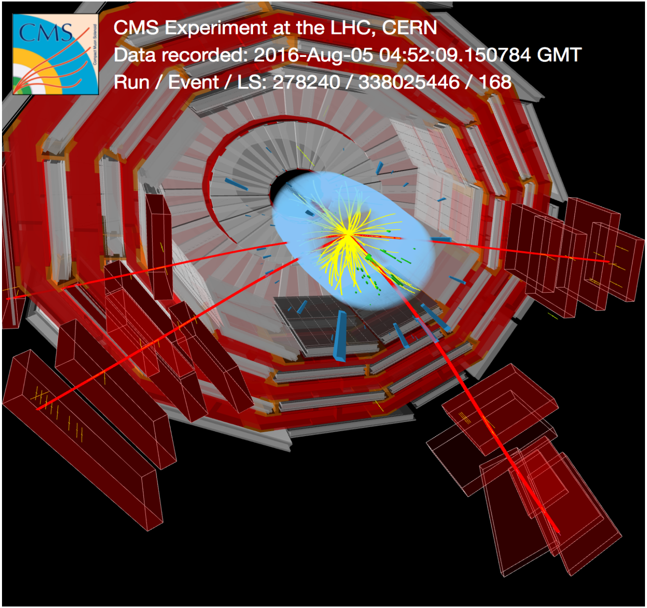
"MONTE CARLO TOOLS for HIGH ENERGY PHYSICS"

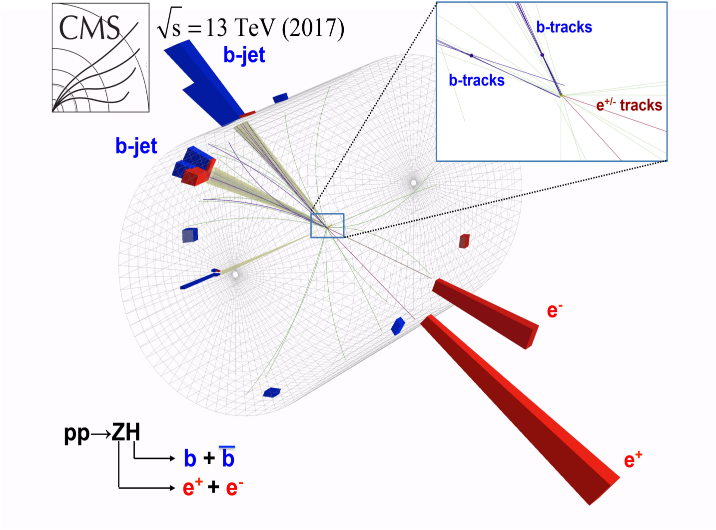
Dr. Ilirjan Margjeka

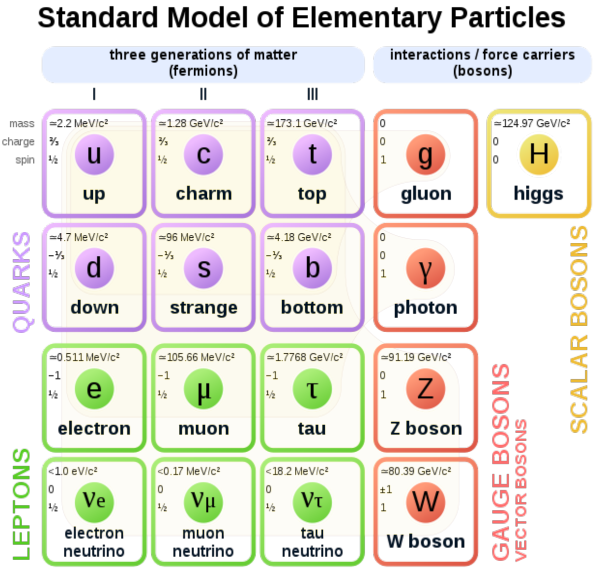
Data Science Applications in Physics, Balkan School, Tirana 2024

