INSTITUTE FOR RESEARCH IN FUNDAMENTAL SCIENCES

IPM

Particle Physics Data Analysis Tools

Gholamhossein Haghighat

School of Particles and Accelerators, IPM

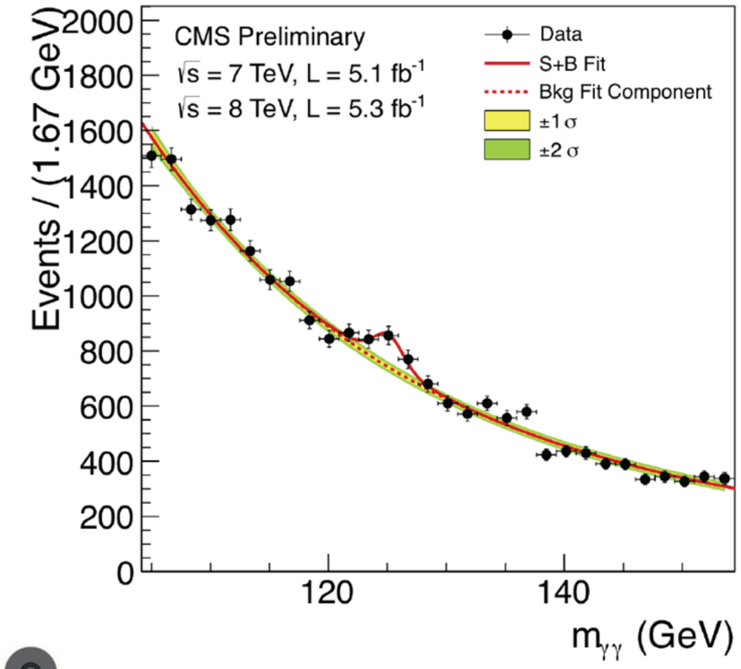
4th IPM Workshop on

Experimental Particle Physics

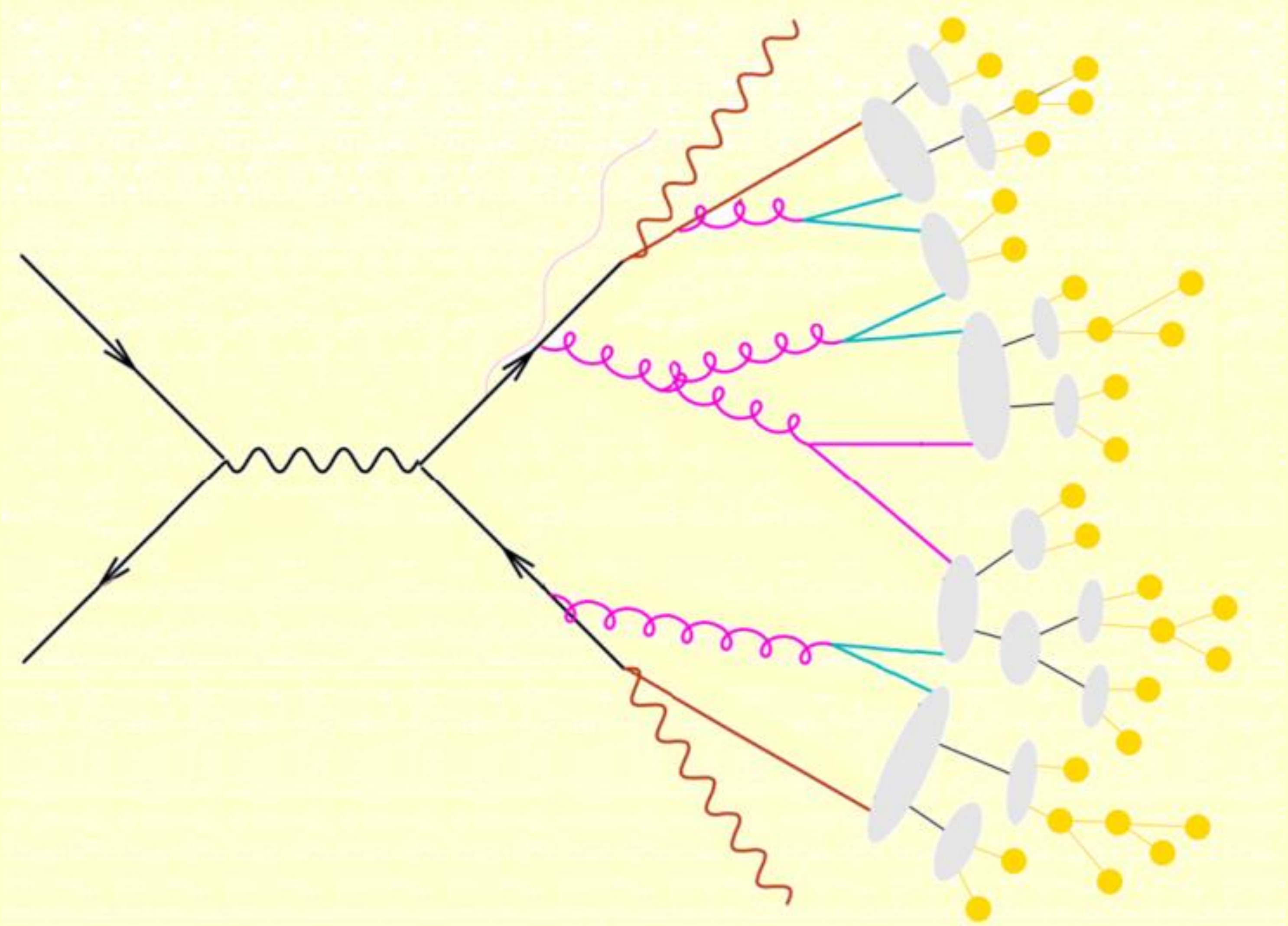
27 Dec 2023

Strategy for particle physics searches

- ***Find excess over SM background***
- ***Identify models compatible with excess***
- ***Look for predicted excesses in other channels***
- ***Determine underlying model***



