



Update on BSM issues and new SIGFPEs

Andrea Valassi (CERN)

4th June 2024

(attached to Madgraph on GPU development meeting, 11th June 2024)

<https://indico.cern.ch/event/1355154>

(previous update was on May 28 – only mentioning changes since then)

Updates and recap – BSM & other processes

- Brief update on my PRs (since May 14 meeting) – Beyond Standard Model models
 - **New WIP PR #847 (EWdim6)** – adds EWdim6 “ $u d \sim \rightarrow w^+ z$ ” to the repo (Zenny’s process)
 - Verified that my earlier BSM patches for SUSY/EFT fix code generation for this too ([#615](#) is fixed)
 - Code also builds ok, but HRDCOD=(0|1) builds give different results at runtime (*new issue #846*)
- Recap of pending process-specific issues (mainly BSM, but not only)
 - SUSY: [#825](#) (*susy_gg_tt madevent tests – xsec mismatch Fortran vs cudacpp*)
 - SUSY: [#826](#) (*susy_gg_t1t1 madevent tests – no xsec in cudacpp madevent*)
 - HEFT: [#833](#) (*heft_gg_bb madevent tests – LHE mismatch Fortran vs cudacpp, FPTYPE=f*)
 - EWdim6: [#846](#) (*ewdim6_ud_wz cudacpp tests – ME mismatch HRDCOD=0 vs HRDCOD=1*)
 - SM: [#806](#) (*gq_ttq cudacpp tests – segmentation fault on AMD GPUs on LUMI*)
 - Anyone interested in taking a look? Otherwise we can fix these after the release?

From systematic tests of all processes and all FPTYPE combinations using ‘tput’ and ‘tmad’ scripts

Today I report on investigations about this in collaboration with Olivier



Some “easy” bits

- SUSY: [#825](#) (*susy_gg_tt madevent tests – xsec mismatch Fortran vs cudacpp*)
 - Fixed by Olivier in PR [#850](#) – missing “Ccoeff” in some code (CODEGEN)
 - Completed by AV in PR [#860](#) – update runTest reference files (mg4gpu) for susy processes
- Minor enhancements by AV (mg4gpu only, no CODEGEN)
 - PR [#851](#) – add back bin/madevent script for all processes
 - PR [#854](#) – add matrix1.pdf (from matrix1.ps) for all processes to debug channel issues

A much more complex bit

- Olivier's draft PR [#852](#) investigating the zero xsec issue 826 opened a Pandora box
 - I tested Olivier's patch and other related issues in PR [#853](#)
 - *Main finding by OM: in cudacpp we miss a mapping of "iconfig" and "channel" (e.g. for colors)*
- Note: all issues on my first slide come from my 'tput' or 'tmad' test suite
 - tmad test suite runs "madevent < input.txt" for Fortran/C++/CUDA, compares xsec and LHE
 - But... so far I have always only been running these for "iconfig=1"
 - *Some nasty problems seem to be lurking only at iconfig>1 (including SIGFPE crashes)*
- Just for reference: there are many different entities and variables involved
 - And the fact that Fortran (1 to N) and C (0 to N-1) indexes differ does not help
 - This is my proposal how to reference them in the code

```
// Summary of numbering and indexing conventions for the relevant concepts (see issue #826 and PR #852)
// - Diagram number (no variable) in [1, N_diagrams]: all values are allowed (N_diagrams distinct values)
//   => this number is displayed for information before each block of code in CPPProcess.cc
// - Channel number ("channelId" in C, CHANNEL_ID in F) in [1, N_diagrams]: not all values are allowed (N_conf
//   => this number (with F indexing) is passed around as an API argument between cudacpp functions
// - Channel number in C indexing: "channelIdC" = channelId - 1
//   => this number (with C indexing) is used as the index of the channelIdC_to_iconfig array below
// - Config number ("iconfig" in C, ICONFIG in F) in [1, N_config]: all values are allowed (N_config <= N_diagr
// - Config number in C indexing: "iconfigC" = iconfig - 1
```

SIGFPE crash in rotxxx

- There is a SIGFPE crash [#855](#) in Fortran function rotxxx (aloha_functions.f)
 - Only in optimized `-O3` code: relevant variables in gdb show up as `<optimized out>`
 - Disabling optimization (IIRC `-O1` is enough?) makes the crash disappear
 - My proposed workaround [#857](#): add the *volatile* keyword for a few Fortran variables
 - Disable optimizations of very specific lines of code (related to Fortran SIMD?)
 - *This technique is extensively used in cudacpp SIMD ixx/oxx: volatile prevents many crashes*
 - Issue and fix are fully reproducible (crashes without, does not crash with volatile)
 - Not clear why it appears only for some iconfig – but I would fix this independently
 - And fixing this issue then makes it possible to see further issues down the line...

A snapshot of other issues

- There is a different SIGFPE crash [#845](#) in cudacpp function sigmakin
 - Intermittent: same binary executable sometimes crashes and sometimes does not...
 - This seems most likely related to the wrong/missing iconfig-ichannel map for colors?
- There is a color mismatch [#856](#) in LHE files
 - This is clearly related to the wrong/missing iconfig-ichannel map for colors
- There is still the zero cross section I reported in [#826](#)
 - Olivier's patch does not fix this for me
- Olivier mentioned a few issues he identified in tests with Stefan
 - Can you provide a detailed reproducer please?
 - Some of these may be related to what I described, some may not
- Last point: IMO it is imperative that we have QUICK systematic tests of “launch”
 - See discussion in [#711](#): allow generate_events with lower precision i.e. fewer events
 - We did not agree on this last year – I think the issues we see now are a consequence of that
 - This would have tested systematically all iconfig for all processes
- My opinion: we need tests, tests, tests...