MG5aMC tutorial

NLO / Loop induced / MLM

Olivier Mattelaer
IPPP/Durham

What is MadGraph5_aMC@NLO

MadGraph5

- Cross-section @LO
- event generation @LO
  - including loop-induced
- Interface to
  - Pythia6/8
  - Delphes
  - MadSpin
- Any BSM model supported

_

aMC@NLO

- Cross-section @NLO
- event generation @NLO+PS
- Interface to
  - PY6/PY8/HW6/HW++ HW7
  - MadSpin
- BSM in quick expansion
  - Framework ready

Ex. I: Install MadGraph 5!

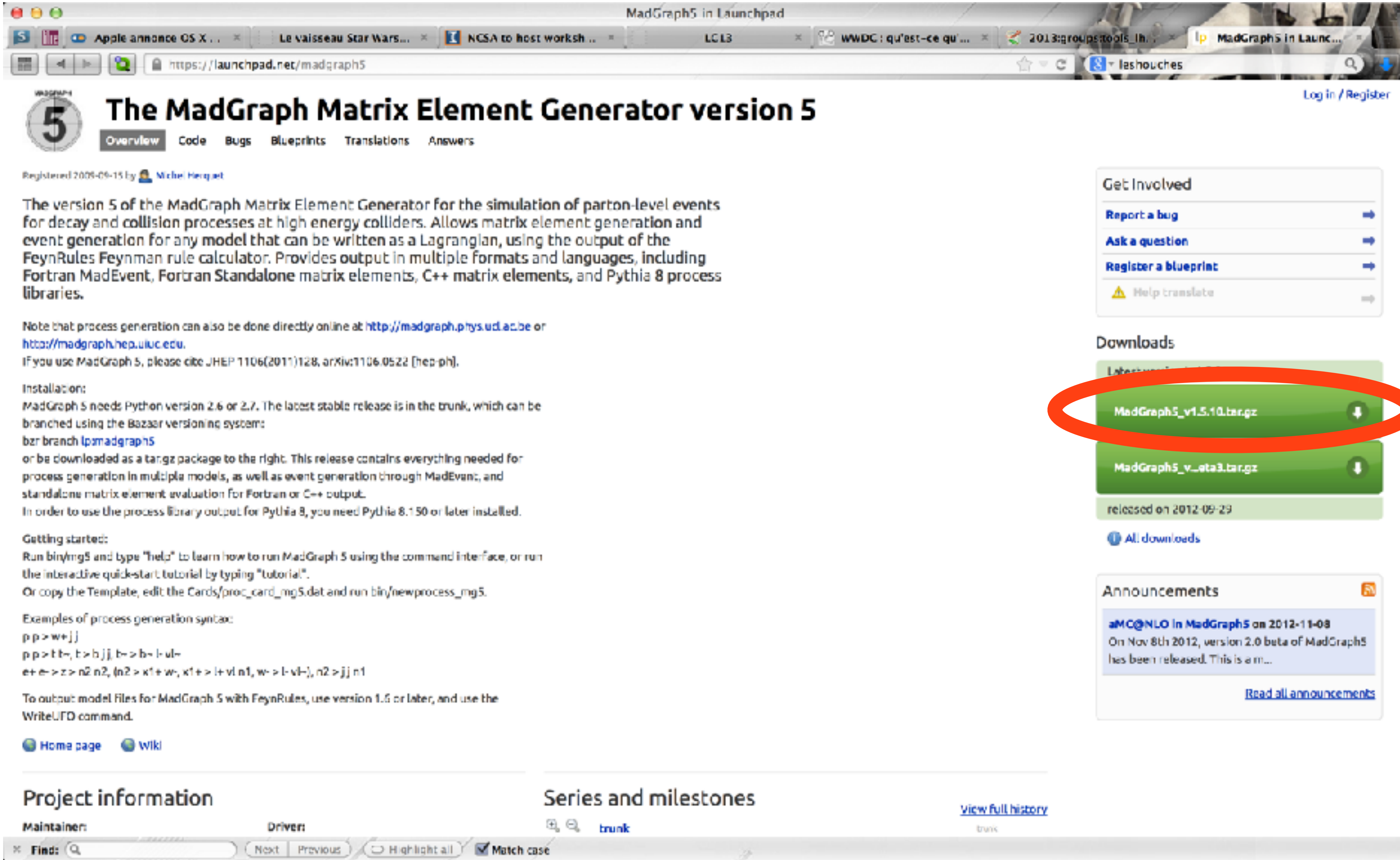
- https://launchpad.net/madgraph5
- untar it (tar -xzpvf MG5_XXX.tgz)
- launch it ($ ./bin/mg5_amc)

