ExHuME: A Monte Carlo Event Generator for Exclusive Diffraction.

J. Monk and A. Pilkington
School of Physics and Astronomy, University of Manchester
Manchester M13 9PL, England

Abstract

We have written the Exclusive Hadronic Monte Carlo Event (ExHuME) generator. ExHuME is based around the perturbative QCD calculation of Khoze, Martin and Ryskin of the process $pp \to p + X + p$, where X is a centrally produced colour singlet system.

Introduction

Central exclusive production events are those in which the colliding protons remain intact after an interaction and all of the energy transferred from the protons goes into a central colour singlet system. Unlike inclusive double pomeron exchange, in which pomeron remnants are present, there is no radiation other than that from the central system. Furthermore, in the limit that the transverse momentum of each of the outgoing protons vanishes, no angular momentum is exchanged between the proton lines and the quantum numbers of the central system are constrained to be $J_Z = 0$ with even parity. The leading order diagram for the central exclusive production of system X is shown in figure 1.

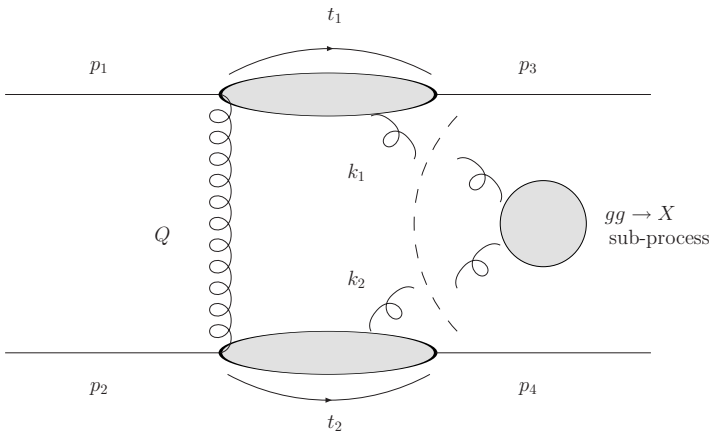

Figure 1: The Exclusive Diffractive Process

The invariant mass of the central system can be determined if proton detectors mounted very forward of the central detector are used to measure the longitudinal momentum losses, $x_{1,2}$, and momentum transfers squared, $t_{1,2}$.

ExHuME factorises central exclusive production as indicated by the dashed line in figure 1. We use gluon fusion cross sections - taking the colour singlet and spin zero projections of the amplitude [1] - but do not fix the collision energy.

The luminosity, $\mathcal{L}$, for the fusing gluons is dependent on both the invariant mass ($\sqrt{\hat{s}}$) and rapidity, ($y = \frac{1}{2}\log(x_1/x_2)$) of the central system [1, 2, 3]:

$$\hat{s}\frac{\partial\mathcal{L}}{\partial\hat{s}\partial y\partial t_1\partial t_2} = e^{b(t_1+t_2)}\left(\frac{\pi}{8}\int\frac{dQ_\perp^2}{Q_\perp^4}f_g(Q_\perp,\ \mu,\ x_1,\ x_1')f_g(Q_\perp,\ \mu,\ x_2,\ x_2')\right)^2. \tag{1}$$

The forward limit implies that $k_{1\perp} \simeq -k_{2\perp} \simeq -Q_\perp$, which simplifies the integral to the form in equation (1). The f_g are the unintegrated skewed gluon distributions (which can be calculated from the integrated gluons [4]). We choose $\mu = 0.618\sqrt{\hat{s}}$ [5] for the hard scale and $b = 4$ GeV^{-2} [1]. The x' are the longitudinal momentum fractions of the gluon that is not connected to the central system, $x_1' = x_2' = \mathcal{O}(Q_\perp^2/s)$ for symmetric proton beams of energy $\sqrt{s}/2$. The integral in equation (1) converges in the infra-red due to the presence of a Sudakov factor that imposes the requirement that there be no radiation emitted between the scales of $Q_\perp$ and μ. The leading term in $k_{1\perp}\cdot k_{2\perp}$ in the integrand of equation (1) is $Q_\perp^2$ and has no dependence on the azimuthal angle ϕ between the outgoing proton directions. For complete details of the luminosity calculation we refer to the calculation of V. A. Khoze et al [1, 2, 3, 4, 5], upon which ExHuME is based.

