How to use MadGraph/MadEvent?

Takeo Moroi (Tohoku)

Outline

1. Introduction

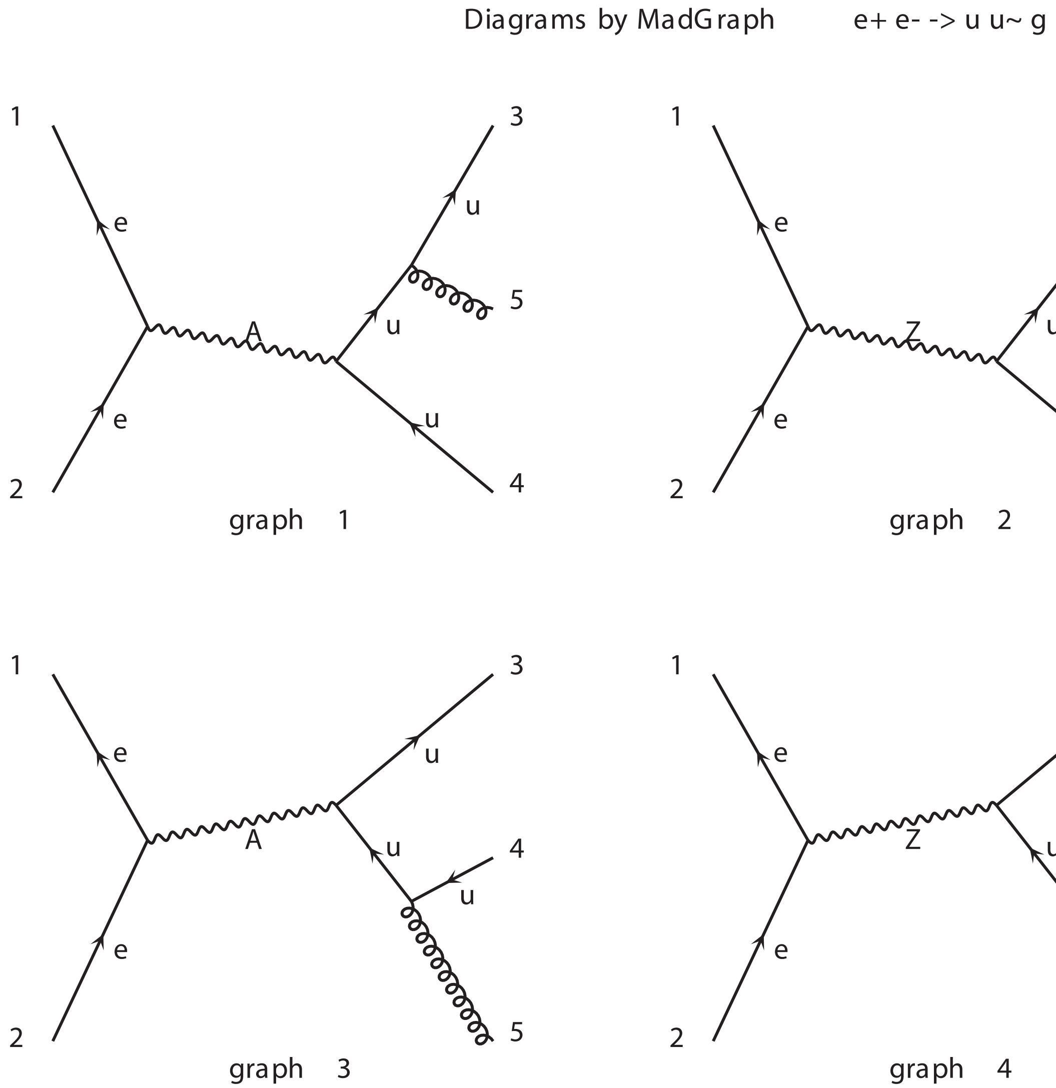
MadGraph

- Automatic calculation of helicity amplitude
  - For a given process, all the relevant diagrams are automatically listed up
  - 4-momenta and helicities of external legs are fixed
- It can deal with only tree-level processes