- learn it!
  - ➡ Type tutorial and follow instructions

- install external package
  - ➡ install pythia8
  - ➡ install MadAnalysis

- put MC4BSM UFO model from indico in the models directory

Install


MadGraph5 in Launchpad
https://launchpad.net/madgraph5
Log in / Register
The MadGraph Matrix Element Generator version 5
Overview
Code
Bugs
Blueprints
Translations
Answers
The version 5 of the MadGraph Matrix Element Generator for the simulation of parton-level events for decay and collision processes at high energy colliders. Allows matrix element generation and event generation for any model that can be written as a Lagrangian, using the output of the FeynRules Feynman rule calculator. Provides output in multiple formats and languages, including Fortran MadEvent, Fortran Standalone matrix elements, C++ matrix elements, and Pythia 8 process libraries.
Note that process generation can also be done directly online at http://madgraph.phys.ucl.ac.be or http://madgraph.hep.uiuc.edu.
If you use MadGraph 5, please cite JHEP 1106(2011)128, arXiv:1106.0522 [hep-ph].
Installation:
MadGraph 5 needs Python version 2.6 or 2.7. The latest stable release is in the trunk, which can be branched using the Bazaar versioning system:
bzr branch lp:madgraph5
or be downloaded as a tar.gz package to the right. This release contains everything needed for process generation in multiple models, as well as event generation through MadEvent, and standalone matrix element evaluation for Fortran or C++ output.
In order to use the process library output for Pythia 8, you need Pythia 8.150 or later installed.
Getting started:
Run bin/mg5 and type "help" to learn how to run MadGraph 5 using the command interface, or run the interactive quick-start tutorial by typing "tutorial".
Or copy the Template, edit the Cards/proc_card_mg5.dat and run bin/newprocess_mg5.
Examples of process generation syntax:
To output model files for MadGraph 5 with FeynRules, use version 1.6 or later, and use the WriteUFO command.
Home page
Wiki
Project information
Maintainer:
Driver:
Series and milestones
View full history
trunk
Get Involved
Report a bug
Ask a question
Register a blueprint
Help translate
Downloads
MadGraph5_v1.5.10.tar.gz
released on 2012-09-29
All downloads
Announcements
aMC@NLO in MadGraph5 on 2012-11-08
On Nov 8th 2012, version 2.0 beta of MadGraph5 has been released. This is a m...
Read all announcements
Find:
Next
Previous
Highlight all
Match case

Where to find help?

- Ask me
- Use the command "help" / "help XXX"
  - ➡ "help" tell you the next command that you need to do.
- Launchpad:
  - ➡ https://answers.launchpad.net/madgraph5
  - ➡ FAQ: https://answers.launchpad.net/madgraph5/+faqs

Exercise II: Cards Meaning

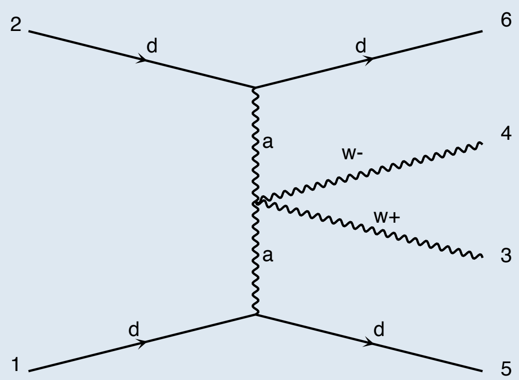
- How do you change
  - ➡ top mass
  - ➡ top width
  - ➡ W mass
  - ➡ beam energy
  - ➡ pt cut on the lepton

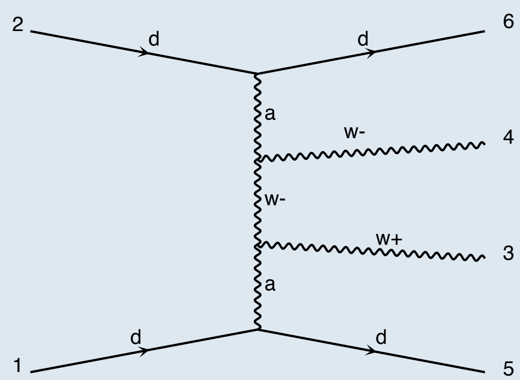
Ex. II : Order

Goal
- What's the default choice for QED/QCD order

Learn
- What's the difference between
  - ➡ p p > t t~
  - ➡ p p > t t~ QED=0
  - ➡ p p > t t~ QED<=2
- Compute the cross-section for each of those and check the diagram

Check
- Generate VBF process (two jet + two W in final state) only the diagram!
- check that you have the QED diagram that you want:





Solution I : Syntax

- What's the meaning of the order QED/QCD
  - ➡ By default MG5 takes the lowest order in QED!

```
INFO: Trying coupling order WEIGHTED<=2: WEIGTHED IS 2*QED+QCD
```

  - ➡ p p > t t~ IS the same as p p > t t~ QED=0
  - ➡ p p > t t~ QED=2 has additional diagrams (photon/z exchange)

p p > t t~

Cross section (pb)
555 ± 0.84

p p > t t~ QED=2

Cross section (pb)
555.8 ± 0.91

No significant QED contribution

Solution I Syntax

- generate p p > w+ w- j j
  - ➡ 76 processes
  - ➡ 1432 diagrams
  - ➡ None of them are VBF

- generate p p > w+ w- j j QED = 2
  - ➡ 76 processes
  - ➡ 1432 diagrams
  - ➡ None of them are VBF

- generate p p > w+ w- j j QED = 4
  - ➡ 76 processes
  - ➡ 5332 diagrams
  - ➡ VBF present! + those not VBF

- generate p p > w+ w- j j QCD = 0
  - ➡ 60 processes
  - ➡ 3900 diagrams
  - ➡ VBF present!

- generate p p > w+ w- j j QCD = 2
  - ➡ 76 processes
  - ➡ 5332 diagrams

- generate p p > w+ w- j j QCD = 4
  - ➡ 76 processes
  - ➡ 5332 diagrams

Ex. III: Syntax

- Generate the cross-section and the distribution (invariant mass) for
  - ➡ p p > e+ e-
  - ➡ p p > z, z > e+ e-
  - ➡ p p > z > e+ e-
  - ➡ p p > e+ e- $ z
  - ➡ p p > e+ e- / z

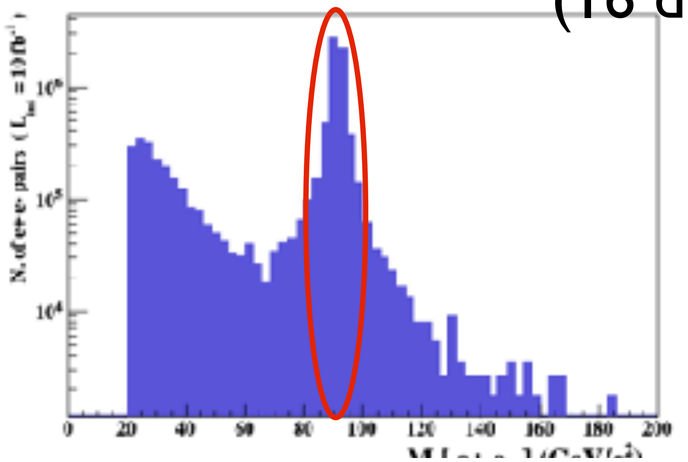
Hint :To plot automatically distributions:

mg5> install MadAnalysis

- Use the invariant mass distribution to determine the meaning of each syntax.