A survival factor, $\mathcal{S}^2$, is also required to ensure that there are no additional interactions between the proton lines. We use a constant value for $\mathcal{S}^2$ that is fixed for a given collision energy, although we anticipate that further work will lead to a parameterisation of the survival factor in terms of t_1, t_2 and ϕ [4]. We note that the ϕ dependence in that model is approximately flat for values of $p_{3\perp},\ p_{4\perp} \lesssim 200$MeV. The default differential luminosity curves used by ExHuME are shown in figure 2.

In this initial release of ExHuME we provide the following sub-processes: $gg \to H$, $gg \to Q\bar{Q}$ and $gg \to gg$, where H is a Standard Model Higgs and Q is a massive quark. We require a width for resonance production and use the Higgs line shape described in [6] with the Higgs width and branching ratios calculated using the program Hdecay [7]

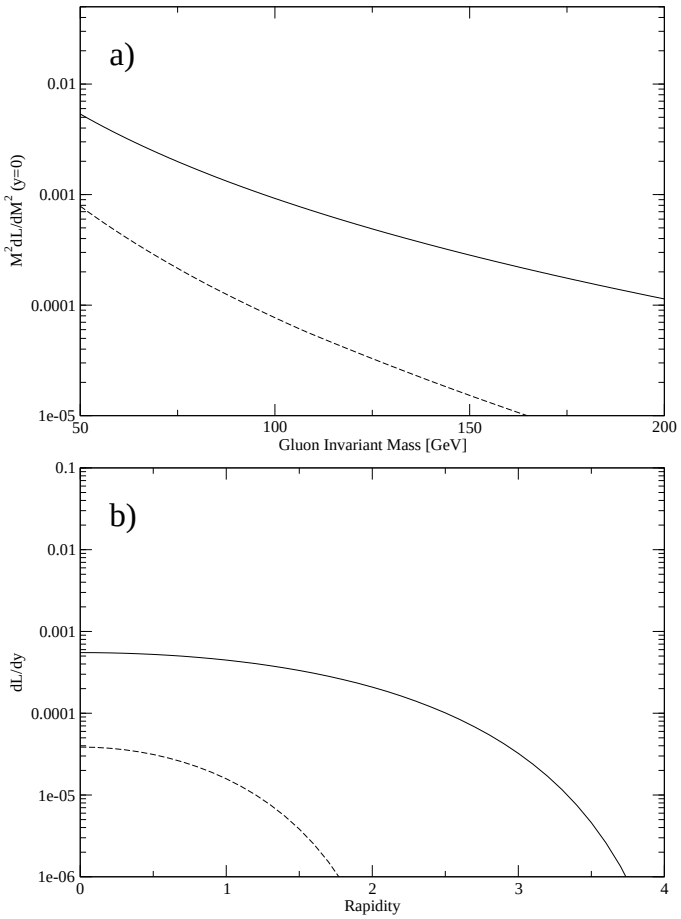

Figure 2: The default luminosity used in ExHuME at collider energies of 1.96 TeV (dashed line) and 14 TeV (solid line). a) shows the luminosity vs. the invariant mass of the fusing gluons at zero rapidity. b) shows the rapidity dependence of the luminosity when the invariant mass of the fusing gluons is fixed at 120 GeV. A fixed survival factor of 0.03 (14 TeV) or 0.045 (1.96 TeV) has been included and the luminosity has been integrated over the $p_\perp$ of the outgoing protons.

Design

ExHuME is written in a modular way using C++. There are two main classes that are needed to generate central exclusive events. The first is a `CrossSection` class that calculates the differential luminosity, the gluon fusion sub-process and the kinematics of any outgoing particles. The second class needed is an `Event` class that generates the events. In addition there is a `Weight` class that provides random numbers distributed according to any given function and from which the `Event` class inherits.

