

G4E における EIC detector の simulation

EIC 勉強会

16 June 2020

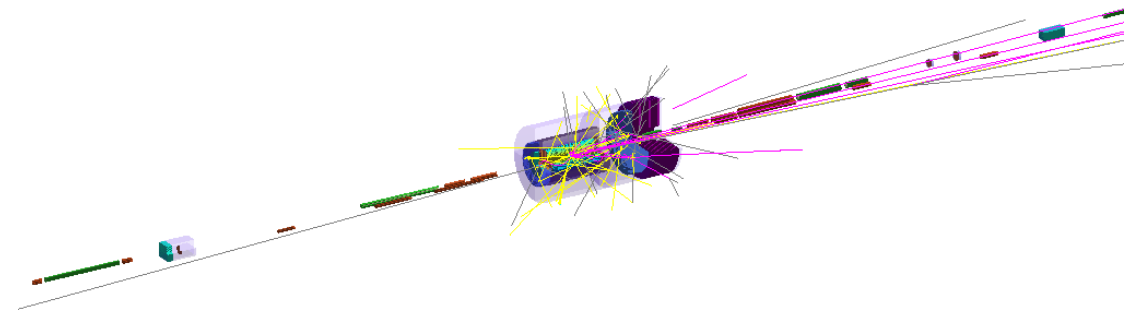
Yuya Ohsumi (Kobe university)

今後の目標と取り組んでいること

- 目標
ZDC の放射線耐性を調べる
 - * どれだけの放射線損傷があるか？
 - * それにより光量はどうなる？
- 取り組んでいること
ZDC でのエネルギー損失の見積もり
⇒ g4e

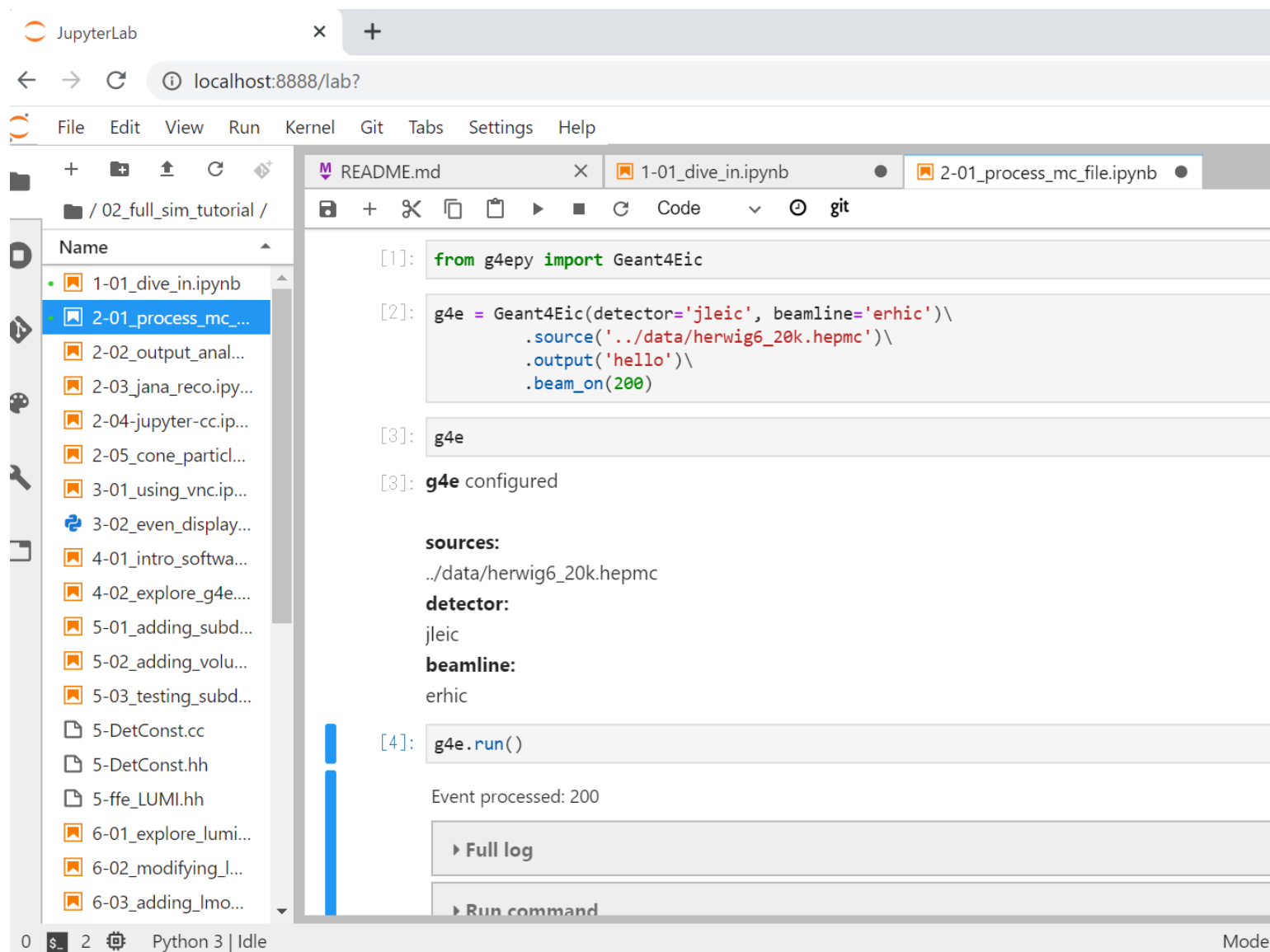
g4e (Geant4 for EIC)

