



Mirror processes and other fixes (master) Channelid reimplementation (master_june24) *Getting closer to a release*

Andrea Valassi (CERN)

Madgraph on GPU development meeting, 30th July 2024

<https://indico.cern.ch/event/1355157>

(previous update was on July 08 – only mentioning changes since then)

Outline

- Progress in [madgraph4gpu master](#) (and [mg5amcnlo gpucpp](#))
 - Our default branch, where I have shown all results in previous meetings
 - Fixes and improvements from myself and Olivier, with mutual reviews/help (thanks!)
- Progress in [madgraph4gpu master_june24](#)
 - A new parallel branch created by Stefan and Olivier for channelid #830 and warps #756
 - Associated to mg5amcnlo gpucpp_june24 and gpucpp_wrap branches (and lone commits)
 - Olivier asked me to look at this with high priority, towards a merge into master
 - (I used github fragments and reverse engineering – was this described in this meeting?)

Overview: following up on three weeks ago



- (1) master
 - all of the blocking issues have been analysed and fixed (some pending PR reviews)
- (2) master_june24
 - *this is where I spent most of the last three weeks*
 - I identified a large number of bugs and missing features in channelid
 - I fixed all of these (pending PR review): *essentially, a full reimplementaion of channelid*



master branch

Pending physics issues in master

- **No cross section in susy gg_t1t1 #826**
 - OM/SR investigating if coupling ordering is responsible [#862](#)
- **Cross section mismatch in pp_tt012j #872**
- Much better than two weeks ago!!! Many crashes and complex issues fixed
- Both issues are now visible in the CI (6 failures: two times three FPTYPE's)
 - Note: the new CI has an option to bypass/ignore these two issues, check if enabled...
- *There is also WIP by OM on another CI extension, and some issues found there*
 - *Apologies, I did not have time to look at that yet...*

Master: following up on three weeks ago



- ([#826](#)) No cross section in susy_gg_t1t1 – a.k.a. bug in the order of couplings
 - **Fix by OM/SR merged on Mon 29 Jul** in PR [#918](#) (closes [#826](#) and [#862](#))
 - **Additional patch by AV (reviewed by OM) merged on Mon 29 Jul** in PR [#934](#)
- ([#872](#)) Cross section mismatch in pp_tt012j
 - **Fix by AV under review by OM** in PR [#935](#)
 - See details on the next slide



pp_tt012j xsec mismatch – mirror processes

- ([#872](#)) Fortran and cudacpp cross sections differ for (gu_ttxgu within) pp_tt012j
- Analysis by AV (with contributions from OM and SR)
 - This only happens for processes with ‘mirror processes’
 - It happens for gu_ttxgu within pp_tt012j (mirror is ug_ttxgu – swap g/u from left/right beam protons)
 - It does not happen for gu_ttxgu standalone (g from left beam, u from right beam)
 - It also happens for uux_ttx within pp_tt (now added as a much simpler test)
 - Code signatures: MIRRORPROCS=true in Fortran, nprocesses=2 in cudacpp
 - These must be kept, else the cross section is a factor two off (OM patch [#754](#) August 2023)
 - Note: nprocesses=2 is only used for static asserts in cudacpp, there is no array(2) for this...
 - *Cross sections are not bit-by-bit the same because different numbers of events are processed*
 - Fortran computes helicities twice (once per mirror), cudacpp computes helicities once (overall)
 - Specifically: Fortran helicity recomputation leads to one more RESET_CUMULATIVE_VARIABLE call
- *Fix by AV (under review by OM) in PR [#935](#)*
 - *Add one extra RESET_CUMULATIVE_VARIABLE call during cudacpp helicity computation*
 - *IMO, **huge** benefit (fortran and cudacpp xsecs agree bit-by-bit) for no cost (process few events more)*
 - IMO, these bit-by-bit tests are the main reason we have a reasonably solid code now
 - *To do (address OM comment): add sanity check that the two fortran helicity lists are identical*