CMS Collision Events



CMS Collision Events

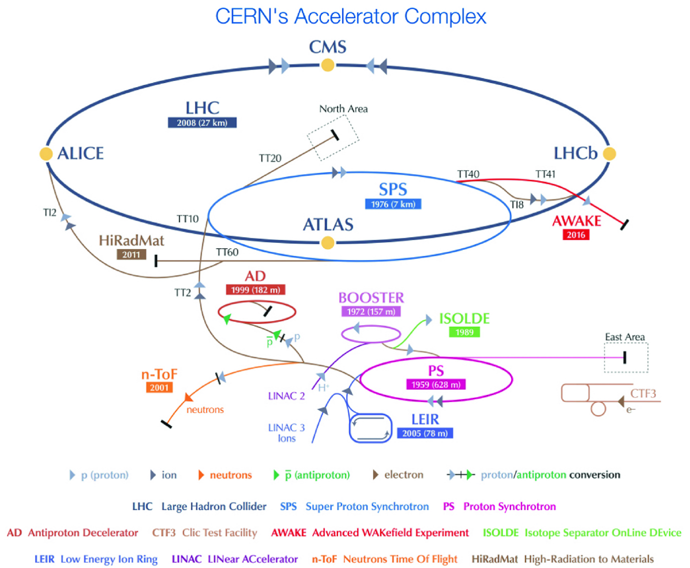


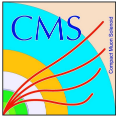

Standard Model of Elementary Particles
three generations of matter (fermions)
interactions / force carriers (bosons)
I
II
III
mass
charge
spin
≃2.2 MeV/c²
⅔
½
u
up
≃1.28 GeV/c²
⅔
½
c
charm
≃173.1 GeV/c²
⅔
½
t
top
0
0
1
g
gluon
≃124.97 GeV/c²
0
0
H
higgs
≃4.7 MeV/c²
−⅓
½
d
down
≃96 MeV/c²
−⅓
½
s
strange
≃4.18 GeV/c²
−⅓
½
b
bottom
0
0
1
γ
photon
≃0.511 MeV/c²
−1
½
e
electron
≃105.66 MeV/c²
−1
½
μ
muon
≃1.7768 GeV/c²
−1
½
τ
tau
≃91.19 GeV/c²
0
1
Z
Z boson
<1.0 eV/c²
0
½
νe
electron neutrino
<0.17 MeV/c²
0
½
νμ
muon neutrino
<18.2 MeV/c²
0
½
ντ
tau neutrino
≃80.39 GeV/c²
±1
1
W
W boson
QUARKS
LEPTONS
GAUGE BOSONS
VECTOR BOSONS
SCALAR BOSONS

From the theory to the actual observation

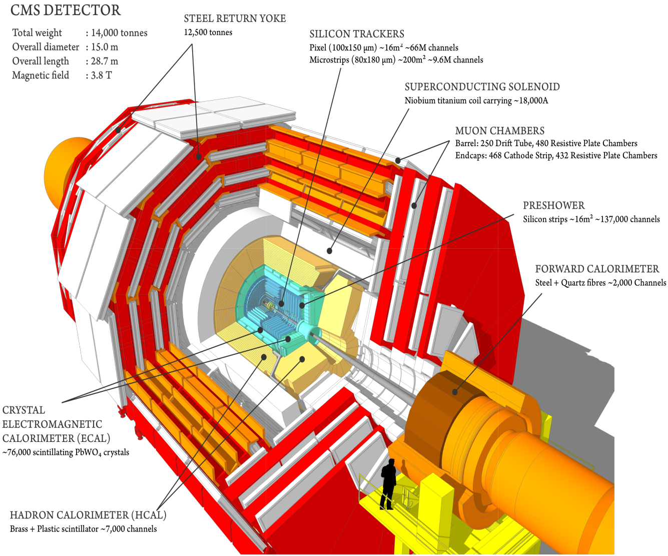
- A problem in a specific matter should appear (e.i. spontaneous symmetry breaking)
- Theoretical model needs to be formulated (e.i. HEFT: Higgs Effective Field Theory)
- Need of computational power and programming skills (C++, python, batch/bash system)
- Compatibility of the theoretical computational model with the framework of the HEP experiment
- Observation of the Physics with $> 5\sigma$ significance (Eureka! $\Rightarrow$ Nobel Prize !!!)