p p > e+ e-

(16 diagrams)

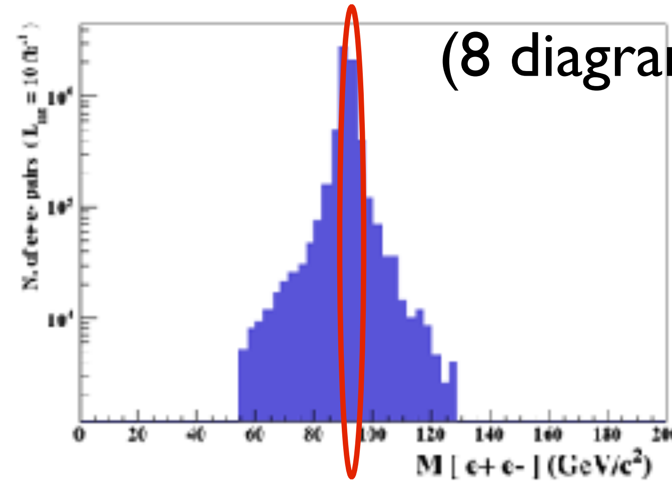


Correct Distribution

Z Peak

p p >z , z > e+ e-

(8 diagrams)



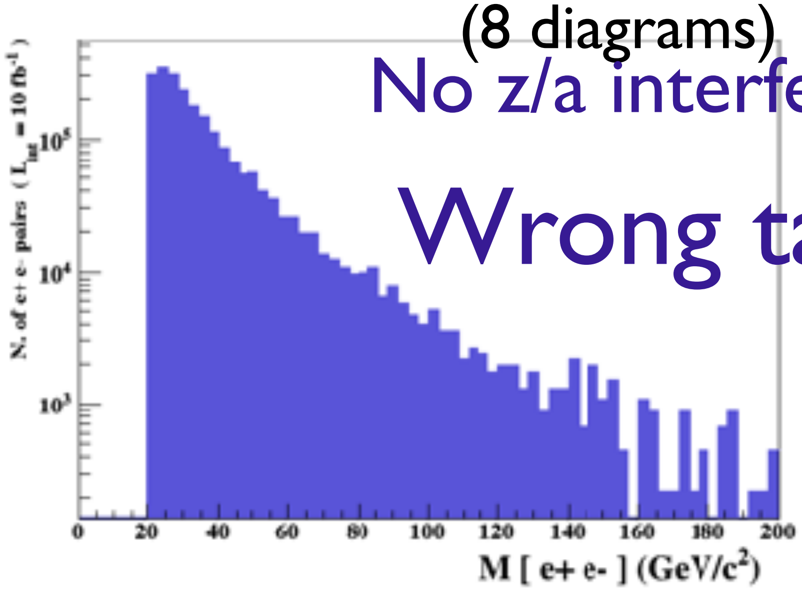
p p > e+ e- /z

(8 diagrams)

No z/a interference

NO Z Peak

Wrong tail



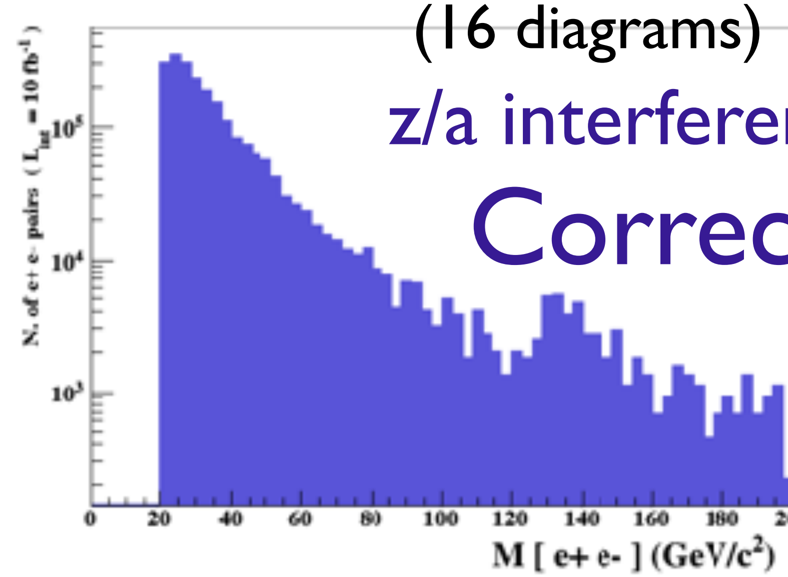
No Z

p p > e+ e- $ z

(16 diagrams)

