Copy as Markdown

em2em: mixed NLO corrections are wrong for μ^+ (`EM2EM_EM_part` hard-codes the μ^- charge sign)

- **Reported by:** C. Zhang (University of Liverpool), 6 November 2026
- **McMule version:** `release` branch, commit `de32e762` (2 October 2026). `EM2EM_EM_part` is the same in `devel` (checked 6 November 2026).
- **Build:** AlmaLinux 9.8, GFortran 11.5.0, Python 3.10, Meson 1.11, all default process groups.

Summary

With a μ^+ beam (flavour `"muone+"`), the mixed electron–muon NLO correction to μe scattering (`em2emFEM` + `em2emREM`) comes out about 10 times too large. With μ^- (`"muone-"`), the same setup reproduces arXiv:2212.06481. The soft subtraction for the mixed pieces uses fixed charges that are only right for μ^-, while the mixed matrix elements take the muon charge from the flavour.

Numbers

160 GeV beam, setup S1 of arXiv:2212.06481 ($E_e > 1$ GeV, $\theta_\mu > 0.3$ mrad), $\xi_c = 0.5$, `bandcut` 0. Raw McMule output ($\alpha = 1$, MeV^{-2}); μb = raw × α^3 × conv = raw × 151.310.

Flavour	`em2emFEM` (raw)	`em2emREM` (raw)	$\sigma_{e\mu}^{(1)}$ [μb]	arXiv:2212.06481, Table 2
muone-	−1.23770e-3	−1.76943(36)e-4	−0.214050(5)	−0.21404 ✓
muone+	−2.81905e-3	+1.72384(1)e-2	+2.1818	+0.21404 ✗

- `em2emREM` for μ^+ used 4 seeds, each with 5M × 10 adaptation and 20M × 20 main calls. χ^2 was 0.9–1.4 and the seeds agree to 2×10^{-4}, so the integral is stable but wrong.
- In the same μ^+ setup, the electron-line, muon-line and vacuum-polarisation contributions agree with the paper (short test runs): −0.6186(45), −0.02853(7) and

1.401403(3) μb, against −0.61211, −0.02843 and 1.40140 μb.

Cause

In `mue/mue.f95`, all four charges in the mixed particle string are hard-coded to +1:

```
FUNCTION EM2EM_EM_part(p1, p2, p3, p4)
...
em2em_em_part = parts((/part(p1, 1, 1, 1), part(p2, 1, 1, 2), part(p3, 1, -1, 1)
```

`EIK_ALL` in `pre/eikonal.f95` weights each pair with `sgn = -effcharge_i*effcharge_j`, so the electron–muon cross terms always carry the sign of Q_e·Q_μ = +1, which is $e^-\mu^-$.

The mixed matrix elements instead take the charges from the flavour (`#include "charge_config.h"` sets Q_μ = +1 for a trailing `+`):

- `em2emg_em` (`mue/mue_mat_el.f95`): `128*pi**3*alpha**3 * Qe * Qmu * ...`
- `em2eml_em` (`mue/mue_mat_el.f95`): `... * Qe * Qmu`

For `"muone+"` the matrix elements flip sign, but the subtraction term in `em2emREM` and the integrated eikonal in `em2emFEM` do not. The soft singularity of `em2emREM` is then no longer cancelled. It is only cut off by `softcut = 1e-10`, which gives a finite, stable, wrong result.

The regression tests (`muevegas`, `muepointwise`) use flavour `muone` (Q_μ = −1), so they don't cover this case.

Affected pieces

Every piece that calls `set_partfunc(em2em_em_part)` in `src/integrands.f95`, for μ^+ only: `em2emFEM`, `em2emREM`, `em2emREM15`, `em2emREM35`, `em2emREMco`, `em2emAFEM`, `em2emAREM`.

Not affected: the electron-line and muon-line pieces (`em2em_ee_part` and `em2em_mm_part` only enter quadratically in the charges), `em2emA`, and the NNLO helpers `EM2EM_31_part`, `EM2EM_22_part`, `EM2EM_13_part` and `EM2EM_mixd_part`, which already use `Qe` and `Qmu`.