Simplified LHC event

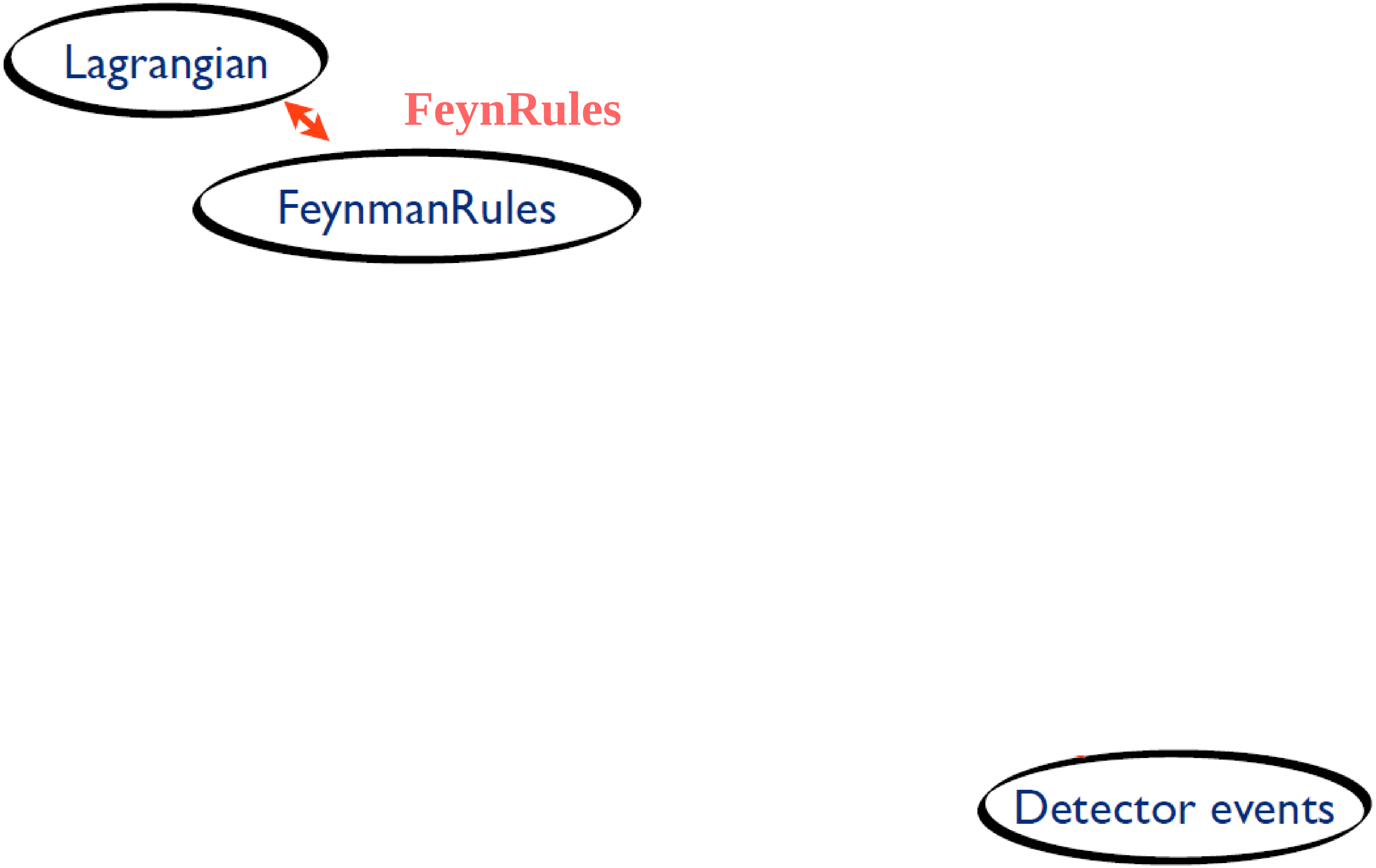


From theory to detector

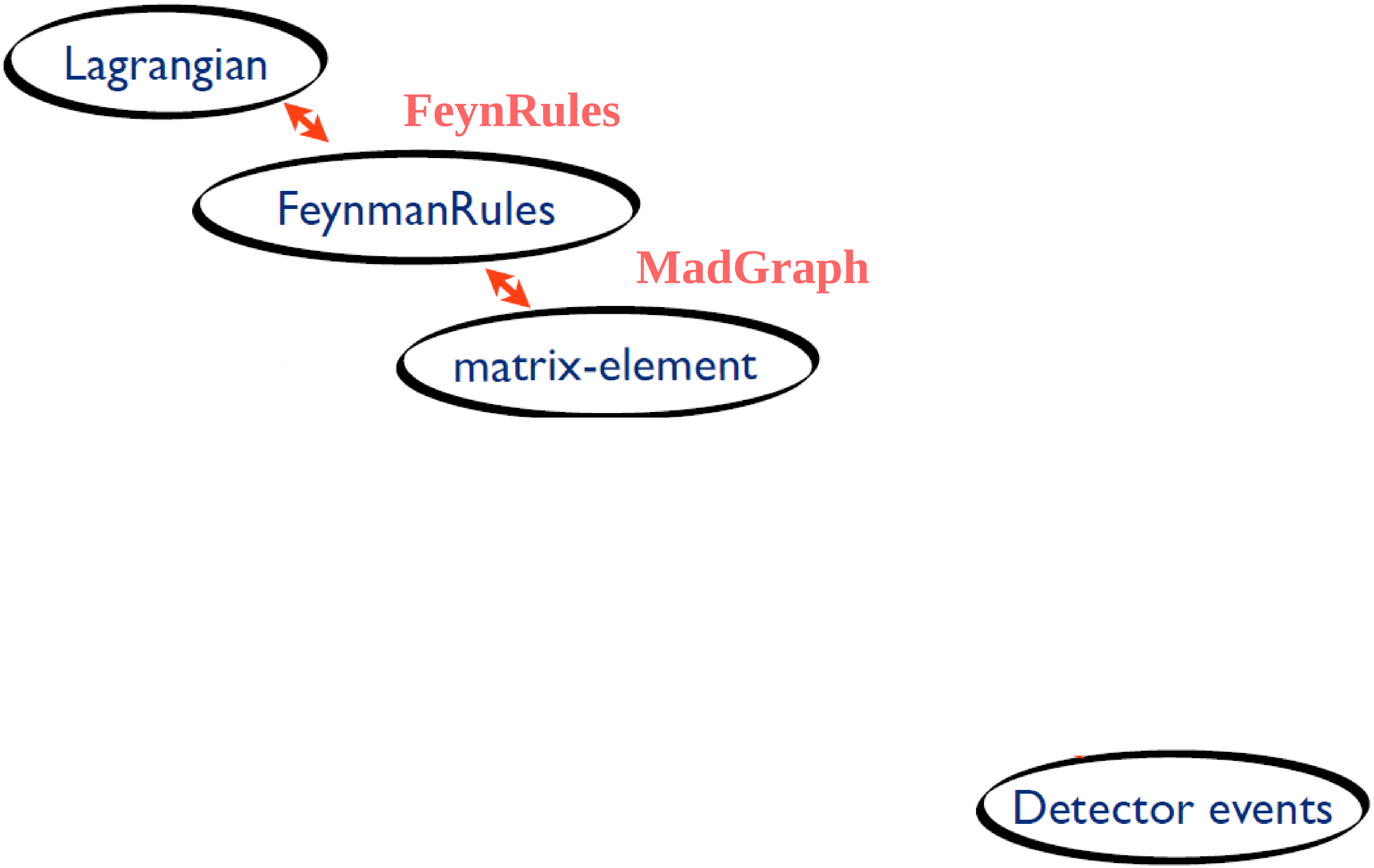
Lagrangian

Detector events

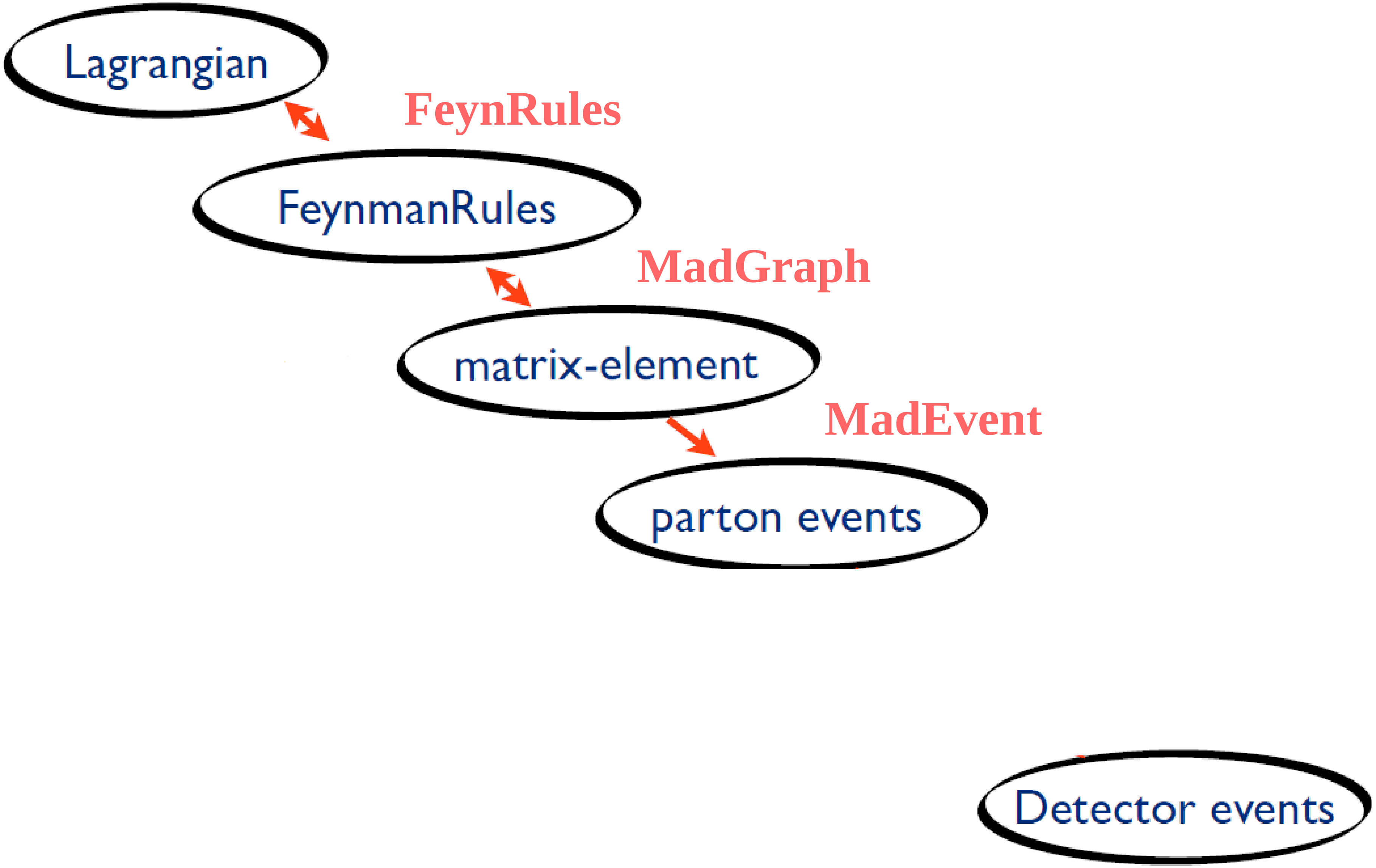
From theory to detector



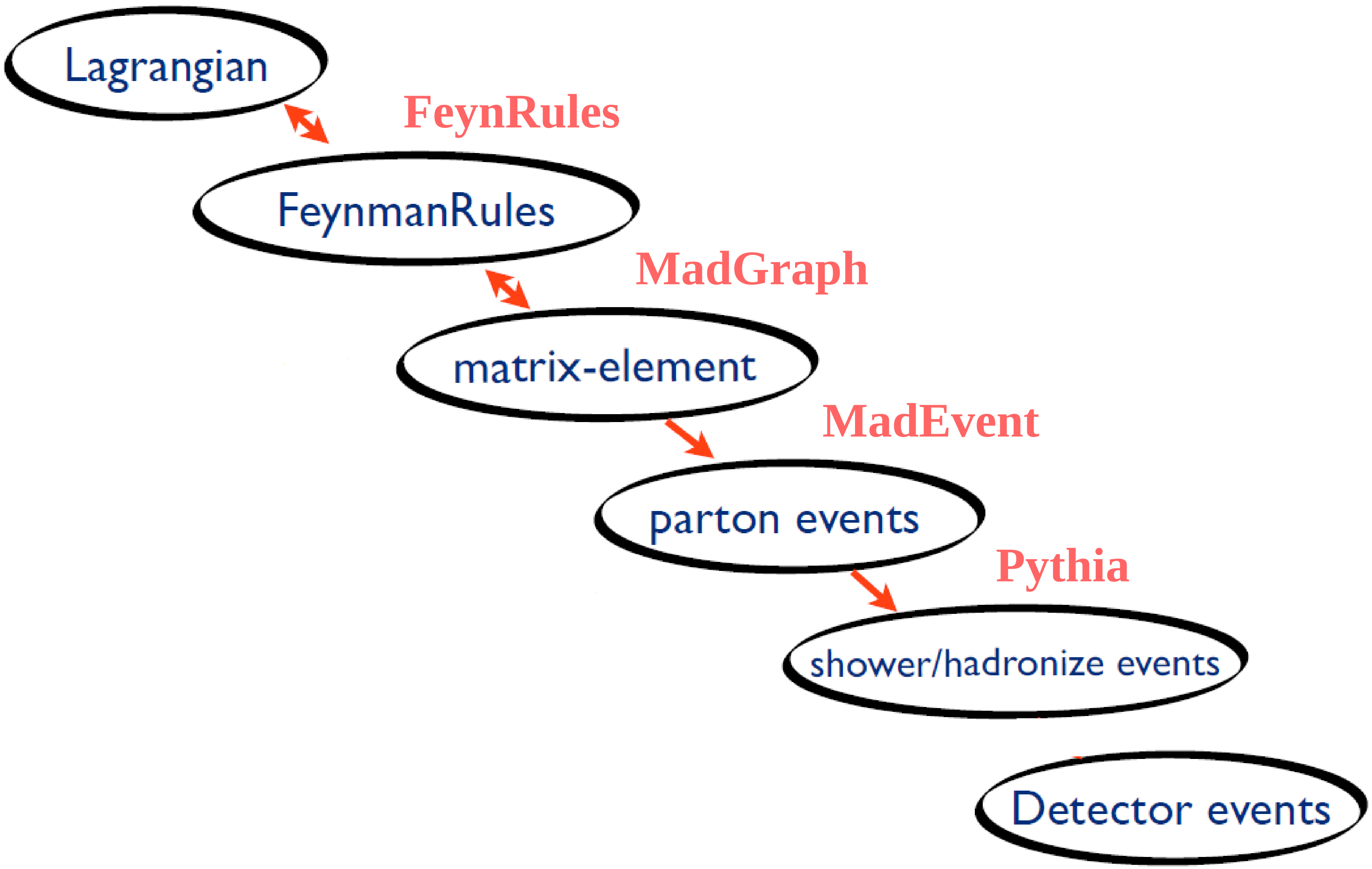
From theory to detector



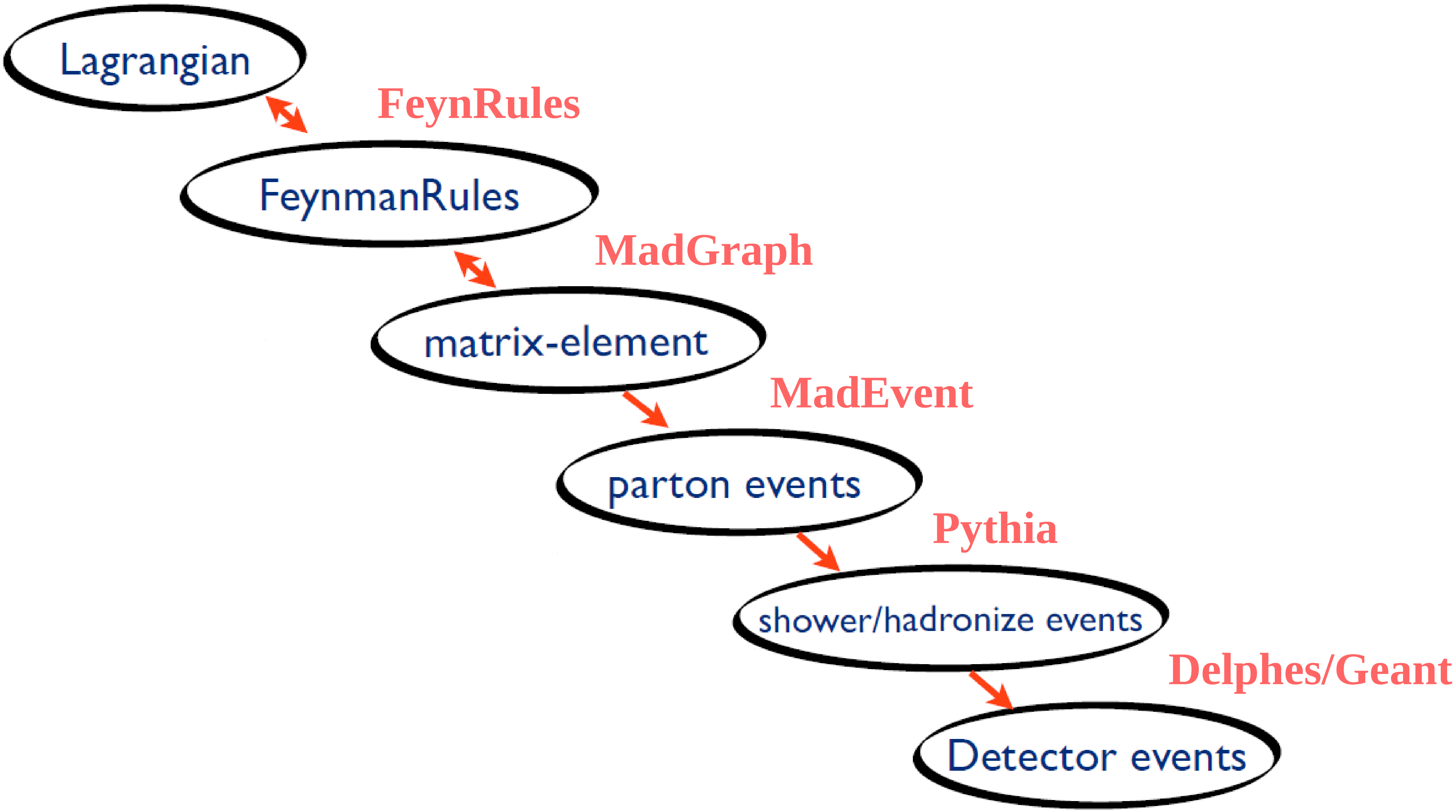
From theory to detector



From theory to detector



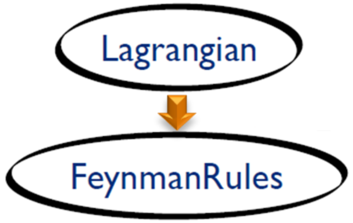
From theory to detector



FeynRules

[arXiv:1310.1921]

- ❖ ***Mathematica package to derive Feynman rules from a Lagrangian***
- ❖ ***Available at feynrules.irmp.ucl.ac.be***



Gauge symmetries	Particles	Parameters	Lagrangian

```
M$GaugeGroups = {
  U1Y  == {
    Abelian              -> True
    C
    G
    C
  } [
```

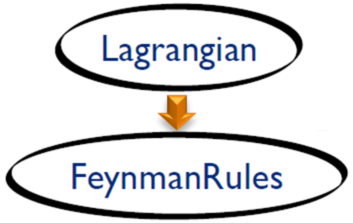
```
SU2L == {
  Abelian               -> False,
  CouplingConstant      -> gw,
  GaugeBoson            -> Wi,
  StructureConstant     -> Eps,
  Representations       -> {Ta,SU2D},
  Definitions           -> {Ta[a_
                              FSU2L
} [...]
```

```
SU3C == {
  Abelian               -> False,
  CouplingConstant      -> gs,
  GaugeBoson            -> G,
  StructureConstant     -> f,
  Representations       -> {T,Colour},
  SymmetricTensor       -> dSUN
} [...]
```

