UCLouvain
Institut de recherche en mathématique et physique
Centre de Cosmologie, Physique des Particules et Phénoménologie

GPU

Olivier Mattelaer

Helicity recycling problem

1 scan1:[1 if i in [0,3] else 0 for i in range(4)] #
2-7 0.000
8 scan1:[1 if i == 1 else 0 for i in range(4)] #
9 scan1:[1 if i == 2 else 0 for i in range(4)] #
10 0.0000

Cross-section was not identical for the two identical benchmark

Issue was because for the first parameter, they are only one helicity contributing so helicity recycling is bypassed (ok first time)

Benchmark #2 is normal -> helicity recycling used

Benchmark #3 has vanishing matrix-element -> special treatment

Benchmark #1 has one helicity but failed the bypassed due to the special treatment of previous benchmark and the makefile fail to recompile the matrix-element -> bias in the cross-section

Aloha module

- Helicity wave functions are represented by an array
  - $[P_0 + iP_3, P_1 + iP_2, W_1, W_2, W_3, W_4]$
- Change it to a fortran structure

```
module ALOHA_OBJECT
   TYPE ALOHA
      double complex::W(6)
      double precision :: P(0:3)
   END TYPE ALOHA
end module ALOHA_OBJECT
```

- Aloha modified accordingly (but no real change)
  - gg>tt~3g is 10% faster

Flavor

- Starting to have a function merge_flavor

- Merging multiple particle to a generic particle with flavor index (like helicity)
  - Coupling will depend of the flavor index of particles (ckm will be challenging)
- Model will have "generic" interaction for such generic particle

```
MG5_aMC>display interactions _quark
15 : _anti_quark _quark a  {'QED': 1} base
17 : _anti_quark _quark g  {'QCD': 1} base
20 : _anti_quark _quark z  {'QED': 1} base
22 : _anti_quark _quark w+  {'QED': 1} base
31 : _anti_quark _quark w-  {'QED': 1} base
```

WARNING

- This is WIP
- Only DIAGRAM generation is working
  - Or is it?
    - Not working for CKM model
- Not yet implemented the structure for the coupling/...
- No standalone validation (neither madevent)

Flavor

- Starting to have a function merge_flavor

Current MG5

Define j = g u u~ d d~ c c~ s s~
generate g g > w+ w- p p QED=2
Output

Two directories

2*31 diagrams

Flavor version

Define j = g _quark _anti_quark
generate g g > w+ w- j j QED=2
Output

One directory

46 diagrams

Flavor

- Why 46 and not 31 diagrams?

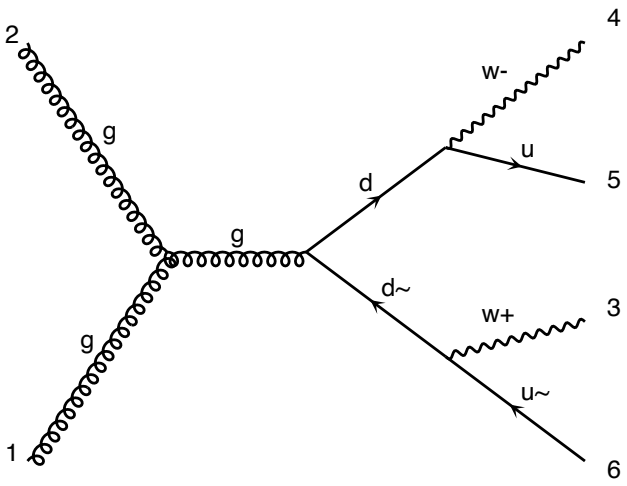


diagram 6 QCD=2, QED=2

Only exists for u u~ final state

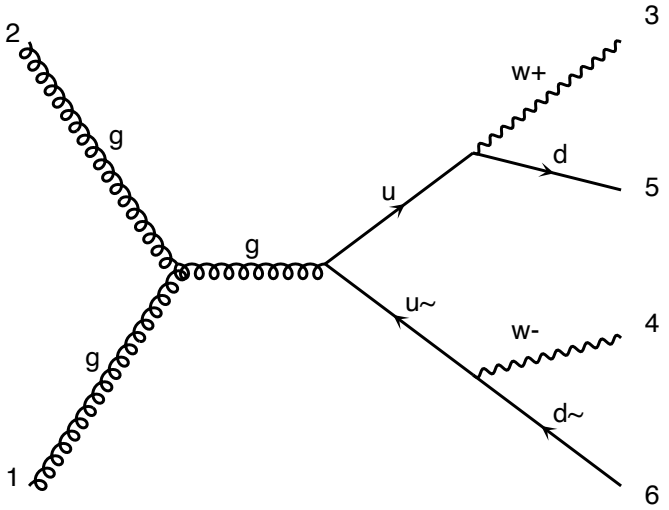
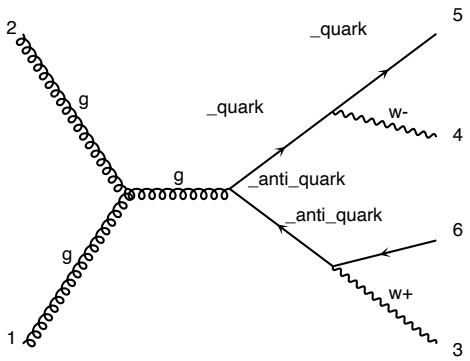
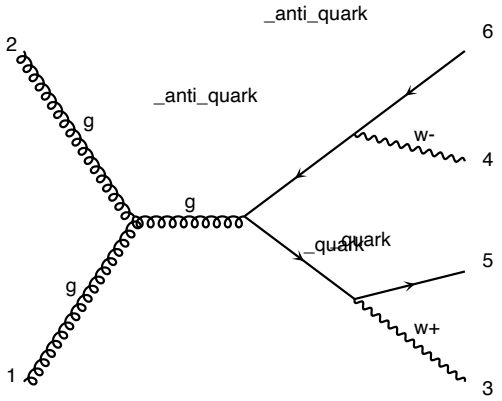


diagram 6 QCD=2, QED=2

Only exists for d d~ final state





So we need BOTH in generic qq~ final state (if not smart?)

Flavor p p > j j QCD=0

Current MG5

Directory	# Diagrams	# Subprocesses
P1_qq_qq	4	2
	4	4
	4	2
	4	4
	4	2
	4	2
	3	4
	3	4
	3	4
	3	4
	3	4
	3	4
	2	2
	2	4
	2	4
	2	4
	2	4
	2	2
	2	4
	2	4
	2	4
	2	4
	2	2
	2	4
	2	2
	1	4
	1	8
	1	4
	1	4
	1	4

So in GPU version:

31 directories

Flavor version

Directory	# Diagrams	# Subprocesses
P1_qq_qq	8	1
	8	2
	8	1

So in GPU version: 3 directories

Number of directory

	MG5 matrix GPU dir	Fortran dir	Flavor	Q -> q~ (Master Student)	S->T (No plan)
pp>jj QED=0	12	5	8	5	3
pp> jj QCD=0	31	1	3	1	1
DY3j	30	5	9	5	2
Tt~3j	16	6	10	6	3
W+w-jj QCD=0	32	1	3	1	1