- g4e とは
EIC detector のための Geant4 シミュレーション
コード等の基本的な枠組みは Geant4 のものと極めて近い
⇒ Geant4 の知識をそのまま活かせる
- docker を用いて g4e を動かすことが出来る
- g4e ではユーザーが自分のパソコンで detector を追加したり、既存のものをカスタマイズすることができる
- EIC User Group が JupyterLab 上で動作する g4e を tutorial とともに提供している



JupyterLab

- データ解析、可視化をブラウザ上で実行できる総合開発環境
- 実行環境は Python
- g4e を実行、その結果を全て JupyterLab 上で見ることができる



The screenshot displays the JupyterLab web interface in a browser window. The address bar shows the URL `localhost:8888/lab?`. The interface includes a menu bar (File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help) and a file browser on the left side. The file browser shows a directory structure for `/02_full_sim_tutorial/` with various IPYNB files, including `2-01_process_mc...` which is currently selected.

The main area shows a code editor with the following Python code:

```
[1]: from g4epy import Geant4Eic
[2]: g4e = Geant4Eic(detector='jleic', beamline='erhic')\
      .source('../data/herwig6_20k.hepmc')\
      .output('hello')\
      .beam_on(200)
[3]: g4e
[3]: g4e configured

sources:
../data/herwig6_20k.hepmc
detector:
jleic
beamline:
erhic
[4]: g4e.run()

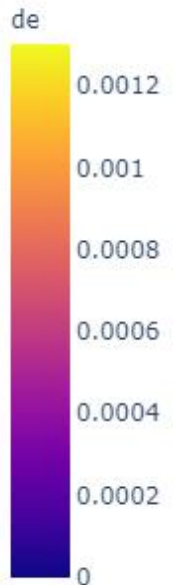
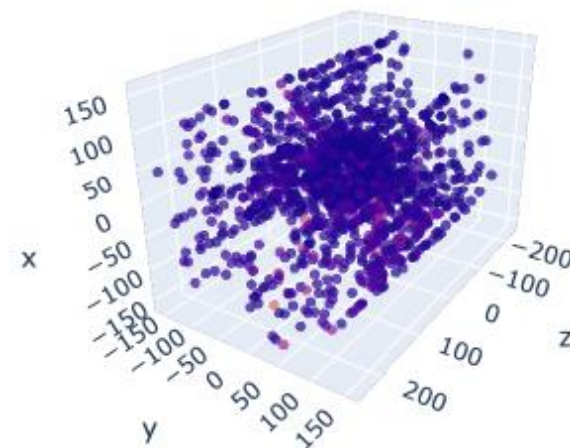
Event processed: 200
```