FeynRules

[arXiv:1310.1921]

- ***Mathematica package to derive Feynman rules from a Lagrangian***
- ***Available at feynrules.irmp.ucl.ac.be***



Gauge symmetries	Particles	Parameters	Lagrangian

```
M$ClassesDescription = {
  V[1] == {
    ClassName        -> A,
    SelfConju
    Mass
    Width
    ParticleN
    PDG
    Propagato
    Propagato
    Propagato
    FullName
  },[...]
```

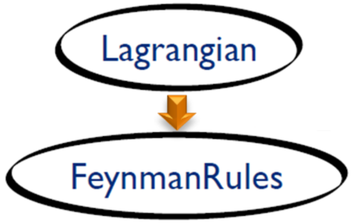
```
V[4] == {
  Clas
  Self
  Indi
  Mass
  Widt
  Part
  PDG
  Prop
  Prop
  Prop
  Full
},[...
```

```
F[1] == {
  ClassName         -> vl,
  ClassMembers      -> {ve,vm,vt},
  Indices           -> {Index[Generation]},
  FlavorIndex       -> Generation,
  SelfConjugate     -> False,
  Mass              -> 0,
  Width             -> 0,
  QuantumNumbers    -> {LeptonNumber -> 1},
  PropagatorLabel   -> {"v", "ve", "vm", "vt"} ,
  PropagatorType    -> S,
  PropagatorArrow   -> Forward,
  PDG               -> {12,14,16},
  ParticleName      -> {"ve","vm","vt"},
  AntiParticleName  -> {"ve~","vm~","vt~"},
  FullName          -> {"Electron-neutrino", "Mu-neutrino",
                        "Tau-neutrino"}
},[...]
```

FeynRules

[arXiv:1310.1921]

- ***Mathematica package to derive Feynman rules from a Lagrangian***
- ***Available at feynrules.irmp.ucl.ac.be***



Gauge symmetries	Particles	**Parameters**	Lagrangian

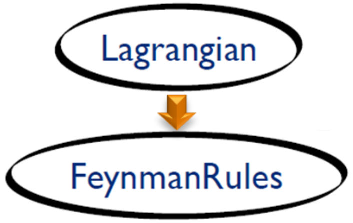
```
M$Parameters = {
  aS == {
    ParameterType    -> External,
    BlockName        -> SMINPUTS,
    OrderBlock       -> 3,
    Value            -> 0.1184,
    InteractionOrder -> {QCD,2},
    TeX              -> Subscript[\[Alpha],s],
    Description      -> "Strong coupling constant at the Z pole"
  },
  gs == {
    ParameterType    -> Internal,
    Value            -> Sqrt[4 Pi aS],
    InteractionOrder -> {QCD,1},
    TeX              -> Subscript[g,s],
    ParameterName    -> G,
    Description      -> "Strong coupling constant at the Z pole"
  },[...]
```

```
CKM == {
  ParameterType -> Internal,
  Indices       -> {Index[Generation], Index[Generation]},
  Unitary       -> True,
  Value         -> {CKM[1,1] -> Cos[cabi],  CKM[1,2] -> Sin[cabi], CKM[1,3] -> 0,
                    CKM[2,1] -> -Sin[cabi], CKM[2,2] -> Cos[cabi], CKM[2,3] -> 0,
                    CKM[3,1] -> 0,          CKM[3,2] -> 0,         CKM[3,3] -> 1},
  TeX         -> Superscript[V,CKM],
  Description -> "CKM-Matrix"}
```

FeynRules

[arXiv:1310.1921]

- ***Mathematica package to derive Feynman rules from a Lagrangian***
- ***Available at feynrules.irmp.ucl.ac.be***



Gauge symmetries | Particles | Parameters | **Lagrangian**

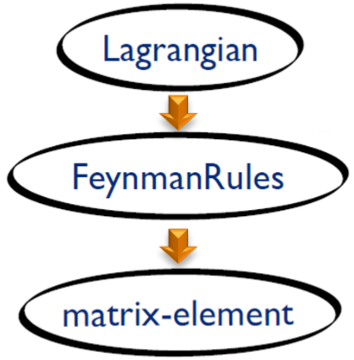
```
LFermions := Block[{mu},
  ExpandIndices[I*(
    QLbar.Ga[mu].DC[QL, mu] + LLbar.Ga[mu].DC[LL, mu] +
    uRbar.Ga[mu].DC[uR, mu] + dRbar.Ga[mu].DC[dR, mu] +
    lRbar.Ga[mu].DC[lR, mu]), FlavorExpand->{SU2W,SU2D}]
```

```
WriteUFO[LSM] →
```

UFO becoming the standard

MadGraph5

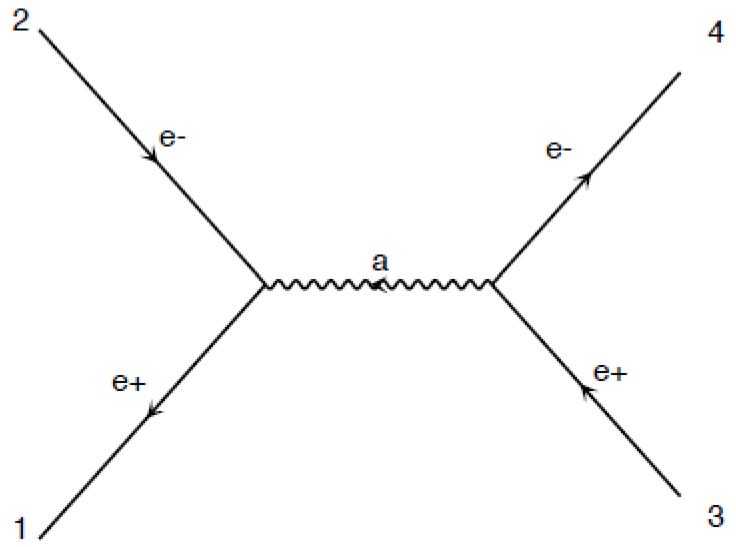
[arXiv:1405.0301]



Computing amplitudes with HELAS

(HELicity Amplitude Subroutine)

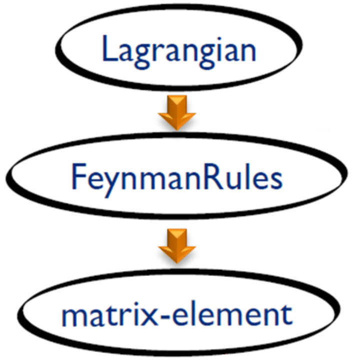
Evaluate $\mathcal{M}$ *for fixed helicity of external particles*



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

MadGraph5

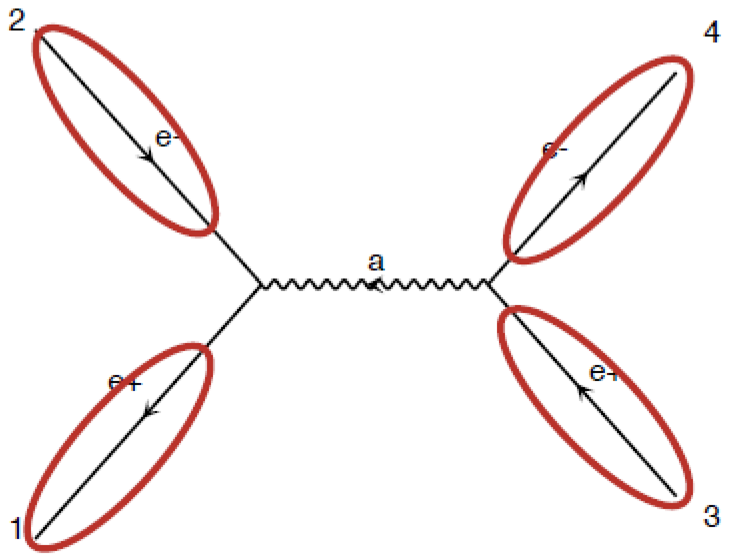
[arXiv:1405.0301]



Computing amplitudes with HELAS

(HELicity Amplitude Subroutine)

Evaluate $\mathcal{M}$ for fixed helicity of external particles

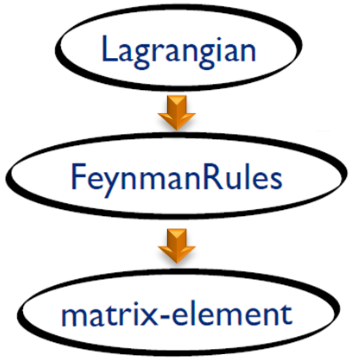


$$\mathcal{M} = \bar{u}\gamma^{\mu}v P_{\mu\nu}\bar{u}\gamma^{\nu}v$$

Particles

MadGraph5

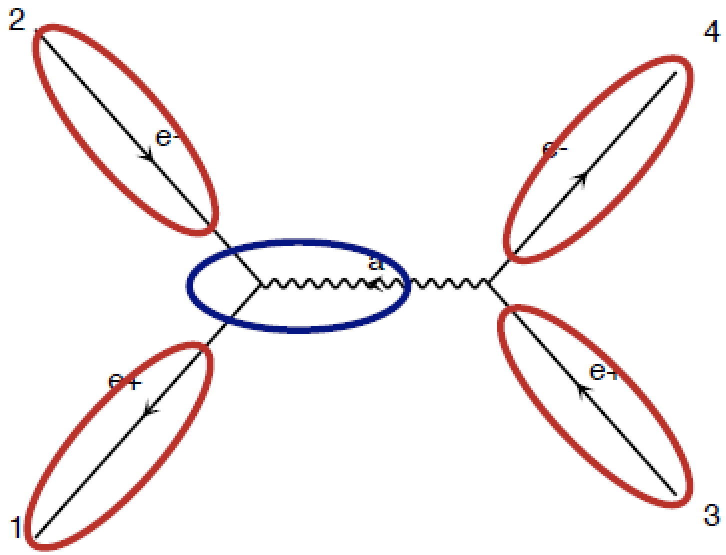
[arXiv:1405.0301]



*Computing amplitudes with **HELAS***

*(**HEL**icity **A**mplitude **S**ubroutine)*

Evaluate $\mathcal{M}$ for fixed helicity of external particles



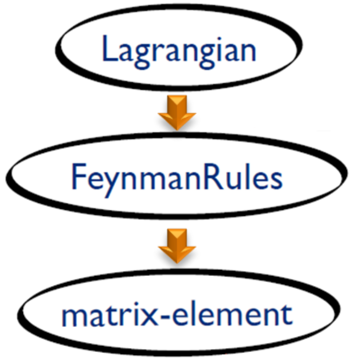
$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Particles

Propagators

MadGraph5

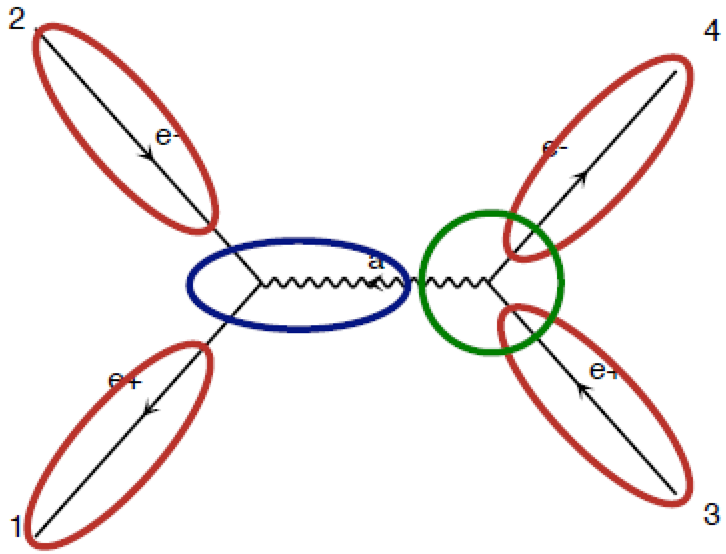
[arXiv:1405.0301]



*Computing amplitudes with **HELAS***

*(**HEL**icity **A**mplitude **S**ubroutine)*

Evaluate $\mathcal{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Particles

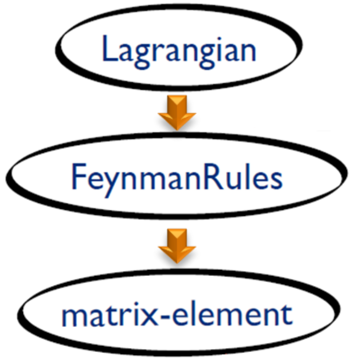
Propagators

Amplitude

- *Helicity amplitude routines needed for the Standard Model, MSSM, …, in hand-written library*
- *Any new Lorentz structure needs addition by hand ➡ restriction on types of models that could be implemented in MadGraph*

