MadEvent

a multi-purpose event generator
powered by MadGraph

Minimal User Guide

madgraph version: V4.1

Authors:

J. Alwall[1], P. Demin[2], S. de Visscher[3], R. Frederix[4], M. Herquet[5], F. Maltoni[6], T. Stelzer[7]

Web Pages:

`http://madgraph.phys.ucl.ac.be`
`http://madgraph.hep.uiuc.edu`
`http://madgraph.roma2.infn.it`

Updated: 01th March 2007

1:johan.alwall@tsl.uu.se,2:pavel.demin@fynu.ucl.ac.be 3:devisscher@fynu.ucl.ac.be,
4:frederix@fyma.ucl.ac.be, 5:mherquet@fyma.ucl.ac.be, 6:fabio.maltoni@cern.ch,
7:tstelzer@uiuc.edu

1 Introduction

MadEvent [1] is a multi-purpose, tree-level event generator which is powered by the matrix element creator MadGraph [2].

In the present version, a process-dependent, self-consistent code for a specific simulation in Standard Model or beyond (MSSM, 2HDM,...) at any collider, *e.g.*, e^-e^+, $\gamma\gamma$, γp ep, pp or still $p\bar{p}$ generated upon the user's request on a web form on following websites:

- `http://madgraph.phys.ucl.ac.be` in Belgium
- `http://madgraph.hep.uiuc.edu` in USA
- `http://madgraph.roma2.infn.it` in Italy

where other websites are also indicated. Given the process, MadGraph automatically creates the amplitudes for all the relevant subprocesses and produces the mappings for the integration over the phase space. This process-dependent information is packaged into MadEvent, and a self containted code is produced that can be downloaded from the web site and allows the user to calculate cross sections and to obtain unweighted events. Alternatively, events can be also generated directly from the web[1], by filling a form and letting the code run over our clusters. Once the events have been generated – event information, (*e.g.* particle id's, momenta, spin, color connections) is stored in the "Les Houches Event Files" (LHEF) format [3]–, may be passed directly to a shower Monte Carlo program (interface is available for Pythia [5]) or be used as inputs for combined matrix element-shower calculations. A series of standard plots and a rootfile contaning the parton level events are also automatically created for events generated over the web. Optionally, rootfiles containing event at pythia or after a generic simulation of a detector (PGS) can be available.

The code is written in Fortran77 and has been developed using the g77 compiler under Linux. The code is parallel in nature and it is optimized to run on a PC farm. At present, the supported PBS batch managing systems are respectively Torque, proPBS and Condor at Italian, American and Belgian clusters. Process specific codes are self contained and therefore do not need any external library. In principle the matrix-element creator can handle any user's request, however, they are limitations of the code are related to the maximum number of final state QCD particles. Currently, the package is limited to ten thousands diagrams per subprocess. So, for example, W+5 jets which has been calculated, is close to its practical limit. The

[1]At present, access possibility as to be asked to the authors

available models are Standard Model with or without Higgs boson or Cabibbo angle, full and simplified 2HDM and MSSM. In addition, effective higgs couplings (HEFT) are implemented (as an option) for 2HDM and SM. For further information about model parametrizations, the user may check materials on `Documents` or `Manual` page on websites.

2 Installation

After the process-dependent code has been generated on the web page (follow the instructions there), download the zipped tar file `madevent.tar.gz` into an empty directory and type:

```
tar -zxvf madevent.tar.gz
```

The following directory structure will appear:

Dir name	Role or Content
`lib`	libraries
`Events`	event files, results, logfiles, plots.
`bin`	executables and scripts
`Source`	process independant source files
`Cards`	steering cards
`SubProcesses`	directories containing subprocesses and process dependant source files
`HTML`	html files to surf on results

The directory `Source` contains itself some subdirectories detailed here below:

- `MODEL`: model implementation.
- `PDF`: parton-distribution libraries.
- `DHELAS`: HELAS 3.1 library.

3 Running

3.1 Basic Scripts

There is only one basic script which should be used for event generation, `generate_events`. This is equivalent to running `survey, refine, unweight_events` one after the other. The script should be called with the following format from main directory:

```
./bin/generate_events run_mode [job_name] run_name
```

where

`run_mode=0,1` the run mode: in parallel (1) or on a single machine (0). When running in parallel (PBS system is supported) the name of the job has to be provided.

`run_name` a word giving the PBS name of the run.

The number of events requested is actually set in the `run_card.dat` file.

