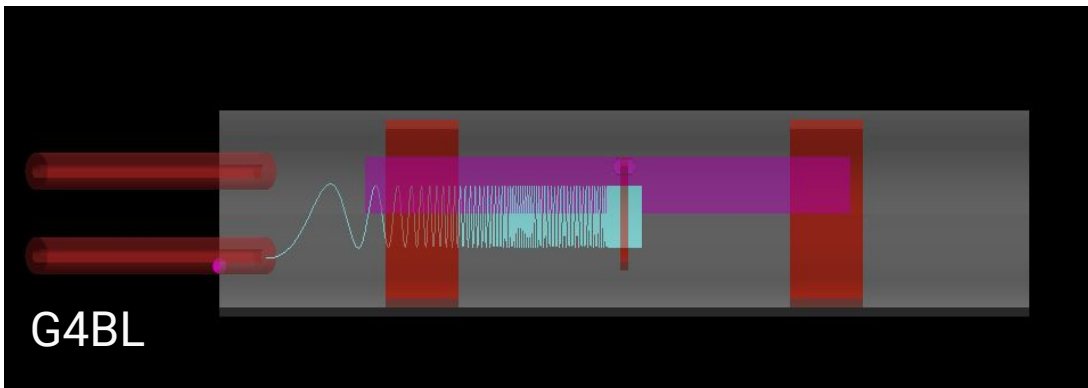
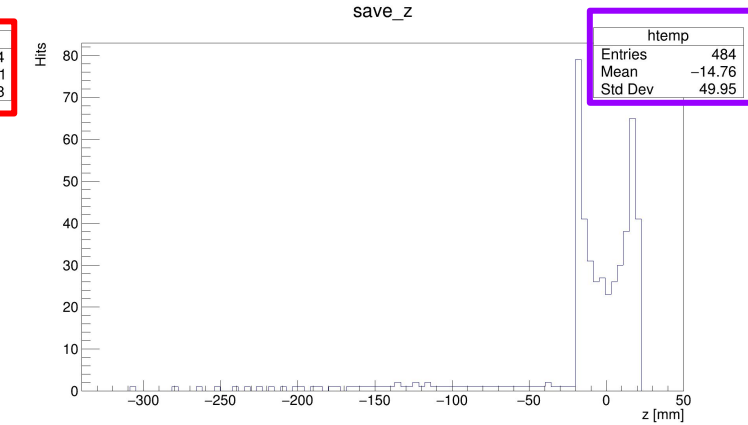
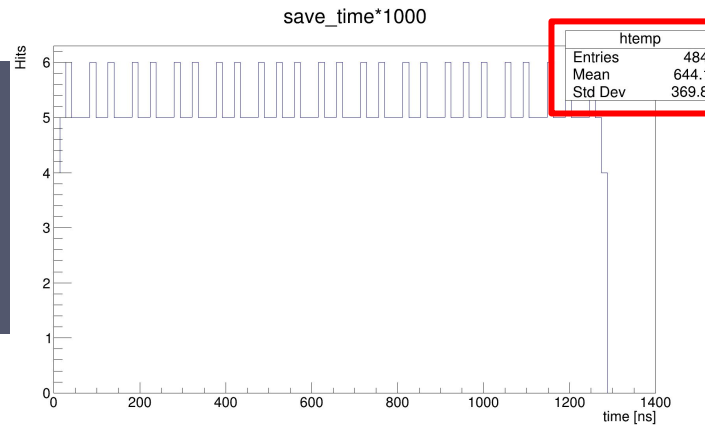
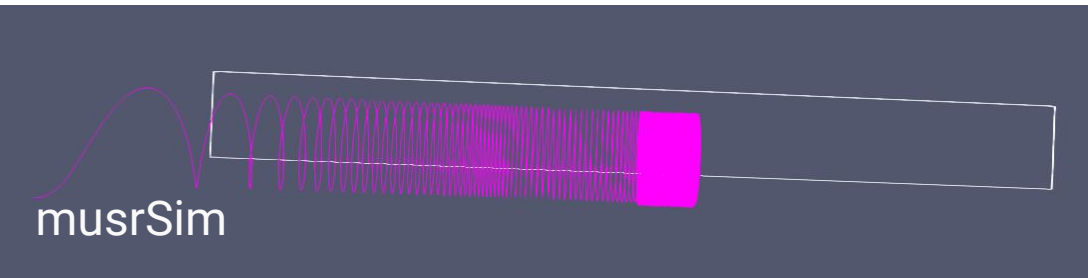
Fields included: PSC + Split coils + weak + Frozen E