MadGraph5

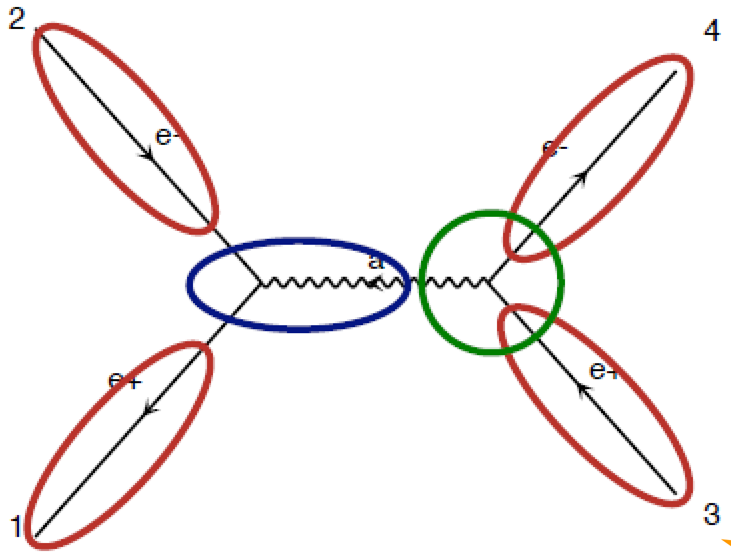
[arXiv:1405.0301]



*Computing amplitudes with **HELAS***

*(**HEL**icity **A**mplitude **S**ubroutine)*

Evaluate $\mathcal{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Particles

Propagators

Amplitude

ALOHA

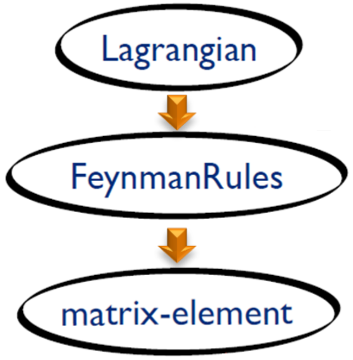
*(**A**utomatic **L**ibraries **O**f **H**elicity **A**mplitudes)*

*translates a **UFO** Lorentz structure into pseudo-**HELAS** subroutine*

MadGraph5

[arXiv:1405.0301]

Weighting experimental events with MadWeight



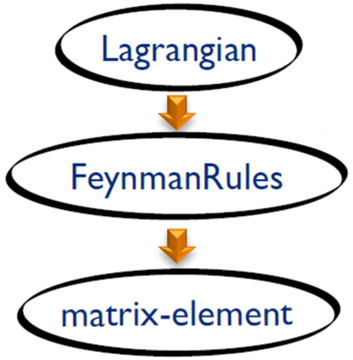
$$P(\boldsymbol{x}, \alpha) =$$

Probability of observing ***x***
predicted by the model ***α***
x: *experimental measurements*

MadGraph5

[arXiv:1405.0301]

Weighting experimental events with MadWeight



$$P(\boldsymbol{x}, \alpha) = \qquad |M_\alpha|^2(\boldsymbol{x})$$

$|M_\alpha|^2(\boldsymbol{x})$ → *Squared matrix element*

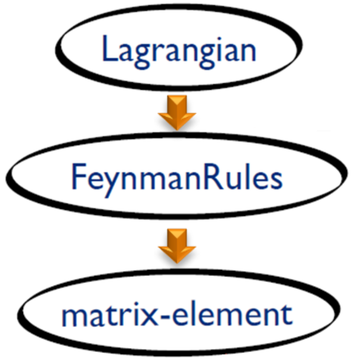
$P(\boldsymbol{x}, \alpha)$ → *Probability of observing **x** predicted by the model **α***

***x**: experimental measurements*

MadGraph5

[arXiv:1405.0301]

Weighting experimental events with MadWeight



$$P(\boldsymbol{x}, \alpha) = \qquad |M_\alpha|^2(\boldsymbol{y}) W(\boldsymbol{x}, \boldsymbol{y})$$

Probability of observing ***x***
predicted by the model ***α***
x: *experimental measurements*

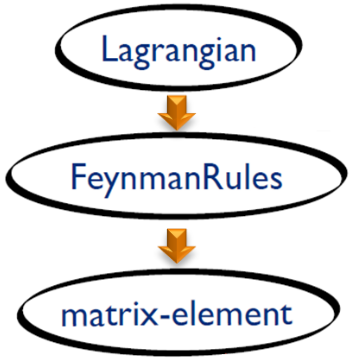
Squared matrix element

Resolution function
y*: partonic momenta*
(experimental extraction)

MadGraph5

[arXiv:1405.0301]

Weighting experimental events with MadWeight



$$P(\boldsymbol{x},\alpha) = \frac{1}{\sigma}\int d\phi(\boldsymbol{y}) \qquad |M_\alpha|^2(\boldsymbol{y}) W(\boldsymbol{x},\boldsymbol{y})$$

Partonic phase-space measure

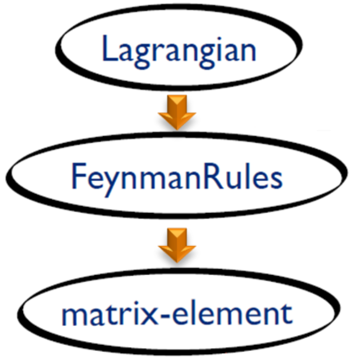
Squared matrix element

Probability of observing ***x***
predicted by the model ***α***
x: *experimental measurements*

Resolution function
y*: partonic momenta*
(experimental extraction)

MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

$$P(\boldsymbol{x},\alpha) = \frac{1}{\sigma}\int d\phi(\boldsymbol{y})dw_1 dw_2 f_1(w_1)f_2(w_2)|M_\alpha|^2(\boldsymbol{y})W(\boldsymbol{x},\boldsymbol{y})$$

Partonic phase-space measure

Squared matrix element

Probability of observing ***x***
predicted by the model ***α***
x: *experimental measurements*

Parton Distribution Functions

Resolution function
y: *partonic momenta*
(experimental extraction)

MadGraph5

[arXiv:1405.0301]

Lagrangian

FeynmanRules

matrix-element

Weighting experimental events with MadWeight

$$P(\boldsymbol{x}, \alpha) = \frac{1}{\sigma} \int d\phi(\boldsymbol{y}) dw_1 dw_2 f_1(w_1) f_2(w_2) |M_\alpha|^2(\boldsymbol{y}) W(\boldsymbol{x}, \boldsymbol{y})$$

Partonic phase-space measure

Squared matrix element

*Probability of observing **x***
*predicted by the model **α***
***x:** experimental measurements*

Parton Distribution Functions

Resolution function
***y**: partonic momenta*
(experimental extraction)

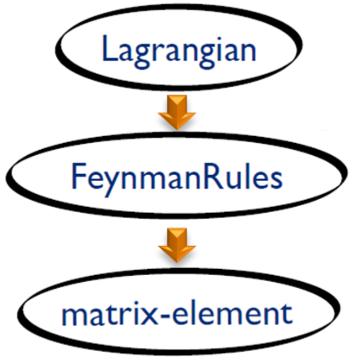
Peaks

$$W(\boldsymbol{x}, \boldsymbol{y}) \approx \prod_{i=E,\phi,\eta} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x_i - y_i)^2}{2\sigma_i^2}}$$

Propagators: $\frac{1}{|q^2 - M^2 + iM\Gamma|^2}$

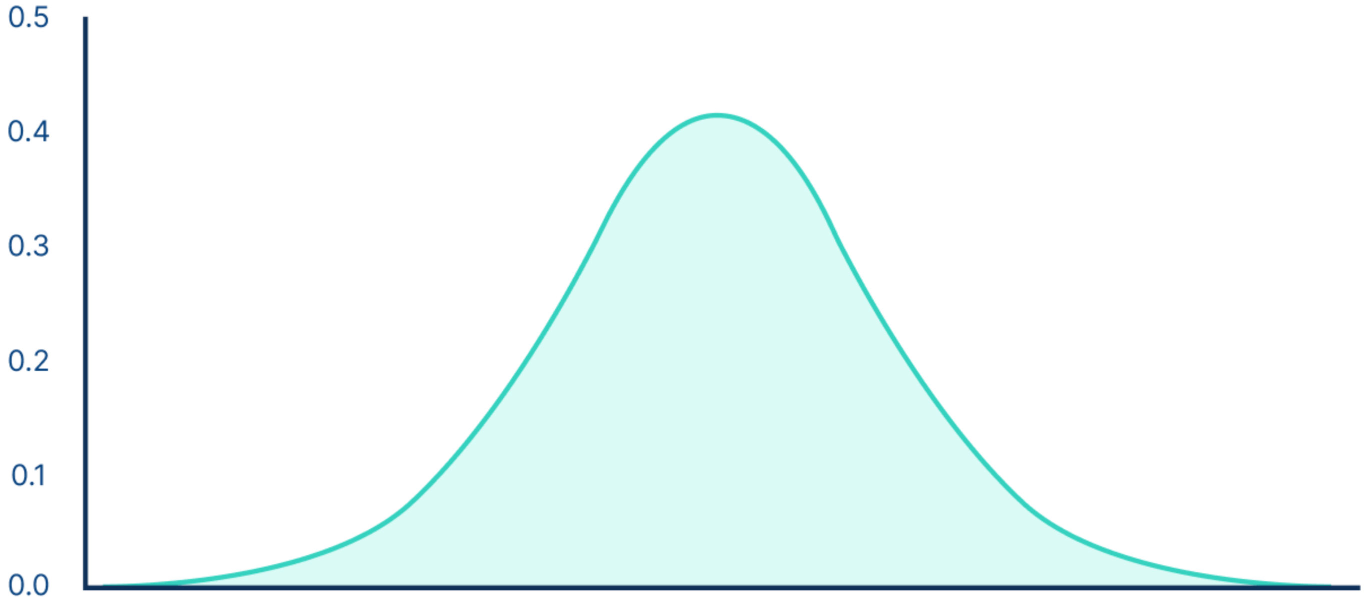
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

Monte-Carlo integration

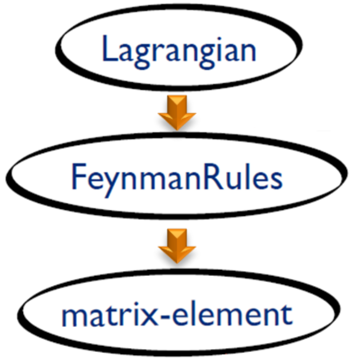


Monte Carlo (MC) method:

a method to obtain deterministic results from random values

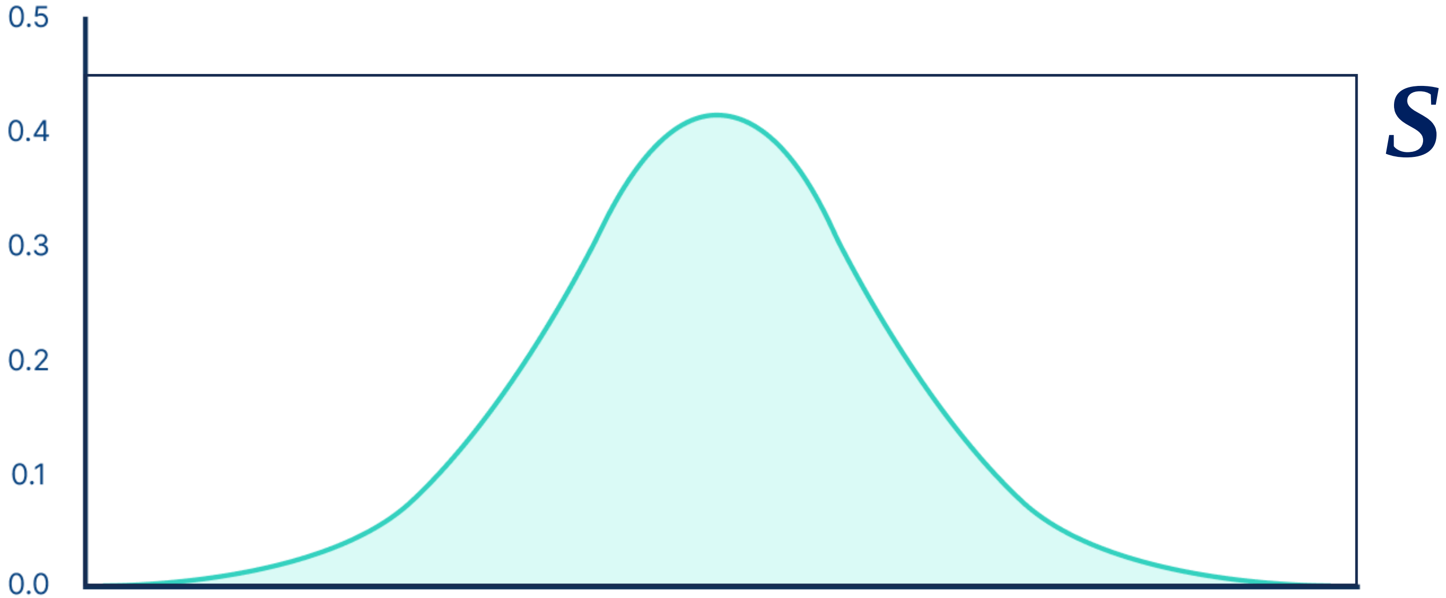
MadGraph5

[arXiv:1405.0301]



