

Short usage introduction to MadGraph 5

Jorge C. Romão
CFTP & Physics Department
Instituto Superior Técnico
Av. Rovisco Pais, 1,1049-001 Lisbon, Portugal

Monday 24th January, 2022

Abstract

We give a short usage introduction to MadGraph 5.

1 MadGraph

MadGraph [1] is a very powerful software framework that aims at providing all the elements necessary for SM and BSM phenomenology, such as the computations of cross sections, the generation of hard events and their matching with event generators, and the use of a variety of tools relevant to event manipulation and analysis. **MadGraph5_aMC@NLO** is the new version of both **MadGraph5** and **aMC@NLO** that unifies the LO and NLO lines of development of automated tools within the **MadGraph** family. As described in their wiki page **MadGraph5_aMC@NLO Wiki**

<https://cp3.irmp.ucl.ac.be/projects/madgraph/>

the processes can be simulated to LO accuracy, and to NLO accuracy in the case of QCD corrections to SM processes. Matrix elements at the tree- and one-loop-level can also be obtained. It can intake models in the **UFO** format that can be produced by other packages such as **FeynRules** and **SARAH**. The software can be downloaded from

<https://launchpad.net/mg5amcnlo>

After downloading follow the instructions on the web page for the installation and read the manual that can be obtained at the the site

<https://cp3.irmp.ucl.ac.be/projects/madgraph/wiki/ManualAndHelp>

In this introductory text we are only to give a very brief introduction on how to use it to evaluate simple processes. As an example we will calculate the tree level cross section for the process $e^-e^+ \rightarrow \mu^-\mu^+$, that we have evaluated with **CalcHEP** and shown in Fig. ??.

MadGraph provides an interactive python interface, which can be accessed by running the main executable. In this interactive interface the command `tutorial` will guide new users to the basic syntax, which we also cover here. We think the tutorial in **MG5** is quite complementary to these notes and talk through a bit more about the parameter and run cards, the extra steps (analysis, hadronisation, and detector simulation) which are not relevant for theoretical inclusive cross sections calculations. Here we will explain how to write and run **MG5** through a script. Suppose we call such a script `eEmM.txt`. Then we in the base directory of **MadGraph5_aMC@NLO** we run the script

```
$ ./bin/mg5_aAC eEmM.txt
```

A minimal working script would be

```
# File eEmM.txt for MadGraph input
generate e- e+ > mu- mu+
output
launch
```

This script illustrates the three main steps needed, `generate`, `output` and `launch`. **MG5** has a plenitude of default options which are a hindrance for this particular exercise, but are reasonable if you study (either an experimental analysis or a phenomenological sensitivity test) requires more features. For instance the browser is set to default because people are supposed to run the Tutorial in the interactive session before doing anything else more complicated. But if you are making a scan, it is better to look only in the end. For this you have to uncomment the line

```
# automatic_html_opening = False
```

in `input/mg5_configuration.txt` by removing the `#`. Second, as you have not given a name for the output, **MadGraph** chooses one. In this case `PROC_sm_0`. If you run again you get `PROC_sm_1` and so on. Notice `_sm` meaning the Standard Model. As one has not given a model it defaults to `sm`. This script is then equivalent to

```
# File eEmM.txt for MadGraph input
import model sm
generate e- e+ > mu- mu+
output
launch
```

The next point is the **ECM**. **MadGraph** defaults to **ECM=1000 GeV** for lepton colliders. This can be changed with the following input file

```
# File eEmM.txt for MadGraph input
import model sm
generate e- e+ > mu- mu+
set lep 100.0
output
launch
```

for an **ECM=100 GeV**. There are several ways of getting the same result. The following commands are all equivalent

```

set lep 100.0
set ebeam 50.0
set ebeam1 50.0
set ebeam2 50.0

```

The next point is that `MadGraph` has some default cuts, that in this case will give a result that does not correspond to what we are looking for. Also we want just the cross section for a given ECM written in a file. The file with all these criteria would look like

```

# File eEmM.txt for MadGraph input
import model sm
generate e- e+ > mu- mu+
output eEmMdir
launch
analysis=OFF
set nevents 1000
set aEWM1 128.023
set MZ 91.29264
set gf 1.1951e-05
set drll 0.0 # No cut on distance between letons
set etal -1.0 # No eta cut for leptons
set ptl -1.0 # No Pt cut for leptons
set lep 100.0
set run_tag 100.0
launch eEmMdir -i
print_results --path=./eEmMdir/cross_section_eEmM.txt --format=short

```

Let us explain the meaning of the new commands. First we gave a name to directory where the information is stored, in this case `eEmMdir`. This stores the Fortran code that will be compiled and used to perform the MC generation and calculation of the cross section, `MG5` is only a python interface for a lot of different C/C++/Fortran routines that are otherwise different software packages. Next we make `analysis=OFF` to disable the `MadAnalysis` as we just want the cross section. Notice that this is optional, you do not need `MadAnalysis` for a full simulation. It exists to derive/compute histograms and to allow for an early-stage data exploration, which for p-p or other discovery machines can be quite interesting (as you have a range of ECM and therefore richer distributions of the final state observables). The following line `set aEWM1 128.023` it is only important because we want to compare with the result of `CalcHEP` that uses $\alpha^{-1} = 128.023$ while `MadGraph` uses $\alpha^{-1} = 132.507$. Beware that `MadGraph` recalculates the W mass for consistency. In fact `MadGraph` allows for the so-called "internal" parameter which is a parameter given by an analytical expressions of other parameters. In the SM this is one of the few cases, but if you have a BSM model, many parameters are often related to another and this way you can generate events without having to numerically enforce those relations. The following table gives the `MadGraph` and `CalcHEP` defaults and the parameters you have to use in `MadGraph` for the SM to compare with `CalcHEP`. So to compare the results with those of `CalcHEP` you have to use the inputs for `MadGraph` in the last column. This correspond to

