

Parçacık Fiziği Veri Analizi Kış Okulu 2025 (TBAE PFVA'25)

Olay Üreticiler (MadGraph, Pythia8) ve DELPHES

İlkay TÜRK ÇAKIR
Ankara Üniversitesi
Hızlandırıcı Teknolojileri Enstitüsü

İÇERİK

- Monte-Carlo (MC)
- Olay ve Olay Sayısı
- Partonik Seviye Yazılım
 - MadGraph
- Hadronik Seviye Yazılım
 - Pythia8
- Dedektör Benzetimi Yazılımı
 - Delphes
- Analiz Yazılımı
 - ROOT
 - Uygulama

MOTE CARLO NEDİR?

- MC simülasyonu veya MC yöntemi, adını Monako Prenslığı'ndeki ünlü kumarhaneden dolayı almıştır. Oynanan rulet oyunu rastgele sayılar üretme mekanizmasının en basit örneğidir. Bu isim Stanislaw Marcin Ulam, Enrico Fermi, John von Neumann ve Nick Metropolis gibi ilk kullanıcıları tarafından popüler hale getirilmiştir.
- MC yöntemleri başta “**istatistiksel örnekleme**” gibi daha jenerik isimler altında uygulanmıştır. **MC yöntemi, matematik veya fizik problemlerinin olasılık modellerine uygulanan rasgele denemeler yoluyla çözülmesi yöntemidir.** Bu rasgele yöntemin en önemli kullanımı 1930 yılında Fermi tarafından, o yıllarda yeni keşfedilen nötronların özelliklerini hesaplamak için kullanılmıştır.



- Belirlenimci (deterministic) algoritmaların kurulmasının olanaksız olduğu ya da çok karmaşık algoritmalar gerektiren problemlerin çözümünde başvurulan bir sayısal çözümleme tekniği olan MC yöntemi, sistemli olarak Manhattan Projesi'nin Los Alamos'taki laboratuvarında nötronların çeşitli maddelerden geçebilme özelliklerinin belirlenmesi çalışmalarında kullanılmıştır (1943).

SİMÜLASYON (BENZETİM) ve (MC) NEDİR?

MC Uygulama Alanları

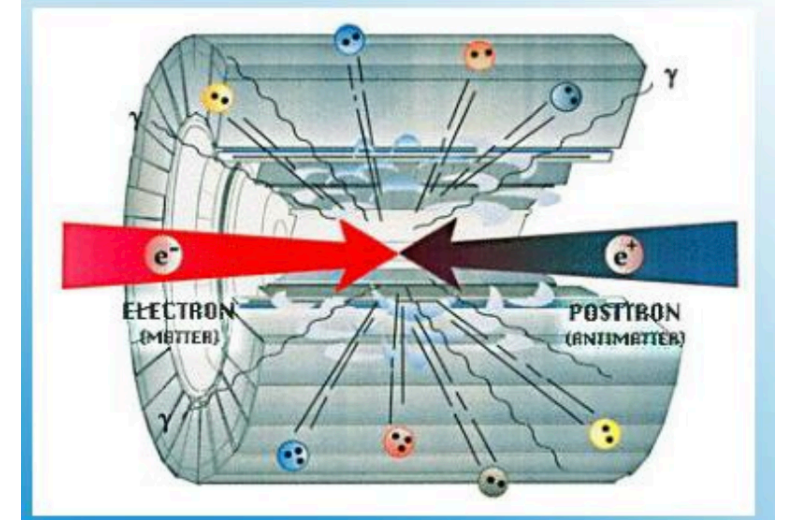
- Sayısal analiz, atom ve molekül fiziği, nükleer fizik, yüksek enerji fiziği, dedektör simülasyonu, hücre simülasyonu, borsa modelleri, dağılım fonksiyonları uygulamalarında kullanılır.
- Çok katlı integrallerin hesaplanmasında, kısmi diferansiyel denklemlerin, doğrusal denklem sistemlerinin çözümünde, nötron yayını, gama ışınımı soğurulması problemlerinde, parçacık fiziğinde bozunum genişliklerinin ve saçılma kesitlerinin hesaplanması gibi bir çok yerde kullanılır.

SİMÜLASYON (BENZETİM) ve (MC) NEDİR?

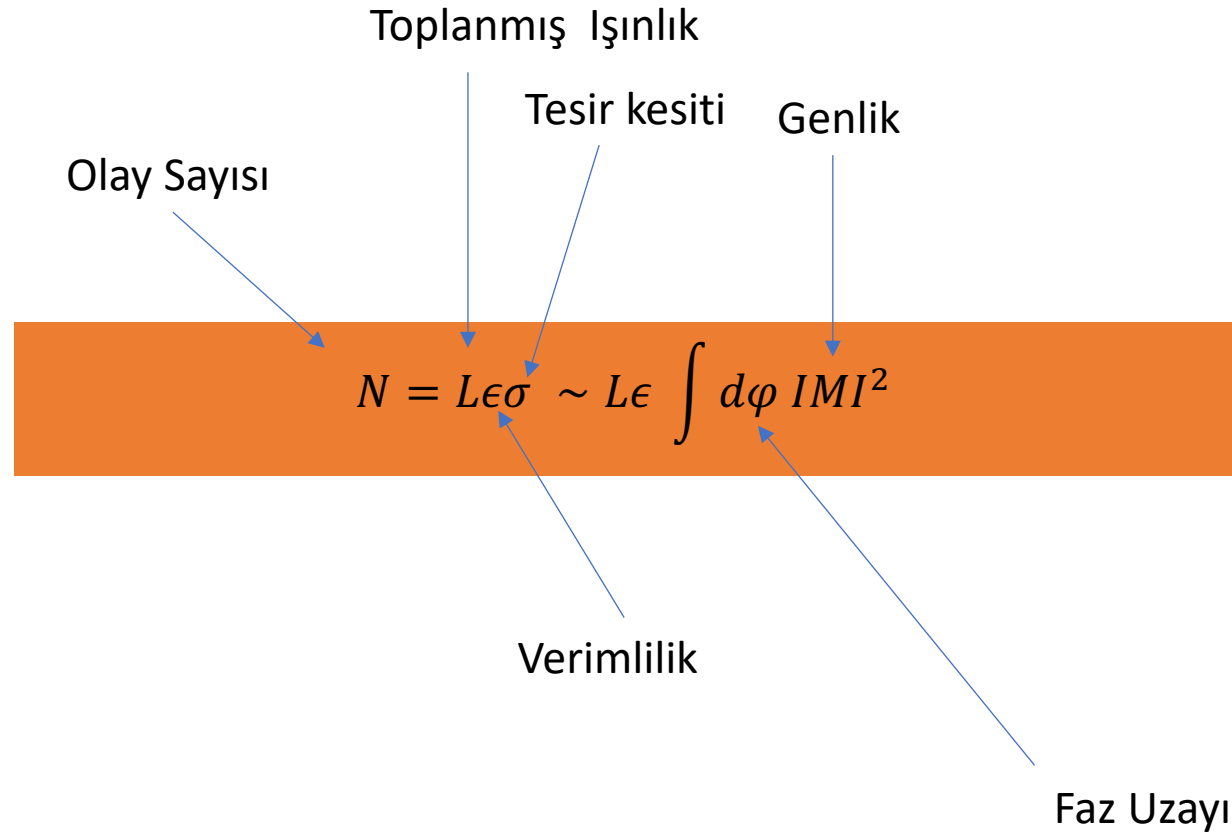
- Rasgele Sayı: Bir kümenin veya dizinin elemanlarından bir kısmını istatistik olarak rasgele seçilmesi yolu ile üretilmiş sayıya rasgele sayı denir. Örneğin, zar atma, tombala çekmek, rulet, kumar makineleri v.b.
- Bilgisayar belirlenimli bir aygıt olduğundan üretilen sayılar aslında tam olarak nitelikli değildir. Rasgele sayılarda bulunması gereken özellikleri belli ölçüde içeren bu sayılar bu nedenle sözde rasgele sayılar olarak adlandırılır.

OLAY NEDİR?