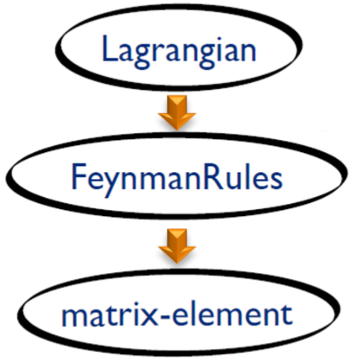
Weighting experimental events with MadWeight

Monte-Carlo integration



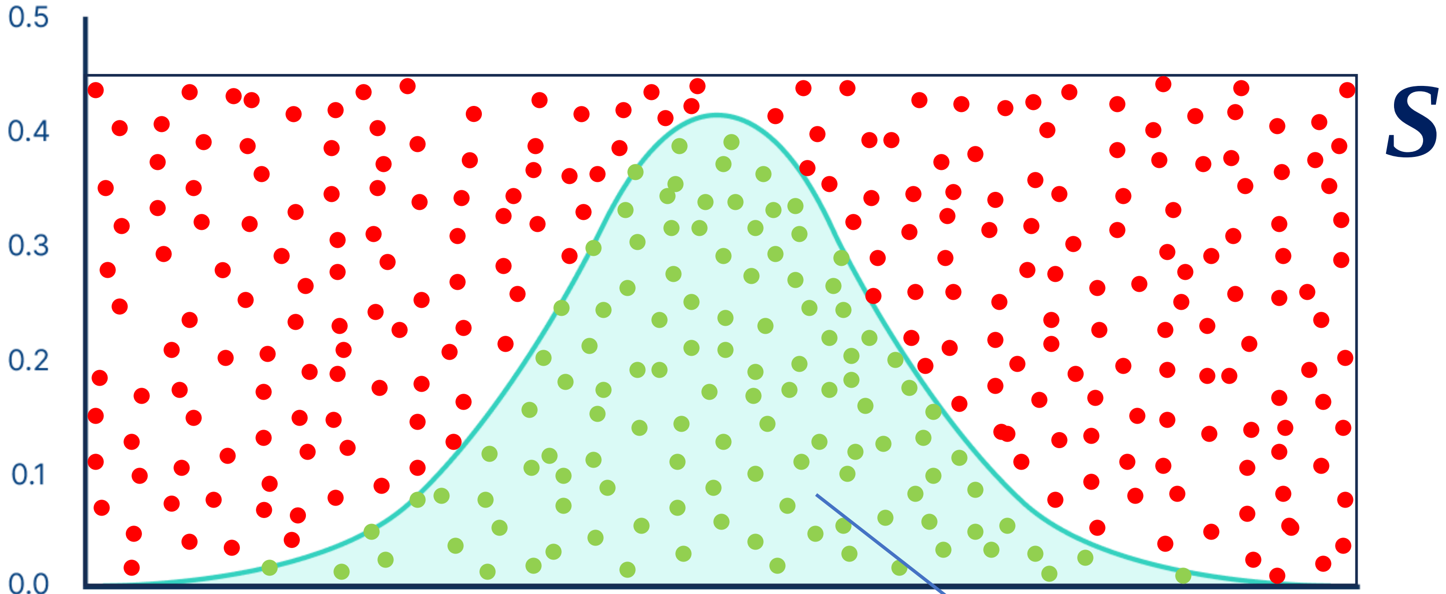
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

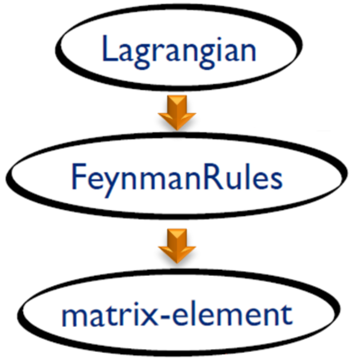
Monte-Carlo integration



$$\frac{N_{accepte}}{N_{total}} \; S$$

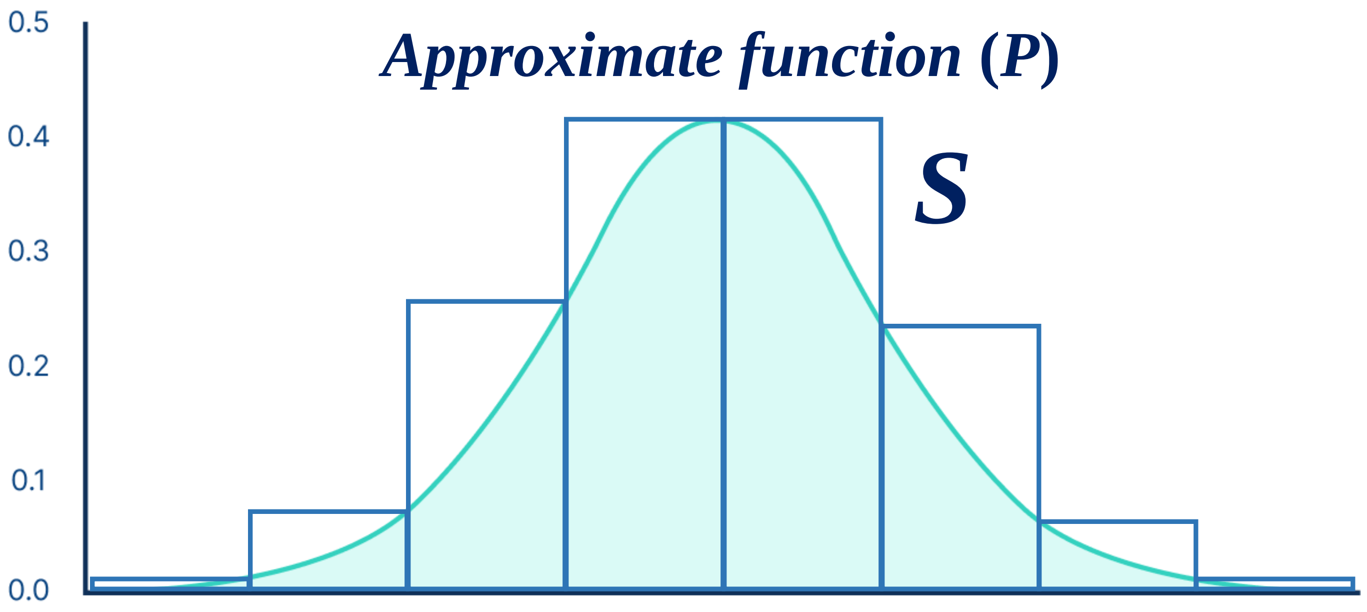
MadGraph5

[arXiv:1405.0301]



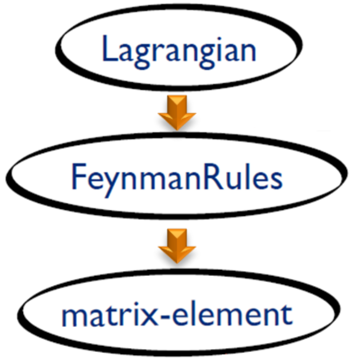
Weighting experimental events with MadWeight

Importance sampling



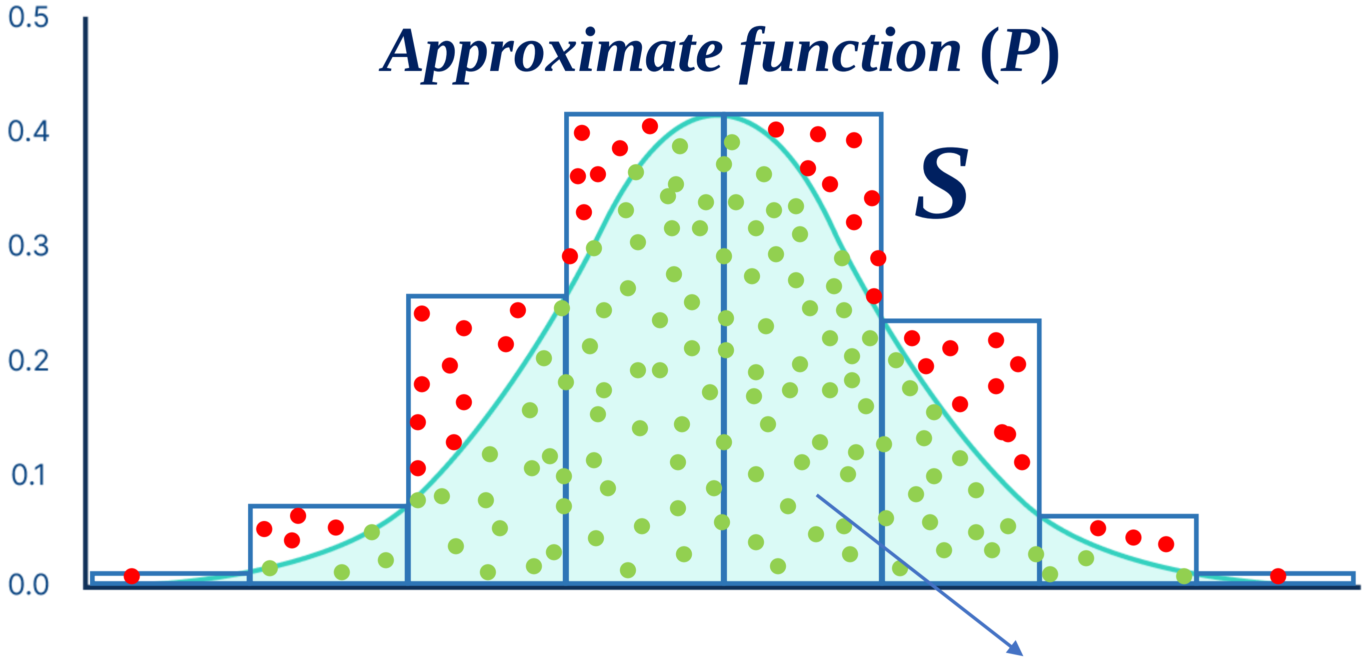
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

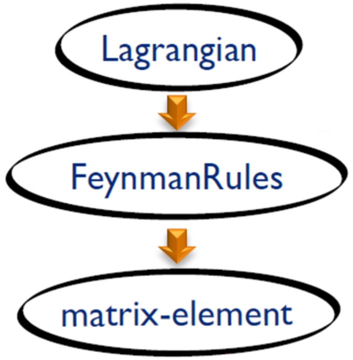
Importance sampling



$$\frac{N_{accepte}}{N_{total}} \, S$$

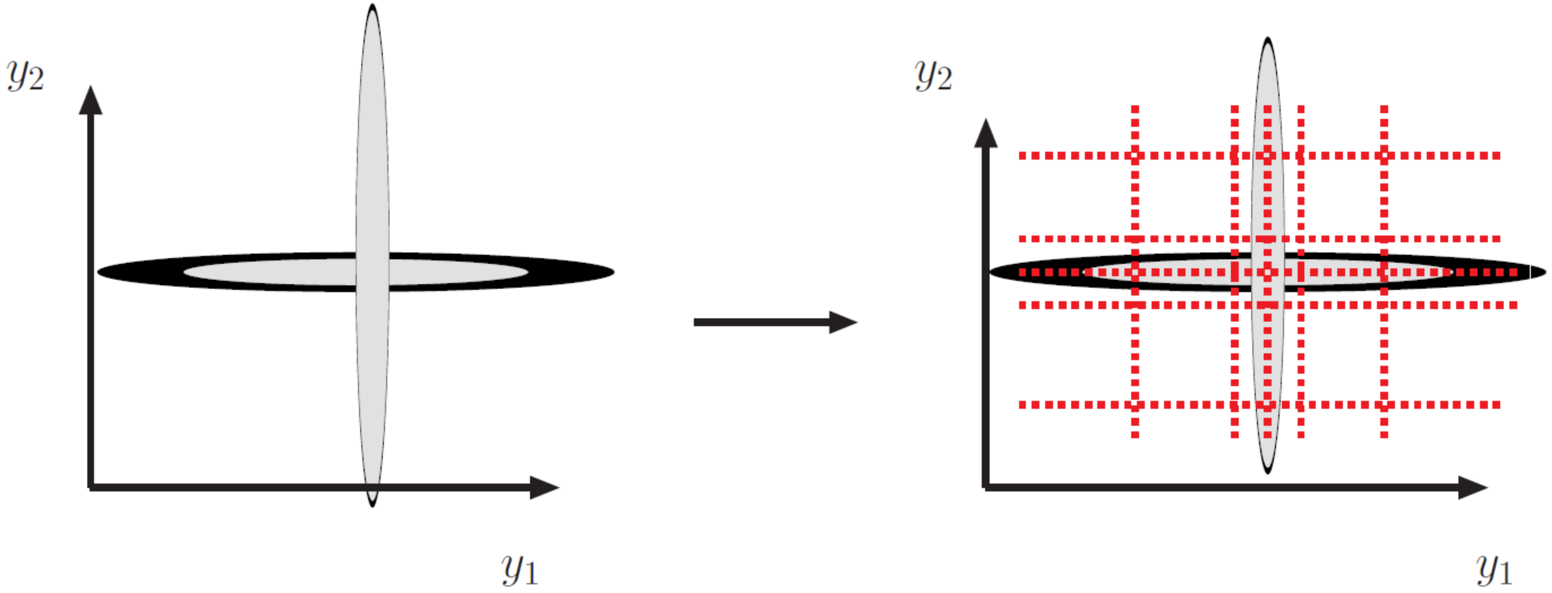
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