Below the code, there are buttons for `Full log` and `Run command`. The status bar at the bottom indicates the current kernel is `Python 3 | Idle`.

g4e を (tutorial のまま) 実行 (1)

- JupyterLab で tutorial に沿って走らせる
- Run event 200 の結果
- central detector 内の vtx detector での energy deposit

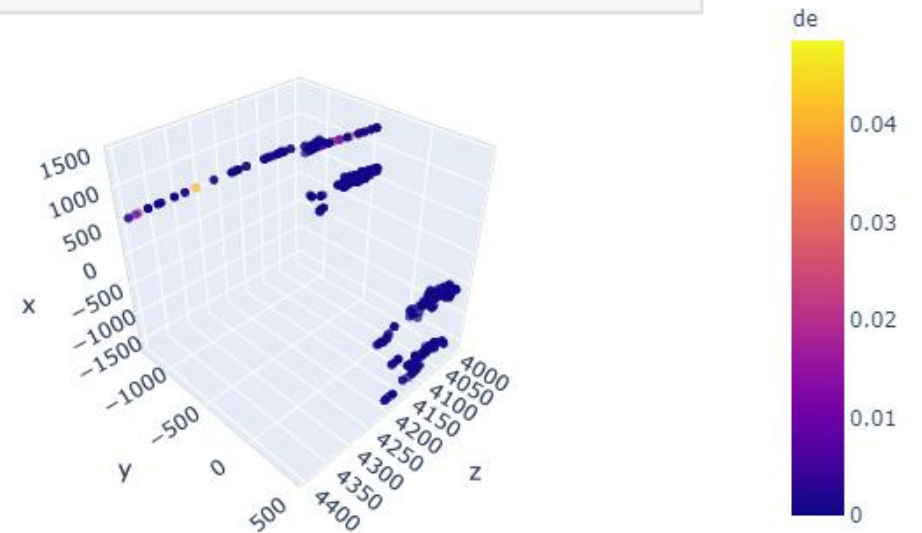
```
[16]: fig = px.scatter_3d(vtx_hits.head(2000), x='z', y='y', z='x', color='de', size_max=18, opacity=0.7)
fig.update_traces(marker=dict(size=2) )
fig.show()
```



g4e を (tutorial のまま) 実行 (2)

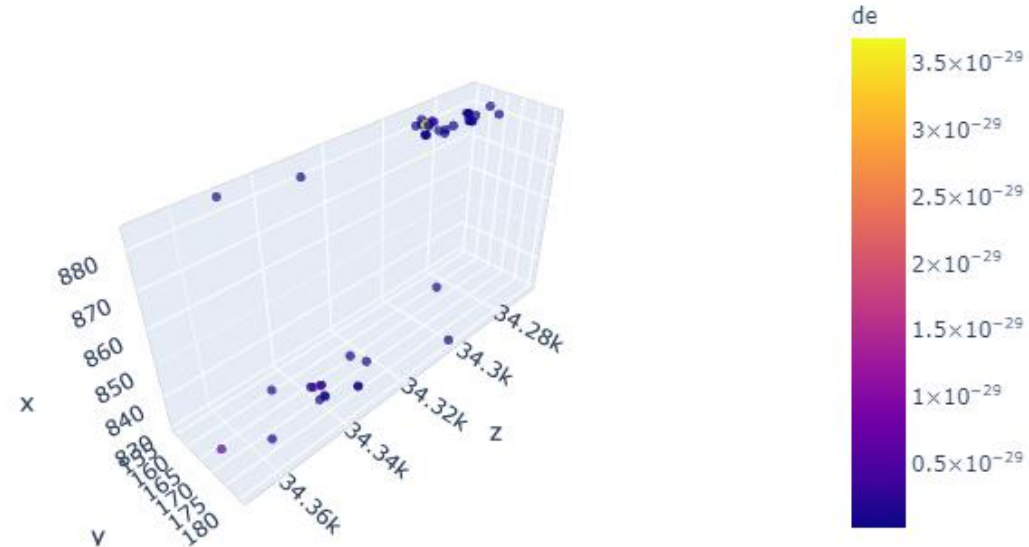
- Run event 200 の結果
- end cap 内の EMcalorimeterでの energy deposit

```
[22]: fig = px.scatter_3d(ci_emcal_hits.head(2000), x='z', y='y', z='x', color='de', size_max=18, opacity=0.7)
fig.update_traces(marker=dict(size=2) )
fig.show()
```