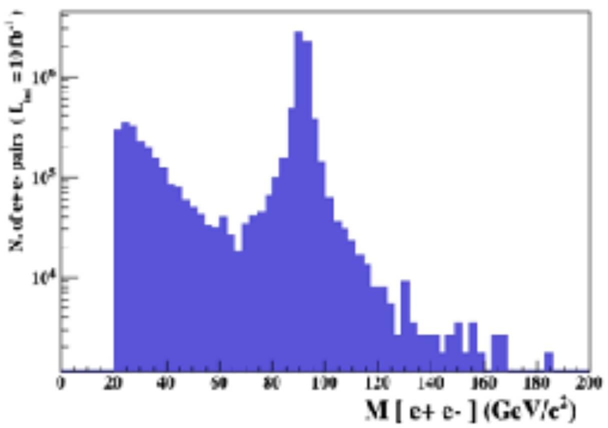
z/a interference

Correct tail



Z- onshell veto

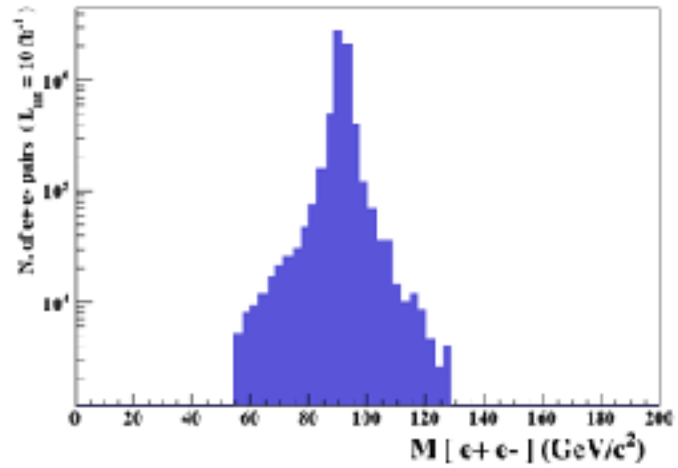
p p > e+ e-



(16 diagrams)

=

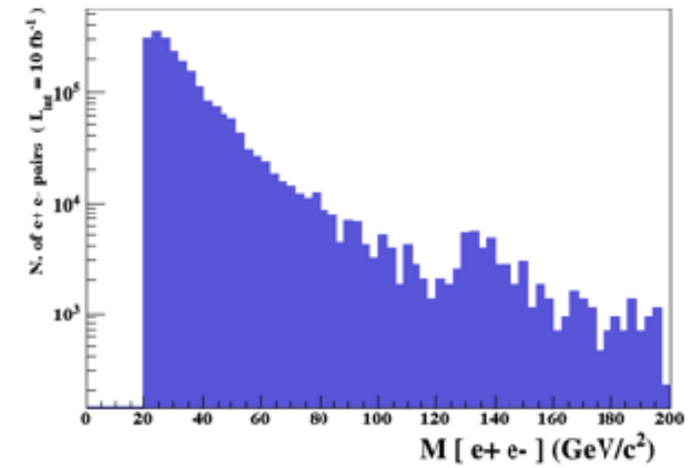
p p >z , z > e+ e-



(8 diagrams)

+

p p > e+ e- $ z



(16 diagrams)

Onshell cut: BW_cut

$$|M^* - M| < BW_{cut} * \Gamma$$

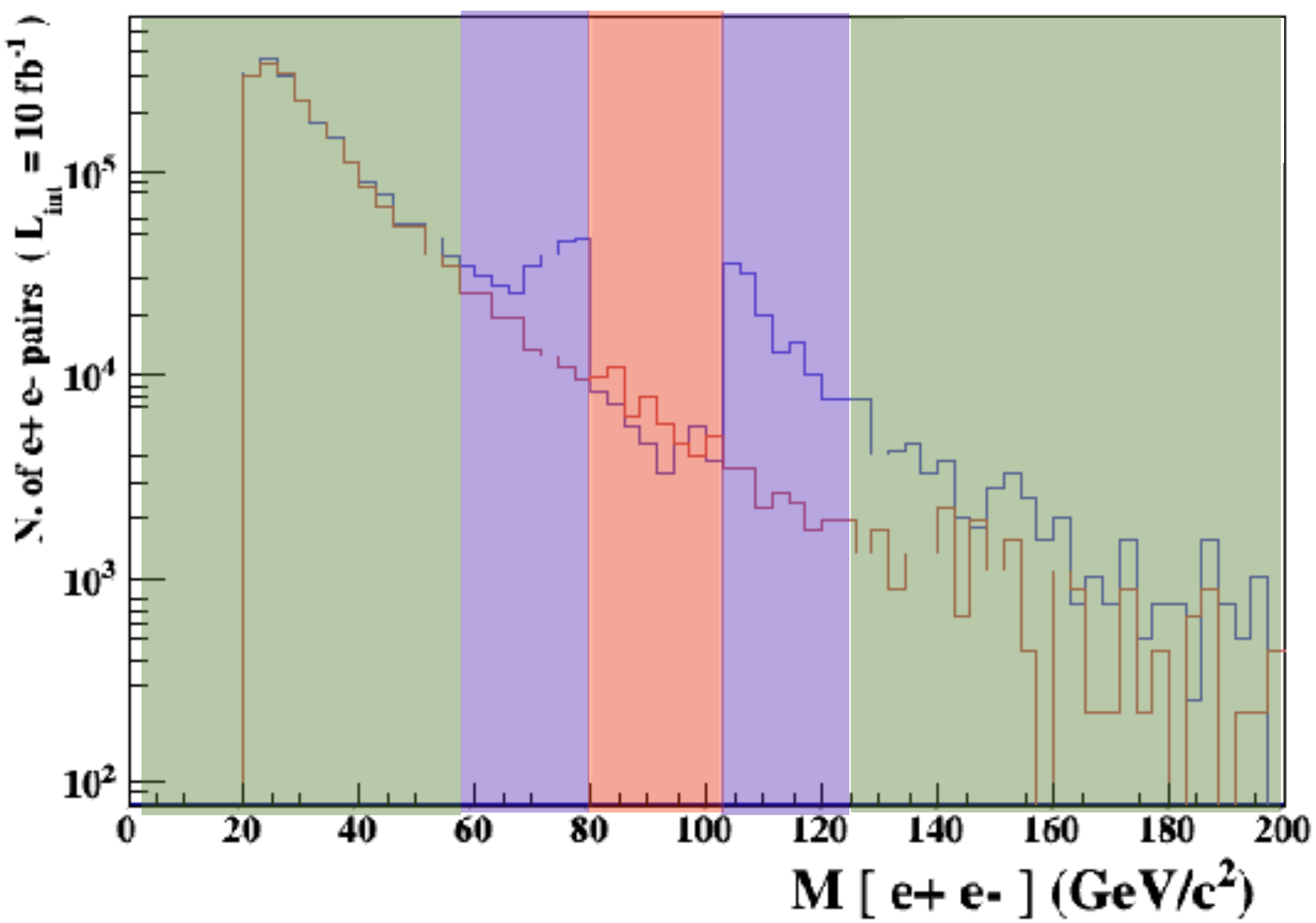
- The Physical distribution is (very close to) exact sum of the two other one.
- The "$" forbids the Z to be onshell but the photon invariant mass can be at MZ (i.e. on shell substraction).
- The "/" is to be avoid if possible since this leads to violation of gauge invariance.