VEGAS (Adaptative Monte-Carlo)

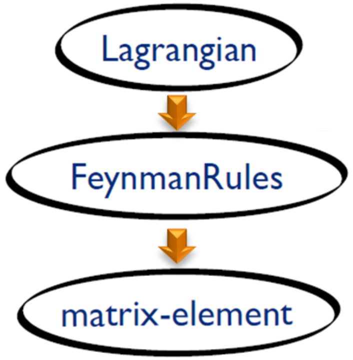


Any peak is aligned along a single direction of the P-S parameterization

Integration is very efficient

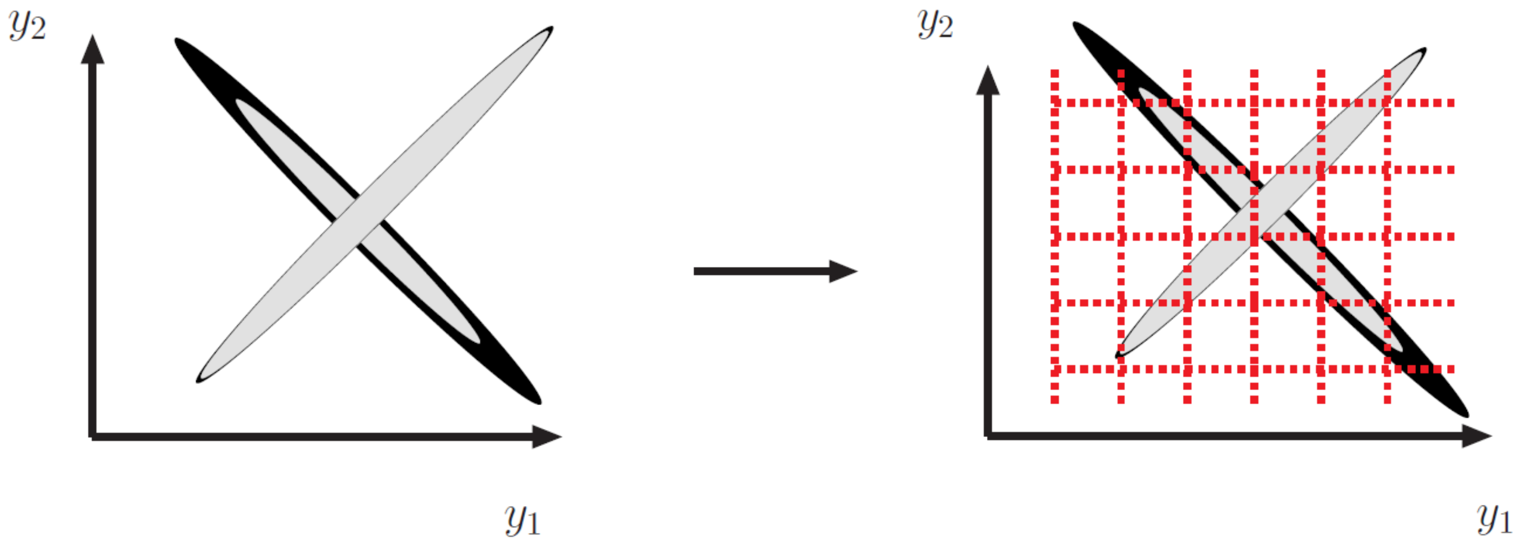
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

VEGAS (Adaptative Monte-Carlo)

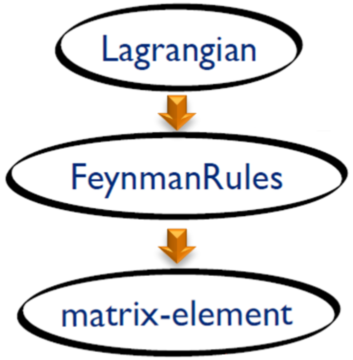


*Some peaks are **not aligned** along a single direction of the P-S parameterization*

Integration converges slowly

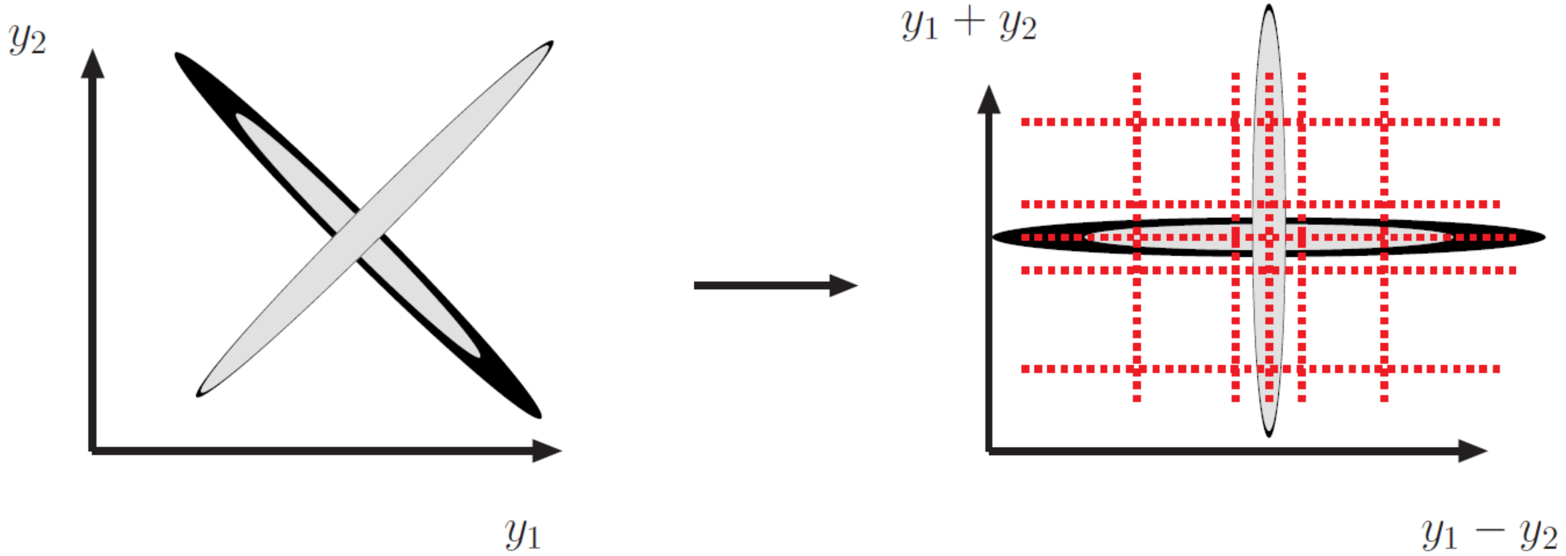
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

VEGAS (Adaptative Monte-Carlo)



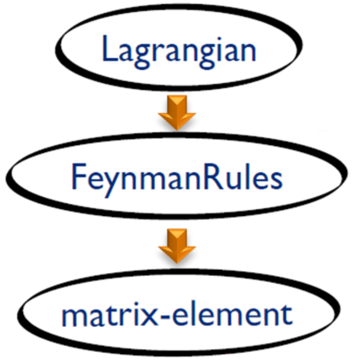
Some peaks are not aligned along a single direction of the P-S parameterization

Integration converges slowly

Solution: perform a change of variables

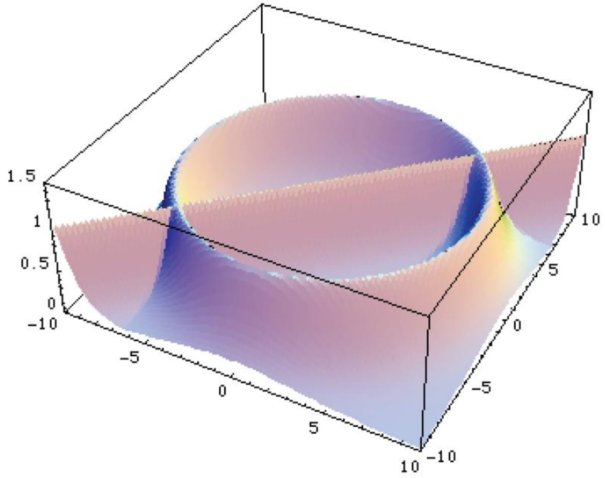
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

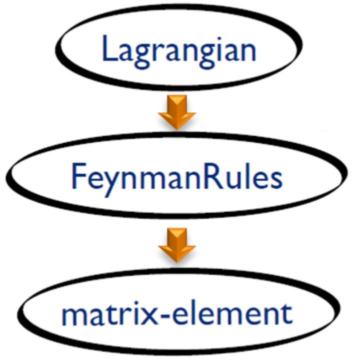
Multi-channel Monte-Carlo



What if there is no transformation that aligns all integrand peaks to the chosen axes?

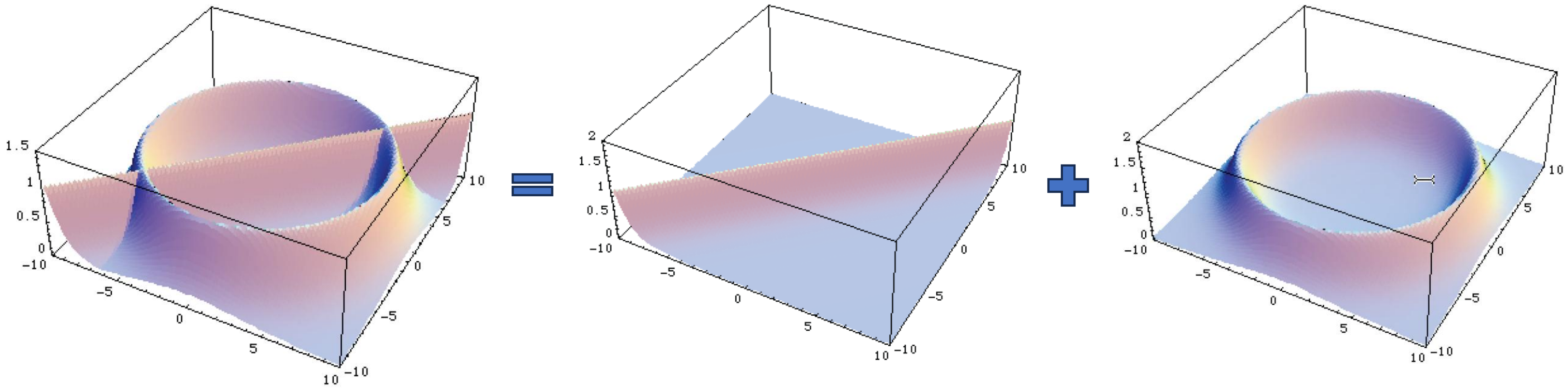
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

Multi-channel Monte-Carlo



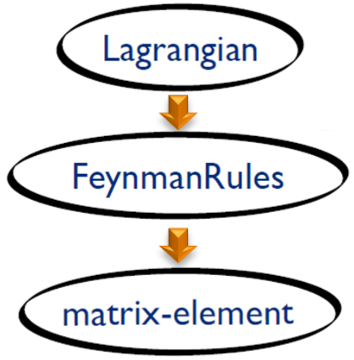
What if there is no transformation that aligns all integrand peaks to the chosen axes?