The CrossSection Class

The `CrossSection` class exploits the factorisation of figure 1. `CrossSection` is an abstract base class containing the calculation of the effective luminosity of the gluon-gluon collision with a virtual method for the gluon fusion sub-process. Complete processes are created by inheriting from the `CrossSection` and explicitly defining a sub-process. This makes it relatively simple to implement new sub-processes in addition to the currently implemented Standard Model Higgs, gg and $Q\bar{Q}$ sub-processes. Instructions for creating a new sub-process are posted at [8] and users are invited to submit any new process to the authors for inclusion in future updates to ExHuME.

CrossSection Methods

The constructor for the `CrossSection` class is

```
CrossSection(int, char**),
```

which allows the user to pass a card file from the command line that changes the default values used in the luminosity calculation (see Appendix A). The constructors for the derived classes are `Higgs(int, char**)`, `QQ(int, char**)`, `GG(int,char**)` or `Dummy(int, char**)`.

The invariant mass and rapidity of the central system and the momentum transfers and azimuthal angles, $\phi_{1,2}$, of the outgoing protons can be set by the `CrossSection` method

```
void SetKinematics(
   const double &mass, const double &rapidity,
   const double &t1, const double &t2,
   const double &phi1, const double &phi2);.
```

The method

```
double Differential()
```

then returns the differential cross section. The 4-vectors of the outgoing protons can be accessed by the methods

`HepLorentzVector GetProton1()` and
`HepLorentzVector GetProton2()`.

The sub-process level information can be extracted by the method

```
std::vector<Particle> GetPartons()
```

which returns the outgoing particles from the gluon fusion matrix element prior to decay, parton showering and hadronisation. The `Particle` class that `GetPartons()` returns contains the particle momentum, `p`, the PDG ID code, `id` and vertex, `vtx`. It is also possible to access the kinematics of the cross section by the following methods:

`double GetRoot_s();`
Returns the invariant mass of the colliding beams.

`double GetsHat();`
Returns the invariant mass squared of the central system.

`double GetSqrtsHat();`
Returns the invariant mass of the central system.

`double Getx1();`
Gets the value of x_1.

`double Getx2();`
Gets the value of x_2.

`double Gett1();`
Gets the value of t_1.

`double Gett2();`
Gets the value of t_2.

`double GetPhi1();`
Gets the value of ϕ_1, the azimuthal angle between the outgoing proton 1 and the x direction.

`double GetPhi2();`
Gets the value of ϕ_2, the azimuthal angle between the outgoing proton 2 and the x direction.

`double GetEta();`
Gets the rapidity, y, of the central system.

Hadronisation is performed via the method

```
void Hadronise()
```

which places the event record into an external `hepevt` common block that can be written to file.

There are a number of sub-process specific methods. The Higgs decay type can be set by the method

```
SetHiggsDecay(const int&)
```

with the PDG code of the decay products as the argument. Similarly the quark type in $q\bar{q}$ can be set by the method

```
SetQuarkType(const int&)
```

which defaults to $b\bar{b}$ production. In addition, the `GG` and `QQ` classes contain

the method

`SetThetaMin(const double&)`

which sets the minimum (and maximum) polar angle of an outgoing parton relative to the beamline in the rest frame of the central system. The default value is $\cos\theta = 0.95$.

The Event Class

The `Event` class generates events, calculates the total cross section and reports the efficiency with which events were generated once event generation has finished. `Event` contains a pointer to a `CrossSection` and the user specifies which sub-process is to be generated in the constructor for the `Event`. The `Event` class also inherits from the `Weight` class. The numerical weighting algorithm is initialised to return the mass distribution of the differential cross section at a rapidity zero and is effective even for a narrow resonance such as the Higgs. The variables t_1 and t_2 are distributed according to $e^{b(t_1+t_2)}$ whilst ϕ_1, ϕ_2 and y are uniformly distributed.

Event Methods

The event is defined by calling the constructor

`Event(CrossSection& P, const unsigned int R)`,

where P is a `CrossSection` with the sub-process defined and R is a random number seed. There are a number of methods that can be used to set the kinematic ranges of the parameters used to define the event:

`void Setx1Max(const double&);`
Sets the upper limit of x_1.