Par	MadGraph	CalcHEP	Input MadGraph	Input CalcHEP	MadGraph compatible w. CalcHEP
α^{-1}	132.507	128.023	Yes	Yes	128.023
M_Z	91.188	91.29264	Yes	No	91.29264
G_F	1.1664×10^{-5}	1.1951×10^{-5}	Yes	No	1.1951×10^{-5}
M_W	80.419	80.385	No	Yes	Calculated
s_W	Calculated	0.474	No	Yes	Calculated

Table 1: Input values for MadGraph and CalcHEP.

```
set aEWM1 128.023
set MZ 91.29264
set gf 1.1951e-05
```

The next three lines remove the default cuts on leptons. These are mainly used for the experimental situation, but if we want to calculate the theoretical exact cross section these have to be removed, otherwise the phase space will not be completely covered. Finally we want to save the results in a file to be able to use it to make a plot. This is achieved by the last two lines. When we run this script we get

```
# run_name tag cross error Nb_event cross_after_matching
# nb_event_after_matching
run_01 100.0 56.8 0.09264843789832616 1000
```

which means that for $ECM=100$ GeV the cross section is 56.8 ± 0.09 pb. Now we want to repeat this for a range of values for ECM. There is a command `scan`, but it is not very useful here. The best way is to create a script file with all the values of ECM. As this is quite repetitive one can use a shell script to produce the MadGraph script. For instance the following code will give the desired result

```
#!/bin/bash
export LC_ALL=C

file="eEmM_scan.txt"
rm -f $file

touch $file
echo "generate e- e+ > mu- mu+" >> $file
echo "output eEmMdir" >> $file

for i in `seq -f "%.1f" 5 2.5 100`
do
ECM=`expr "$i + $i" | bc`
echo "launch" >> $file
echo "analysis=OFF" >> $file
echo "set nevents 10000" >> $file
echo "set aEWM1 128.023">> $file
echo "set MZ 91.29264">> $file
```

```

echo "set gf 1.1951e-05">> $file
echo "set drll 0.0 # retira a distancia minima entre leptoes" >> $file
echo "set etal -1.0 # retira o cut no eta dos leptoes" >> $file
echo "set ptl -1 # retira o cut no Pt dos leptoes" >> $file
echo "set lep $ECM" >> $file
echo "set run_tag $ECM" >> $file
echo "done" >> $file
echo "" >> $file

done

echo "launch eEmMdir -i" >> $file
echo "print_results --path=./eEmMdir/cross_section_eEmM.txt
--format=short" >> $file

```

This will produce the file `eEmM_scan.txt` that we illustrate below

```

generate e- e+ > mu- mu+
output eEmMdir
launch
analysis=OFF
set nevents 10000
set aEWM1 128.023
set MZ 91.29264
set gf 1.1951e-05
set drll 0.0 # retira a distancia minima entre leptoes
set etal -1.0 # retira o cut no eta dos leptoes
set ptl -1 # retira o cut no Pt dos leptoes
set lep 10.0
set run_tag 10.0
done

launch
...
set lep 15.0
set run_tag 15.0
...
set lep 200.0
set run_tag 200.0
...
launch eEmMdir -i
print_results --path=./eEmMdir/cross_section_eEmM.txt --format=short

```

where the dots mean that we repeat the code. Running this code

```
./bin/mg5_aMC eEmM_scan.txt
```

we get a file `cross_section_eEmM.txt` that we illustrate here

```

# run_name tag cross error Nb_event cross_after_matching
# nb_event_after matching
ECM_2.0 tag_1 24947.000718 90.005 1000
...

```

```
ECM_200.0 tag_1 2.98247 0.0059477821791 1000
```

Using these values we can compare with the result of CalcHEP. This is shown in Fig. 1, showing a perfect agreement.

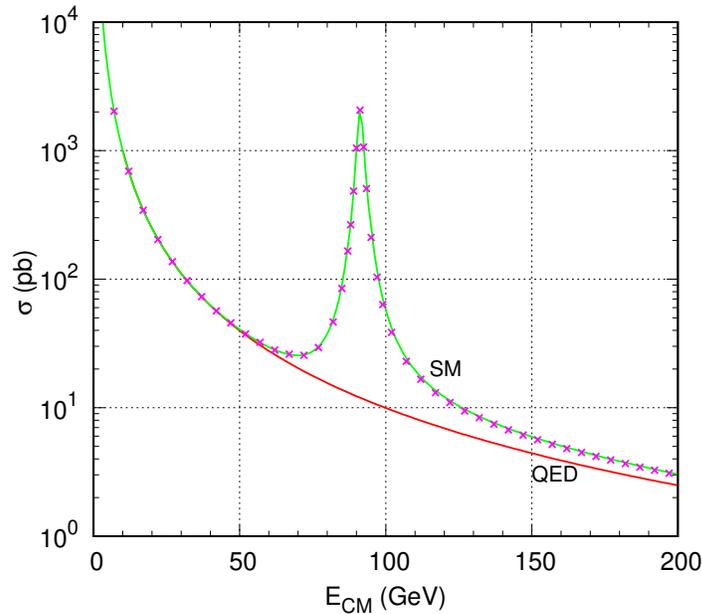


Figure 1: Cross section for $e^+e^- \rightarrow \mu^+\mu^-$ in QED (blue) and the SM (red) calculated with CalcHEP. The result of MadGraph are the magenta crosses.

The reader should read the manual as well as a variety of online presentations to be able to understand the full power of MadGraph.

References

- [1] J. Alwall *et al.*, JHEP **07**, 079 (2014), [1405.0301].