The Large Hadron Collider





The CMS Dectector



The Monte Carlo Argorithm and method

The Montecarlo method is a ramdom sampling method, which has specific parameters:

- Class of computational algorithms that rely on repeated random sampling to obtain numerical results.
- It uses randomness to solve problems that might be deterministic in principle
- It is used to used for : optimization, numerical integration, and generating draws from a probability distribution.
- In physics-related problems, Monte Carlo methods are useful for simulating systems with many coupled degrees of freedom, such as the interactions of parton in pp-collisions

Monte Carlo methods vary, but tend to follow a particular pattern:

1. Define a domain of possible inputs
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs
4. Aggregate the results

MC Event Generators

- The need for MC event generators is related to how we extract predictions (stochastic processes) from fundamental theories QCD and EW and BSM models;
- Event Generation:

1. Hard scattering (where new physics lies): high Q^2 scale (short time scale, where Q is momentum transfer), first principles decription, can be improved (NLO, NNLO, etc), process dependent
2. Parton Shower:Well known QCD physics, first principles description, process independent
3. Hadronization: low Q^2 physics (long time scale), model based, process independent
4. Underlying Event: low Q^2 physics, model based, energy and process dependent

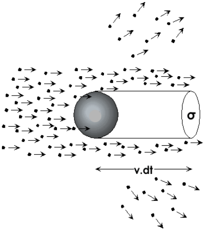
Cross Section

• Most experimental observations in particle physics corresponds to cross section and decay rates measurements. From an experimental point of view the cross section can be defined considering a beam of particles colliding with a target of cross sectional area σ.

$\Rightarrow$ **CROSS SECTION:**

The number of collisions dN observed in an time interval dt is given by:

$$\boxed{dN = \rho\sigma(vdt) \quad \rightarrow \quad \sigma = \frac{W}{F}} \tag{1}$$



where σ is the cross section, $F = \rho v$ is the incident flux and $W = \dfrac{dN}{dt}$ the collision rate.

• For a quantum scattering, the cross section σ represents an effective area of interaction between the particles and it measures the probability of a scattering to occur

Hard Scattering

From the theory side we can relate the cross section (observable) with the process matrix element (quantum scattering amplitude)

$\Rightarrow$ **SCATTERING GOLDEN RULE:**

- The relation between the cross section σ and matrix element $\mathcal{M}_{ij}$ can be obtained from Fermi golden rule. In the case of a scattering process $1 + 2 \to 3, 4, ..., n$ we have:

$$\boxed{d\sigma \frac{\parallel \mathcal{M}_{ij} \parallel}{F} d\phi} \tag{2}$$

where $d\phi$ is the phase space and F is the incoming flux factor

The phase space $d\phi$ can be written in the following Lorentz invariant form:

$$d\phi = \left[\left(\frac{d^3 p_3}{(2\pi)^3 2E_3}\right)\left(\frac{d^3 p_4}{(2\pi)^3 2E_4}\right)\dots\left(\frac{d^3 p_n}{(2\pi)^3 2E_n}\right)\right](2\pi)^4 \delta^4(p_1 + p_2 - (p_3 + p_4 + \dots p_n)) \tag{3}$$

while the flux factor can also be written in the invariant form:

$$F = 4\sqrt{(p1 \cdot p2)^2 - (m1m2)^2} \tag{4}$$

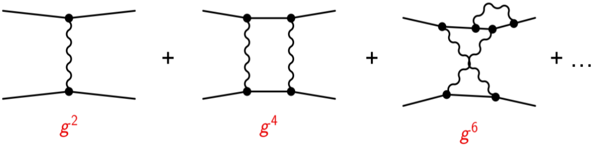
Matrix Elements and Feynman Diagrams

Theory gives the solutions of scattering problems as a perturbative series in the coupling constant. The terms of the series can be obtained directly from the Feynman diagrams through the application of the Feynman rules, without the need to calculate them explicitly (QFT).