If the user prefers not to generate events but only to calculate a cross section, he/she can just run `survey` and `refine`.

- `survey` performs a quick integration over the phase space without adjusting the grids. For instance:

  ```
  ./bin/survey 0
  ```

  means that the user has requested on a single machine

  ```
  ./bin/survey 1 goofy
  ```

  means that the user has requested to run in parallel, giving the jobs name "goofy". The results of this first run (as well as all the others) can be looked up with a browser in the file `SubProcesses/results.html`.

- `refine` performs an integration based on the information gathered by `survey`. It adjusts the grids to improve the efficiency. It asks the user to provide either the number of desired unweighted events (> 1) or the percent accuracy of the integration (< 1); in that case no event is written out. Then, it asks the run mode, parallel (1) or on one machine (0). When running in parallel the name of the job has to be provided. In the latest version, the first time `refine` does not write out events, but only adjust the grids. The second time it keeps the grids fixed and generates events. For instance:

```
./bin/refine 10000 0
```

means that the user has requested to produce 10K unweighted events, running on a single machine.

```
./bin/refine 0.01 1 pippo
```

means that the user has requested a cross section with 1% accuracy, running in parallel, giving the jobs name "pippo". The results of the run can be looked up with a browser in the file `SubProcesses/results.html`. Note that weighted events are stored in `Events/[run_name]_events.lhe.gz` in the Les Houches format.

- `unweight_events` performs the unweighting of the events. The events are stored in the Les Houches format in the file `Events/[run_name]_unweighted_events.dat`.

4 Input Parameters

Running options and cuts are set in the `run_card.dat` data file in the `Card` directory. This superseeds the old way of setting the parameters and cuts, which entailed to modify fortran files in `Source` directory. This now should be avoided, unless special needs (such as a particular event-by-event scale choice or complicated cuts) have to be considered.

4.1 run_card.dat

```
#*********************************************************************
# Tag name for the run (one word)                                    *
#*********************************************************************
  'fermi'     = run_tag ! name of the run
#*********************************************************************
# Number of events and rnd seed                                      *
#*********************************************************************
  10000       = nevents ! Number of unweighted events requested
      0       = iseed   ! rnd seed (0=assigned automatically=default))
#*********************************************************************
# Collider type and energy                                           *
#*********************************************************************
```

```
       1     = lpp1  ! beam 1 type (0=NO PDF)
       1     = lpp2  ! beam 2 type (0=NO PDF)
    7000     = ebeam1  ! beam 1 energy in GeV
    7000     = ebeam2  ! beam 2 energy in GeV
#*********************************************************************
# PDF CHOICE: this automatically fixes also alpha_s and its evol.    *
#*********************************************************************
 'cteq6l1'    = pdlabel     ! PDF set
#*********************************************************************
# Renormalization and factorization scales                           *
#*********************************************************************
 T            = fixed_ren_scale  ! if .true. use fixed ren scale
 T            = fixed_fac_scale  ! if .true. use fixed fac scale
  91.1880     = scale            ! fixed ren scale
  91.1880     = dsqrt_q2fact1    ! fixed fact scale for pdf1
  91.1880     = dsqrt_q2fact2    ! fixed fact scale for pdf2
#*********************************************************************
```

4.1.1 Number of Events and seed

The user can determine the number of unweighted events to be generated by setting the variable `nevents`. `iseed` sets the random number offset of the generation. It should be always left to zero, unless the user aims to reproduce a specific set of results. If left to zero, it is incremented automatically in each run so that successive runs are statistically independent.

4.1.2 Collider energy and beams

The `lpp1` and `lpp2` define the nature of incoming particles and their PDF. If the value is

- -1: PDF of anti-proton
- 0: PDF is switched-off
- 1: PDF of a proton
- 2: PDF for photons

For what concerns photon PDF, the equivalent photon approximation structure function using the improved Weizsaecker-Williams formula[12] has been added in `MadEvent` in order to simulate low virtuality photon emission by proton beams. So for instance, a simulation for Tevatron Run II, is obtained by the following choice:

```
##*****************************************************************
## Collider type and energy                                       *
##*****************************************************************
# 1 = lpp1 !  beam 1 type
# -1 = lpp2 !  beam 2 type
# 980 = ebeam1 !  beam 1 energy in GeV
# 980 = ebeam2 !  beam 2 energy in GeV
```