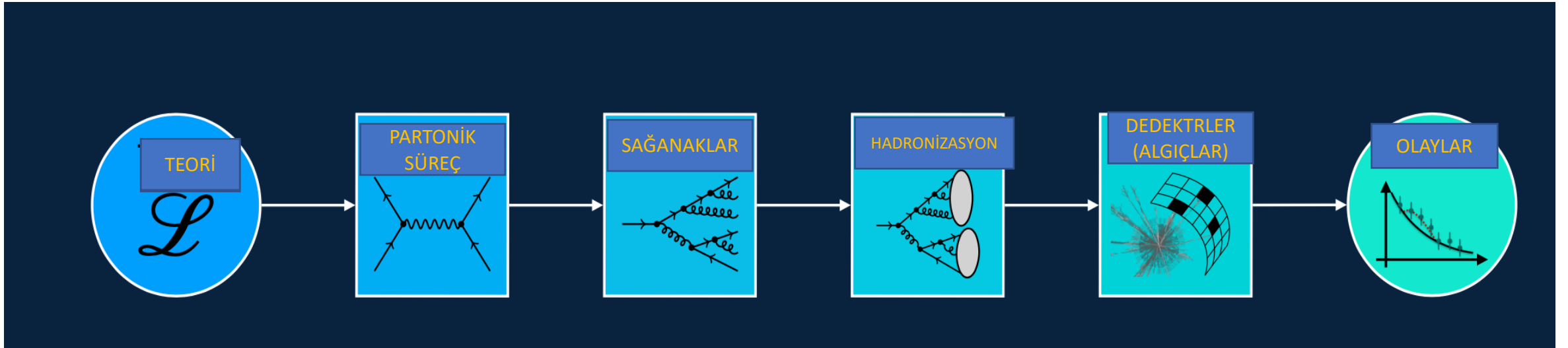
- Doğrusal çarpıştırıcısı (LC) veya Büyük Hadron Çarpıştırıcısı (LHC) yüksek enerji fiziği deneyleri, yüksek performanslı hesap ortamı için en çok istenen alt yapılardır. Parçacıklara yeteri kadar enerji verdikten sonra, bu parçacıklar ya bir hedefle yada birbirleriyle çarpıştırılırlar. Bu çarpışmaların her biri, **OLAY** olarak adlandırılır. Bir yüksek enerji fizikçisinin amacı, her bir olayı ayıklayıp, bu tek olaydaki verileri toplayıp, bu olaydaki parçacık sürecininin test edilen teori ile uyuşup uyuşmadığını incelemektir. Her olay, bir çok parçacık üretildiğinden çok karmaşıktır. Dolayısıyla, başka parçacıklara bozunmadan önce çok kısa mesafeler gittiklerinden gözlenebilir izler bırakmazlar. Dedektörler (algıçlar) bu izleri ve bu sinyalleri algılayan aletlerdir.



OLAY SAYISI NEDİR?

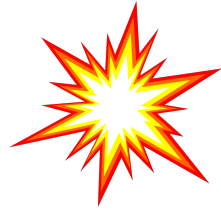


FİZİK OLAYLARI

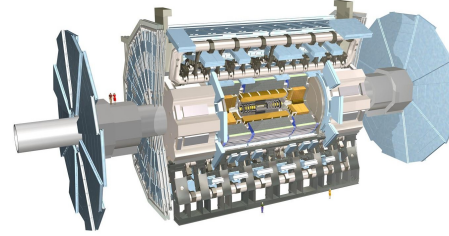


MONTE CARLO BENZETİM ADIMLARI

Veriler

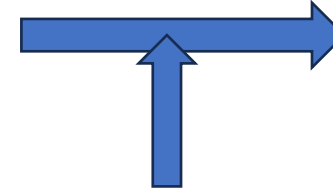


Çarpışma



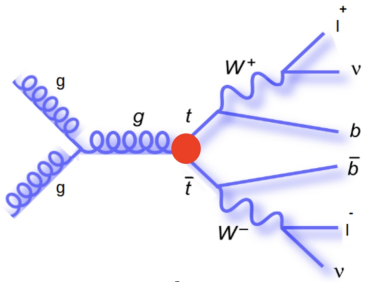
Algılama /
Algıç

Yeniden oluşturma
/ Yapılandırma

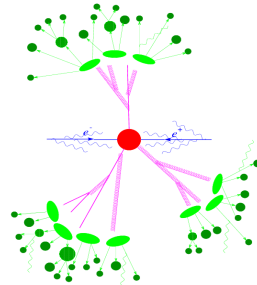


Analiz
girdileri

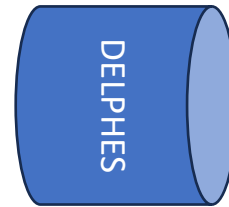
MC



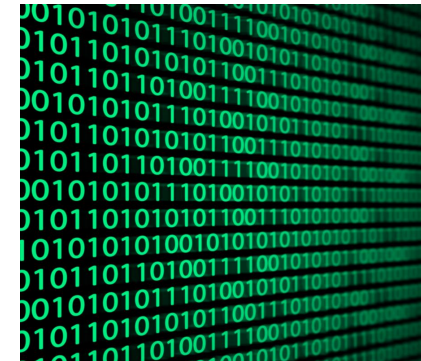
Matris elemanı
hesaplanması



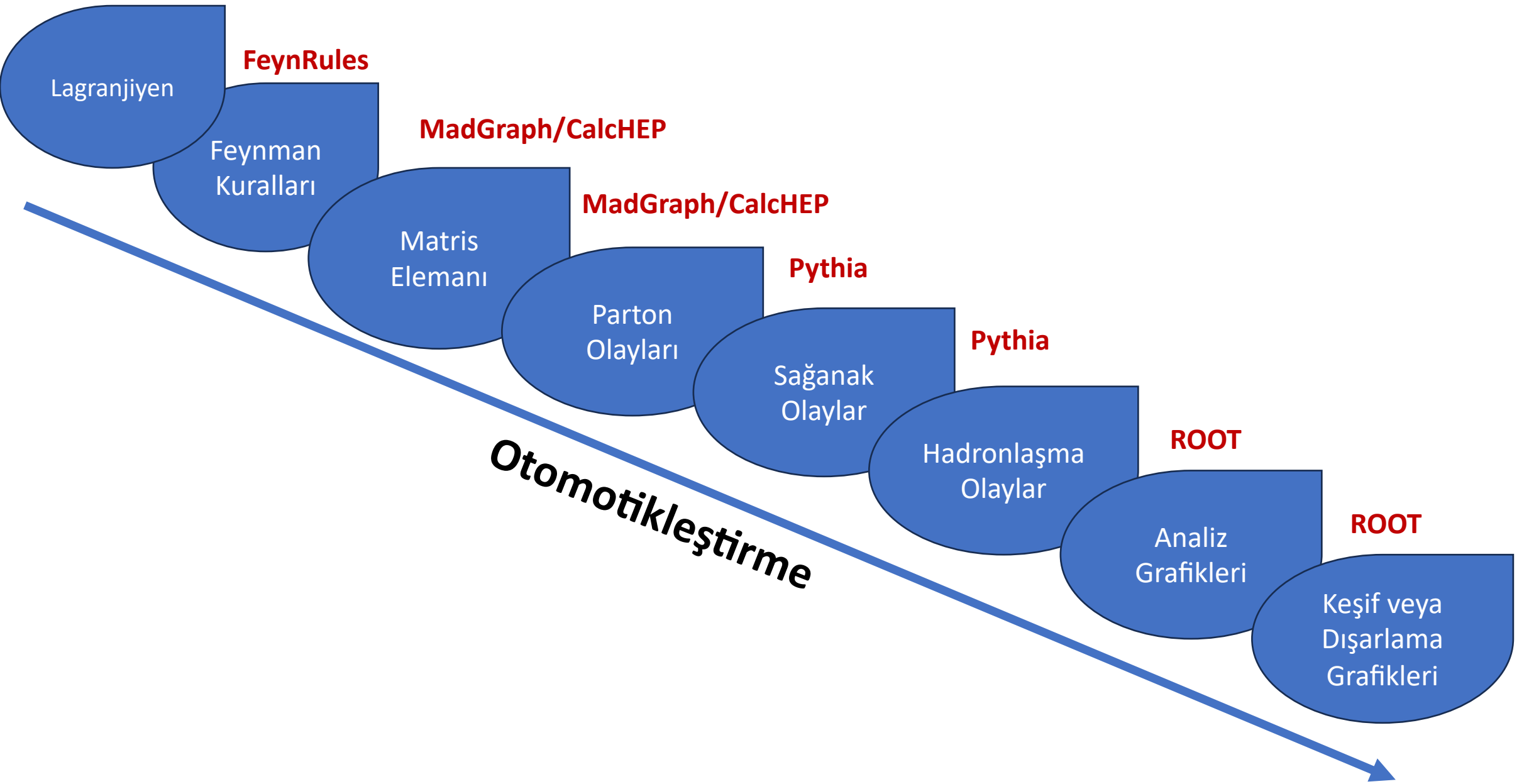
Parton sağanağı /
hadronlaşma



Dedektör
benzetim



Dijitalleştirme



OLAY ÜRETİMİDEKİ ADIMLAR

CalcHEP, MadGraph ve PYTHIA olay üretimine giden son hesap aşamalarını otomatikleştirmeye çalışan programlardır. Verilen bir saçılma süreci için temel basamaklar aşağıdaki gibidir:

- Süreçte bulunan Feynman diyagramlarının çizilmesi
- Matris elemanı ifadelerinin bulunması
- Faz uzayı üzerinden integralin alınması
- Toplam veya diferensiyel tesir kesitinin bulunması
- Herbiri son durum parçacıklarına karşı gelen dörtlü enerji-momentum vektörleri kümesi olarak toplam diferensiyel tesir kesitine göre olaylar **rasgele** olarak üretilir.