$\Rightarrow$ **Matrix Element:**

- Matrix elements can be written as a power series, where each series term can be associated with a diagram

$$\mathcal{M}_{ij} = \mathcal{M}_{ij}^{(1)} + \mathcal{M}_{ij}^{(2)} + \mathcal{M}_{ij}^{(3)} + \ldots \tag{5}$$



Example: QED Feynman Rules

- **External Lines**

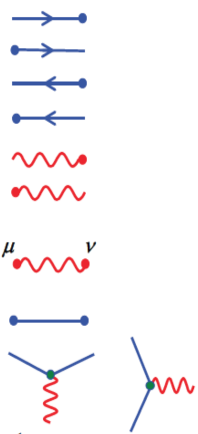
spin 1/2	incoming particle	$u(p)$	
	outgoing particle	$\bar{u}(p)$	
	incoming antiparticle	$\bar{v}(p)$	
	outgoing antiparticle	$v(p)$	
spin 1	incoming photon	$\varepsilon^{\mu}(p)$	
	outgoing photon	$\varepsilon^{\mu}(p)^{*}$	

- **Internal Lines (propagators)**

spin 1	photon	$-\dfrac{ig_{\mu\nu}}{q^2}$	
spin 1/2	fermion	$\dfrac{i(\gamma^{\mu}q_{\mu}+m)}{q^2-m^2}$	

- **Vertex Factors**

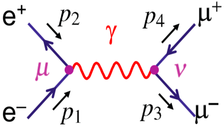
spin 1/2	fermion (charge -|e|)	$ie\gamma^{\mu}$



- **Matrix Element** $-iM$ = **product of all factors**

Matrix Element (ex: $e^+e^- \rightarrow \mu^+\mu^-$)

- **Matrix Element calculation:** $e^+e^- \rightarrow \mu^+\mu^-$



- Draw the Feynman diagrams
- Use Feynman rules to get matrix element:

$$\mathcal{M} = e^2 \, [\bar{v}(p_2)\gamma^{\mu} u(p_1)] \left((\frac{g_{\mu\nu}}{q^2} \right) [\bar{u}(p_3)\gamma^{\nu} v(p_4)] \quad (6)$$

- Average and sum over polarizations (trace theorems):

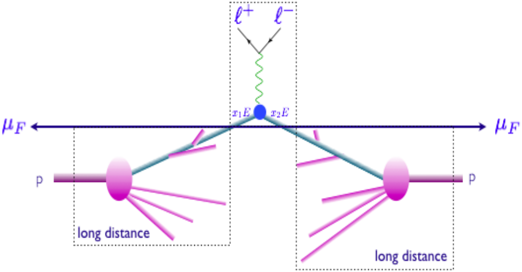
$$\frac{1}{4}\sum_{pol} \parallel \mathcal{M} \parallel^2 = \frac{8e^4}{q^4}[(p_1 \cdot p_3)(p_2 \cdot p_4) + (p_1 \cdot p_4)(p_2 \cdot p_3)] \quad (7)$$

- Number of terms rise as N^2 (efficient only for small number of final states)

Hard Scattering

LHC beams are made of protons (not partons). QCD factorization theorem relates the short-distance partonic cross section $\sigma_{ab\to X}$ and long-distance parton distribution function (PDF), obtained from experiment

$$\sigma_X = \sum_{a,b} \int_0^1 dx_1 dx_2 f_a(x_1, \mu_f) f_b(x_2, \mu_f) \hat{\sigma}_{ab\to X}(\hat{s}, \mu_f, \mu_r) \quad (8)$$



MC Event Generation

• Calculations of cross section or decay widths involve integrations over high-dimensional phase space of very complex (peaked) functions

$\rightarrow$ **Cross Section Integral:**

$$\hat{\sigma} = \frac{1}{2s} \int \parallel \mathcal{M} \parallel^2 d\phi(N) \tag{9}$$

where for N final states the integration over phase-space has $3N$ dimensions.

$\rightarrow$ **Definition of MC Event Generator:**

- Monte Carlo event generator is a program which produces particle physics events with the same probability as they occur in nature (virtual collider). It performs a cross section integrals calculations and unweights events to give the four momenta of the particles.

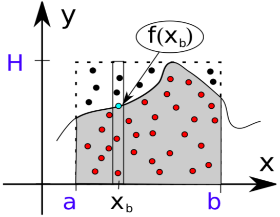
- Among theorists "Monte Carlo" also includes codes which don't provide a fully exclusive information on the final state, but only cross sections or distributions at the parton level (typically at NLO or higher). These codes are also referred as "MC integrators" or "Cross Section integrators".