Step-by-Step Comparison

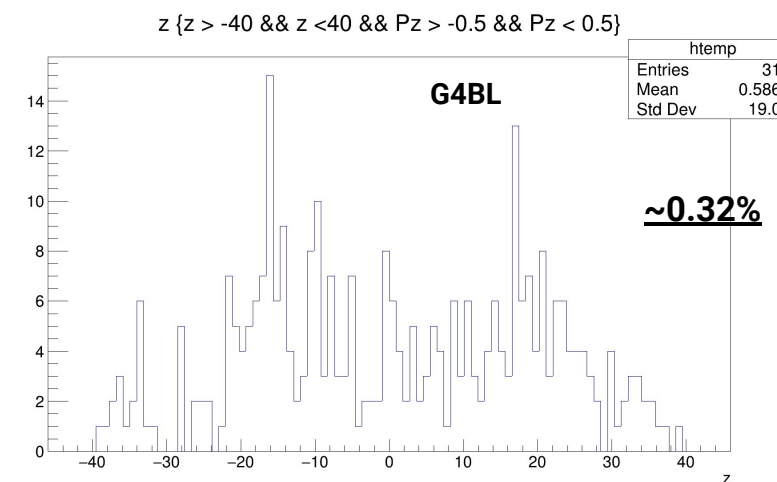
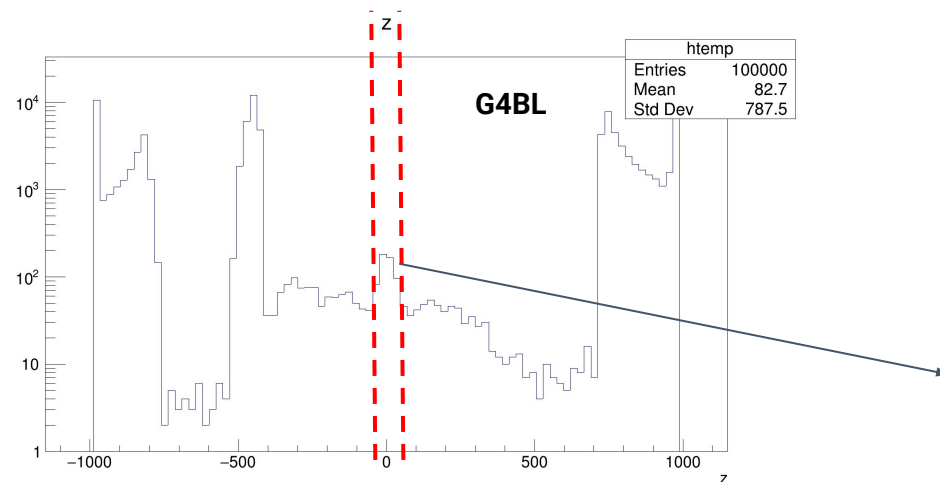
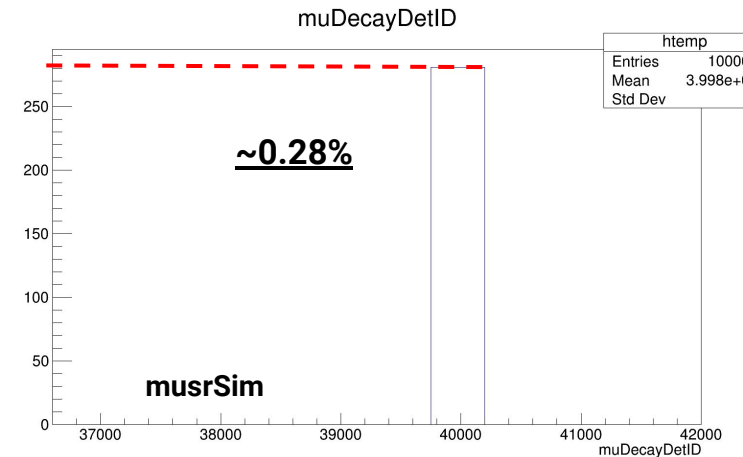
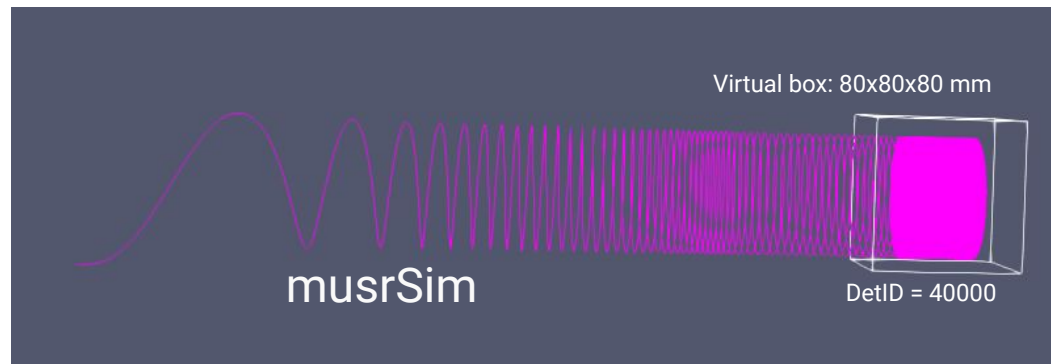
Fields included: PSC + Split coils + weak + Frozen E + kicker