`void Setx2Max(const double&);`
Sets the upper limit of x_2.

`void Sett1Max(const double&);`
Sets the upper limit of t_1 ($t \leq 0$).

`void Sett1Min(const double&);`
Sets the lower limit of t_1.

`void Sett2Max(const double&);`
Sets the upper limit of t_2.

`void Sett2Min(const double&);`
Sets the lower limit of t_2.

`void SetMassRange(const double & minimum, const double & maximum);`
Sets the lower and upper limits of the mass range respectively.

The last method that is used before event generation is

`void SetParameterSpace()`,

which must be called in order to initialise the event generation. Individual events are generated by the method

`void Generate()`.

Finally, the total cross section from the events generated can be calculated by the method

`double CrossSectionCalculation()`

and the efficiency via

`double GetEfficiency()`.

Using ExHuME

Installing ExHuME

The ExHuME source code is available from [8] or on request from the authors. In its standard form ExHuME must be linked at compilation to Pythia [9], CLHEP [12] and either LHAPDF [11] or the CERN PDFLIB [13]. It would also be possible to modify ExHuME to use Herwig [10] instead of Pythia for the hadronisation or to use a stand alone PDF instead of either LHAPDF or PDFLIB. By default ExHuME sets the location of the directory containing the grid or parameter files for LHAPDF to be wherever the program is executed from. A symbolic link should be created to wherever the grid and parameter files actually reside. For further information please see the respective documentation for each of these programs .

Example Main Program

In this section we demonstrate a simple main program that generates 5000 $H \rightarrow WW^*$ events for a Higgs with the default mass of 120 GeV. We also show how to extract simple information from the hepevt record.

```
   #include <iostream>
```

The following headers are for ExHuME:

```
   #include "Event.h"
   #include "Higgs.h"

int main(int argc, char** argv){
```

Declare a new Higgs `CrossSection`:

```
Exhume::Higgs higgs(argc,argv);
```

and set the Higgs to decay only to W bosons:

```
higgs.SetHiggsDecay(24); // 24 is the PDG code for W
```

Declare an event with the Higgs as the cross section and a random number seed of 1111.

```
Exhume::Event HiggsEvent(higgs,1111);
```

The allowed range of gluon fusion invariant masses must be set. As long as the range is much bigger than the width of the resonance the results will not be sensitive to the range chosen. This is, of course, not the case whenever the central system does not have a narrow width, for example in gg and $Q\bar{Q}$ production. The efficiency of event generation will drop if a large mass range is chosen for a narrow resonance such as the Higgs, so for this example where the width is 0.0036 GeV we set the mass range to be between 115 GeV and 125 GeV.

```
HiggsEvent.SetMassRange(115,125);
```

This must be called before event generation can begin:

```
HiggsEvent.SetParameterSpace();

double x1;
int Nobj;
std::vector<Exhume::Particle> HiggsInfo;

for ( int i = 0 ; i < 5000 ; i++ ){
```

The next line generates a single event:

```
    HiggsEvent.Generate();
```

Access the longitudinal momentum loss of proton 1 for this event:

```
    x1 = process.Getx1();
```

and the information about the Higgs:

```
    HiggsInfo = process.GetPartons();
```

Get the number of particles in the hepevt common block:

```
      Nobj = hepevt_.nhep;
   }
   std::cout<<" Cross section = "
   <<HiggsEvent.CrossSectionCalculation()<<std::endl;
   std::cout<<" Efficiency of event generation = "
   <<HiggsEvent.GetEfficiency()<<std::endl;
   return(0);
}
```

The program allows (but does not demand) a card file to be given on the command line that overrides the default parameters. Such a card file could look like

```
HiggsMass    140
TopMass    180
```

which would be appropriate for investigating the effects of varying the higgs and top masses.

Acknowledgements

We would like to thank Brian Cox, Jeff Forshaw, Valery Khoze and Misha Ryskin for many useful and interesting discussions and suggestions. We also thank the U.K. Particle Physics and Astronomy Research Council for funding this work.