MadEvent

- Automatic event generation (at the parton level)
- For your favorite model, you obtain cross sections (and many more which I don't understand yet)

You can do something like this: $e^+e^- \to u\bar{u}g$



2. Installation

You should install:

- MadGraph V4.x (which also contains MadEvent)

  `http://madgraph.roma2.infn.it/`

- ROOT (Version 5.xx)

  `http://root.cern.ch/`

- Pythia and PGS package

  `http://madgraph.roma2.infn.it/`

Necessary files:

- If you just want to calculate cross section at the parton level, you only need MadGraph / MadEvent
- If you are interested in something more (like hadronization, detector simulation, also so on), you need all

Notice:

- They run on Linux machine
- ROOT should be installed before Pythia and PGS package

Installation of MadGraph / MadEvent

1. Get source file from Roma server
2. Untar it at, for example, ~/, then the directory `MG_ME_V4.0` (or whatever) is made
3. Do `make` in ~/`MG_ME_V4.0`

Installation of ROOT

1. Make `~/ROOT` directory
2. Download the source file from CERN into `~/ROOT`
3. Go into `~/ROOT/root` and do `./configure`
4. Do `./make`; this will take a few hours
5. Do `./make install`

Binary files are also available, but I haven't checked if they work well or not...

Variables ROOTSYS, LD_LIBRARY_PATH, and PATH should be properly defined

- Add the following lines in your `.cshrc` file

```
setenv ROOTSYS ~/ROOT/root
setenv PATH $PATH:$ROOTSYS/bin
setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:$ROOTSYS/lib
```

- For those who use bash: do something...

Run `root`; something happens if it's working

If you run `root` in `~/ROOT/root/tutorials/`, you can see what you can (potentially) do with ROOT

Installation of Pythia and PGS package

1. Get source file from Roma server
2. Untar it in the directory `~/MG_ME_V4.0`
3. Do `make`

The directory `pythia-pgs/` should be in the same directory as `Template/`

3. Define your own model

First, let us define a new model

- Your model should be defined in a new directory under:

  `~/MG_ME_V4.0/Models/`

- Following models are already defined and available

  sm, smckm, sm_nohiggs, mssm, 2hdm, $\cdots$

Hereafter, I show how new models can be analyzed

$\Rightarrow$ As an example, I consider case with an exotic fermion

1. Make a new directory

- Template is available in `~/MG_ME_V4.0/usrmod`
- You should better leave the original `usrmod` directory untouched

```
> cd ~/MG_ME_V4.0/Models
> cp -R usrmod MyModel
```
(or whatever name you like)

2. Modify `.dat` files

- `particles.dat`

  List of particles and properties

- `interactions.dat`

  Interactions

- `VariableName.dat`

  List of fundamental variables to calculate coupling constants

- These files are in ˜/`MG_ME_V4.0/Models/MyModel`

particles.dat: MODEL EXTENSION section

```
#MODEL EXTENSION
xf   xf~   F   S   XFMASS XFWIDTH   S   XF   8
# END
```

1: Particle name

2: Anti-particle name (“tilde” is usually for anti-fermion)

3: `F` for fermion (`S`: scalar / `V`: vector)

4: `S` for straight line in Feynman diag. (`D`: dotted / `W`: wavy)

5: Variable name for the mass of new particle

6: Variable name for the width of new particle

7: `S` for color singlet (`T`: triplet / `O`: octet)

8: Name used in Feynman diagram

9: Particle id. number (8 is actually for t' in PDG code)

`interactions.dat`: `USRVertex` section (at the end)

```
#   USRVertex
xf xf a GAXF QED
```

1: particle 1

2: particle 2

3: particle 3 (photon, in this case)