WARNING

- NEXT SLIDE is generated with bw_cut =5
- This is TOO SMALL to have a physical meaning (15 the default value used in previous plot is better)
- This was done to illustrate more in detail how the "$" syntax works.

See previous slide warning

p p > e+ e- / Z (red curve) adding p p > e+ e- $ Z (blue curve)



- Z onshell veto
- In veto area only photon contribution
- area sensitive to z-peak
- very off-shell Z, the difference between the curve is due to interference which are need to be KEPT in simulation.

5 times width area

15 times width area

>15 times width area

The “$” can be use to split the sample in BG/SG area

- Syntax Like
  - ➡ p p > z > e+ e- (ask one S-channel z)
  - ➡ p p > e+ e- / z (forbids any z)
  - ➡ p p > e+ e- $$ z (forbids any z in s-channel)
- ARE NOT GAUGE INVARIANT !
- forgets diagram interference.
- can provides un-physical distributions.

Avoid Those as much as possible!

check physical meaning and gauge/Lorentz invariance if you do.

- Syntax like
  - p p > z, z > e+ e- (on-shell z decaying)
  - p p > e+ e- $ z (forbids s-channel z to be on-shell)
- Are linked to cut $|M^* - M| < BW_{cut} * \Gamma$
- Are more safer to use
- Prefer those syntax to the previous slides one

Ex. IV : BSM

Goal
- Handling model

Import
- type: `import model MC4BSM`
- Automatic switch to 4/5 flavor computation according to the mass of the b quark
- MGaMC renames sm/susy particles such that their names are identical for any model. This can be avoid by `import model MC4BSM —modelname`

Check
- type:

```
define bsm = uv uv~ ev ev~ p1 p2
check p p > bsm bsm
```

- checks internal validity of the BSM part and consistency of the model (lorentz/gauge)

Ex.V: Decay-Chain

Goal
- understanding decay-chain handling

Exercise
- Compare the cross-section for

```
define evdec = bsm / ev ev~
generate p p > ev ev~
output; launch
```

```
generate p p > ev ev~, ev > evdec all
output; launch
```

This is called the decay chain syntax

```
generate p p > ev > evdec all ev~
output; launch
```

- Use Automatic width computation (for all 3 cases)

```
set width wev Auto
```

To enter at the time of the edition of the cards

- Change the "cut_decays" parameter

```
set cut_decays T
```

Decay-Chain Solution

Goal • understanding decay-chain handling

	Default	Correct width	+cut_decays=T
define bsm = bsm / ev ev~ generate p p > ev ev~ output; launch	19.7 pb	19.6 pb	19.7 pb
generate p p > ev ev~, ev > bsm all output; launch	0.1 pb	19.3 pb	11.8 pb
generate p p > ev > bsm all ev~ output; launch	0.07 pb	11.9 pb	11.9 pb

Remember

- We do not use the BR information. The cross-section depends of the total width
- particle from on shell decay do not have cut by default

Ex VI: Width Computation

Goal
- present the various way to compute the width

- Check with MG the width computed with FR:
  - ➡ generate uv > all all; output; launch
  - ➡ generate ev > all all; output; launch
  - ➡ generate p1 > all all; output; launch
  - ➡ generate p2 > all all; output; launch

FR Number
0.0706 GeV
0.00497 GeV
0 GeV
0.0224 GeV

- Compare with `compute_widths bsm`
- Why the width of uv is zero here?

Function called when width on Auto

- Muv = 400 GeV   Mev = 50 GeV λ=0.1
- m1 = 1GeV   m2 = 100GeV   m12 = 0.5 GeV

Ex VII: Automation

Goal
- script and scan

Parameter scan:
- compute the cross-section for a couple of mass

```
generate p p > ev ev~
```

- for that you can enter for the ev mass:

```
set mev scan:[100,200, 300]
```

```
set mev scan:[100*i for i in range(1,4)]
```

Any python syntax is valid!!

scripting/ other scan:
- write in a file (./MYFILE)
- run it as ./bin/mg5_aMC ./MYFILE

```
import model MC4BSM
generate p p > ev ev~
ouput TUTO
launch
  set nevents 5000
  set LHC 13
launch
  set LHC 14
```

Automation

Goal
- script and scan

Parameter scan:
- compute the cross-section for a couple of mass

```
generate p p > ev ev~
```

- for that you can enter for the ev mass:

```
set mev scan:[100,200, 300]
```

```
set mev scan:[100*i for i in range(1,4)]
```

Any python syntax is valid!!

Comment:
- ONLY for param_card entry!! Use scripting for other type of parameters (run_card,…)
- synchronized scan can be done via

```
set mev scan1:[100,200, 300]
set muv scan1:[200,300,400]
```

Three value will be computed!!

