Luminosity simulation with Guinea-Pig(++)

Antoine Laudrain (antoine.laudrain@desy.de)
HALHF WG meeting – 08.06.2023

HELMHOLTZ

DESY.

GuineaPig setup

Several GuineaPig versions available:

- Mikael's package:
  - C language version, v1.4.2
  - 2007, not maintained
  - Typical run: ~20 minutes

- From Key4HEP stack: (source /cvmfs/sw.hsf.org/spackages7/key4hep-stack/2023-04-08/x86_64-centos7-gcc11.2.0-opt/urwcv/setup.sh)
  - C++ version, v1.2.2 (https://gitlab.cern.ch/clic-software/guinea-pig/https://gitlab.cern.ch/clic-software/guinea-pig/)
    - A few new features added.
  - Not very active dev (but EDM4HEP support added in 2022)
  - Typical run: ~3 minutes (compiled with FFTW3)

GuineaPig setup

- New features:
  - Much cleaner and readable output
  - Automatic grid size (I had a lot of warnings due to particles going out of the default simulation grid) (https://trac.lal.in2p3.fr/GuineaPig/raw-attachment/wiki/PrincipalesPresentations/gridSizing.pdf)
  - EDM4HEP output might prove useful in the future.
- Some were back-ported to the C version, but not all...

- Documentation is approximative for both versions...

- I would lean towards using the C++ version in the long term.
  - I did all tests using both Mikael's (C) and Key4HEP (C++) versions for comparison.
  - => very good agreement

Luminosity simulations

First step: (try to) reproduce the results from the proposal

(https://arxiv.org/abs/2303.10150)

E (GeV)	σ_z (µm)	N (10^{10})	ϵ_{nx} (µm)	ϵ_{ny} (nm)	β_x (mm)	β_y (mm)	$\mathcal{L}$ (µb^{-1})	$\mathcal{L}_{0.01}$ (µb^{-1})	P/P_0
125 / 125	300 / 300	2 / 2	10 / 10	35 / 35	13 / 13	0.41 / 0.41	1.12	0.92	1
31.3 / 500	300 / 300	2 / 2	10 / 10	35 / 35	3.3 / 52	0.10 / 1.6	0.93	0.71	2.13
31.3 / 500	75 / 75	2 / 2	10 / 10	35 / 35	3.3 / 52	0.10 / 1.6	1.04	0.71	2.13
31.3 / 500	75 / 75	4 / 1	10 / 10	35 / 35	3.3 / 52	0.10 / 1.6	1.04	0.60	1.25
31.3 / 500	75 / 75	4 / 1	10 / 40	35 / 140	3.3 / 13	0.10 / 0.41	1.01	0.58	1.25
31.3 / 500	75 / 75	4 / 1	10 / 80	35 / 280	3.3 / 6.5	0.10 / 0.20	0.94	0.54	1.25
31.3 / 500	75 / 75	4 / 1	10 / 160	35 / 560	3.3 / 3.3	0.10 / 0.10	0.81	0.46	1.25
45.6 / 45.6	109 / 109	2 / 2	10 / 10	35 / 35	4.7 / 4.7	0.15 / 0.15	1.12	0.93	1
31.3 / 66.5	75 / 75	2.9 / 1.4	10 / 21	35 / 75	3.3 / 3.3	0.10 / 0.10	1.06	0.78	1.07
11.4 / 182	27 / 27	4 / 1	10 / 160	35 / 560	1.2 / 1.2	0.04 / 0.04	0.81	0.46	1.25

Table I. GUINEA-PIG simulations showing the luminosity per bunch crossing of both symmetric and asymmetric collisions. The first number in each pair refers to the positron bunch, the second to the electron bunch. Tabulated are, from left to right, beam energies, bunch lengths, number of particles per bunch, normalised emittances in the horizontal and vertical planes, interaction-point beta functions in the horizontal and vertical planes, the calculated full luminosity and that with energy within 1% of the nominal peak in inverse microbarns, and the relative power increase required compared to symmetric collisions. Numbers in the upper table represent HZ operation, whereas the lower table represents Z operation. The first row in each section of the table represents ILC-like parameters. Simulations include a vertical waist shift (equal to the bunch length), but assume zero transverse offsets and crossing angle.