4.1.3 Parton distributions

The value and the evolution of $\alpha_S(M_Z)$ are by default hardwired with the parton distribution. The user can choose from a collection of modern parton distribution functions, that are included in `Source/PDF`, together with the relative table files in `lib/Pdfdata`.
The distributions, together with their names (the naming scheme is borrowed from `MCFM` [7]), labels, data file name, associated $\alpha_S(M_Z)$ and running order are given in table 1.

name	pdflabel	data file	$\alpha_S(m_Z)$	nloop
MRST2002NLO	`mrs02nl`	mrst2002nlo.dat	0.1197	2
CTEQ6M	`cteq6_m`	cteq6m.tbl	0.118	2
CTEQ6D	`cteq6_d`	cteq6d.tbl	0.118	2
CTEQ6L	`cteq6_l`	cteq6l.tbl	0.118	2
CTEQ6L1	`cteq6l1`	cteq6l1.tbl	0.130	1
CTEQ5M	`cteq5_m`	cteq5m.tbl	0.118	2
CTEQ5L	`cteq5_l`	cteq5l.tbl	0.127	1

Table 1: The MRST and CTEQ family of parton distribution functions.

4.1.4 Scales

Renormalization and factorization scales can be chosen from the `run_card.dat` card. First of all the user has to choose to use a fixed scale fixed or a scale calculated on event-by-event basis.

If the user opts for fixed scales, then the flag in the `run_card.dat` had to be `T` and then the by-default value for scale is used. On the contrary, the user can choose to have a different scale for each event. In that case

$$\mu_F^2 = \mu_R^2 = Q^2(M_{FS}^2 + \sum_j P_T^2) \qquad (1)$$

where Q, M_{FS} and j are respectively the scale factor, the maximum mass of among final states and the index for running over jets and massless visible particles.

In this case, she/he should set the above flags to `F` in the card and edit the routines `set_ren_scale(P,scale)` and `set_fac_scale(P,q2fact)` in `Subprocesses/cuts.f` to calculate the desired value. Several examples are given in the file to help the user.

4.2 Standard Cuts in the `run_card.dat`

Beside prescriptions related to beams characteristic, `run_card.dat` also contains information about cuts (on P_T, $|\eta|$, ΔR,...) to apply at parton level.

```
##*********************************************************************
## Minimum pt's                                                       *
##*********************************************************************
# 20 = ptj !  minimum pt for the jets
# 0 = ptb !  minimum pt for the b
# 20 = pta !  minimum pt for the photons
# 20 = ptl !  minimum pt for the charged leptons
##*********************************************************************
## Maximum rapidity                                                   *
##*********************************************************************
# 2.5 = etaj !  max rap for the jets
# 1000 = etab !  max rap for the b
# 2.5 = etaa !  max rap for the photons
# 2.5 = etal !  max rap for the charged leptons
##*********************************************************************
## Minimum DeltaR distance                                            *
##*********************************************************************
# 0.4 = drjj !  distance between jets
# 0.0 = drbb !  distance between b's
# 0.4 = drll !  distance between leptons
# 0.4 = draa !  distance between gammas
# 0.0 = drbj !  distance between b and jet
# 0.4 = draj !  distance between gamma and jet
# 0.4 = drjl !  distance between jet and lepton
# 0.0 = drab !  distance between gamma and b
# 0.0 = drbl !  distance between b and lepton
# 0.4 = dral !  distance between gamma and lepton
##*********************************************************************
```

```
## Minimum invariant mass for pairs                                  *
##*******************************************************************
# 0 = mmjj !  min invariant mass of a jet pair
# 0 = mmbb !  min invariant mass of a b pair
# 0 = mmaa !  min invariant mass of gamma gamma pair
# 0 = mmll !  min invariant mass of l+l- (same flavour) lepton pair
##*******************************************************************
## Inclusive cuts                                                    *
##*******************************************************************
# 0 = xptj !  minimum pt for at least one jet
# 0 = xptb !  minimum pt for at least one b
# 0 = xpta !  minimum pt for at least one photon
# 0 = xptl !  minimum pt for at least one charged lepton
##*******************************************************************
```

By default a simple set of cuts is applied to **all** final state particles in the event depending on their nature. They are the usual minimal P_T requirement and ΔR defined as $\Delta R = \sqrt{\eta^2 + \phi^2}$ with η the pseudo-rapidity and ϕ the azimutal angle. There are a few remarks:

- By default, only visible leptons and jets (u,d,s,g and b) in final states are submitted to cuts, for example if the process is $pp \to t\bar{t}$ no cuts is applied, on the contrary of $pp \to t\bar{t} \to e^+\nu_e jjb\bar{b}$.
- b quarks are always considered separately than light ones, even if they are considered with the label j in the diagrams generation. So related cuts are `drbb, drbj, drbl, mmbb, xptb, etab` and `ptb`.