自分で edit した現段階での結果

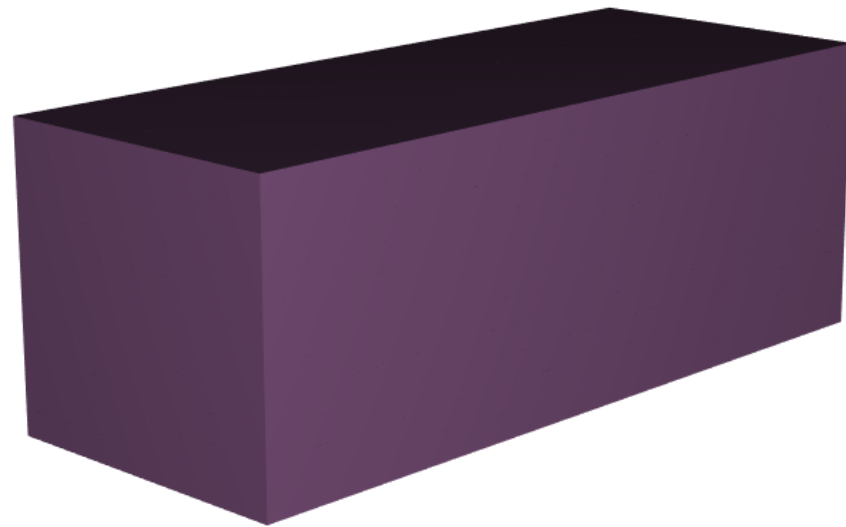
- tutorial にならって ZDC での energy deposit のグラフを取って見たがエネルギーが非常に低い (うまくいっていない) ⇒そもそも default で組み込まれている ZDC は機能しているのか？
- ZDC での energy deposit の simulation 結果をとることが今の目標
- ZDC のsetting を色々いじってみることに取り組んでいる (detector のサイズを変えてみる、カロリメータの層を増やす etc.) この結果からうまくいかない要因を調べる



Back up

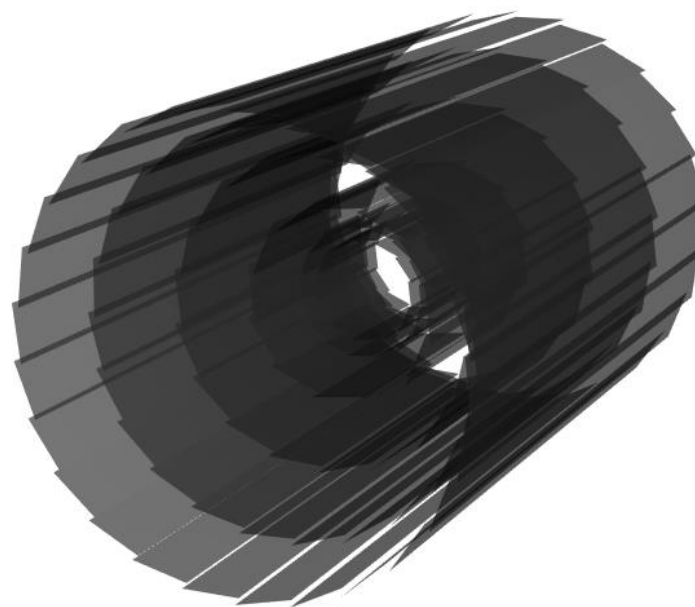
ZDC

- ZDC detector



VTX

- VTX detector



EMcal

- EMcalorimeter
(in end cap)

