Contents

Chapter 1

Introduction

1.1 Project Outline

Chapter 2

Dark Matter

42

2.1 Setup Software on lxplus

2.1.1 Setup Git

source karl's local directory for newer version of git `/afs/cern.ch/user/k/knordstr`

```
/local/git_bin/git clone https://github.com/karlnordstrom/CERN_SummerProje
.git CERN_SummerProject
```

Or put it into `Path` in .bashrc

```
export PATH=/afs/cern.ch/user/k/knordstr/local/git_bin:$PATH
```

Or

```
export PATH=$PATH:/afs/cern.ch/user/k/knordstr/local/git_bin
```

Useful commands:

```
git clone https://github.com/karlnordstrom/CERN_SummerProject
.git CERN_SummerProject
git pull [https://github.com/karlnordstrom/CERN_SummerProject
.git CERN_SummerProject]
```

(??)

```
git status
git checkout file1.txt file2.bla
```

`git add file.*` adds files that should be committed next time.

```
git commit
```

`git commit -a` combines commit and add.

2.1.2 Setup Lxplus

ssh into lxplus with your username: (alternatively lxplus5 or lxplus6 for SLC5 or SLC6 respectively).

```
ssh lmeng@lxplus.cern.ch -Y
```

copy the following script to your home directory (only once)

```
cp ~/doglioni/public/.bashrc .
```

and then source it everytime you log in

```
source ~/.bashrc
```

this gives you a few useful commands, including all the ATLAS software framework setup. To set up ROOT you have to do:

```
setupATLAS
```

and then to set up ROOT:

```
localSetupROOT
```

2.1.3 Setup MadGraph

```
cd "somewhere"(\$HOME or so)
```

1) create all the directories needed ============================

```
mkdir DMSoft
cd DMSoft
mkdir Generators
mkdir Generators/MG5
```

============================

2) download the software ============================

```
wget https://launchpad.net/mg5amcnlo/2.0/2.1.0/+download/MG5_aMC_v2.1.1.tar.gz Generators/MG5/
wget http://cp3.irmp.ucl.ac.be/downloads/Delphes-3.1.2.tar.gz Parametrization/Delphes/
```

============================

3) untar and compile when needed ============================

```
cd Generators/MG5
tar -zxf MG5_aMC_v2.1.1.tar.gz
rm MG5_aMC_v2.1.1.tar.gz

cd ../../Parametrization/Delphes
tar -zxf Delphes-3.1.2.tar.gz
rm Delphes-3.1.2.tar.gz
cd Delphes-3.1.2
cp /afs/cern.ch/user/h/helsens/public/FCC/setup.sh .
source setup.sh
make -j 4
```

============================

4) run mg5

```
cd ../../Generators/MG5/MG5_aMC_v2_1_1/
./bin/mg5_aMC
```

now you are entering the MG5 interactive command line

Unzip the model found in the note "Simplified model FeynRules" in the models/ directory
Import the model
mg5>import model dmV_UFO
Generate your process (careful with spaces) mg5>generate p p > g xi, xi > chi chi –> not working mg5>generate p p > g chi chi –> not working
[If you want to decide how many orders you want for the calculation use the following, only if it's inside the model though
mg5>generate p p > g xi, xi > g chi chi QED=99 QCD=99 @2]
Set the output directory naming - you'll find
mg5>output dm_mediator_g_monojet
Start the production
mg5>launch
Tell it you want to shower with Pythia when asked by typing "1"
mg5>1
Change the parameter card by typing 1: mass of the chi (1 TeV for now) Change the run card by typing 2: CM energy (ok), PDF and Madgraph-level cuts (set qCut=80) ??? Can also change with Pythia6
mg5>done
KN: Could we have the dmV_UFO folder to download somewhere/sent by email?
start MadAnalysis
edit ma_card.dat
./plot_events
enter event (.lhe file) path
plot file for topdrawer in MadAnalysis folder.
go to td (topdrawer folder)
./td ../MadAnalysis/plots.top
to generate plots.ps
open plots.ps

2.1.4 Setup Rivet

Use the project website
https://rivet.hepforge.org/trac/wiki/GettingStarted
Installation with afs:
The description below is based on a build from CERN's lxplus6 SLC6 machines.
1) Download the bootstrap-lcg script into a temporary working directory, and make it executable:

```
cd /scratch/rivet
(cd $HOME)
wget http://rivet.hepforge.org/hg/bootstrap/raw-file/2.1.1/rivet-bootstrap-lcg
chmod +x rivet-bootstrap-lcg
```

or for Rivet 1.x:

```
cd /scratch/rivet
wget http://rivet.hepforge.org/hg/bootstrap/raw-file/1.9.0/rivet-bootstrap-lcg
chmod +x rivet-bootstrap-lcg
```

2) Check/edit the script. Look in the script to see its target setup and make edits if you need to: you may want to change the LCG tag, the compiler environment that is set up, whether LCG packages are to be used from AFS, and the install and build locations:

```
less rivet-bootstrap-lcg \#\# and read...
nano rivet-bootstrap-lcg \#\# and edit...
```