Automation

scripting/ other scan:

- write in a file (./MYFILE)
- run it as ./bin/mg5_aMC ./MYFILE

```
import model MC4BSM
generate p p > ev ev~
ouput TUTO
launch
  set nevents 5000
  set LHC 13
launch
  set LHC 14
```

Comment on scripting

- Do not use ./bin/mg5_aMC < ./MYFILE
- If an answer to a question is not present: Default is taken automatically
- EVERYTHING that you type can be put in the entry file

Exercise VIII: MadSpin

Goal
- Learn MadSpin for Onshell Decay

What is MadSpin

arXiv:1212.3460

- Program to decay on-shell particles
  - Use the NWA and the Branching-ratio
  - keep full spin-correlation
  - keep off-shell effect (up to cut-off)
  - keep unweighted event

Exercise
- generate all decay from ev pair production via MadSpin (and compare with decay-chain syntax)

Exercise VIII: MadSpin

Goal
- Learn MadSpin for Onshell Decay

How to

```
The following switches determine which programs are run:
/------------------------------------------------------------------------------------\
|  1. Choose the shower/hadronization program:                shower = OFF           |
|  2. Choose the detector simulation program:               detector = OFF           |
|  3. Run an analysis package on the events generated:      analysis = MADANALYSIS_5 |
|  4. Decay particles with the MadSpin module:               madspin = OFF           |
|  5. Add weights to events for different model hypothesis: reweight = OFF           |
\------------------------------------------------------------------------------------/
```

When you see this text, type madspin=ON

- Then edit the madspin_card and include

  `decay ev > all all`

- You are done

Note
- Also valid for NLO processes
- sometimes faster/slower than decay-chain

Ex IX: Loop-Induced

Goal
- Learn loop-induced syntax

Ex.
- Compare Large stop limit and full loop

```
import model heft
generate g g > h
output; launch
```

```
import model sm
generate g g > h [QCD]
output; launch
```

```
import model sm-no_b_mass
generate g g > h [QCD]
output; launch
```

Note
- Interface fully identical to LO one
- No decay-chain/MadSpin allowed

Ex 10: NLO

Goal
- Learn NLO syntax

Ex.
- Run the pair-production at NLO

```
import model MC4BSM
generate p p > ev ev~ [QCD]
output; launch
```

Note
- Interface close but different to LO one
  - different options
  - different cuts
- No decay-chain but MadSpin allowed
- Need dedicated model (not all model valid@NLO)

NLO

```
The following switches determine which operations are executed:
 1 Perturbative order of the calculation:                            order=NLO
 2 Fixed order (no event generation and no MC@[N]LO matching):   fixed_order=OFF
 3 Shower the generated events:                                     shower=ON
 4 Decay particles with the MadSpin module:                        madspin=OFF
 5 Add weights to the events based on changing model parameters:  reweight=OFF
  Either type the switch number (1 to 5) to change its default setting,
  or set any switch explicitly (e.g. type 'order=LO' at the prompt)
  Type '0', 'auto', 'done' or just press enter when you are done.
 [0, 1, 2, 3, 4, 5, auto, done, order=LO, ... ][60s to answer]
```

order=LO / order=NLO

- Use this switch to compute K-factor with the exact same settings

fixed_order=ON / fixed_order=OFF

- if ON, we perform a pure NLO computation of the cross-section — no event generation—
- if OFF, we run NLO+PS, with the MC counter-term for a given parton shower —with event generation

Exercise XI: Matching

1. Generate p p > w+ with 0 jets, 0,1 jets and 0,1,2 jets
   (Each on different computers - use the most powerful computer for 0,1,2 jets)
   a. Generate 20,000 events for a couple of different xqcut values.
   b. Compare the distributions (before and after Pythia) and cross sections (before and after Pythia) between the different processes, and between the different xqcut values.
   c. Summarize: How many jets do we need to simulate? What is a good xqcut value? How are the distributions affected?

How to.

- generate the diagram with
  - ➡ generate
  - ➡ add process
- output
- launch
  - ➡ ask to run pythia
  - ➡ In run_card: put ickkw=1
    - ✦ set the value for xqcut
  - ➡ In pythia_card set a value for qcut

- Qcut is the matching scale (the separation between the shower and the matrix element)
- xqcut should be strictly lower (by at least 10-15GeV) than qcut

Solution MLM

Exercise VI: Matching+Merging

	w+0j	w+1j	w+2j	w+3j
no matching	8,35E+04	1,58E+04	8,7E+03	3,5E+03

	1GeV	10GeV	20GeV	50GeV	100GeV	500GeV
w+0	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04
0+1	1,07E+05	9,09E+04	8,91E+04	8,61E+04	8,40E+04	8.35+04
0+1+2	1,12E+05	9,29E+04	9,03E+04	8,66E+04	8,44E+04	8,35E+04
0+1+2+3	1,20E+05	9,47E+04	9,07E+04	8,68E+04	8,40E+04	8,35E+04

Slow → Fast

low efficiency → High efficiency

Exercise VI: Matching+Merging

	w+0j	w+1j	w+2j	w+3j
no matching	8,35E+04	1,58E+04	8,7E+03	3,5E+03

	1GeV	10GeV	20GeV	50GeV	100GeV	500GeV
w+0	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04
0+1	1,07E+05	9,09E+04	8,91E+04	8,61E+04	8,40E+04	8.35+04
0+1+2	1,12E+05	9,29E+04	9,03E+04	8,66E+04	8,44E+04	8,35E+04
0+1+2+3	1,20E+05	9,47E+04	9,07E+04	8,68E+04	8,40E+04	8,35E+04

- No effect of the matching for 0 jet sample.

Exercise VI: Matching+Merging

	w+0j	w+1j	w+2j	w+3j
no matching	8,35E+04	1,58E+04	8,7E+03	3,5E+03

	1GeV	10GeV	20GeV	50GeV	100GeV	500GeV
w+0	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04
0+1	1,07E+05	9,09E+04	8,91E+04	8,61E+04	8,40E+04	8.35+04
0+1+2	1,12E+05	9,29E+04	9,03E+04	8,66E+04	8,44E+04	8,35E+04
0+1+2+3	1,20E+05	9,47E+04	9,07E+04	8,68E+04	8,40E+04	8,35E+04

- matching scale too high only the 0 jet sample contributes => all radiations are from pythia