Master: other ‘smaller’ fixes/improvements

- Various build improvements (largely developed while debugging physics bugs)
 - Disable OpenMP by default as agreed 3 weeks ago – enable it only if USEOPENMP=1
 - [Patch by AV \(reviewed by OM\) merged on Tue 16 Jul](#) in PR [#900](#) (closes [#758](#))
 - Add support for clang16 and clang17 (while debugging with clang address sanitizer)
 - [Fix by AV \(reviewed by OM\) merged on Tue 16 Jul](#) in PR [#905](#) (closes [#904](#))
- Various fixes and improvements in tests and CIs
 - Fix segfault from constexpr_math.h and increase testmisc.cc tolerances within valgrind
 - [Fix by AV \(reviewed by OM\) merged on Tue 16 Jul](#) in PR [#908](#) (closes [#903](#) and [#906](#))
 - Fix runTest segfault (remove cudaDeviceReset) and simplify googletest template usage
 - [Fix by AV \(reviewed by OM\) merged on Tue 16 Jul](#) in PR [#909](#) (closes [#907](#))
 - Fix the “tput_test” in AV’s “2nd CI” (runTest.exe and other tests were not executed at all)
 - [Fix by AV \(reviewed by OM\) merged on Wed 24 Jul](#) in PR [#938](#) (closes [#937](#))
 - Add a “3rd CI” by OM
 - Pending: [patch by OM \(under review by AV – apologies for the delay\)](#) in PR [#865](#)
- Various fixes and improvements in code generation and MG5AMC submodule
 - Upgrade master MG5AMC to latest gpucpp (OM [mg5amcnlo#118](#) and DM [mg5amcnlo#107](#))
 - [Patch by AV \(reviewed by OM\) merged on Tue 16 Jul](#) in PR [#897](#)
 - [Additional patch by AV \(reviewed by OM\) merged on Tue 16 Jul](#) in PR [#913](#) (without squashing)
 - Remove gen_ximprove.py and madevent_interface.py from patch.common
 - [Patch by OM \(reviewed by AV\) merged on Mon 29 Jul](#) in PR [#849](#) (closes [#844](#))
 - [Additional patch by AV \(reviewed by OM\) merged on Mon 29 Jul](#) in PR [#939](#)

master_june24 branch

Merging master and master_june24

- Olivier asked me to look into the merge of master and master_june24
- I created my branch 'june24' in WIP PR [#882](#)
 - starting at master_june24
 - with the idea of progressively merging master into it
- To start with, I looked at master_june24 as-is, or with minimal modifications
 - I regenerated all processes with master_june24 codegen (so that the old CI can test them)
 - Processes had not been regenerated with the latest codegen, unlike what we had agreed long ago
 - I included the new CI tmad tests (so that any issues there immediately show up)
 - NB: tmad tests (cross section and LHE comparisons) were ALREADY available via manual scripts
- Many issues showed up, including *trivial build errors*, and crashes (see next slide)
 - **My opinion: channelid PR #830 was not sufficiently tested upfront**
 - Some issues may also arise because 'warp' modifications in mg5amcnlo for [#765](#) are incomplete
 - **We should agree on a procedure to avoid this happening again in the future...**
 - (at the very least: regenerate the code and ensure all CI tests pass...)
- In the meantime, I am working on fixing these issues, so that we can move on...
 - I will come back to Stefan or Olivier when/if I have questions (I already asked some...)

Master_june24: following up on three weeks ago



- *I confirm my opinion: PR #830 (Sep 2023 – Jun 2024) was insufficiently tested*
 - *There are issues that could have been spotted with existing tests*
 - *There are new features for which new specific tests should have been added*
 - *There are usage assumptions for which new sanity checks should have been added*
 - *Especially, the SIMD implementation in #830 was almost completely wrong*
 - *Some parts of the code were modified and there was no need for that*
- ***Therefore: I essentially reimplemented channelid from scratch in 2 weeks***



Main issues in #830 channelid

(From <https://github.com/madgraph5/madgraph4gpu/pull/882#issuecomment-2239469088>)

Linked issues (limited to 10): the 10 main *major* issues from PR [#830](#) in master_june24

- [🟢 master_june24 tmad tests crashes in dsig1_vec \(CUDA_CPP_RUNTIME_VECSIZEUSED not correctly propagated?\) #885](#) (implement nb_warp_used in mg5amcnlo to avoid crashes when VECSIZE_MEMUSED is set)
- [🟡 master_june24: nomultichannel should be an array nullptr, not an array full of 0s #892](#) (clearly distinguish between 0 and nullptr in multichannel APIs and implementations)
- [🟡 master_june24: wrong/bugged and unnecessary SIMD handling of numerators and denominators #894](#) (fix/remove wrong and unnecessary SIMD handling of numerators and denominators)
- [🟡 master_june24 add runTest \(ME comparison\) tests for two warps with different channel #896](#) (add ME comparison test on warps with different channels to be able to test the channelid functionality)
- [🟡 master_june24: missing sanity checks that all channelids are the same in a "warp" \(or at least inside each SIMD vector\) #898](#) (add sanity checks that the same channel is used in one SIMD vector)
- [🟡 master_june24: major bug in memory access for channelids in C++ calculate_wavefunctions \(missing ieventAccessRecordConst\) #899](#) (add missing ieventAccessRecordConst for channelid in calculate_wavefunction)
- [🟡 master_june24: wrong memory access for channelids in C++ sigmakin \(missing ieventAccessRecordConst\) #911](#) (add missing ieventAccessRecordConst for channelid in sigmaKin)
- [🟡 master_june24: fbridgesequene_nomultichannel is not used #914](#) (use the fbridgesequene_nomultichannel function)
- [🟡 master_june24: fbridge.inc interface was not modified #916](#) (fix the fbridge.inc interface describing the Bridge call sequence in Fortran)
- [🟡 master_june24: add LHE color/helicity comparison test for 2 warps with different channel, fix mixed mode \(two SIMD vectors must have same channel\) #924](#) (check that the same channel is used in two SIMD vectors in mixed precision mode)

Other issues in #830 channelid

(From <https://github.com/madgraph5/madgraph4gpu/pull/882#issuecomment-2239469088>)