MadGraph5 +Pythia8+ Delphes+ ROOT

- MadGraph5
 - Süreç tanımlama, diyagramlar, parametreler
- Pythia8
 - Hadronlaşma, bozunum
- Delphes
 - Dedektör hızlı simülasyonu, bilinen dedektörler için kartlar, parametreler
- ROOT
 - Analiz

MadGraph Özellikleri

- MadGraph programı, Standart Model (SM) ve SM ötesi fenomenoloji için gerekli tüm bileşenleri sağlayan bir pakettir.
- Ağaç seviyesinde matrix elemanı hesaplayıcısı, tesir kesiti hesabı ve olay üretimi yapar.
- Uzun süre C++/Fortran ile hazırlanan MG4 kullanılmış, 2010 yılından bu yana “Python” dilinde hazırlanan MG5 kullanılmaktadır.
- Diğer MC üreticilerine göre; Grid üzerinde, çok çekirdekli ya da küme bilgisayarlarda (pbs) daha başarılı hesaplama yapabilir.
- Terminal üzerinde arayüz sunar. Sonuçlar html olarak verilebilir. Günümüzde olayları simüle etmek için yaygın olarak kullanılmaktadır.

MadGraph İLE NE YAPILIR?

- Bu takım, parton seviyesindeki basit bir tesir kesiti

$$\sigma = \frac{1}{F} \int dPS^{(n)} \overline{|M_{fi}|^2}$$

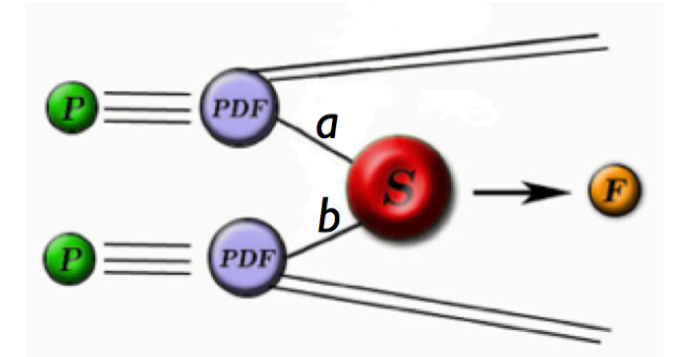
- Matris elemanı karesi: Feynman diagramlarının genlikleri yoluyla gelir, diagramlar modelinin tüm etkileşim ve parçacık bilgisini taşıyan Lagrangieni ile elde edilir.
- F başlangıç durumları üzerinden alınan ortalama akı faktörlerdir. (Flux)
- Tüm faz uzayı (PS) üzerinden integral alınarak hesaplanır.

MadGraph İLE NE YAPILIR?

- Hadron Çarpıştırıcısında bir tesir kesiti hesabı;

$$\sigma = \frac{1}{F} \sum_{ab} \int d\text{PS}^{(n)} dx_a dx_b f_{a/p}(x_a) f_{b/p}(x_b) \overline{|M_{fi}|^2}$$

- Tüm partonlar üzerinden toplam alınır (a ve b).
- f: parton dağılım fonksiyonları (PDF)



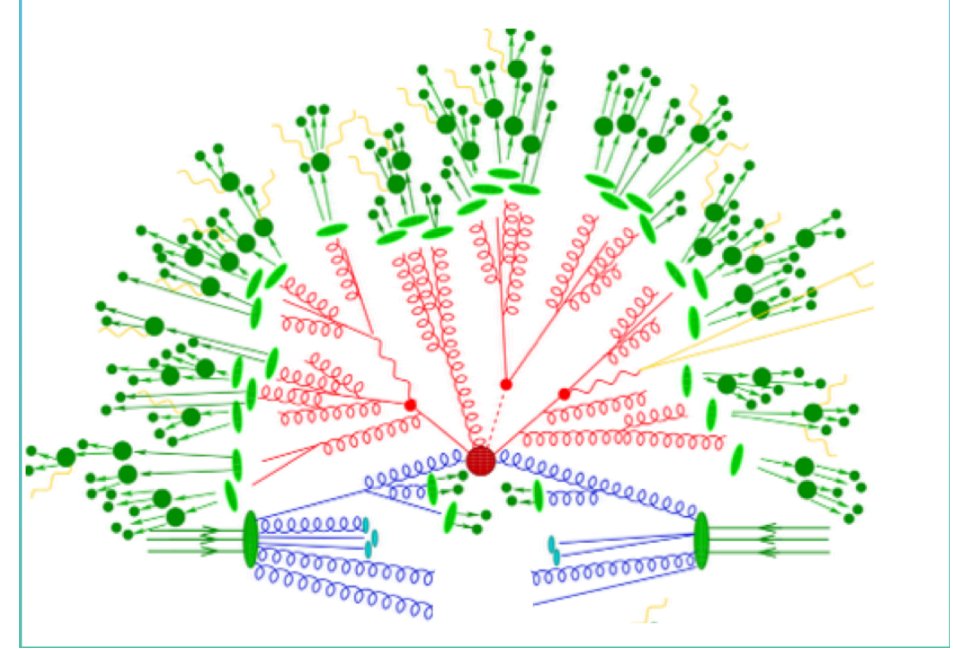
- a ve b protondan gelen birer parton ve x parton fraksiyonları olmak üzere f(x) partonların protonlardan gelme olasılığını gösterir.

MadGraph NE İÇERİR?

- MadGraph yazılımı, çeşitli programları içerecek şekilde genişletilebilir:
 - Bir rastgele olay üretici, parton duşu ve hadronizasyon için kullanılan kod Pythia ve iki dedektör simülatörü (PGS ve Delphes).
- İlgilendiğimiz simülasyon türüne bağlı olarak, bu ek parçalardan bazıları gereksiz olabilir. Örneğin, sadece parton seviyesinde bir tesir kesiti hesaplamak istiyorsak, temel formülü kullanmak yeterlidir. Bununla birlikte, bunun ötesine geçmek ve hadronizasyon veya detektör simülasyonunu dahil etmek istiyorsak, **Pythia** ve PGS veya **Delphes**'i de kullanmak zorundayız. Bu biraz karmaşık gelebilir, ancak pratikte bu araçların kombinasyonu basittir ve aslında MadGraph çerçevesi bunu otomatik olarak yapmaya hazırdır.

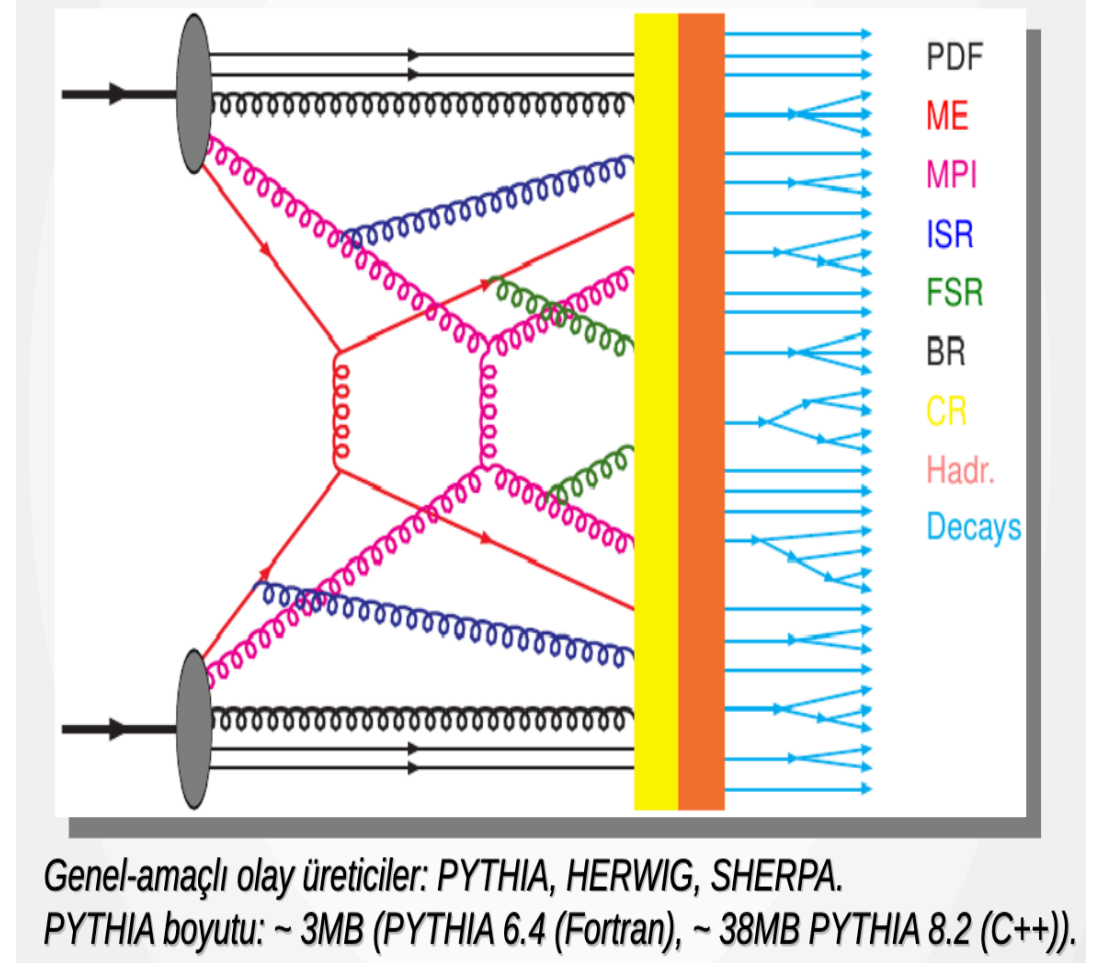
MadGraph NE İÇERİR?

