

# Mismatch DY

## DY+nj problems

See [MADGRAPH5/MADGRAPH4GPU#944](#).

## Introduction

See [PREVIOUS REPORT](#).

## Run setup

Gridpacks are in

```
/cvmfs/cms.cern.ch/phys_generator/gridpacks/  
GPUtest
```

, and they are packaged as `.tar.xz` files. Inside them one can find the cards in `InputCards/` directory.

## Generation setup

Process card:

```
import model sm-no_b_mass  
define p = g u c d s b u~ c~ d~ s~ b~  
define j = p  
define ell+ = e+ mu+ ta+  
define ell- = e- mu- ta-  
define nu = ve vm vt  
define nubar = ve~ vm~ vt~  
generate p p > ell+ ell- j j j @@  
output DYnj
```

When running, set the options in the `run_card.dat` accordingly:

- `sde_strategy` to 1 or 2;
- `fixed_ren_scale` and `fixed_fac_scale` to `True` or `False`.

## DY+3j

### Tests setup

- run on local machine
- `mg5amcnlo@2acd31be96fa9045f3f319bb2e5058c7ba1a5e2a`, tag: `3.5.5`
- LHAPDF 6.5.4 and the `NNPDF23_lo_as_0119_qed` PDF set
- Python 3.7.17
- Fortran (GCC) 14.1.1
- 10k generated events

Runs were executed using the same `run_card.dat` that has been used by Jin, in particular, when using the upstream FORTRAN, I copied all the cuts, for example:

- [run\\_card.dat used](#);
- [Jin's run\\_card.dat](#).

Additionally, I performed runs with 4 configurations:

- `sde_strategy` = `[1, 2]`;
- `fixed_ren_scale`, `fixed_fac_scale` = `[False, True]`.

### Results

fixed_ren_scale, fixed_fac_scale	sde_strategy	
	1	2
False	1380 +- 3.2	1506 +- 4.1

fixed_ren_scale, fixed_fac_scale	sde_strategy	
	1	2
	1385 +- 3.3	1512 +- 4.5
	1391 +- 3	1519 +- 3.8
	1394 +- 3.4	1511 +- 3.7
	1388 +- 3.1	1512 +- 3.8
average:	1387.6 +- 1.4	1512.0 +- 1.8
True	1477 +- 3.5	1542 +- 4.1
	1475 +- 3.5	1545 +- 4.5
	1474 +- 3.5	1544 +- 4.2
	1475 +- 3.6	1545 +- 3.7
	1471 +- 3.4	1547 +- 4.3
average:	1474.4 +- 1.6	1544 +- 1.9

### Jin's results:

- Jin's FORTRAN: 1365 +- 1.265 pb
- Jin's CUDA: 1385 +- 0.3288 pb

## DY+4j

### Tests setup

- Done on `itscrd-v100.cern.ch`
- `mg5amcnlo@2acd31be96fa9045f3f319bb2e5058c7ba1a5e2a`, tag: `3.5.5`
- LHAPDF 6.5.4 and the `NNPDF23_lo_as_0119_qed` PDF set
- Python 3.9.18

- Fortran (GCC) 14.1.1
- 10k generated events

**Important:** need to use LHAPDF with the correct PDF also for the calculation of the cross-section, because otherwise the results are not comparable.

Results with the same `run_card.dat` as Jin's:

- [run\\_card.dat used](#);
- [Jin's run\\_card.dat](#).

## Results

fixed_ren_scale, fixed_fac_scale	sde_strategy	
	1	2
False	839.3 +- 2	938.1 +- 2.2
True	932.8 +- 2.1	973.9 +- 2.5

### Jin's results:

- Jin's FORTRAN: 883.4 +- 0.3813 pb
- Jin's CUDA: 843.8 +- 0.2022 pb

## Comments

It seems to me the `sde_strategy = 1` with not fixed scale reproduces well the CUDA results. However, I'm a bit worried about the differences with the choice of `sde_strategy` values.