



Madgraph4gpu progress: colour and helicity choice, to-do towards alpha release

Andrea Valassi (CERN)

Madgraph on GPU development meeting, 9th January 2023

<https://indico.cern.ch/event/1228816>

Random choice of helicity/colour is done: overview

- Done in the .mad directories in madgraph4gpu from 16 Dec 2022 (MR [#573](#))
- A LOT of work needed on the “upstream” mg5amcnlo
 - Pick up Olivier’s helicity/colour changes from nuvecMLM to vecMLM (madgraph4gpu MRs)
 - Adapt my own CODEGEN to the new mg5amcnlo (madgraph4gpu MRs)
 - Start porting my CODEGEN into mg5amcnlo (mg5amcNLO and madgraph4gpu MRs)
 - More details in the next slide...
 - ... but a lot of cleanup is needed! (even more than I thought in December)
- Random choice of helicity and of colour are now both done
 - Additional madgraph4gpu MRs on top of the above
 - NB: color numbering is copied brute-force from Fortran (coloramps.inc to coloramps.h)
 - Same cross section and **same LHE files in Fortran and in CUDA or C++**, at last!
 - No need to “massage” the LHE files to filter out helicity/colour comparison any more
 - This is for the few usual processes we work with (e.g. gg to ttg*, NOT pp to ttg* yet...)

TO-DO before an alpha release in random order

- See my suggested list of items on issue [#576](#)
- (1) Validate that the numbering in coloramps.inc/h is the correct one (OM)
- (2a) Agree (AV+OM) on what we want in the Fortran
 - En passant, integrate Olivier’s “lo_vect_diag” if still needed (my bad, I missed that...)
 - Cleanup of my mg5amcnlo MRs...
 - NB: this is needed even if we give pre-cooked gridpacks to the experiments
- (2b) Merge as much as possible of my changes upstream
 - Strictly needed only if we do not give gridpacks to the experiments
 - BUT: not much more effort than above, and desperately needed to avoid further divergence
- (3) Clean up build and runtime environment for seamless “launch” integration
 - madevent input files, executable names, SIMD build options...
- (4) Cross check all is ok for physics parameter choices
- (5) TEST! The “launch” command as in the experiment frameworks
- (6) More processes, in particular pp to gg etc

Aside: quick tests of OMP multithreading

- Done in the .mad directories in madgraph4gpu from 19 Dec 2022 (MR [#577](#))
 - This is currently the latest status of master
- Motivation
 - OMP MT seemed to work nicely in the latest Fortran code too
 - The OMP pragmas had to be moved anyway with the latest code, minimal effort
- Details in issue [#575](#)
 - *Using 32 threads with AVX2 on a node with 16 physical cores and 2x hyperthreading*
 - *Maximum speedup of x14.5 with respect to a single core with AVX2*
 - Scalability depends a lot on number of events processed (it gets better with more events)
 - No comparison to multi process yet, no plots yet, no comparison to Kokkos/Sycl yet...
 - IMO this is still a proof of concept prototype which needs more detailed studies