Validating McMule

- Before searching for lepton-flavour-violating ALPs at MUonE (FairMuonE). First we need a McMule installation with numbers we can trust.
- Setups
  - Liverpool server (AlmaLinux 9.8)
  - GFortran 11.5; 3.10 via conda
  - Meson + ninja -j1
  - ‘Latest’ release branch (de32e762)

Running McMule

- user.f95 → user.so: cuts, observables and beam energy, compiled with gfortran
- Standard input: statistics, seed, ξc and the piece, e.g. em2em0
- **mcmule** writes one .vegas file per run into out/
- **pymule** merges seeds and converts to μb

- It prints raw numbers: $\sigma\,[\mu b] = \text{raw} \times \alpha^n \times 3.894 \times 10^8$
  - It computes with α=1 and reports cross sections in MeV^{-2}.
  - Multiply by α^2 at LO and α^3 for NLO pieces.
  - E.g. LO at 150 GeV: 5.85594×10^{-3} raw → 121.42 μb

Running McMule

- $e^- \mu \rightarrow e^- \mu$ (em2em)
- μ^+ or μ^- from the flavour: muone+ or muone-

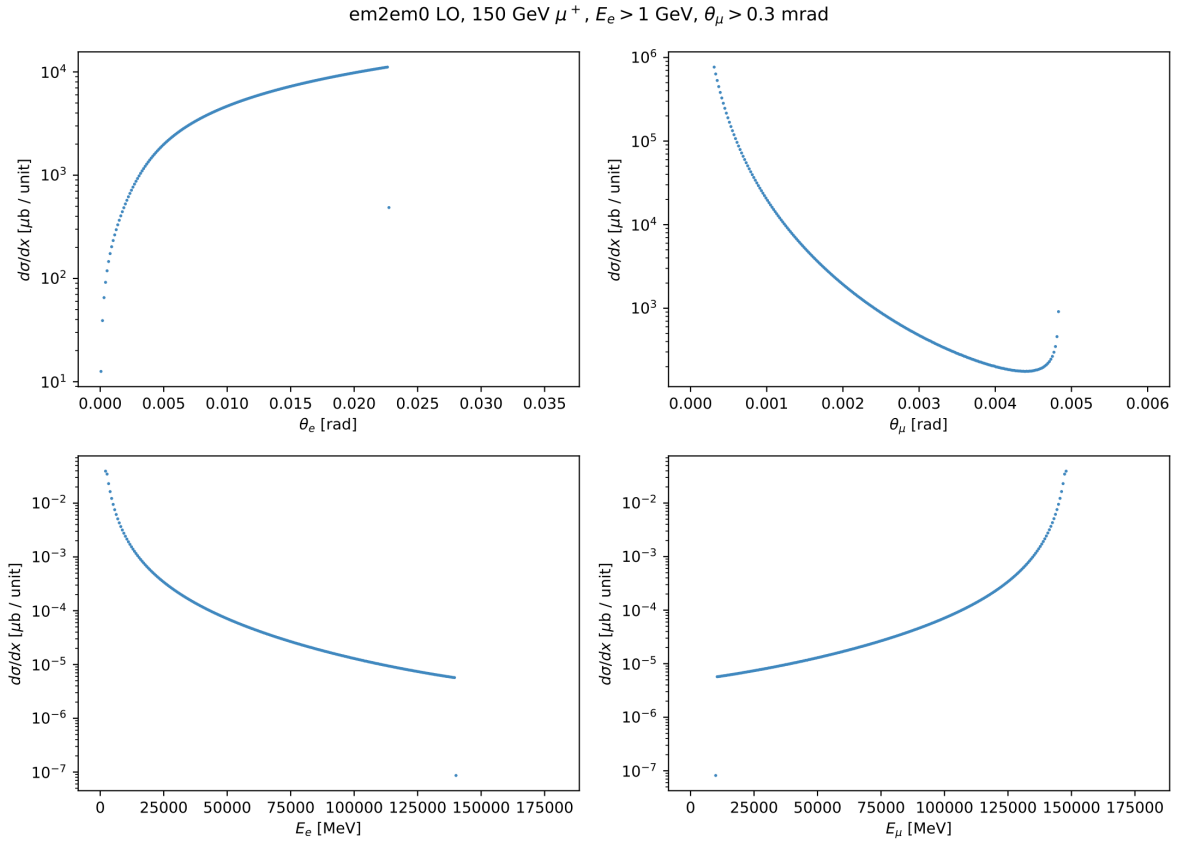
Setup	Cuts	User file
S1	**E_e > 1 GeV, θμ > 0.3 mrad**	**bandcut 0**
S2	**S1 + 0.9 < θμ / θμ,el < 1.1**	**bandcut 1**
A	**E_e > 1 GeV only**	**θμ cut removed**