Exercise VI: Matching+Merging

	w+0j	w+1j	w+2j	w+3j
no matching	8,35E+04	1,58E+04	8,7E+03	3,5E+03

	1GeV	10GeV	20GeV	50GeV	100GeV	500GeV
w+0	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04
0+1	1,07E+05	9,09E+04	8,91E+04	8,61E+04	8,40E+04	8.35+04
0+1+2	1,12E+05	9,29E+04	9,03E+04	8,66E+04	8,44E+04	8,35E+04
0+1+2+3	1,20E+05	9,47E+04	9,07E+04	8,68E+04	8,40E+04	8,35E+04

- matching scale too low. Only highest multiplicity sample contributes and low efficiency

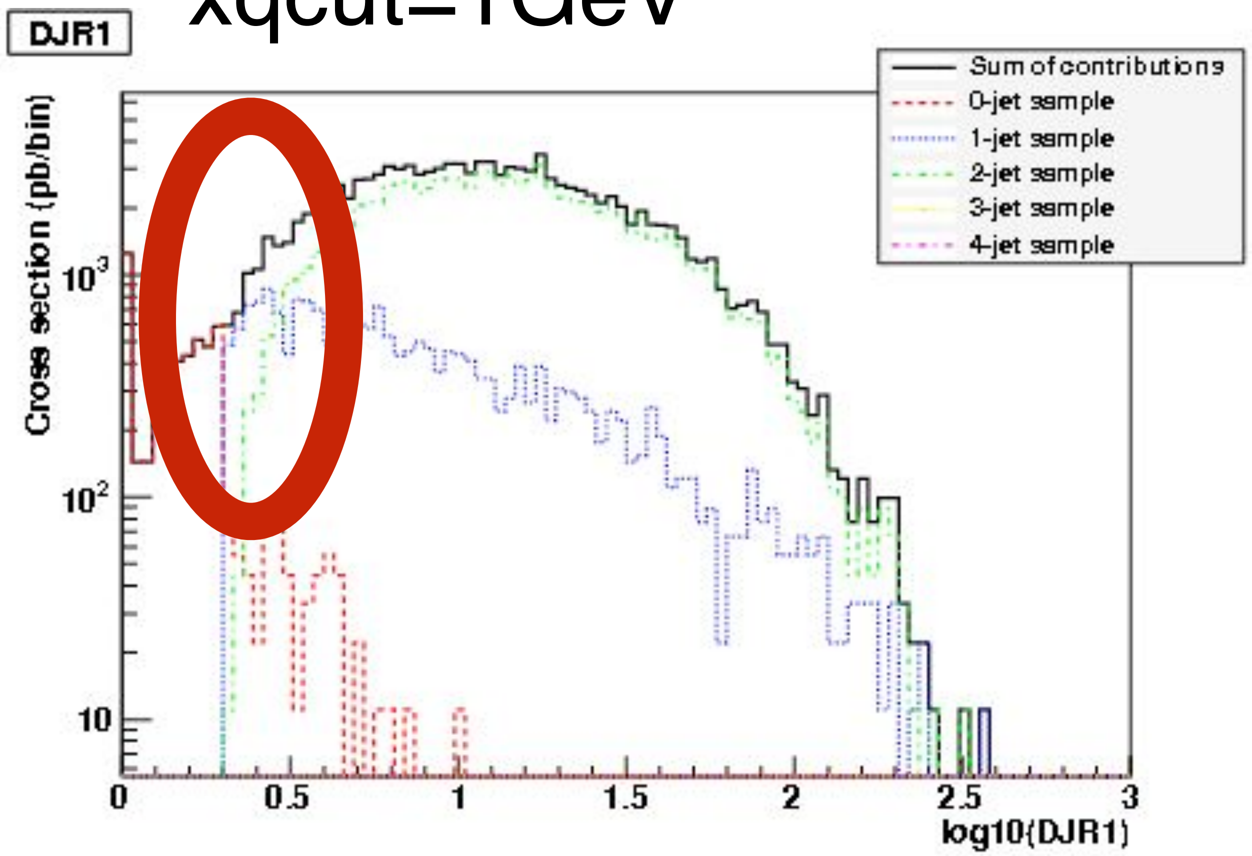
Exercise VI: Matching+Merging

	w+0j	w+1j	w+2j	w+3j
no matching	8,35E+04	1,58E+04	8,7E+03	3,5E+03

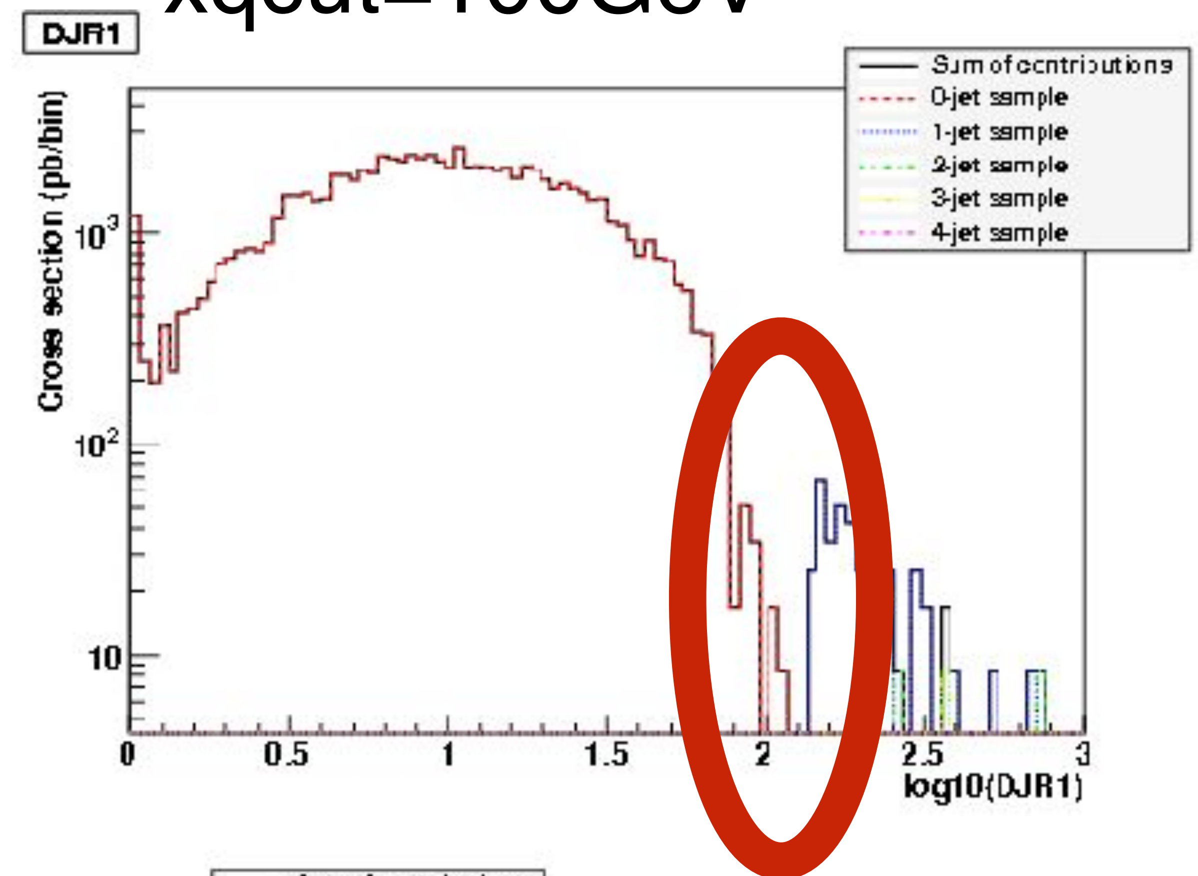
	1GeV	10GeV	20GeV	50GeV	100GeV	500GeV
w+0	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04
0+1	1,07E+05	9,09E+04	8,91E+04	8,61E+04	8,40E+04	8.35+04