- Yüksek enerjili çarpışmalarda, detektörlerde son aşamada ortaya çıkan hadronlaşma gözlemlenir.
- **Hadronlaşma:** Başlangıçta sürece giren kuark ya da gluonların, son durumda bozunarak asimptotik özgürlüğün izin verdiği ölçüde bir araya gelerek yine kuark ya da gluon ışınımı yapmasıdır.



PYTHIA

- Seçilen fizik modeli çerçevesinde MC teknikleri kullanarak yüksek enerji fiziği **olaylarını üretme** programıdır.
- Parçacık çarpışmalarının ve etkileşmelerinin benzetiminde kullanılır.
- PYTHIA, orta ve yüksek momentum aktarımlı etkileşmeleri, etkileşmede çıkan parçacıkların bozunma ve dallanmalarını, saçılma tesir kesitini, ilk durum ve son durum ışımalarını, çoklu etkileşmeleri, parton dağılım fonksiyonlarını ve partonların hadronlaşması için gerekli alt programları içermektedir.





DELPHES
fast simulation

DELPHES : Parametrik Hızlı Benzetim

- Genel bir çarpıştırıcı deneyinin hızlı benzetimi için bir çerçeve programıdır.
- Delphes, hızlı ve çok amaçlı bir algıç yanıt benzetimi gerçekleştiren bir C++ program çerçevesidir. Benzetim manyetik alana gömülü bir izleme sistemi, kalorimetreler ve bir müon sistemi içerir. Bu çerçeve, standart dosya formatlarına (örneğin, Les Hauches Olay Dosyası (LHE) veya High Energy Monte Carlo (HepMC) bağlanır ve ayrı ayrı analizler için kullanılacak izole leptonlar, kayıp enerji ve jet toplama gibi gözlemlenebilir çıktılar üretir. Algıç yanıtının benzetimi manyetik alanın etkisini, kalorimetrelerin tanecikliğini ve alt algıç çözünürlüklerini dikkat alır. Son durum parçacıklarının görselleştirilmesi, aynı zamanda karşılık gelen ROOT kütüphanesi kullanılarak gerçekleştirilebilir.

DELPHES

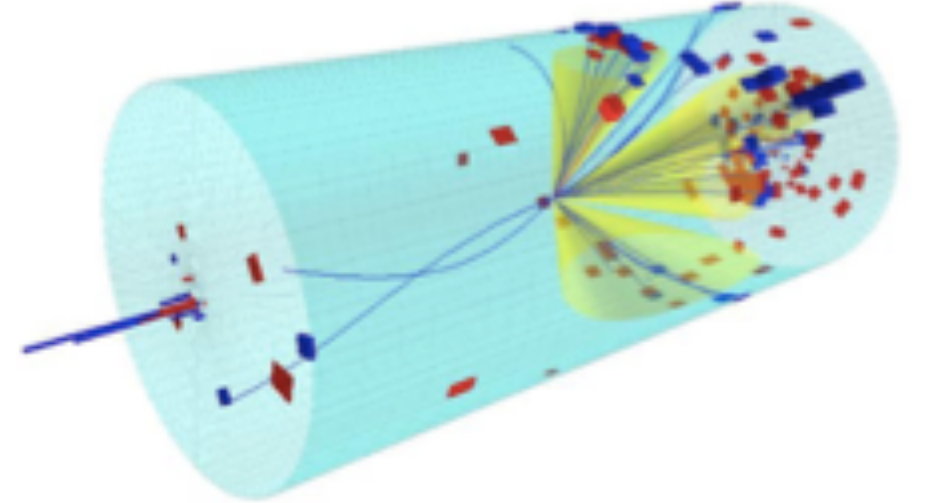
- Delphes, parametrize edilmiş bir yapısalılık içinde çok amaçlı algıçların tepkilerinin benzetimini yapabilen modüler bir çerçevedir.

NE İÇERİR?

- birikme (pile-up)
- manyetik alanda yüklü parçacık iletimi
- elektromanyetik ve hadronik kalorimetreler
- müon sistem

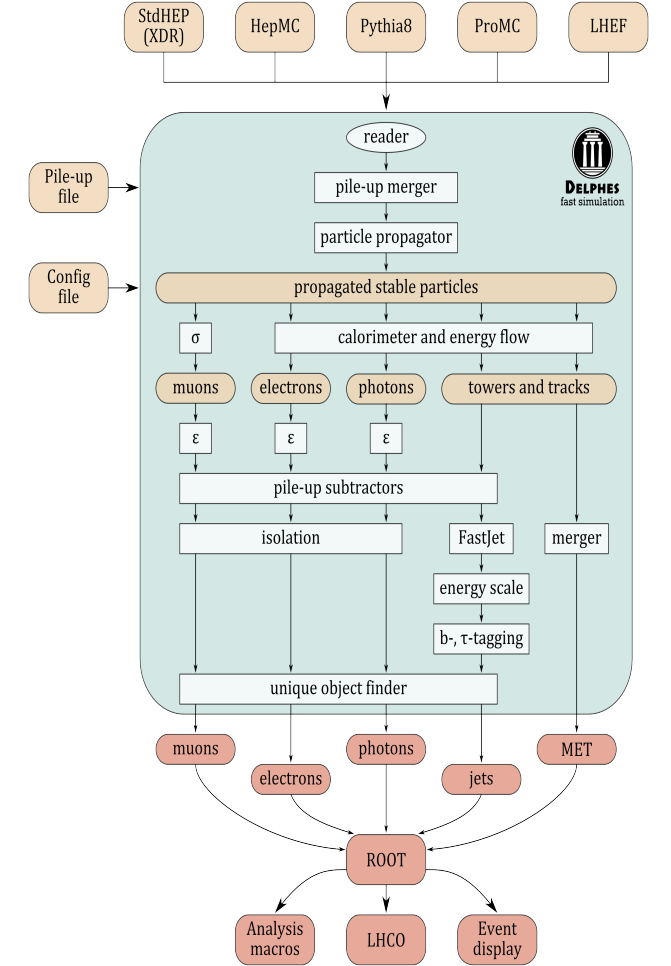
NE SAĞLAR?

- lepton (elektronlar ve müonlar)
- fotonlar
- jetler ve kayıp enine enerji (parçacık- akış)
- tau'lar ve b'ler



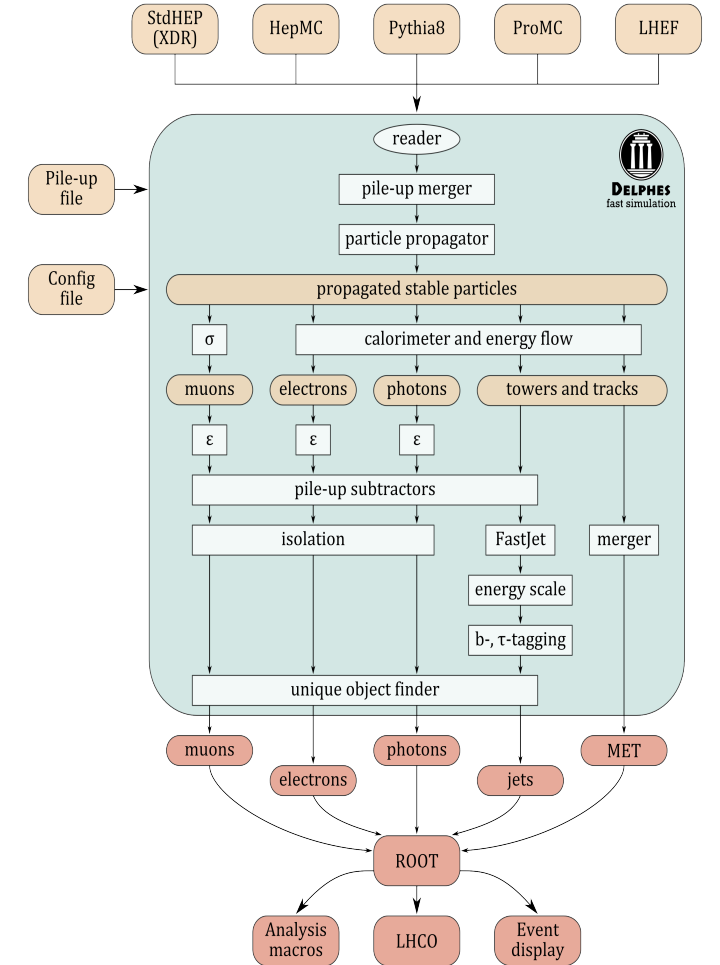
DELPHES: VERİ AKIŞ DİYAGRAMI

- Harici Monte-Carlo jeneratörlerinden gelen olay dosyaları ilk önce bir okuyucu aşama tarafından işlenir.
- Yığılma olayları daha sonra sert saçılma olayına üst üste biner.
- Uzun ömürlü parçacıklar, homojen bir manyetik alan içinde kalorimetrelere yayılır.
- Kalorimetrelere ulaşan parçacıklar enerjilerini kalorimetrelerde biriktirir.