4: Variable name for the coupling

5: Class: `QED` or `QCD`

You can introduce all the renormalizable interactions

`VariableName.dat`

- Define variables which will be used in `couplings.f` later
- You can make a comment using `#`

```
xfcharge # charge of xf
```

3. Go one-step further

- Run perl script to make `param_card.dat` and `couplings.f`

```
> ConversionScript.pl
```

- Modify `param_card.dat`
  - Define the size of fundamental parameters (numerical values)
- Modify `couplings.f`
  - Define how the coupling constants are related to fundamental parameters

param_card.dat: various places for new particles

```
Block MASS      #  Mass spectrum (kinematic masses)
...
             8     5.00000000e+02   # XFMASS
#           PDG        Width
...
DECAY        8     1.00000000e+00   # XFWIDTH


...


BLOCK MGUSER
             1     1.00000000e+00   # xfcharge
```

`couplings.f`: `UserMode couplings` section

```
c**************************************
c UserMode couplings
c**************************************

      GAXF(1) = ee * xfcharge
      GAXF(2) = ee * xfcharge
```

- Variable `ee` is defined as electric charge in this program
- In HELAS, `GAXF` is used as (chiral) gauge couling constant
  - `GAXF(1)`: for left-handed
  - `GAXF(2)`: for right-handed

4. Cross section

1. Make a new directory for the study

- All the files to be used are in `Template` directory
- It is better to leave `Template` directory clean

```
> cd ~/MG_ME_V4.0
> cp -R Template NewModel (or what-so-ever)
```

2. Modify `xxx_card.dat` files in `Cards` directory

- All the information should be in Card files

  ```
  > cd ~/MG_ME_V4.0/NewModel/Cards
  ```

- At least, modify the following files
  - `proc_card.dat`

    Define the process you study
  - `run_card.dat`

    Define the collider and MC parameters
- `param_card.dat` is automatically replaced later
- There exist other Card files

  `pythia_card.dat`, `plot_card.dat`

proc_card.dat: Define the process

```
#*********************************************************
# Process(es) requested : mg2 input                      *
#*********************************************************
# Begin PROCESS  # This is TAG. Do not modify this line

p p > xf xf~     # First Process
QCD=99           # Max QCD couplings
QED=99           # Max QED couplings
end_coup         # End the couplings input

done             # this tells MG there are no more procs

# End PROCESS    # This is TAG. Do not modify this line
```

proc_card.dat: Model information / multiparticle definitions

```
#*********************************************************
# Model information                                      *
#*********************************************************
# Begin MODEL  # This is TAG. Do not modify this line
MyModel
# End   MODEL  # This is TAG. Do not modify this line
#*********************************************************
# Start multiparticle definitions                        *
#*********************************************************
# Begin MULTIPARTICLES # This is TAG.
P uu~dd~ss~cc~g
J uu~dd~ss~cc~g
L e+e-mu+mu-ta+ta-veve~
# End  MULTIPARTICLES # This is TAG.
```

`run_card.dat`: Number of event sample

⇒ Sample files for LEP, Tevatron, and LHC are available

- `run_card-LEP-3.dat`
- `run_card-TEV-20.dat`
- `run_card-LHC-40.dat`

```
#*********************************************************
# Number of events and rnd seed                          *
#*********************************************************
10000 = nevents ! Number of unweighted events requested
    0 = iseed   ! rnd seed
```

run_card.dat: Initial particles

```
#*********************************************************
# 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
```

$\Rightarrow$ `lpp1 / lpp2 = 0` for electron, 1 for proton

3. Now, you can run the program

```
> cd ~/MG_ME_V4.0/NewModel

> ./bin/newprocess

> ./bin/generate_events
```

Use the NetScape to see what's going on

5. And beyond...

There are much more to be done with MG / ME

- Hadronization
- Event generation
- MC Simulation including detector effects (with PGS)
- . . .

So, play with MG / ME, and have fun!