MC Integration: from Acceptance-Rejection to Averages

⇒ Acceptance-Rejection Method:

The MC integration by the Acceptance-Rejection method has a simple geometrical interpretation as:

$$I = \left(\frac{N_{accept}}{N_{tot}}\right) A_{box} \tag{10}$$



where σ is the cross section, $F = \rho v$ is the incident flux and $W = \frac{dN}{dt}$ the collision rate.

⇒ From Acceptance-Rejection to Averages:: The Aceptance-Rejection method is not optimal. Consider a narrow strip (bin) of height H , around a point x_b:

- For this bin $\frac{N_{accept}}{N_{tot}} H$ is just an estimate for $f(x_b)$
- We can estimate integral $f(x)$ in $[a, b]$ by sampling N points $\{x_i\}$ uniformly in the interval and summing over each bin contibution

$$I \simeq \sum_{i=1}^{N} \left(\frac{b-a}{N}\right) f(x_i) = (b-a)\left[\frac{1}{N}\sum_{i=1}^{N} f(x_i)\right] = (b-a)\langle f \rangle \tag{11}$$

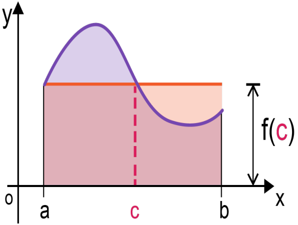
MC Integration as Averages

Another way of understanding the MC integration as an average is through the mean value theorem

⇒ **Mean Value Theorem for Integrals:**
Let $f : [a, b] \to R$ be a continuous function. Then there exists c in (a, b) such that:

$$\int_a^b f(x)dx = (b - a)f(c) \tag{12}$$

where the value of $f(c)$ is the mean value in $[a, b]$



- We can estimate the mean value of $f(x)$ in $[a, b]$ directly by sampling the function uniformly in the interval and taking the average $f(c) \simeq \frac{1}{N}\sum_{i=1}^{N} f(x_i)$

•Integrals as an Average:
The integral of $f(x)$ and its variance can be estimated by sampling N points $\{x_i\}$ in the interval $[a, b]$

$$\begin{cases} I_N = \int_a^b f(x)dx \simeq (b-a)\left[\frac{1}{N}\sum_{i=1}^{N} f(x_i)\right] \\ V_N \simeq \frac{(b-a)^2}{N}\sum_{i=1}^{N}[f(x_i)]^2 - I^2 \end{cases} \Rightarrow I = I_N \pm \sqrt{\frac{V_N}{N}} (CLT - Theorem)$$

MC Integration in n-dimensions

The generalization of the MC integration to n-dimension is given by:

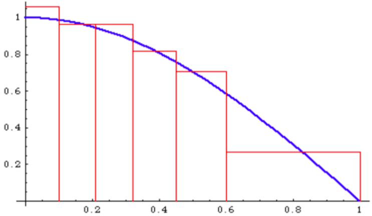
- Consider the n-dimentional integration of a function $f(\vec{x})$ over a volume V:

$$I = \int dVf(\vec{x}) \simeq \frac{1}{N}\sum_{i=1}^{N} Vf(\vec{x_i}), \quad w_i = Vf(\vec{x_i}) \quad event \quad weight \qquad (13)$$

Comparing the convergence of different integration methods, we can see the power of MC integration for high dimensional cases (Numerical Integration Methods in n-dimensions):

- Monte Carlo: $\frac{1}{\sqrt{N}}$ (dimension independent)
- Trapezium : $\frac{1}{N^{(2/n)}}$
- Simpson : $\frac{1}{N^{(4/n)}}$

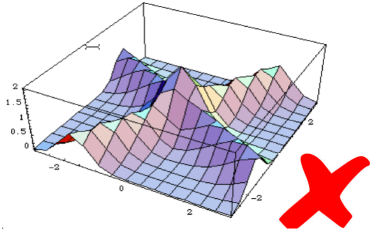
- To increase the integration efficiency, instead of sampling $[a, b]$ uniformly, we sample more points, where $f(x)$ is large. For that we use an importance sampling distribution $g(x)$ that aproximates the integrand
- VEGAS altgorithm: is an adaptative importance sampling algorithm. It determines the sampling distribution iteractively while sampling the integrand $f(x)$, by building a step function approximation(histogram)