References

[1] V.A.Khoze, A. D. Martin and M. G. Ryskin, Prospects for New Physics observations in diffractive processes at the LHC and Tevatron, Eur. Phys. J. C23 (2002) 311, hep-ph/0111078

[2] V.A.Khoze, A. D. Martin and M. G. Ryskin, The rapidity gap Higgs signal at LHC, Phys.Lett. B401 (1997) 330, hep-ph/9701419

[3] V. A. Khoze, A. D. Martin and M. G. Ryskin, Can the Higgs be seen in rapidity gap events at the Tevatron or the LHC? hep-ph/0002072

[4] V. A. Khoze, A. D. Martin and M. G. Ryskin, Physics with tagged forward protons at the LHC, Eur. Phys. J. C24 (2002) 581, hep-ph/0203122

[5] A. B. Kaidalov, V. A. Khoze, A. D. Martin and M. G. Ryskin, Extending the study of the Higgs sector at the LHC by proton tagging, Eur. Phys. J. C33 (2004) 261, hep-ph/0311023

[6] Michael H. Seymour, The Higgs Boson Lineshape and Perturbative Unitarity, Phys. Lett. B354 (1995) 409, hep-ph/9505211.

[7] A. Djouadi, J. Kalinowski and M. Spira, HDECAY: A program for Higgs boson decays in the standard model and its supersymmetric extension, Comp. Phys. Comm. 108 (1998) 56 [hep-ph/9704448].

[8] http://www.exhume-me.com

[9] T. Sjostrand et al, Comp. Phys. Comm. 135 (2001) 238.

[10] HERWIG 6.5, G. Corcella, I.G. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, M.H. Seymour and B.R. Webber, JHEP 0101 (2001) 010, hep-ph/0210213.

[11] http://durpdg.dur.ac.uk/lhapdf/

[12] L. Lönnblad, Comp. Phys. Comm. 84 (1994) 307. `http://wwwinfo.cern.ch/asd/lhc++/clhep/`

[13] H. Plothow-Besch, PDFLIB: a library of all available parton density functions of the nucleon, the pion and the photon and the corresponding alpha-s calculations, Comp. Phys. Comm. 75 (1993) 396.

A The ExHuME Control Parameters

ExHuME can be controlled by passing a card file from the command line that can contain the control parameters given in table 1. The collider can be set to the LHC (1), the Tevatron (0) or neither (-1). Choosing a collider sets the proton collision energy, $\sqrt{s}$, the survival factor $\mathcal{S}^2$ and R_g, which accounts for the skewed effect in the un-integrated gluons. Choosing neither means that the user must set these parameters. The PDF values are the PDF set numbers accepted by the LHAPDF library. `Freeze` is a scale below which α_s is frozen and $Q_{\perp min}$ is the lower bound on the integral in equation (1) - the integral is formally convergent, but the necessary freezing of α_s requires a lower bound to be introduced.

Parameter	Name	Type	Default
α	`AlphaEW`	`double`	0.0072974
M_W	`WMass`	`double`	80.33
M_Z	`ZMass`	`double`	91.127
M_H	`HiggsMass`	`double`	120.0
M_t	`TopMass`	`double`	175.0
M_b	`BottomMass`	`double`	4.6
M_c	`CharmMass`	`double`	1.42
M_s	`StrangeMass`	`double`	0.19
M_τ	`TauMass`	`double`	1.77
M_μ	`MuonMass`	`double`	0.1057
v	`HiggsVev`	`double`	246.0
$Q^2_{\perp min}$	`MinQt2`	`double`	0.64
Λ_{QCD} (MeV)	`LambdaQCD`	`double`	80 (MeV)
	`Freeze`	`double`	$Q_{\perp min}$
b	`B`	`double`	4.0
collider	`FNAL_or_LHC`	`int`	1
s	`s`	`double`	1.96×10^8
$s^{\frac{1}{2}}$	`root_s`	`double`	14000.0
R_g	`Rg`	`double`	1.2
$\mathcal{S}^2$	`Survive`	`double`	0.03
PDF	`PDF`	`int`	20250

Table 1: ExHuME control parameters