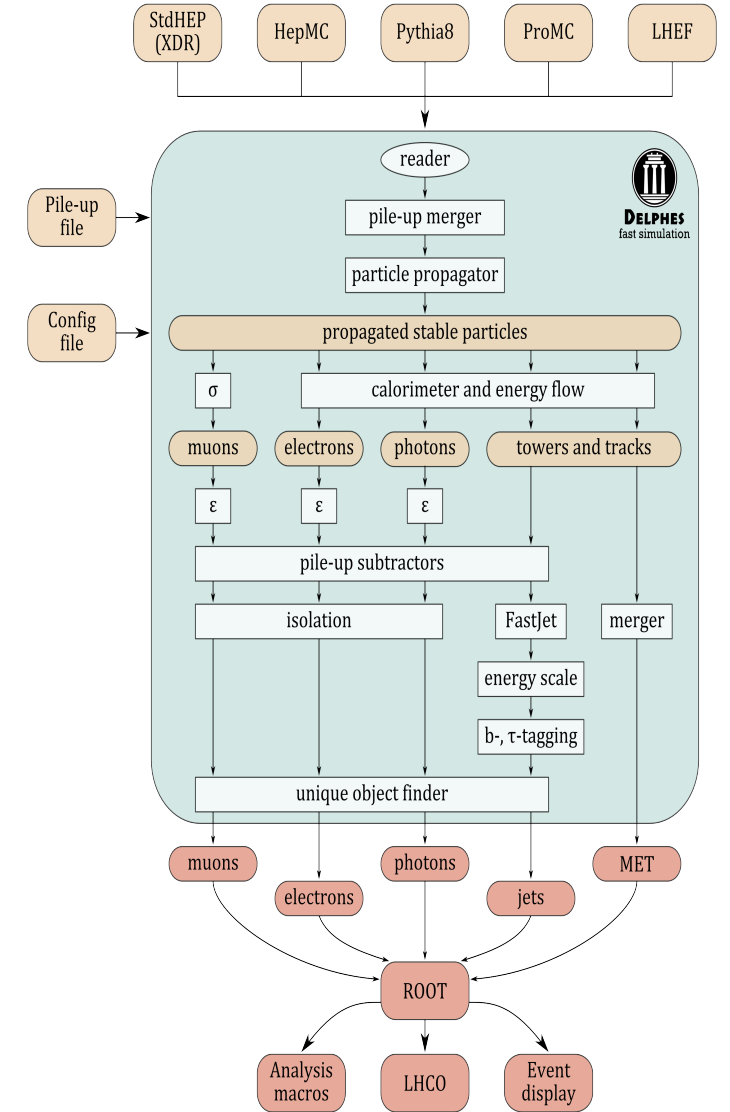
DELPHES: VERİ AKIŞ DİYAGRAMI

- Parçacık akış algoritması 4 vektörden oluşan koleksiyon üretir: elektronlar, fotonlar, parçacık akış izleri ve kuleler
- Muonlar kararlı parçacıklardan seçilir ve 4 vektörleri bulunur.
- Yığılma tepe noktalarından gelen yüklü hadronlar atılır ve kalan olay yığılma yoğunluğu hesaplanır.
- Yığılma yoğunluğu daha sonra, jetler üzerinde ve müonlar, elektronlar ve fotonlar için izolasyon parametresi üzerinde biriktirme çıkarma işlemini gerçekleştirmek için kullanılır. Eksik enerjide birikme çıkarma işlemi gerçekleştirilmez.



DELPHES: VERİ AKIŞ DİYAGRAMI

- Son aşamada, yeniden yapılandırılmış nesnelerin kopyaları kaldırılır.
- Çıktı verileri bir **ROOT** ağacı biçiminde saklanır ve **ROOT** çerçevesi yardımıyla analiz edilebilir. Kök ağaç dosyaları da LHCO dosya formatına dönüştürülebilir.
- Her adımda yapılanma dosyası tarafından kontrol edilir.



ROOT

- ROOT içinde olan fakat C++'da olmayan özel komutlar bir . işareti ile birlikte yazılır.

Örnekler:

- ROOT'dan çıkmak için: `.q`
 - kabuk komutu kullanmak için: `!.!`
 - ROOT'u çalıştırmak için: `root` veya `root -l`
 - bir makro yüklemek için: `.L` dosya adı
 - bir makro çalıştırmak için: `.x` dosya adı
 - yardım için: `.help` veya `?.?`
-
- NOT: Makro nedir? Belli bir işi yapmak üzere her istediğinizde çalıştırabileceğiniz bir dosya içerisine yazılmış komutlar ve fonksiyonlar dizisidir.

ROOT KURULUMU

<https://root.cern> sayfasından bilgisayarınıza uygun platformu olan ROOT'u seçip indirebilirsiniz.

ROOT: analyzing petabytes of data, scientifically.
An open-source data analysis framework used by high energy physics and others.

[Learn more](#) [Install v6.28/04](#)

LATEST STABLE Release 6.28/04 - 7 May 2023

Version 6

Release 6.28/04 - 07 May 2023

Release 6.28/02 - 23 Mar 2023

Release 6.28/00 - 04 Feb 2023

Release 6.26/10 - 16 Nov 2022

Release 6.26/06 - 29 Jul 2022

Release 6.26/04 - 07 Jun 2022

Release 6.26/00 - 13 May 2022

Release 6.10/04 - 28 Jul 2017

Release 6.10/02 - 06 Jul 2017

Release 6.10/00 - 13 Jun 2017

Release 6.09/02 - 08 Mar 2017

Release 6.08/06 - 02 Mar 2017

Release 6.08/04 - 13 Jan 2017

get the latest binaries, features and quick user support.

Download a pre-compiled binary distribution

We distribute pre-compiled ROOT for several major Linux distributions as well as MacOS and (as a beta) Windows. The steps to install a pre-compiled binary are simple:

1. Install all [required dependencies](#) with the system package manager
2. [Download the release](#) for the desired platform and ROOT version
3. Unpack the archive
4. Add the ROOT libraries and executables to your environment by sourcing the appropriate `thisroot.*` script. These setup scripts can be found in the ROOT binary release, in the `bin` directory.

For example, on CentOS8, a user could execute the following bash commands to install ROOT v6.28/04, after installing all [required dependencies](#):

Binary distributions

Platform	Files	Size
CentOS 8	root_v6.28.04.Linux-centos8-x86_64-gcc8.5.tar.gz	225M
Fedora 36	root_v6.28.04.Linux-fedoraVERSION_ID=36-x86_64-gcc12.2.tar.gz	301M
Ubuntu 18.04	root_v6.28.04.Linux-ubuntu18-x86_64-gcc7.5.tar.gz	243M
Ubuntu 20.04	root_v6.28.04.Linux-ubuntu20-x86_64-gcc9.4.tar.gz	244M
Ubuntu 22.04	root_v6.28.04.Linux-ubuntu22-x86_64-gcc11.3.tar.gz	291M
macOS 11.7 arm64 Xcode 12	root_v6.28.04.macos-11.7-arm64-clang120.pkg	344M
macOS 11.7 arm64 Xcode 12	root_v6.28.04.macos-11.7-arm64-clang120.tar.gz	223M
macOS 11.7 x86_64 Xcode 12	root_v6.28.04.macos-11.7-x86_64-clang120.pkg	359M
macOS 11.7 x86_64 Xcode 12	root_v6.28.04.macos-11.7-x86_64-clang120.tar.gz	235M
macOS 12.6 arm64 Xcode 14	root_v6.28.04.macos-12.6-arm64-clang140.pkg	341M
macOS 12.6 arm64 Xcode 14	root_v6.28.04.macos-12.6-arm64-clang140.tar.gz	223M
macOS 12.6 x86_64 Xcode 14	root_v6.28.04.macos-12.6-x86_64-clang140.pkg	355M
macOS 12.6 x86_64 Xcode 14	root_v6.28.04.macos-12.6-x86_64-clang140.tar.gz	234M
macOS 13.3 arm64 Xcode 14	root_v6.28.04.macos-13.3-arm64-clang140.pkg	354M

MadGraph (LaunchPad)

- <https://launchpad.net/mg5amcnlo>



MadGraph5_aMC@NLO

[Log in / Regi](#)

Overview

Code

Bugs

Blueprints

Translations

Answers

Registered 2009-09-15 by Michel Herquet

MadGraph5_aMC@NLO is a framework that aims at providing all the elements necessary for SM and BSM phenomenology, such as the computations of cross sections, the generation of hard events and their matching with event generators, and the use of a variety of tools relevant to event manipulation and analysis. Processes can be simulated to LO accuracy for any user-defined Lagrangian, and the NLO accuracy in the case of models that support this kind of calculations -- prominent among these are QCD and EW corrections to SM processes. Matrix elements at the tree- and one-loop-level can also be obtained.