S1 and S2: the setups of arXiv:2212.06481, Muon-electron scattering at NNLO.

- What I tested: LO@150 and 160 GeV; NLO@160 GeV

LO at 150 GeV: 121.4229 μb

- 5 seeds all give 5.85594×10^{-3} (raw)
- Histograms integrate back to σ: 121.422–121.423 μb



LO at 150 GeV: 121.4229 µb

- It agrees with two published results

Check	Beam, cuts	McMule [µb]	Published [µb]
arXiv:1811.06743	150 GeV, E_e > 1 GeV	245.038870(23)	245.038910 (MESMER)
arXiv:2212.06481	160 GeV, S1 = S2	106.443563(9)	106.44356 (McMule)

- arXiv:1811.06743 uses slightly older m_e and m_mu (Claude).
-

NLO agrees with arXiv:2212.06481

- 138 jobs at ξ_c = 0.5, 0.25 and 0.125:

Contribution	McMule pieces	S1 paper [ub]	S1 McMule [ub]	S2 paper [ub]	S2 McMule [ub]
Electron line	FEE + REE15 + REE35	−0.61211	−0.61211(4)	−4.66042	−4.66037(3)
Mixed e–μ	FEM + REM	−0.21404	−0.214042(3)	−0.16017	−0.160175(4)
Muon line	FMM + RMM	−0.02843	−0.028429(1)	−0.16134	−0.161336(1)
Vac. polarisation	A	1.40140	1.401402	1.40140	1.401402
NLO total σ_1, μ^-	LO + all of the above	106.99038	106.99038(4)	102.86304	102.86309(3)
NLO total σ_1, μ^+	mixed sign flipped	107.41847	107.41847(4)	103.18338	103.18344(3)