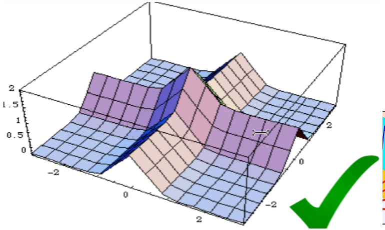
MC Integration : VEGAS Algorithm

- The number of histogram bins in d-dimensions grows as K^n (dimensional curse)
- Vegas avoids this by approximating the distribution by a separable(factorizable) function like: $g(x_1, x_2, ...x_n) = g_1(x_1) \cdot g_2(x_2) ... \cdot g_n(x_n)$
- For factorizable functions the number of bins grows only as K^n
- The efficiency of VEGAS depends on the function factorization

Separability(factorization) is equivalent to aligning the integrand peaks with the coordinate axes (change of variables)

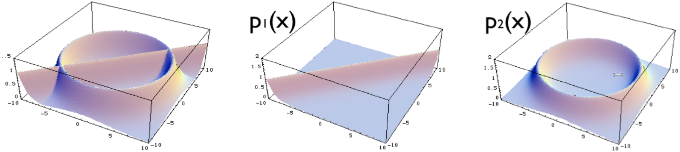


$\Rightarrow$



- If there is no transformation that aligns the integrand peaks to the coordinate axes, VEGAS will mostlikely not work.
- Multichannel integration use different transformations (channels) for separating non-factorizable singularities. Each channel takes care of one peak at the time, such that:

$$\begin{cases} p(x) = \sum_{i=1}^{n} \alpha_i p_i(x), \quad where \quad \sum_{i=1}^{n} \alpha_i = 1 \\ I = \int f(x)dx = \sum_{i=1}^{n} \alpha_i \int p_i(x) \dfrac{f(x)}{p(x)} dx \end{cases}$$



Madgraph uses a multichannel integration based on single diagrams, where the integrand $\mid \mathcal{M}_{tot} \mid^2$ is decomposed as $\mid \mathcal{M}_{tot} \mid^2 = \sum_i f_i$. The basis f_i is defined in terms of individual diagrams amplitude $\mathcal{M}_i$ by:

$$f_i = \frac{\mid \mathcal{M}_i \mid^2}{\sum_j \mid \mathcal{M}_j \mid^2} \mid \mathcal{M}_{tot} \mid^2 = \mid \mathcal{M}_i \mid^2 \tag{14}$$

- The peak structure of each f_i is the same as of that of $\mid \mathcal{M}_i \mid^2$
- The mapping g_i can be derived from diagram propagator structure
- Easy to reweight the channels, so large contributions are evaluated with greater number of MC points
- Decomposes the amplitude into n independent integrations (parallelization)

MC Event Generation

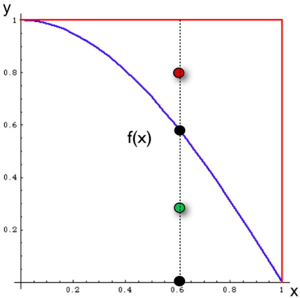
$\Rightarrow$ **Event generation** is done by sampling phase space points (event kinematics). The cross section associates different weights to distinct events and for detector simulation and analysis we want unweighted events like real data.

- Unweighting is performed by applying Acceptance-Rejection method

$\Rightarrow$ **Mean Value Theorem for Integrals:**

Unweighting (Acceptance-Rejection)

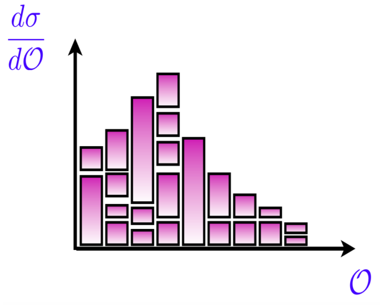
- Random sort x
- Calculate $f(x)$
- Random sort $0 < y < f_{max}$
- If $y \leq f(x)$ accept the event, otherwise reject it