The kicker field implementation is validated



Storage Efficiency Validation

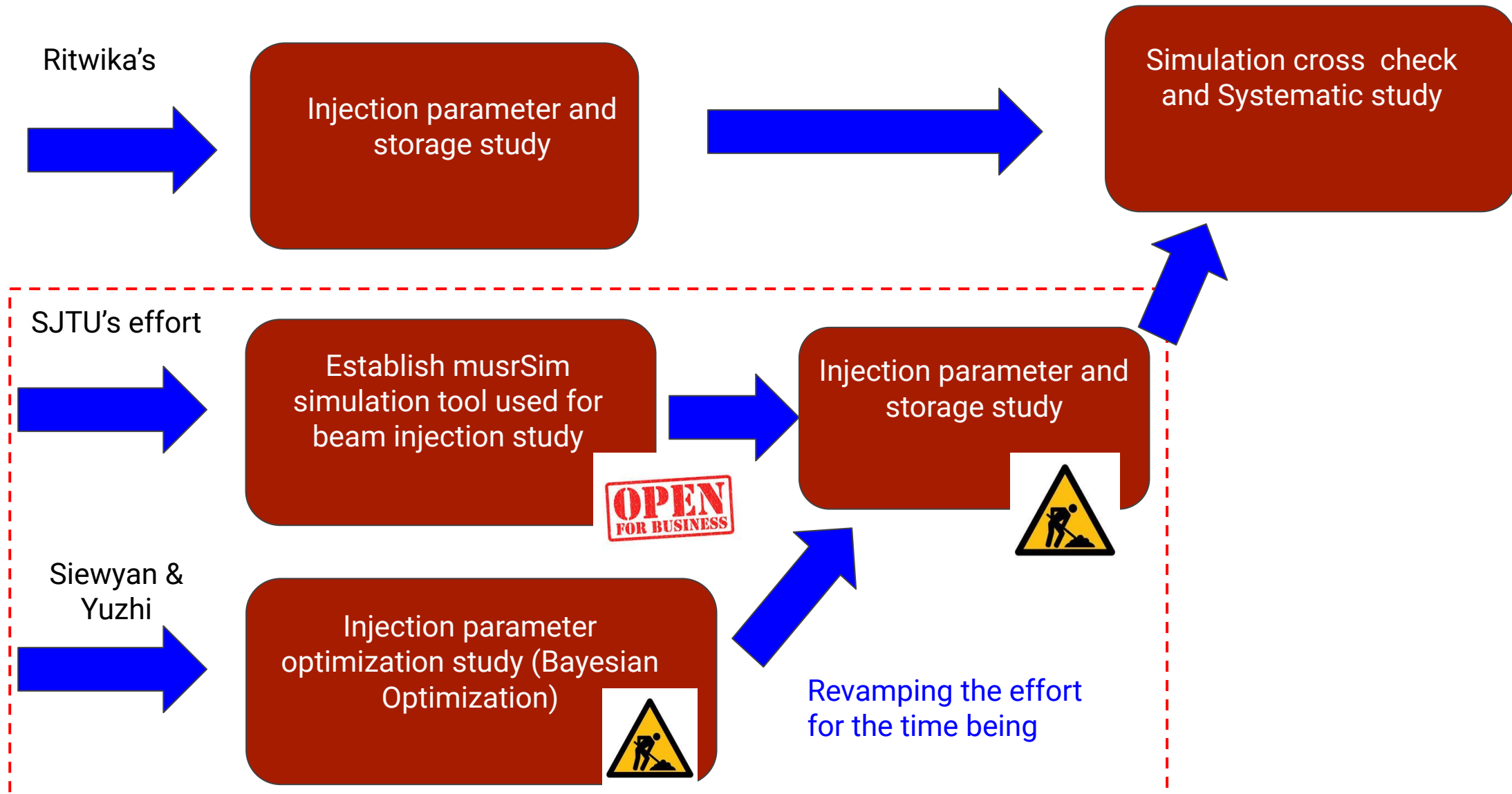
Using 100k injection phase space events
(InputDistAfterCollimator_R75mm_turtle.txt)



Thanks Yusuke!

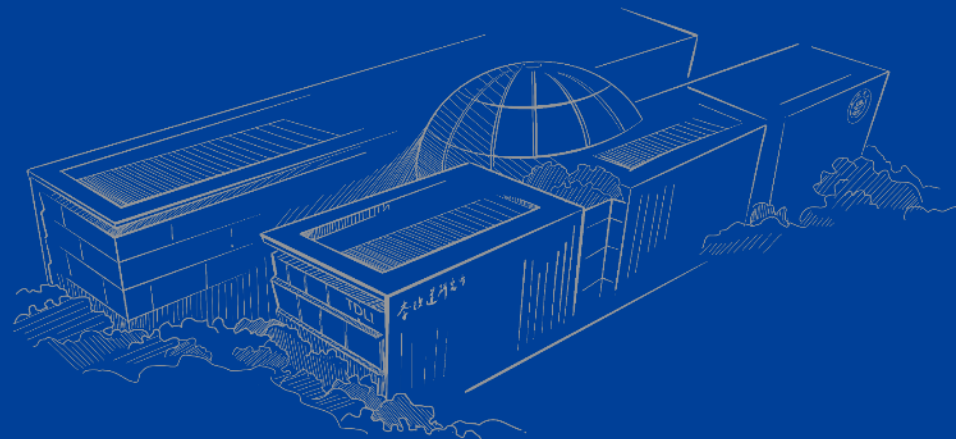
- **musrSim** could not perform tracking on the muon hits (involve placing infinite virtual plane along the trajectory).
- Field map coordinate offset is needed for **musrSim**, due to different technical design of both simulators.
- Interpolation step in field map is sensitive to **musrSim** simulation performance.