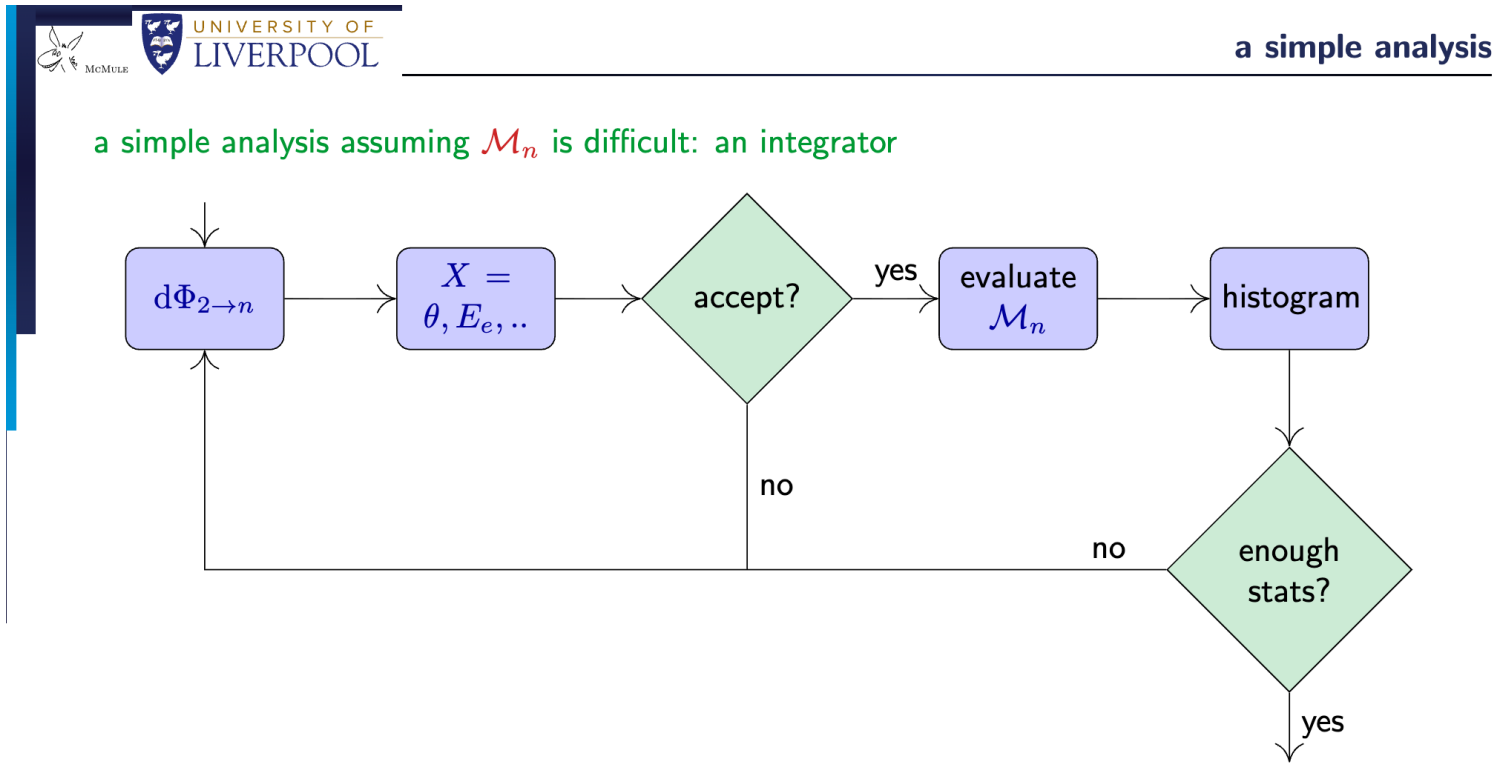
Mixed NLO pieces – a bug in there?

- mu- beam (muone-), S1, $\xi c = 0.5$: –0.21405 ub
  - Paper: –0.21404 ub
- mu+ beam (muone+), S1, $\xi c = 0.5$: 2.18 ub
  - Paper: +0.214 ub

- Claude:
  - Cause: the soft-photon subtraction for the mixed pieces (EM2EM_EM_part) hard-codes the μ^- charge sign, while the matrix elements take the muon charge from the flavour. For μ^+ the soft divergence no longer cancels.
  - Affects em2emFEM and em2emREM (plus REM15, REM35, REMco, AFEM, AREM) for μ^+, the MUonE beam.

Some initial thoughts integrating McMule into FairMUonE

- Thanks for Yannick and Jake's helpful discussion!
- McMule's output is histograms, but no event output.



- After consulting with Claude:

3. Integration options

Option	Pros	Cons and blockers
(a) Offline: event files read by a new `MUonEVertexGenerator` subclass	Matches McMule's batch design, including resampling, which needs a whole buffer of events. No Fortran or OpenLoops build inside FairMUonE. Files can be reused.	McMule fixes the collision energy (`scms`) for a whole run, so every event assumes the same beam energy. Large files: LHE text is about 1 kB/event. The generator is still in testing.
(b) Online library, like MESMER	Uses the real per-event beam. Same workflow as MESMER.	There is no "give me one event" call. Resampling needs a full buffer. Fortran global state, an OpenLoops/meson build, and no C API for generation. Needs major work from the expert.
(c1) HepMC3 pipe from McMule's pipe writer	No linking, events stream into FairMUonE.	The pipe writer is unreachable (suffix bug). Same fixed-beam limit as (a). Needs HepMC3 in FairSoft (not checked).
(c2) MEG-style: reweight MESMER events by McMule K-factors	Cheap, and possible now.	Approximate. The photon kinematics come from MESMER, so it is not a real replacement.

4. Recommendation: option (a)

- **McMule side:** run `--generate`, write LHE or HepMC3, and convert to a flat ROOT tree (n, pdg[], p4[], w[], piece).
- **New FairMUonE class** **`MUonEVertexGeneratorMcMule`** (in `VertexGenerators/McMule/`):