Solution: use different transformations (channels)

$$p(x) = \sum_{i=1}^{n} \alpha_i p_i(x) \qquad \sum_{i=1}^{n} \alpha_i = 1$$

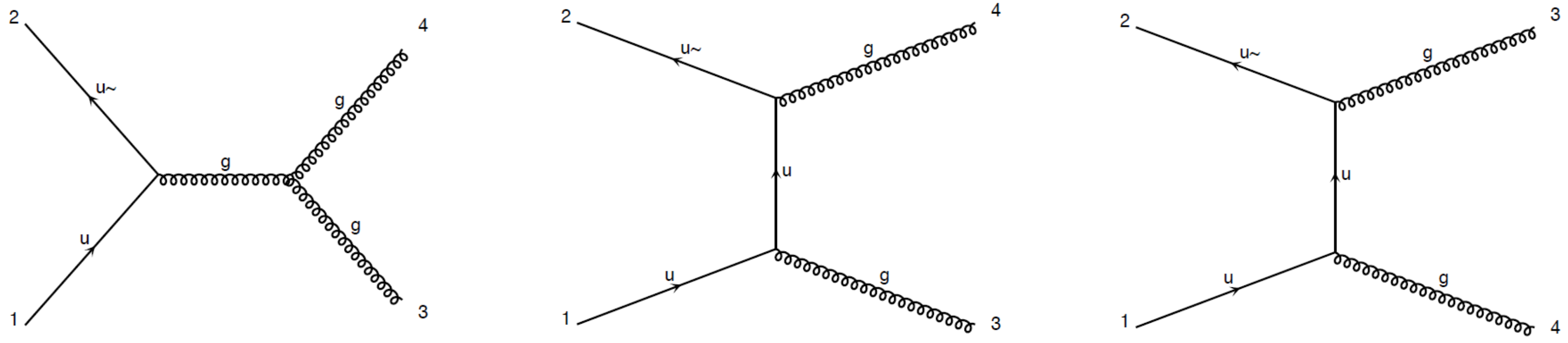
MadGraph5

[arXiv:1405.0301]



Weighting experimental events with MadWeight

Example



Three very different pole structures contributing to the same matrix element

$$p(x) = \sum_{i=1}^{n} \alpha_i p_i(x) \qquad \sum_{i=1}^{n} \alpha_i = 1$$

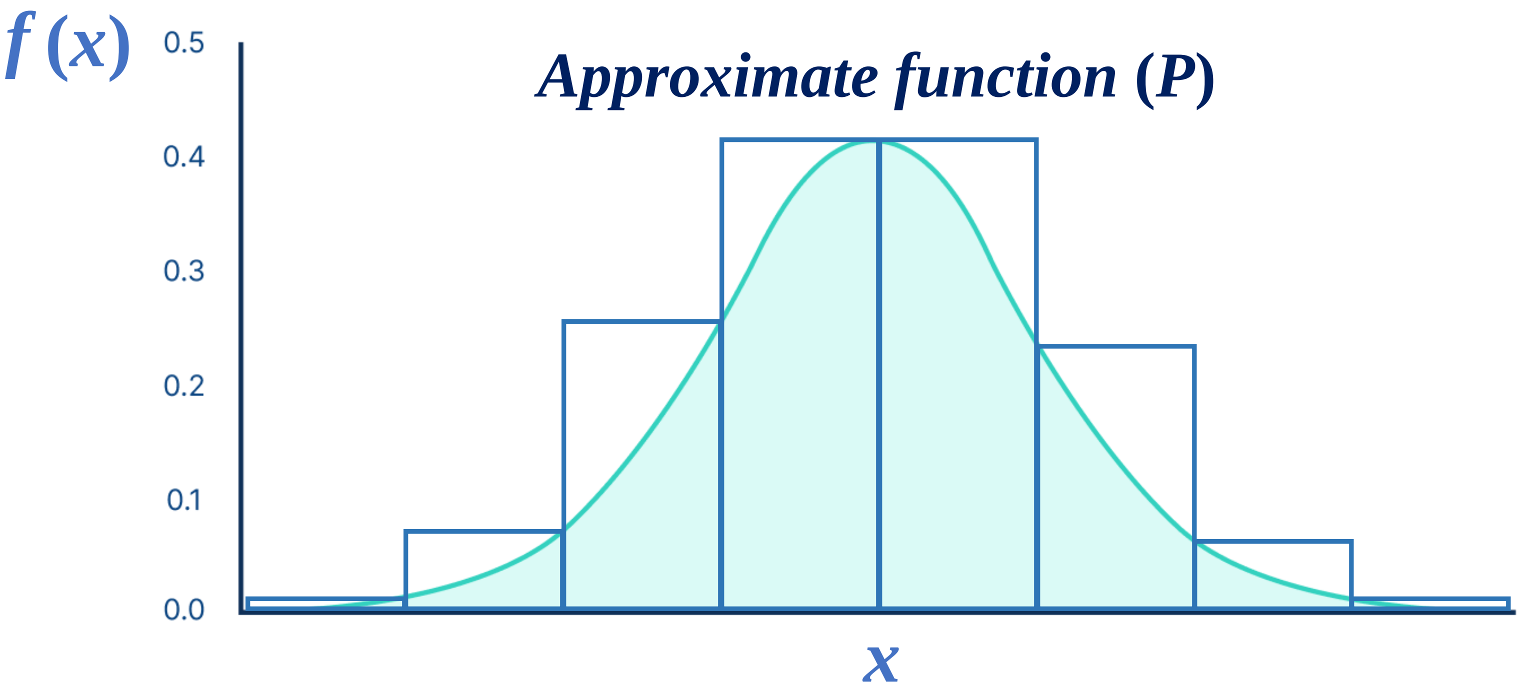
$$\int |M_{tot}|^2 = \int \frac{\sum_i |M_i|^2}{\sum_j |M_j|^2} |M_{tot}|^2 = \sum_i \int \frac{|M_i|^2}{\sum_j |M_j|^2} |M_{tot}|^2$$

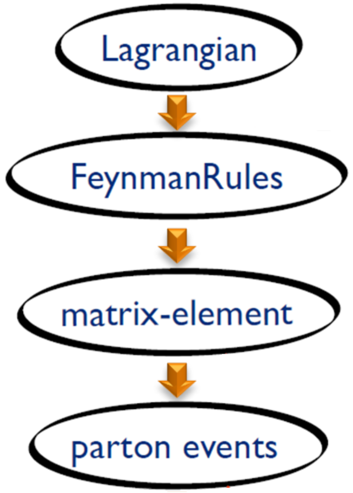
$$\frac{|M_{tot}|^2}{\sum_j |M_j|^2} \approx 1$$

MadGraph5

[arXiv:1405.0301]

Generating events with MadEvent

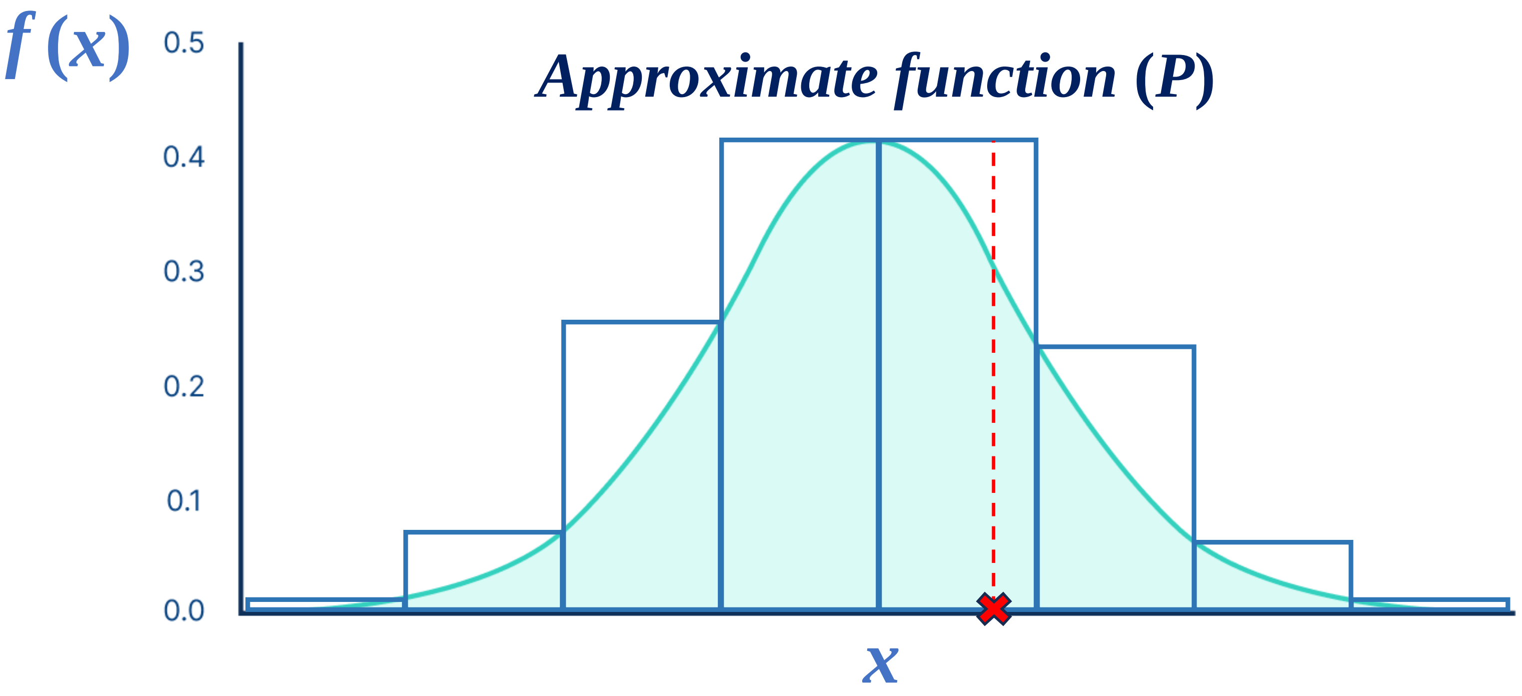




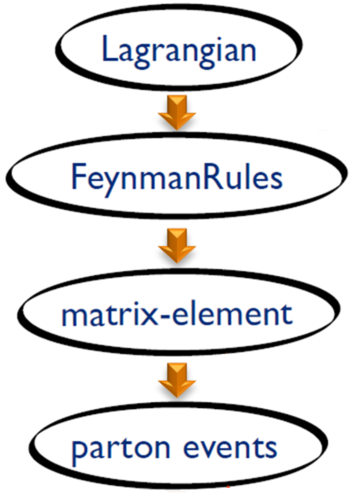
MadGraph5

[arXiv:1405.0301]

Generating events with MadEvent

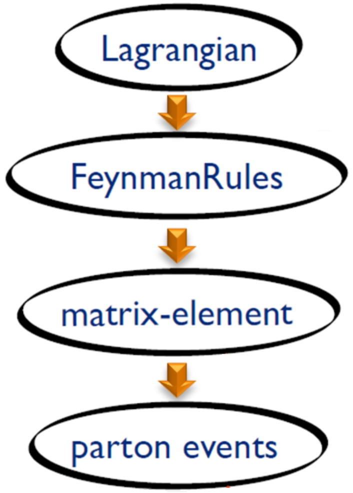


1. pick x distributed as p(x)

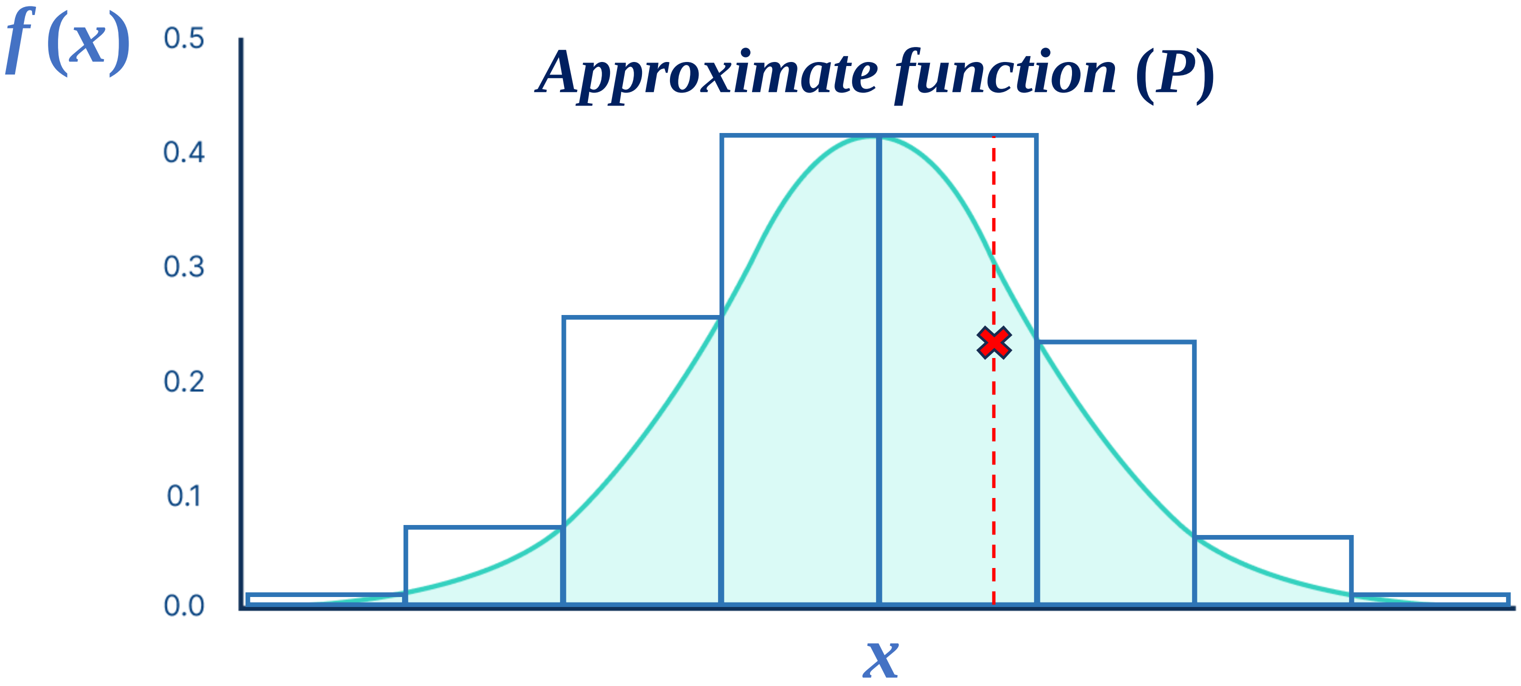


MadGraph5

[arXiv:1405.0301]