Values from the paper

Within 1% of nominal CME

Scenario C++, autogrid	lumi	lumi1%	lumi_fine	lumi_ee	lumi_ee_high	lumi_pp	lumi_eg	lumi_ge	lumi_gg	lumi_gg_high
125/125	1.12	0.92	1.03295	1.0436	0.87912	0	0.40372	0.40245	0.2054	0.2054
31.3/500 | 2/2	0.93	0.71	0.76492	0.77173	0.62071	0	0.22554	0.42807	0.14929	0.14929
31.3/500 | 2/2	1.04	0.71	0.98722	0.99715	0.71125	0	0.35999	0.48273	0.20731	0.20731
31.3/500 | 4/1	1.04	0.60	0.99398	1.00253	0.61835	0	0.21258	0.84196	0.20347	0.20347
31.3/500 | 4/1	1.01	0.58	0.96853	0.98151	0.59906	0	0.21163	0.82866	0.20439	0.20439
31.3/500 | 4/1	0.94	0.54	0.9291	0.93816	0.5687	0	0.20084	0.79815	0.1924	0.1924
31.3/500 | 4/1	0.81	0.46	0.82466	0.82963	0.49592	0	0.18196	0.72346	0.17915	0.17915

Scenario C++, manual	lumi	lumi1%	lumi_fine	lumi_ee	lumi_ee_high	lumi_pp	lumi_eg	lumi_ge	lumi_gg	lumi_gg_high
125/125	1.12	0.92	1.03469	1.06133	0.90052	0	0.40877	0.4032	0.20135	0.20135
31.3/500 | 2/2	0.93	0.71	0.74549	0.75856	0.61583	0	0.21638	0.41555	0.14176	0.14176
31.3/500 | 2/2	1.04	0.71	0.95029	0.96992	0.69893	0	0.35146	0.46708	0.19793	0.19793
31.3/500 | 4/1	1.04	0.60	0.96181	0.98013	0.60748	0	0.20913	0.81915	0.19913	0.19913
31.3/500 | 4/1	1.01	0.58	0.9435	0.96364	0.59547	0	0.20541	0.80763	0.19434	0.19434
31.3/500 | 4/1	0.94	0.54	0.91126	0.92689	0.5666	0	0.19807	0.78576	0.19143	0.19143
31.3/500 | 4/1	0.81	0.46	0.91126	0.92689	0.5666	0	0.19807	0.78576	0.19143	0.19143

Scenario C, manual	lumi	lumi1%	lumi_fine	lumi_ee	lumi_ee_high	lumi_pp	lumi_eg	lumi_ge	lumi_gg	lumi_gg_high
125/125	1.12	0.92	1.03743	1.04801	0.88036	0	0.40492	0.40477	0.20592	0.20592
31.3/500 | 2/2	0.93	0.71	0.76286	0.77025	0.61802	0	0.22217	0.42814	0.14878	0.14878
31.3/500 | 2/2	1.04	0.71	0.98456	0.99604	0.70958	0	0.36645	0.48629	0.2127	0.2127
31.3/500 | 4/1	1.04	0.60	0.99157	1.00343	0.61473	0	0.21592	0.84856	0.20697	0.20697
31.3/500 | 4/1	1.01	0.58	0.96729	0.97615	0.59512	0	0.21419	0.83129	0.20771	0.20771
31.3/500 | 4/1	0.94	0.54	0.92905	0.93702	0.56783	0	0.2061	0.80152	0.20032	0.20032
31.3/500 | 4/1	0.81	0.46	0.82398	0.8292	0.49596	0	0.18176	0.72498	0.17971	0.17971

Conclusions

- Autogrid gives less simulation warnings.

- Luminosity comparison (only e+e- initiated lumi provided in the paper):
  - Good compatibility between C++ w/ autogrid, C++ manual grid and C versions.
  - Some differences for the first two scenarios
    - Not sure we have the exact same parameters.