4.3 Model parameters in `param_card.dat`

For all models, parameters are set in a `param_card.dat` according to the "Les Houches" format. Here below there is an example of what the user can find in the Standard Model one.

```
Block MODSEL # Select Model
     0   1   #  0 1 = SM
Block SMINPUTS      # Standard Model inputs
     1         1.27900000E+02   # alpha_em(MZ)(-1) SM MSbar
     2         1.16600000E-05   # G_Fermi
     3         1.18000000E-01   # alpha_s(MZ) SM MSbar
     4         9.15445065E+01   # Z mass (as input parameter)
Block MGSMPARAM      # Standard Model parameters for MadGraph
     1         2.31200000E-01   # sin(theta_W)^2
     2         8.02673592E+01   # W mass (as input parameter)
```

```
Block MGYUKAWA     # Yukawa masses m/v=y/sqrt(2)
#    PDG          YMASS
     5         4.20000000E+00   # mbottom for the Yukawa  y_b
     4         1.25000000E+00   # mcharm  for the Yukawa  y_c
     6         1.74300000E+02   # mtop    for the Yukawa  y_t
    15         1.77700000E+00   # mtau    for the Yukawa  y_ta
Block MGCKM     # CKM elements for MadGraph
     1   1     9.75000000E-01   # Vud for Cabibbo matrix
Block MASS      #  Mass spectrum (kinematic masses)
#       PDG       Mass
         4     1.40000000E+00   # charm    pole mass
         5     4.20000000E+00   # bottom   pole mass
         6     1.74300000E+02   # top      pole mass
        15     1.77700000E+00   # tau      mass
        23     9.15445065E+01   # Z        mass
        24     8.02673592E+01   # W        mass
        25     1.00000000E+02   # H        mass
#            PDG       Width
DECAY         6     1.51013490E+00   # top width
DECAY        23     2.44639985E+00   # Z   width
DECAY        24     2.03535570E+00   # W   width
DECAY        25     4.27608700E-03   # H   width
#        BR           NDA       ID1       ID2
     7.18385415E-02    2           4        -4   # BR( H -> c  cbar  )
     8.03404251E-01    2           5        -5   # BR( H -> b  bbar  )
     0.00000000E+00    2           6        -6   # BR( H -> t  tbar  )
     4.83591503E-02    2          15       -15   # BR( H -> tau- tau+)
     2.69892403E-05    2          23        23   # BR( H -> Z   Z^(*))
     3.67605190E-03    2          24       -24   # BR( H -> W   W^(*))
     2.43358656E-02    2          21        21   # BR( H -> g   g    )
```

In a general way, parametrization of a model can be tuned from that card. The card is divided in blocks, depending of the content: kinematic mass, yukawa mass, decay properties, etc... Obviously the number of blocks depends on the model, but the principle remains the same.

Note that in Standard Model implementation, CKM mixing matrix is given as follows: $V_{ud} = V_{cs} = 0.975$ ($\texttt{Vud}$), $V_{us} = V_{cd} = \sqrt{1 - \texttt{Vud}^2}$ ($\texttt{Vus}$), $V_{tb} = 1$.
Note that in MSSM, Higgs widths can be calculated using SDECAY [10] and in SM and 2HDM, the calculations are done with local calculator (TwoHiggsCalc code for 2HDM).

5 The format of the event record

The events generated by `MadEvent` are stored in LHEF format. Routines that write and read the events in this format are provided in the file `Source/rw_events.f`. Interfaces for `Pythia` is available on the web site. It start with declaration of accelerator typical final state in two line, surrounded by *init* flags:

```
<init>
IDBMUP(1) IDBMUP(2) EBMUP(1) EBMUP(2) PDFGUP(1) PDFGUP(2) PDFSUP(1) PDFSUP(2) IDWTUP NRUP
SUM XERR MAXWGT 661
</init>
```

The variables are discussed in Ref[11]. On the first line,

- `IDBMUP(i)` are Particle ID of particles in the beam (here two protons)
- `EBMUP(i)` are beam energies
- `PDFGUP(i)` are author groups of PDF's
- `PDFSUP(i)` are PDF set IDs
- `IDWTUP` is a switch referencing to the interpretation of weighting events
- `NPRUP` is the number of differents user subprocesses.