Suggested fix (not tested on our side)

```
  FUNCTION EM2EM_EM_part(p1, p2, p3, p4)
    !! e-(p1) mu(p2) -> e-(p3) mu(p4), muon charge from the flavour
  real (kind=prec) :: p1(4),p2(4),p3(4),p4(4)
  type(particles) :: em2em_em_part
#include "charge_config.h"
  em2em_em_part = parts((/part(p1, Qe, 1, 1), part(p2, Qmu, 1, 2), part(p3, Qe, -1
  END FUNCTION EM2EM_EM_part
```

For μ^-, every electron–muon product of effective charges is unchanged, so μ^- results stay identical. For μ^+, the cross terms flip. With this fix we would expect `em2emFEM` = +1.23770e-3 and `em2emREM` = +1.76943e-4 for `"muone+"` in the setup above.

A cheap regression test would be C symmetry of the mixed pieces: `em2emFEM` (μ^+) = − `em2emFEM` (μ^-), and the same for `em2emREM` .

How to reproduce

User file: `examples/shared_user.f95` (with `min_val` and `max_val` declared `real(kind=prec)`), with the beam set to 160 GeV in `INITUSER` :

```
  call initflavour("muone+")
  call initflavour("muone+", me**2 + mm**2 + 2*me*160.e3_prec)
```

Then run, and repeat with `"muone-"` :

```
printf '%s\n' 1000 5 5000 5 11702 0.5 0.5 em2emFEM 0 | ./mcmule
printf '%s\n' 5000 10 20000 20 11702 0.5 0.5 em2emREM 0 | ./mcmule
```

Other observations (minor)

1. `examples/shared_user.f95` declares `min_val` and `max_val` as default `real` , but `src/load_user.f95` maps them as `real(kind=prec)` pointers, so the histogram bounds are read wrongly. The total cross section is unaffected.
2. Regression tests on this machine: `muevegas` passes 47/57 and `muepointwise` 55/56. Every LO and NLO piece passes, apart from `em2emREE35` at 1.4×10^{-10} (threshold 10^{-10}). The failures above round-off level are NNLO only: `em2emRREEEE3536` 6.5×10^{-3}

(threshold 10^{-5}), `em2emRRMIXD3536` 2.0×10^{-5} and `em2emAR` 1.0×10^{-6}. The rest are at most 2.2×10^{-8} (`em2emNFDISP`, `em2emNFMMDISP`, `em2emAREE35`, `em2emAFEM`, `em2emAR35`, `em2emRF35`).

3. The pymule from the GitLab package registry (in a Python 3.10 environment) does not match the release. `pymule batch` stops in `FortranLoader` with `undefined symbol: load_user_MOD_names` (from `_libmcmule_mock`), and the menu runner expects `.mcmule` output files where the release writes `.vegas`. Analysis with `sigma` and `mergefks` works (with `obs="0"`).
4. The first parallel `ninja` build failed with `Cannot open module file 'hol_initialisation_dp.mod'` in the OpenLoops process libraries. `ninja -j1` worked.

Context: what agreed

Check	McMule	Reference
LO, 150 GeV, E_e > 1 GeV	245.038870(23) µb	245.038910 µb (arXiv:1811.06743, MESMER)
LO, 160 GeV, S1 = S2	106.443563(9) µb	106.44356 µb (arXiv:2212.06481)
NLO total, 160 GeV μ^-, S1	106.99038(4) µb	106.99038 µb (arXiv:2212.06481)
NLO total, 160 GeV μ^-, S2	102.86309(3) µb	102.86304 µb (arXiv:2212.06481)

The 150 GeV difference comes from the older m_e and m_µ values in arXiv:1811.06743. The exact LO cross section is 245.038912 µb with their masses and 245.038901 µb with McMule's.

With μ^-, every NLO contribution in Table 2 of arXiv:2212.06481 agrees within 1.3σ for both S1 and S2: electron line, mixed, muon line and vacuum polarisation. Each F+R sum is independent of ξ_c (0.5, 0.25, 0.125; $\chi^2/\text{dof} \le 1.1$). Flipping the sign of the μ^- mixed term also reproduces the paper's μ^+ totals: 107.41847(4) against 107.41847 µb (S1), and 103.18344(3) against 103.18338 µb (S2).