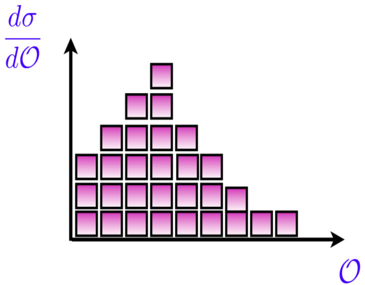
MC Event Generation

Comparinson between a distribution made with weighted versus unweighted events

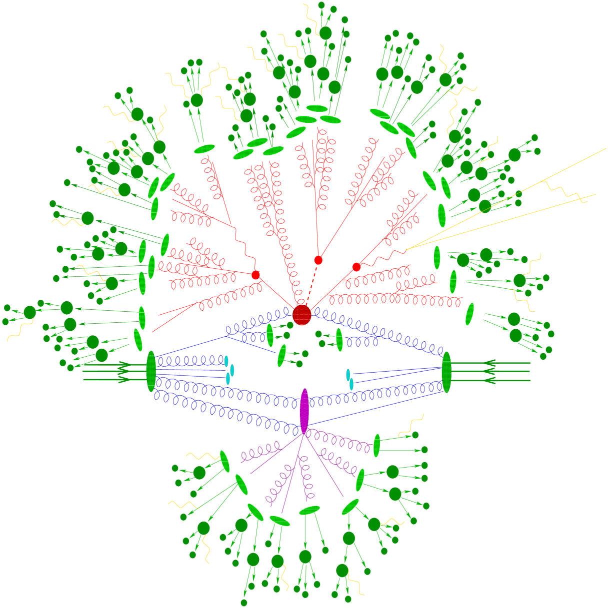
Weighted Events: Aproximately the same number of events in areas of phase space with very different probabilities, so events must have different weights



Unweighted Events: The number of events is proportional to the probability of phase space region, so all events have the same weight. Events distributed as we observe in nature.



Parton Shower

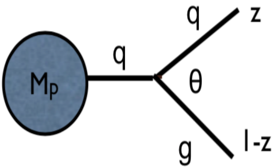


Parton Shower

Acelerated color charged radiates gluons, and as gluons themselves carry colour charges, they can emit further radiation, leading to parton showers

Soft and Collinear Emissions: The QCD matrix elements enhances radiation for soft and collinear emissions (most singular).

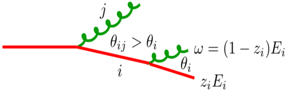
$$\frac{1}{(p_q + p_g)^2} \simeq \frac{1}{2E_q E_g (1 - cos\theta)} \qquad (15)$$



If the n-parton differential cross section before splitting is $d\sigma_n$ the expressions for the collinear and soft aproximations factorizes. For the splittings of a parton (ex: $q \to q + g$) we have:

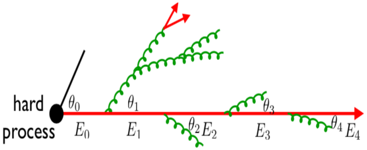
1. Parton Branching (Collinear): $d\sigma_{n+1} \simeq d\sigma_n \frac{\alpha_s}{2\pi} \sum_i P_{ij}(z_i, \phi_j) \frac{d\xi_i}{\xi_i} \frac{d\phi_i}{2\pi}$
2. Parton Branching (Collinear): $d\sigma_{n+1} \simeq d\sigma_n \frac{\alpha_s}{2\pi} \frac{d\omega}{\omega} \sum_i (-\vec{T}_i \cdot \vec{T}_j) \Theta(\xi_{ij} - \xi_i) \frac{d\xi_i}{\xi_i}$

where θ_i is the emission angle , $\xi_i = 1 - cos\theta_i$ and z_i is the parton energy fraction after emission



Hadronization

- Degrees of freedom that reaches detector are hadrons
- After parton shower is terminated quarks and gluons must turn into hadrons
- Need a phenomenological model to describe QCD confinement



$$z = \frac{E_{i+1}}{E_i}$$

1) String Hadronization Model:

- At short distances (large Q), QCD is like QED: colour field lines spread out (1/r potential)
- At long distances, gluon self-attraction gives rise to colour string (linear potential $\Rightarrow$ quark confinement)
- Intense colour field induces quark-antiquark pair creation, which combines into color neutral bound states (hadronization)

2) Cluster Hadronization Model:

- In parton shower, relative transverse momenta evolve from a high scale Q towards lower values
- At a scale near $\Lambda_{QCD} = 200$ MeV , perturbation theory breaks down and hadrons are forme
- Before that, at scales of aproximatley few $\times \Lambda_{QCD}$, there is universal preconfinement of colour