3) Run the script. By default it will install to `\$PWD/local`, where `\$PWD` is the current directory. If you need to change that, edit the file as above.
`./rivet-bootstrap-lcg`
We will refer to the installation root path as $PREFIX.
(this will take a while)
If you have trouble with the Boost library (hopefully you won't) see TroubleshootingBoost.
Setting up the environment
After the script grinds away for a while, it will tell you that it is finished and how to set up a runtime environment (similar to that used inside the build script) for running Rivet. A sourceable rivetenv.(c)sh script is provided for (c)sh shell users to help set up this environment. Here's how to set up the environment and then test the rivet program's help feature and analysis listing:

```
source \$PREFIX/rivetenv.sh
rivet --help
rivet --list-analyses
```

If that works, everything is installed correctly. If you are using the bash shell in your terminal, then Rivet will offer you programmable tab completion: try typing rivet and pressing the Tab key!
You may wish to add the environment variable settings to your /.bashrc shell config file, so that Rivet will work without needing any special session setup.
You can now check out the FirstRivetRun guide.
run it:
`source ~/.bashrc` setupRivet
Rivet runs with HepMC files. To convert the lhe files generated by MadGraph download lhef2hepmc from the repo:
`svn checkout http://rivet.hepforge.org/svn/contrib/lhef2hepmc/ /local_path/`
In the Makefile change the path to HepMC to:
`HEPMC_PREFIX=/afs/cern.ch/sw/lcg/external/HepMC/2.06.06/x86_64-slc6-gcc48-opt`
This is where HepMC is installed on AFS.
Then run `./lhef2hepmc --help`

2.1.5 Other tools

source karl's local directory for newer version of git `/afs/cern.ch/user/k/knordstr`

```
/local/git_bin/git clone https://github.com/karlnordstrom/CERN_SummerProje
.git CERN_SummerProject
```

Or put it into `Path` in .bashrc

```
export PATH=/afs/cern.ch/user/k/knordstr/local/git_bin:$PATH
```

Or

```
export PATH=$PATH:/afs/cern.ch/user/k/knordstr/local/git_bin
```

Useful commands:

```
git clone https://github.com/karlnordstrom/CERN_SummerProject
.git CERN_SummerProject
git pull [https://github.com/karlnordstrom/CERN_SummerProject
.git CERN_SummerProject]
```

(??)

```
git status
git checkout file1.txt file2.bla
```

`git add file.*` adds files that should be committed next time.

```
git commit
```

`git commit -a` combines commit and add.

2.2 Run Analysis

2.2.1 Rivet Analysis

You can get some help info by running rivet-mkanalysis –help, but the basic usage (to generate the files in your current directory) is rivet-mkanalysis MY_ANALYSIS_NAME. A three part name, separated by underscores, is a Rivet convention that we recommend you to use: the first part is the experiment name, the second is the year of publication, and the third is the IS code for the corresponding paper in the SPIRES HEP database (preceded by an "S"). So, for example, ATLAS_2010_S8591806 is the name for the first ATLAS minimum bias paper. You can get the SPIRES ID from a SPIRES page by clicking the "Display again" button, and looking for the number following "FIND KEY " in the search box.

The script will have generated a .cc C++ source file template, and template metadata files for information about the analysis (.info) and specifications of titles, axis labels, etc. for the plots which the analysis will produce (.plot). These templates will include, if possible, extra analysis metadata such as a BibTeX publication entry in the .info file.

run

```
rivet --analysis=myanalysis hepmcfile.hepmc
```

yoda file is created containing the plots

4) You'll notice a Rivet.yoda file in the directory you've run on. This contains all info to make plots. However, to get some non-blank plots out, you have to hack the texfm file in your installation dir:

local/share/Rivet/texmf/cnf/texmf.cnf

and change the following line (see 'more info'):

(main_memory = 700000 % → main_memory = 70000000 %)

Now you can make plots with

```
rivet-mkhtml Rivet.yoda
```

Chapter 3

TCAD

3.1 Setup and run TCAD

TCAD
get a license (ask Mathieu Benoit), agree by replying the mail
get a working space from account manager (`http://account.cern.ch`, max. 100 GB)
copy some scripts
login onto lxplus `ssh lmeng@lxplus.cern.ch -Y`
go to working space
`cd /afs/cern.ch/work/l/lmeng/`
`source setupTCAD.sh` or `. setupTCAD.sh`

```
bla
```

to start `swb &` for *Sentaurus Workbench*
http://web.stanford.edu/class/ee328/swb/swb_d.html for a tutorial
In Sentaurus Workbench, parameters can be defined and multiple values can be assigned to them to create splits in experiments (simulations). Each value of a parameter creates an additional experiment; therefore, for N values of a parameter, there are N experiments. For two parameters, P1 and P2, with M and N values, respectively, there are M x N possible experiments.
The creation of different scenarios is particularly helpful when many parameters are used. The resulting experiments can be classified into separate scenarios to represent different physical situations.
Extensions → Commandline prompt here
All interactive operations associated with running a project can be accomplished at a UNIX prompt. The commands for submitting jobs, preprocessing, and cleaning up project directories are
`gsub (gjob)` submitting jobs `spp` proprocessing `gcleanup`
For more information about the commands executed (at the prompt), refer to the Sentaurus Workbench User Guide or use the UNIX command:
> <command> -h[elp] for example:
> gsub -h
Node navigation

a colon (:) is the operator used to navigate between different nodes within the same tool of the project. a vertical bar (|) is the operator to navigate between different tool nodes in the project.

List of Figures

List of Acronyms

Own Contributions

Acknowledgements