MadGraph5_aMC@NLO is the new version of both MadGraph5 and aMC@NLO that unifies the LO and NLO lines of development of automated tools within the MadGraph family. It therefore supersedes all the MadGraph5 1.5.x versions and all the beta versions of aMC@NLO. As such, the code allows one to simulate processes in virtually all configurations of interest, in particular for hadronic and e+e- colliders; starting from version 3.2.0, the latter include Initial State Radiation and beamstrahlung effects.

The standard reference for the use of the code is: J. Alwall et al, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations", arXiv:1405.0301 [hep-ph]. In addition to that, computations in mixed-coupling expansions and/or of NLO corrections in theories other than QCD (eg NLO EW) require the citation of: R. Frederix et al, "The automation of next-to-leading order electroweak calculations", arXiv:1804.10017 [hep-ph]. A more complete list of references can be found here: http://amcatnlo.web.cern.ch/amcatnlo/list_refs.htm

Download:

Get Involved

[Report a bug](#)

[Ask a question](#)

[Register a blueprint](#)

[Help translate](#)

Downloads

Latest version is 3.5.x

[MG5aMC_LTS_v2.9.16.tar.gz](#)

[MG5_aMC_v3.5.1.tar.gz](#)

released on 2023-05-12

MG5_aMC_v3.5.0.tar.gz indirilir. Python3 ile uyumludur.

MadGraph KURULUMU

- `tar zxvf MG5_aMC_v3.5.0.tar.gz`
- `./bin/mg5_aMC`
- `install pythia8`
- `install Delphes`

(NOT 1: Eğer burada hata verirse, (örneğin require g77 gibi), bilgisayarınız Mac ise uyumlu olan versiyona uyumlu olan gfortran doyası indirilir. Örneğin, macOS Big Sur ise gfortran 11.2 –BigSur.dmg indirdir ve `install Delphes` komutu tekrarlanır.)

NOT 2: Root Delphes'ten önce mutlaka kurulmalıdır. ROOT'un kurulumu MadGraph'dan bağımsızdır.

NOT 3: Eğer “install Delphes” komutu ile yine de başaramaz iseniz bir sonraki slayttaki adımları izlemeniz yeterli olacaktır.

DELPHES KURULUMU

- <https://github.com/delphes/delphes> bağlantısına tıklayıp <https://github.com/delphes/delphes.git> adresi kullanılarak

git clone https://github.com/delphes/delphes.git
komutu ile Delphes indirilir.

Sonra sırası ile aşağıdaki komutları çalıştırılarak kurulur.

```
mv delphes Delphes
```

```
cd Delphes
```

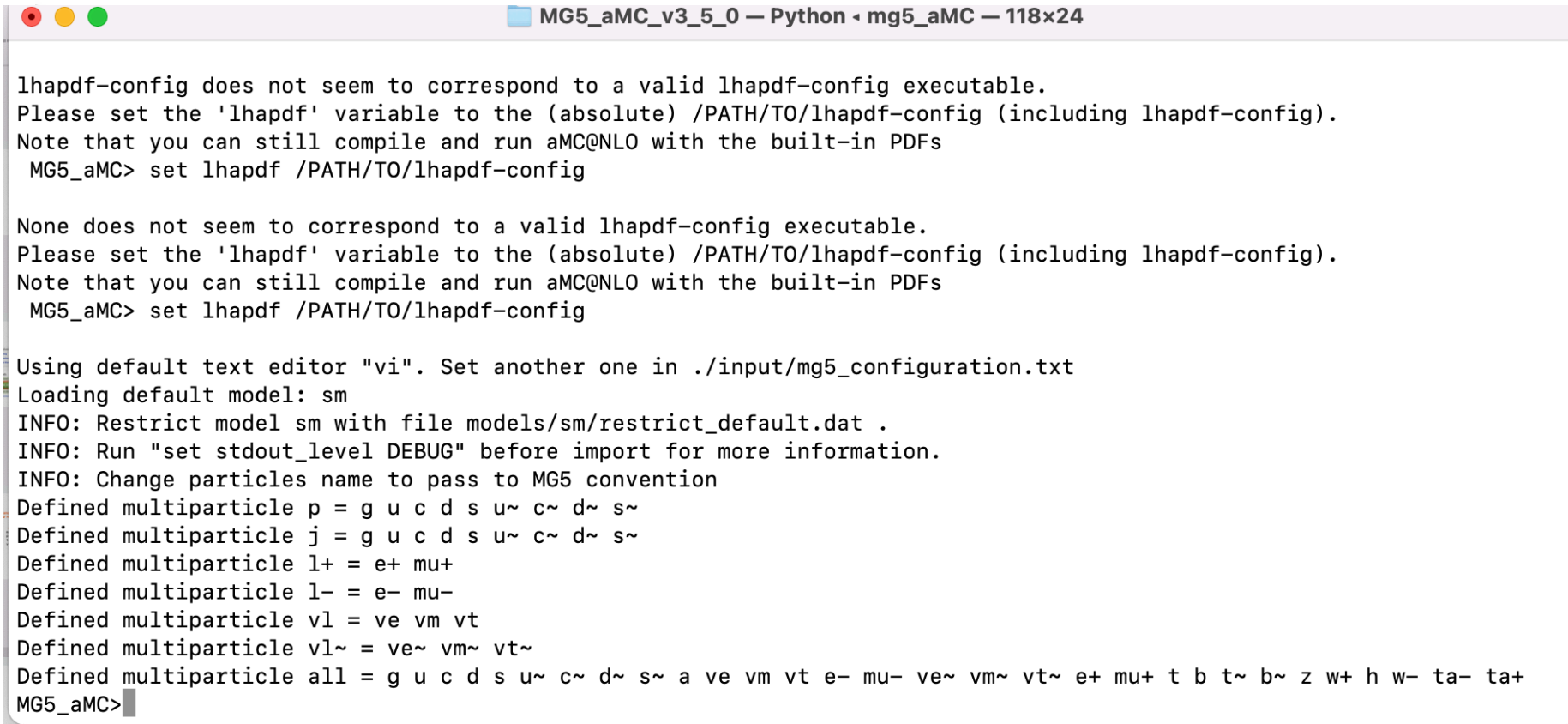
```
make -j4
```

```
make display
```

UYGULAMA

```
MG5_aMC>exit
```

```
ilkay@ilkay1 MG5_aMC_v3_5_0 % ./bin/mg5_aMC
```



```
MG5_aMC_v3_5_0 - Python - mg5_aMC - 118x24
lhapdf-config does not seem to correspond to a valid lhapdf-config executable.
Please set the 'lhapdf' variable to the (absolute) /PATH/TO/lhapdf-config (including lhapdf-config).
Note that you can still compile and run aMC@NLO with the built-in PDFs
MG5_aMC> set lhapdf /PATH/TO/lhapdf-config

None does not seem to correspond to a valid lhapdf-config executable.
Please set the 'lhapdf' variable to the (absolute) /PATH/TO/lhapdf-config (including lhapdf-config).
Note that you can still compile and run aMC@NLO with the built-in PDFs
MG5_aMC> set lhapdf /PATH/TO/lhapdf-config

Using default text editor "vi". Set another one in ./input/mg5_configuration.txt
Loading default model: sm
INFO: Restrict model sm with file models/sm/restrict_default.dat .
INFO: Run "set stdout_level DEBUG" before import for more information.
INFO: Change particles name to pass to MG5 convention
Defined multiparticle p = g u c d s u~ c~ d~ s~
Defined multiparticle j = g u c d s u~ c~ d~ s~
Defined multiparticle l+ = e+ mu+
Defined multiparticle l- = e- mu-
Defined multiparticle vl = ve vm vt
Defined multiparticle vl~ = ve~ vm~ vt~
Defined multiparticle all = g u c d s u~ c~ d~ s~ a ve vm vt e- mu- ve~ vm~ vt~ e+ mu+ t b t~ b~ z w+ h w- ta- ta+
MG5_aMC>
```

```

heft                loop_sm-no_widths          sm-zeromass_ckm
heft-ckm            loop_sm-parallel_test      sm/
heft-full           loop_sm-test               SMScalars
heft-no_b_mass      loop_sm-zeromass_ckm      SMScalars-full
heft-no_masses      loop_sm/                   SMScalars/
heft-no_tau_mass    MSSM_SLHA2-full           taudecay_UFO/
heft-zeromass_ckm  MSSM_SLHA2-no_b_mass      TopEffTh
hgg_plugin/         MSSM_SLHA2-no_masses      TopEffTh-
loop_qcd_qed_sm     MSSM_SLHA2-no_tau_mass    triplet_diquarks
loop_qcd_qed_sm-full MSSM_SLHA2/                triplet_diquarks-
loop_qcd_qed_sm-no_widths nmssm                       uutt_sch_4fermion
loop_qcd_qed_sm-with_b_mass nmssm-full                 uutt_sch_4fermion-
loop_qcd_qed_sm-with_b_mass_no_widths RS                          uutt_tch_scalar
loop_qcd_qed_sm_Gmu RS-                          uutt_tch_scalar-
loop_qcd_qed_sm_Gmu-ckm sextet_diquarks            VLQ_v4_4FNS_UFO-3rd
loop_qcd_qed_sm_Gmu-full sextet_diquarks-          VLQ_v4_4FNS_UFO-3rdL
loop_qcd_qed_sm_Gmu-no_widths sm-c_mass                 VLQ_v4_4FNS_UFO-3rdL_noCKM
loop_sm-c_mass      sm-ckm                    VLQ_v4_4FNS_UFO/
loop_sm-ckm         sm-full
MG5_aMC> import model sm-full
INFO: Change particles name to pass to MG5 convention
Kept definitions of multiparticles p / j / l+ / l- / vl / vl~ unchanged
Defined multiparticle all = g u d s u~ d~ s~ a ve vm vt ve~ vm~ vt~ c t b c~ t~ b~ z w+ h w- e- mu- ta- e+ mu+ ta+
MG5_aMC>