MC Generation Tutorial

Introduction to MC-tools

There are several MC-frameworks able to reproduce SM and BSM phenomenologies, calculation of cross sections and more:

- `MadGraph5_aMC@NLO`, `POWHEG-Box`: to generate events;
- `Pythia-8.2.3.5`: to shower and to hadronize the genrated events;
- It is possible to use the tools mentioned above in different stages of accuracies (LO, NLO, NNLO)
- `Delphes-3.4.1`: to perform fast multipurpose detector response simulations, especially on LHC detectors;

General working-schema

1. Event generation (e.g. ⇒ `p p > h b b~ [QCD]`)
2. Parton shower/hadronization of the generated event (e.g. ⇒ `h > Z Z, Z > l+ l-`)
3. Simulation of the kinematics of a "parton showered/hadronized" generated event in one of the LHC detectors

MC-tools: Part 1

- The MadGraph5_aMC@NLO webpage: `https://cp3.irmp.ucl.ac.be/projects/madgraph/`
- The donload link is: `https://launchpad.net/mg5amcnlo/+download`;
- In oder to properly work, we need `CERN Root, python2.7`;
- It is possibble inside the `MadGraph5_aMC@NLO` interface to include several interfaces, in order to extend the a very large MC analysis of simulated events, as shown below:

```
:~/ mkdir MadGraph
:~/ mv MG5_aMC_v2.6.X.X.tar.gz /path/MadGraph
:~/ cd /path/MadGraph
:~/ tar xf MG5_aMC_v2.6.X.X.tar.gz
:~/ cd MG5_aMC_v2.6.X.X
:~/ mkdir Work_Place
:~/ cd Work_Place
:~/ pwd
/path/MadGraph/MG5_aMC_v2.6.X.X/Work_Place
```

- After the work space is setup, we go further with the interfaced MG5 tools:

```
::~/ MG5_aMC_v2.6.3.2/Work_Place ./../bin/mg5_aMC
MG5_amc > install ExRootAnalysis
MG5_amc > install MadAnalysis5
MG5_amc > install pythia8
MG5_amc > install pythia-pgs
MG5_amc > install Delphes
MG5_amc > exit
```

MC-tools: Part 2

We can shedule at once: generation of the selected collision events, parton shoewer and hadroniztion, and finally multipurpose detector response simulations. The multi-incorporated MadGraph5_aMC@NLO interface offers an executable algorithm for all of this:

```
:∼/ MG5_aMC_v2.6.3.2/Work_Place$ ./../bin/mg5_aMC
MG5_amc > define (specific decay of particles)
MG5_amc > import model (TopEffTh, 2HDM, heft, BSM_gg_hh ...)
MG5_amc > generate p p > h h (BSM_gg_hh)
MG5_amc > output gluon_HH_BSM
MG5_amc > open index.html
MG5_amc > output gluon_HH_BSM
.......................................................................
The following switches determine which programs are run:
/========================================================================|
| 1.  Choose the shower/hadronization program ...  shower = Pythia8 ....|
| 2.  Choose the detector simulation program ..  detector = PGS/Delphes.|
| 3.  Choose an analysis package (plot/convert)..analysis = MadAnalysis5|
| 4.  Decay onshell particles ..................  madspin = OFF ........|
| 5.  Add weights to events for new hypp ......  reweight = OFF ........|
|========================================================================/
```

MC-tools: Part 3

We can download and install even Pythia8 and Delphes separated in the following links:

- `http://home.thep.lu.se/Pythia/`
- `https://cp3.irmp.ucl.ac.be/projects/delphes`

To run `Delphes-3.4.1` we do:

```
 :~/ cd /path/MG5_aMC_v2.6.3.2/Delphes
:~/ MG5_aMC_v2.6.3.2/Delphes$ ./DelphesSTDHEP cards/delphes_cards.tcl
delphes_output_file.root ../path/pythia-pgs_imput_file.hep
:~/ MG5_aMC_v2.6.3.2/Delphes$ root -l -r
examples/EventDisplay.C'("cards/delphes_card_CMS.tcl","delphes_output.root")'
```

References

- https://cp3.irmp.ucl.ac.be/projects/madgraph/
- http://home.thep.lu.se/Pythia/
- https://cp3.irmp.ucl.ac.be/projects/delphes