Other issues from PR [#830](#) in master_june24:

- [✔ master_june24 builds fail for MAC SIMD #883](#) (fix failing MAC SIMD builds)
- [✔ master_june24 builds fail for FPTYPE=m #884](#) (fix failing FPTYPE=m builds)
- [✔ master_june24: Fortran runtime error: Index '32765' of dimension 1 of array 'symconf' above upper bound of 3 #888](#)
(Olivier's clarification about symconf Fortran array out of bounds)
- [✔ master_june24: xsec mismatch in gg_tggg for FPTYPE=f \(must increase tolerance\) #889](#) (increase FP tolerance in cross section comparison)
- [✔ master_june24: hack in counters.cc hides a real issue in auto_dsig1.f #891](#) (hack in counters.cc hides a real issue in auto_dsig1.f)
- [✔ master_june24: __CUDACC__ macros prevent HIP support #893](#) (replace CUDA-only statements to allow AMD HIP support)
- [✔ master_june24: CID_ACCESS::kernelAccessConst should be called only once \(it is now called once per diagram\) #895](#)
(kernelAccess should not be called for each Feynman diagram)
- [✔ master_june24: PR 830 was merged still including merge conflicts #901](#) (code was committed to repo without fixing merge conflicts)
- [✔ master_june24: 'error: unhandled SGPR spill to memory' in gg_tggg builds on LUMI/HIP #933](#) (skip gg_tggg HIP builds at LUMI as they run out of memory)

New issues added (and fixed) in my own PR

(From <https://github.com/madgraph5/madgraph4gpu/pull/882#issuecomment-2239469088>)

Additional feature enhancements (and fixes for my own bugs) that I developed while debugging 830:

- [✔ Add comparison of selected helicity and color to runTest #925](#) (major, add helicity/color comparison in runTest)
- [✔ in codegen, add internal consistency checks for the number of diagrams #910](#) (add sanity checks about ndiagrams from code generation)
- [✔ segfault in gg_ttgg runTest.exe in my 'june24' WIP #917](#) (runTest should only use channels with associated iconfig)
- [✔ Clarified: gg_ttggg has 1240 diagrams in CPPProcess.cc but only 1122 entries in channel2iconfig \(and 945 config\) #919](#) (sanity checks about ndiagrams vs nchannels from code generation)
- [✔ issues in gg_ttggg runTest.exe in my 'june24' WIP #920](#) (fix diagram loop bug in runTest)
- [✔ issues in tput \(and tmad\) tests in my 'june24' WIP #922](#) (improvements in tput/tmad scripts)
- [✔ MG5AMC_CARD_PATH feature fails if it points to the Cards directory #923](#) (fixes for param card path handling)
- [✔ selected color comparison fails for HIP on LUMI #931](#) (fixes for helicity/color selection causing failures on HIP/LUMI)

Pending channelid issues (~for later)

(From <https://github.com/madgraph5/madgraph4gpu/pull/882#issuecomment-2239469088>)

PS I was forgetting. There are also some issues which are NOT YET fixed here, but I suggest that we consider them later, after merging this branch into master.

Unsolved issues from PR [#830](#) in master_june24:

- [🕒 URGENT - master_june24 does not use gpucpp_june24 \(and upgrading fails codegen\) #886](#) (master_june24 does not use the mg5amcnlo gpucpp_jun24 branch -> for Olivier)
- [🕒 master_june24: document WARP_SIZE in vector.inc \(clarified: WARP_SIZE vs VECSIZE_MEMMAX\) #887](#) (document WARP_SIZE vs VECSIZE_MEMMAX in vector.inc)

Followup to discussion on some of the issues solved from 830 (and possible future improvements of those patches)

- [🕒 \(after master_june24\) consider releasing the assumption that two adjacent SIMD vectors have the same channelid in mixed mode #926](#) (consider allowing different channelIds for two adjacent SIMD vectors in mixed precision mode)
- [🕒 \(after master_june24\) ensure that some madevent tests use several channels #927](#) (ensure that some madevent tests use more than one channel, testing the channelid functionality)
- [🕒 \(after master_june24\) add nb_warp_used support in banner.py etc in mg5amcnlo #928](#) (complete the nb_warp_used patch, e.g. with banner.py fixes)
- [🕒 \(after master_june24\) add sanity check that channelid is the same in a warp also for GPU? #929](#) (consider adding a sanity check that channelid is the same in a GPU warp)

outlook

Outlook? (my opinion)

- Master – previous major bugs seem to have been solved (pending PR review...)
- Master_june24 – a new channelid implementation with more tests is ready
 - I still do not see how in practice it will speed up madevent, but I think it is ready
 - A PR under review is ready to be merged into master_june24
 - Then the new master_june24 (or the original PR) can be merged into master
- Master – packaging of the submodule will be my next priority
 - And after that I think a release starts to be on the horizon
- The issues we discussed with CMS remain pending
 - DY+4jets xsec mismatch fortran/cudacpp?
 - DY+3jets cudacpp slower than fortran?
 - But they can probably come after a release?