One the second one,

- `SUM` is the integrated weight
- `XERR` isthe error on cross section
- `MAXWGT` is the maximum weight
- `661` is just a tag

Here below is shown a numerical example for *pp* collision at LHC.

```
<init>
  2212  2212  0.70000000000E+04  0.70000000000E+04 0 0 10042 10042 3 1
  0.75279308564E+03  0.25001000000E+01  0.75384847350E-01 661
</init>
```

Following this header, each event is detailled according the format described in [3].:

```
IC(1,i) IC(6,i) IC(2,i) IC(3,i) IC(4,i)
IC(5,i) P(1,i) P(2,i) P(3,i) P(0,i) P(4,i) P(1,i) IC(7,i)
```

with

```
ic(*,1) = Particle ID
ic(*,2) = First mother particle index
ic(*,3) = Second mother particle index
ic(*,4) = color characteristic
ic(*,5) = color characteristic
ic(*,6) = status (-1=initial state +1=final  +2=decayed)
ic(*,7) = Helicity
```

As an example, consider the following event for the process $pp \to t\bar{t}$ [2]:

```
<event>
   4 661  0.7538485E-01  0.9118800E+02  0.7546772E-02  0.1300000E+00
  21 -1  0  0  501  502  0.000E+00  0.000E+00  0.150E+03  0.150E+03  0.000E+00 0.  1.
  21 -1  0  0  502  503  0.000E+00  0.000E+00 -0.390E+03  0.390E+03  0.000E+00 0. -1.
   6  1  1  2  501    0 -0.164E+03 -0.337E+02 -0.136E+03  0.277E+03  0.174E+03 0. -1.
  -6  1  1  2    0  503  0.164E+03  0.337E+02 -0.103E+03  0.263E+03  0.174E+03 0. -1.
#  0.0000000E+00  0.0000000E+00
</event>
```

6 Off-line applications

The events generated by MadEvent are stored in LHEF [3] format (see previous Section). Within this format enough information for each event is available that some of the data analysis usually performed during the event generation phase, such as plotting, estimating PDF's errors or scale variations can be deferred to a later stage. [3] To this aim, simple routines have been developed to perform some tasks "off-line", *i.e.*, directly on the event files produced. The main reason for doing this is to improve versatility and save time. Generating events is a CPU expensive activity, which, in some cases, can take many hours. Therefore, it is not desirable to have to rerun codes only for making new plots or switching from one scale choice to another. Another important advantage in working directly with the events is that the tools developed are "independent" on how the events were generated and can be used with any event set in the Les Houches format. In this respect, the applications presented below should be considered mainly as examples. The expert user is invited to develop his/her own tools and make them available to the physics community.

[2]Only four digits after coma for 4-momentum numbers have been kept to fit the page.

[3]To be precise, this should be done with weighted events.

References

[1] F. Maltoni and T. Stelzer, JHEP **0302** (2003) 027 [arXiv:hep-ph/0208156].

[2] T. Stelzer and W. F. Long, Comput. Phys. Commun. **81** (1994) 357 [arXiv:hep-ph/9401258].

[3] J. Alwall *et al.*, Comput. Phys. Commun. **176** (2007) 300 [arXiv:hep-ph/0609017].

[4] G. Marchesini and B. R. Webber, *Monte carlo simulation of general hard processes with coherent qcd radiation*, *Nucl. Phys.* **B310** (1988) 461.

[5] H.-U. Bengtsson and T. Sjostrand, *The lund monte carlo for hadronic processes: Pythia version 4.8*, *Comput. Phys. Commun.* **46** (1987) 43.

[6] S. Catani, F. Krauss, R. Kuhn and B. R. Webber, JHEP **0111** (2001) 063 [arXiv:hep-ph/0109231].

[7] J. Campbell and R. K. Ellis, MCFM, A Monte Carlo for FeMtobarn processes at Hadron Colliders, `http://mcfm.fnal.gov`.

[8] M. L. Mangano, M. Moretti, F. Piccinini, R. Pittau and A. D. Polosa, arXiv:hep-ph/0206293.

[9] S. Dittmaier and M. Roth, Nucl. Phys. B **642** (2002) 307 [arXiv:hep-ph/0206070].

[10] M. Muhlleitner and A. Djouadi and Y. Mambrini [arXiv.org:hep-ph/0311167]

[11] E. Boos *et al.*, arXiv:hep-ph/0109068.

[12] V. M. Budnev, I. F. Ginzburg, G. V. Meledin and V. G. Serbo, Phys. Rept. **15** (1974) 181.