Conclusion and To-do



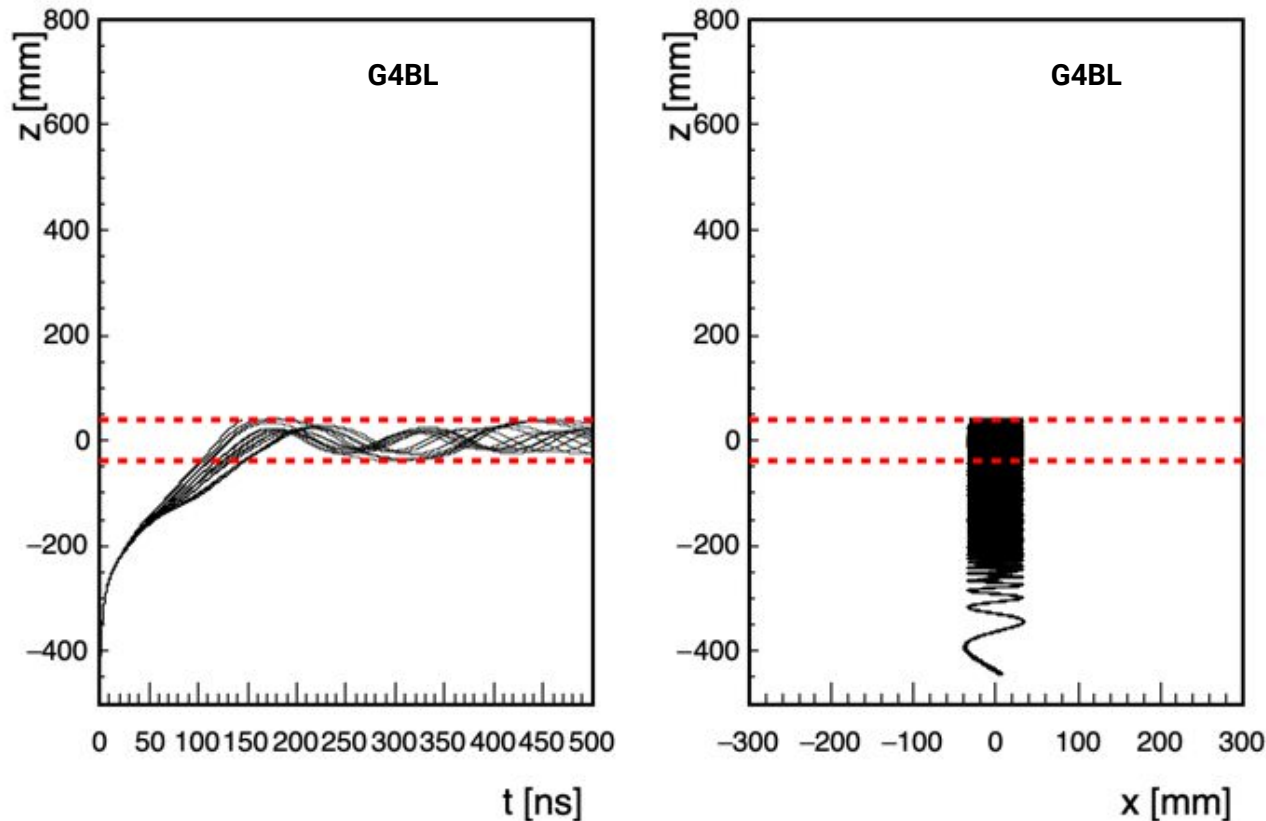


Backup



Good Event Definition

- Event taken from storage phase space.
- Event demonstrated in G4BL, that is successfully stored, using all the fields and injection parameters.



Yusuke' s work