0+1+2	1,12E+05	9,29E+04	9,03E+04	8,66E+04	8,44E+04	8,35E+04
0+1+2+3	1,20E+05	9,47E+04	9,07E+04	8,68E+04	8,40E+04	8,35E+04

- Wrong differential rate plot. so to discard.

xqcut=1GeV

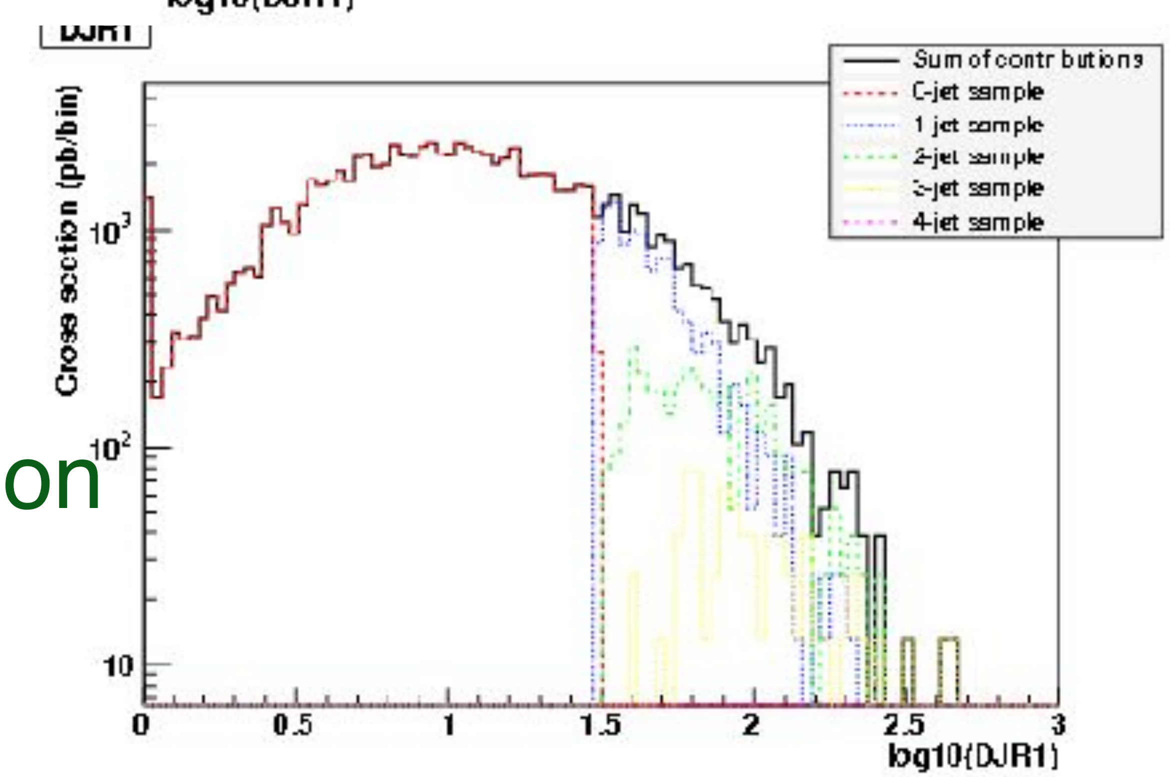


xqcut=100GeV



xqcut=20GeV

smooth transition



Exercise VI: Matching+Merging

	w+0j	w+1j	w+2j	w+3j
no matching	8,35E+04	1,58E+04	8,7E+03	3,5E+03

	1GeV	10GeV	20GeV	50GeV	100GeV	500GeV
w+0	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04	8,35E+04
0+1	1,07E+05	9,09E+04	8,91E+04	8,61E+04	8,40E+04	8.35+04
0+1+2	1,12E+05	9,29E+04	9,03E+04	8,66E+04	8,44E+04	8,35E+04
0+1+2+3	1,20E+05	9,[illegible]7E+04	9,07E+04	8,68E+04	8,40E+04	8,35E+04

- Relatively stable cross-section! Important check.
- Close to the unmatched 0j cross-section