```

```
[MG5_aMC>generate e+ e- > e+ e-
```

```
INFO: Checking for minimal orders which gives processes.  
INFO: Please specify coupling orders to bypass this step.  
INFO: Trying process: e+ e- > e+ e- WEIGHTED<=4 @1  
INFO: Process has 6 diagrams  
1 processes with 6 diagrams generated in 0.012 s  
Total: 1 processes with 6 diagrams
```

```
MG5_aMC>output ilkay
```

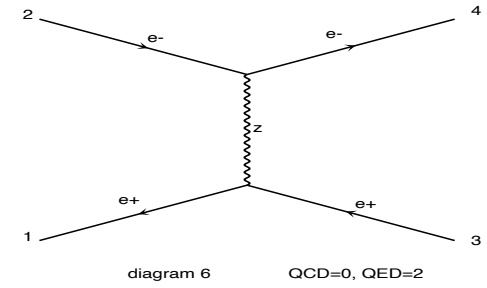
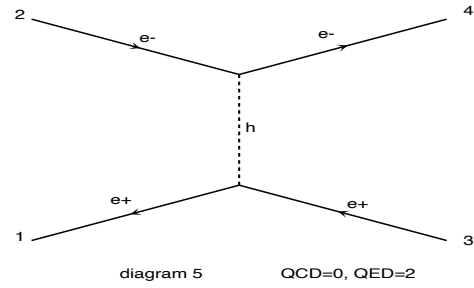
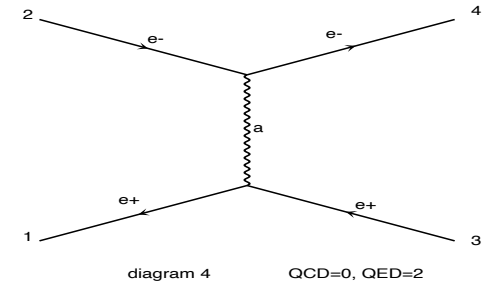
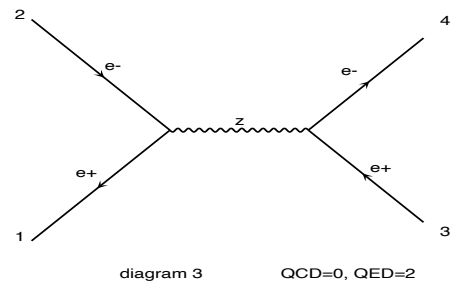
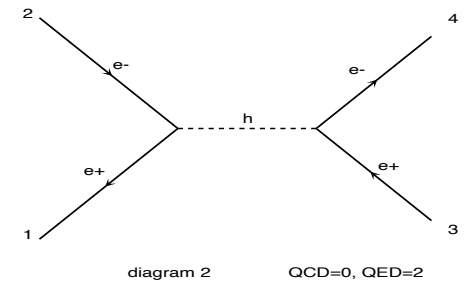
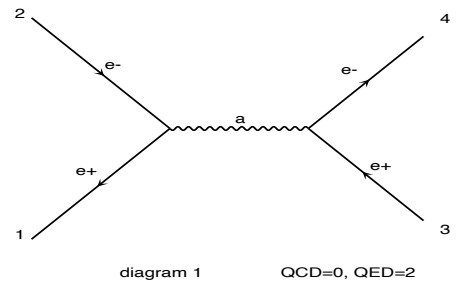
```
>y  
INFO: initialize a new directory: ilkay  
INFO: remove old information in ilkay  
INFO: Organizing processes into subprocess groups  
INFO: Generating Helas calls for process: e+ e- > e+ e- WEIGHTED<=4 @1  
INFO: Processing color information for process: e+ e- > e+ e- @1  
INFO: Creating files in directory P1_epem_epem  
INFO: Some T-channel width have been set to zero [new since 2.8.0]  
if you want to keep this width please set "zerowidth_tchannel" to False  
INFO: Generating Feynman diagrams for Process: e+ e- > e+ e- WEIGHTED<=4 @1  
INFO: Finding symmetric diagrams for subprocess group epem_epem  
Generated helas calls for 1 subprocesses (6 diagrams) in 0.010 s  
Wrote files for 16 helas calls in 0.127 s  
ALOHA: aloha starts to compute helicity amplitudes  
ALOHA: aloha creates 4 routines in 2.785 s  
save configuration file to /Users/ilkay/MG5_aMC_v3_5_0/ilkay/Cards/me5_conf  
INFO: Use Fortran compiler gfortran  
INFO: Use c++ compiler clang  
INFO: Generate web pages  
Output to directory /Users/ilkay/MG5_aMC_v3_5_0/ilkay done.  
Type "launch" to generate events from this process, or see  
/Users/ilkay/MG5_aMC_v3_5_0/ilkay/README  
Run "open index.html" to see more information about this process.  
MG5_aMC>
```

Total: 1 processes with 6 diagrams

```
[MG5_aMC>display diagrams
```

e+ e- > e+ e- WEIGHTED=4

page 1/1



Run "open index.html" to see more information about this process.

MG5_aMC> launch

The following switches determine which programs are run:

```
/=====\  
| 1. Choose the shower/hadronization program      shower = OFF  
| 2. Choose the detector simulation program        detector = OFF  
| 3. Choose an analysis package (plot/convert)    analysis = Not Avail.  
| 4. Decay onshell particles                       madspin = OFF  
| 5. Add weights to events for new hypp.          reweight = OFF  
\=====/
```

Either type the switch number (1 to 5) to change its setting,
Set any switch explicitly (e.g. type 'shower=Pythia8' at the prompt)
Type 'help' for the list of all valid option

ilkey@ilkays-MacBook-Pro ~ % cd MG5_aMC_v3_5_0/il/Events/run_01

NOT: Bu aşamada hızlı hareket etmek gerekiyor, çünkü beklendiği zaman otomatik olarak seçim yapıyor.

The following switches determine which programs are run:

```
/=====\  
| 1. Choose the shower/hadronization program      shower = Pythia8  
| 2. Choose the detector simulation program        detector = Delphes  
| 3. Choose an analysis package (plot/convert)    analysis = Not Avail.  
| 4. Decay onshell particles                       madspin = OFF  
| 5. Add weights to events for new hypp.          reweight = OFF  
\=====/
```

Either type the switch number (1 to 5) to change its setting,
Set any switch explicitly (e.g. type 'shower=OFF' at the prompt)
Type 'help' for the list of all valid option

Type '0', 'auto', 'done' or just press enter when you are done.

>

ilkey TÜRK ÇAKIR, Ankara Üniversitesi

- 1
- 2
- 0

Do you want to edit a card (press enter to bypass editing)?

```
-----\
| 1. param   : param_card.dat
| 2. run     : run_card.dat
| 3. pythia8 : pythia8_card.dat
| 4. delphes : delphes_card.dat
|-----/
```

you can also

- enter the path to a valid card or banner.
- use the 'set' command to modify a parameter directly.
The set option works only for param_card and run_card.
Type 'help set' for more information on this command.
- call an external program (ASperGE/MadWidth/...).
Type 'help' for the list of available command

[0, done, 1, param, 2, run, 3, pythia8, 4, enter path, ...][90s to answer]

>

- 2 (run_card.dat dosyasına girer). Burada i tuşuna basıp gerekli değişiklikler yapılır ve esc tuşuna basılır ve :x kullanılır ise kayıt eder.
- Eğer kayıt etmek istemez isek :q komutu verilir.

```
# Collider type and energy *
# lpp: 0=No PDF, 1=proton, -1=antiproton, *
#          2=elastic photon of proton/ion beam *
#          +/-3=PDF of electron/positron beam *
#          +/-4=PDF of muon/antimuon beam *
#*****
#          0          = lpp1      ! beam 1 type
#          0          = lpp2      ! beam 2 type
#          45.0       = ebeam1    ! beam 1 total energy in GeV
#          45.0       = ebeam2    ! beam 2 total energy in GeV
#*****
```

0 yazılır ise artık sonuçlar alınır:

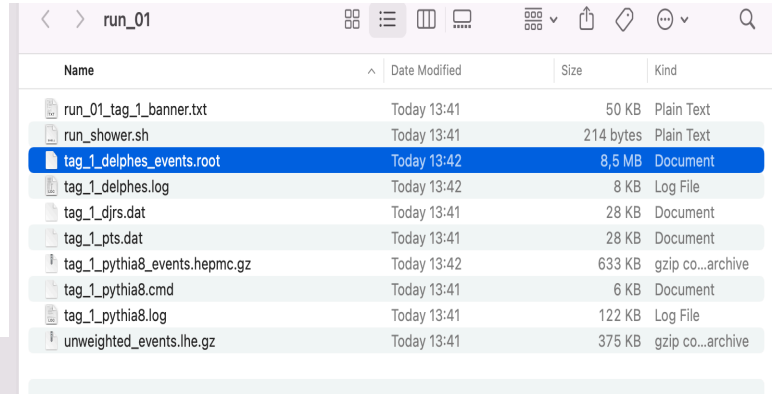


Results in the sm-full for e+ e- > e+ e-

Available Results

Run	Collider	Banner	Cross section (pb)	Events	Data	Output	Action	
run_01	e+ e- 45.0 x 45.0 GeV	tag_1	4185 ± 4.1	10000	parton madevent	LHE	remove run	launch detector simulation
					pythia8	LOG HEPMC	remove run	launch detector simulation
					delphes	LOG rootfile	remove run	

Main Page

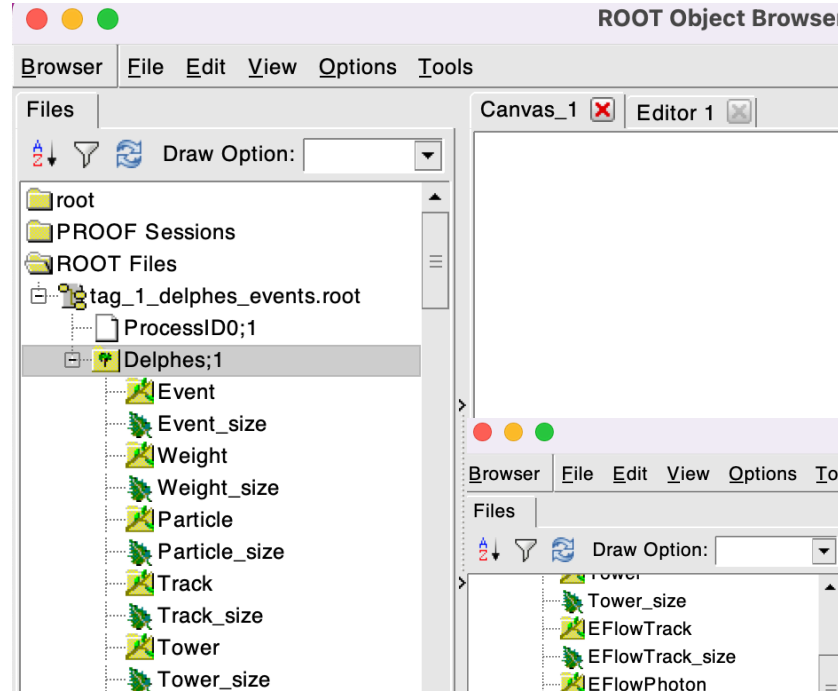


```
ilkay@Ilkays-MacBook-Pro ~ % cd MG5_aMC_v3_5_0/ilkson/Events/run_01
ilkay@Ilkays-MacBook-Pro run_01 % ls
run_01_tag_1_banner.txt      tag_1_pts.dat
run_shower.sh               tag_1_pythia8.cmd
tag_1_delphes.log           tag_1_pythia8.log
tag_1_delphes_events.root   tag_1_pythia8_events.hepmc.gz
tag_1_djrs.dat              unweighted_events.lhe.gz
ilkay@Ilkays-MacBook-Pro run_01 %
```

ROOT İLE ÇALIŞMA

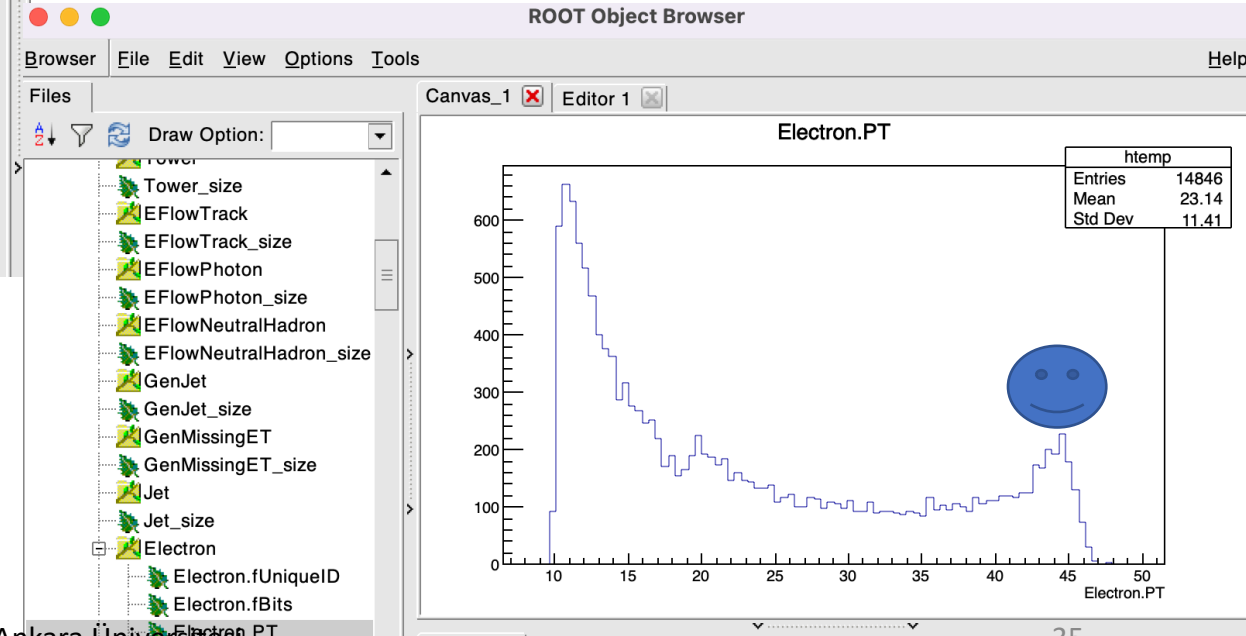
```
ilkay@Ilkays-MacBook-Pro run_01 % root -l tag_1_delphes_events.root
```

```
root [1] lBr
TBranch
TBranchCacheInfo
TBranchClones
TBranchDescriptor
TBranchElement
TBranchObject
TBranchProxy
TBranchProxyClassDescriptor
TBranchProxyDescriptor
TBranchProxyDirector
TBranchProxyHelper
TBranchRef
TBranchSTL
TBrowser
TBrowserImp
TBrowserPlugin
root [1] TBrowser i
```



- root filenin içinde delphes tree var.
- Delphes dedektör sim. programı Delphes içindeki tree'nin adına Delphes denmiş

Ayrıca ; .ls ve new Tbrowser komutları da kullanılabilir.



MadGraph KAYNAKLAR

1- <https://cp3.irmp.ucl.ac.be/projects/madgraph>

2- <https://launchpad.net/mg5amcnlo>

3-

https://indico.cern.ch/event/335370/contributions/781780/attachments/655157/900786/HPFBUO4_MadGraph_2.pdf

ROOT KAYNAKLAR

1- <http://root.cern.ch>

2- <https://root.cern/manual/>

3- <https://root-forum.cern.ch/>

4-Parçacık Fiziği Bilgisayar Uygulamaları Okulu, İstanbul Üniversitesi, İlkay TÜRK ÇAKIR, Ders Notları

https://indico.cern.ch/event/877623/contributions/3707938/attachments/1980072/3297006/session_PFHUU.pdf

PYTHIA KAYNAKLAR

- 1- T. Sjostrand et al., An Introduction to PYTHIA 8.2, arXiv:1410.3012 [hep-ph], <http://home.thep.lu.se/~torbjorn/pdfdoc/pythia8200.pdf>
- 2- T. Sjostrand, S. Mrenna and P. Skands, A Brief Introduction to Pythia 8.1, arXiv:0710.3820, web sayfası:
- 3- <http://home.thep.lu.se/~torbjorn/pythia8/pythia8100.pdf>
- 4- <http://home.thep.lu.se/~torbjorn/php8157/Welcome.php>
- 5- T. Sjostrand, S. Mrenna and P. Skands, PYTHIA 6.4 Physics and Manual, hep-ph/0603175, web sayfası:
<http://www.thep.lu.se/~torbjorn/pythia/lutp0613man2.pdf>

DELPHES KAYNAKLAR

- 1- <https://www.tcm.phy.cam.ac.uk/~mjh261/pdfs/delphes.pdf>
- 2- <https://cp3.irmp.ucl.ac.be/projects/delphes>