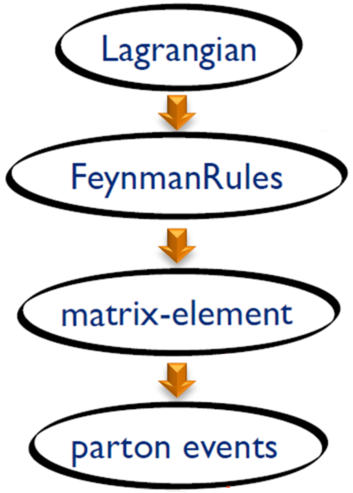
Generating events with MadEvent



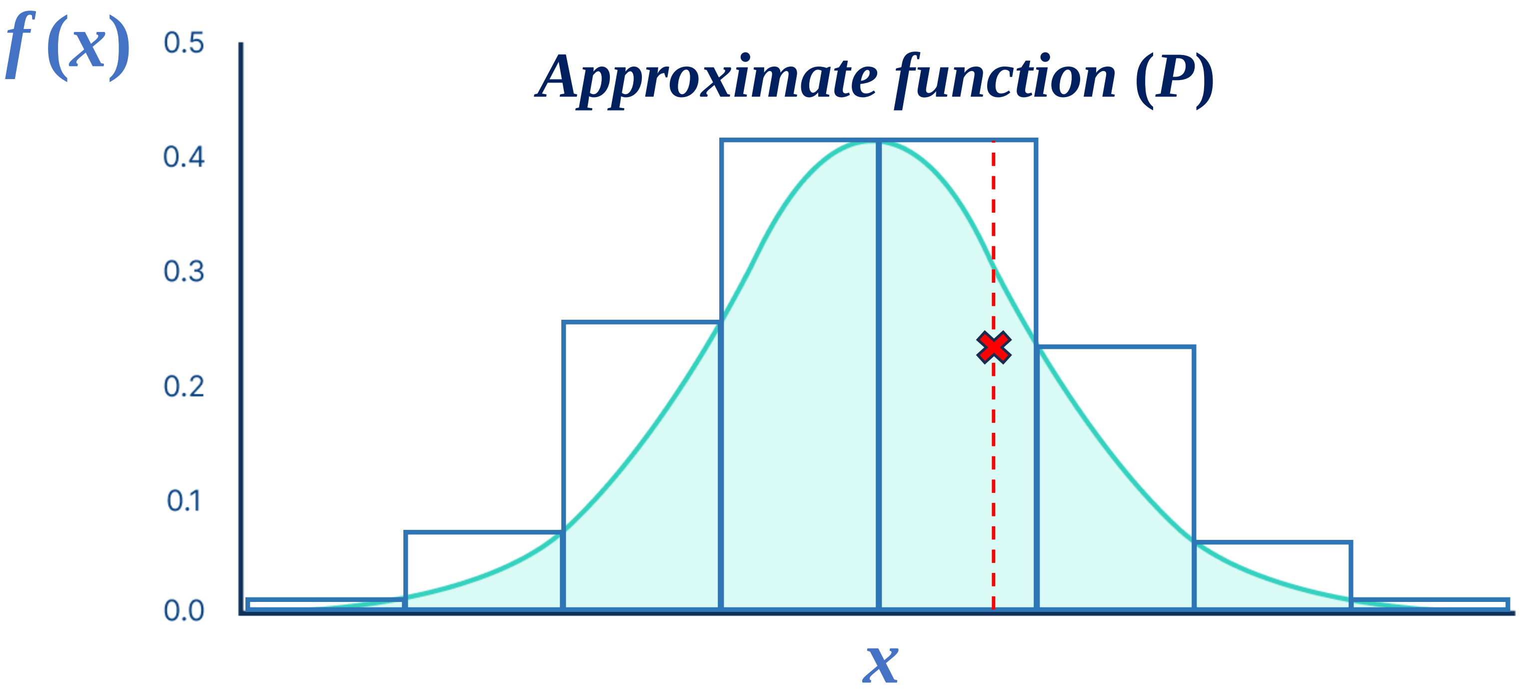
1. pick x distributed as p(x)
2. pick 0<y<p(x)

MadGraph5

[arXiv:1405.0301]



Generating events with MadEvent



1. pick x distributed as p(x)

2. pick 0<y<p(x)

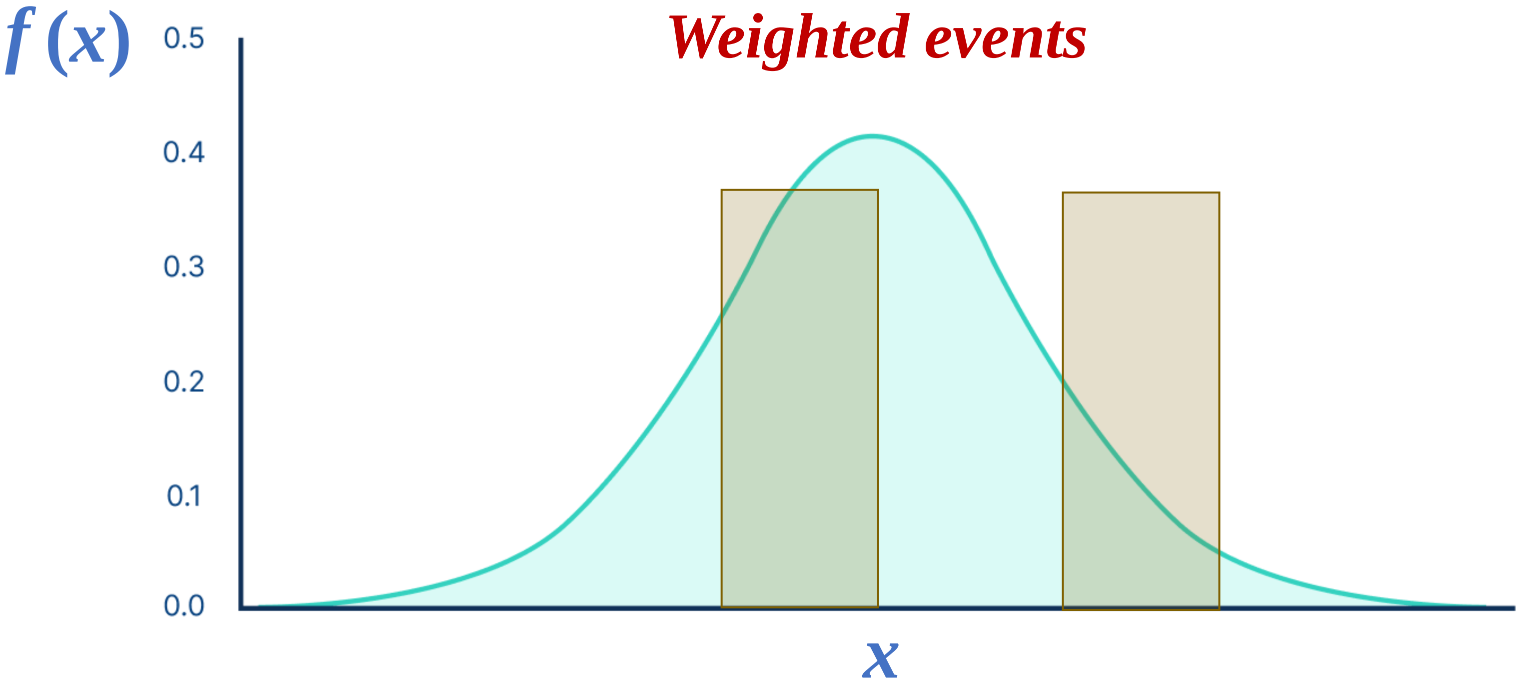
3. if y<f(x) accept event, else reject it

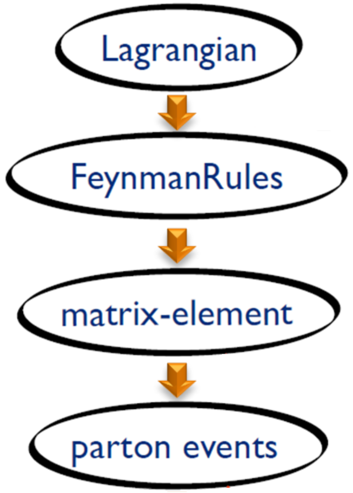
MadGraph5

[arXiv:1405.0301]

Generating events with MadEvent

Weighted events





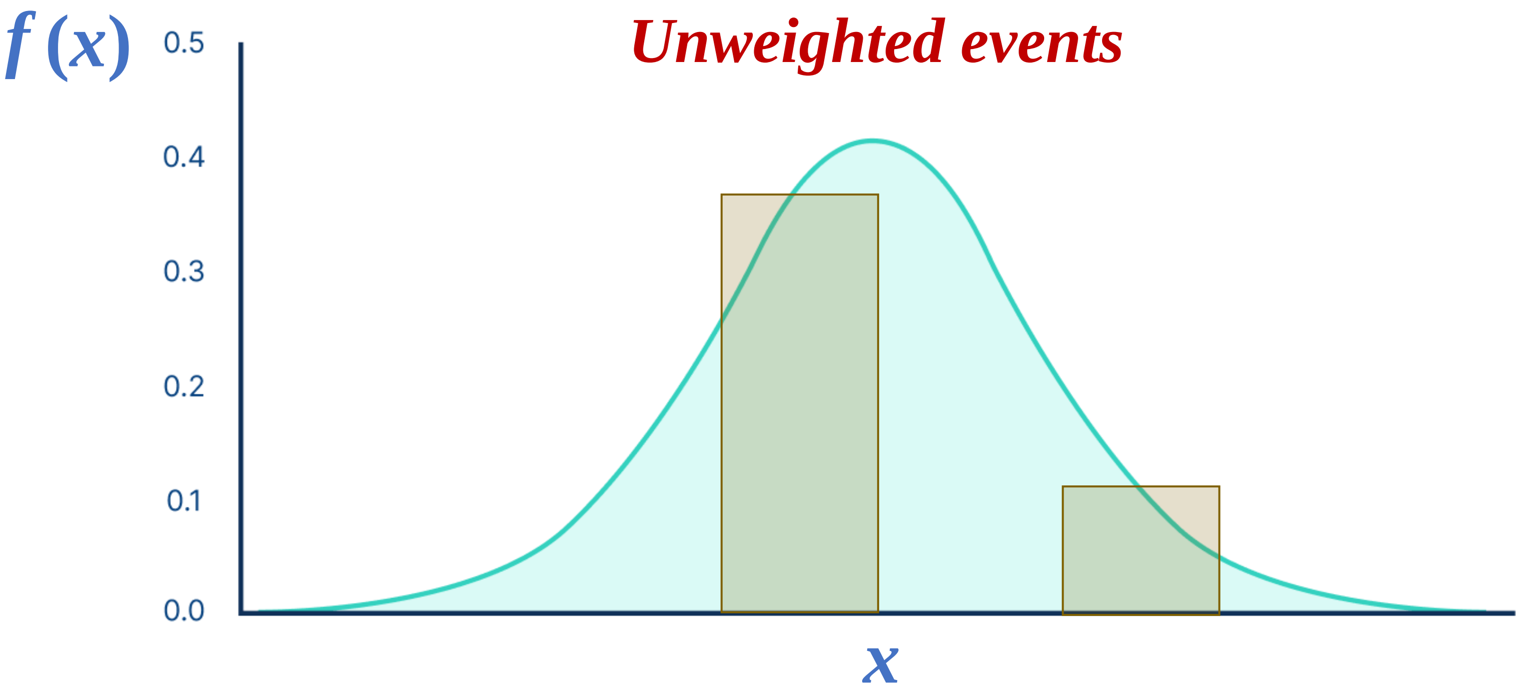
Same number of events in areas of phase space with different probabilities

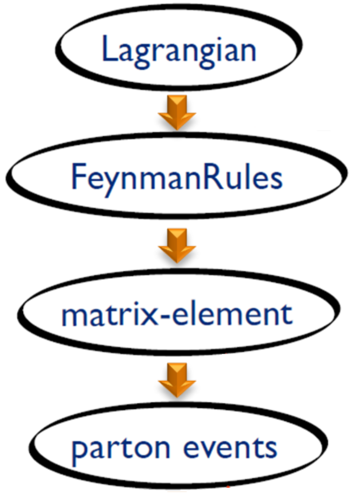
Events must have different weights (weighted)

MadGraph5

[arXiv:1405.0301]

Generating events with MadEvent





Number of events is proportional to the probability of areas of phase space

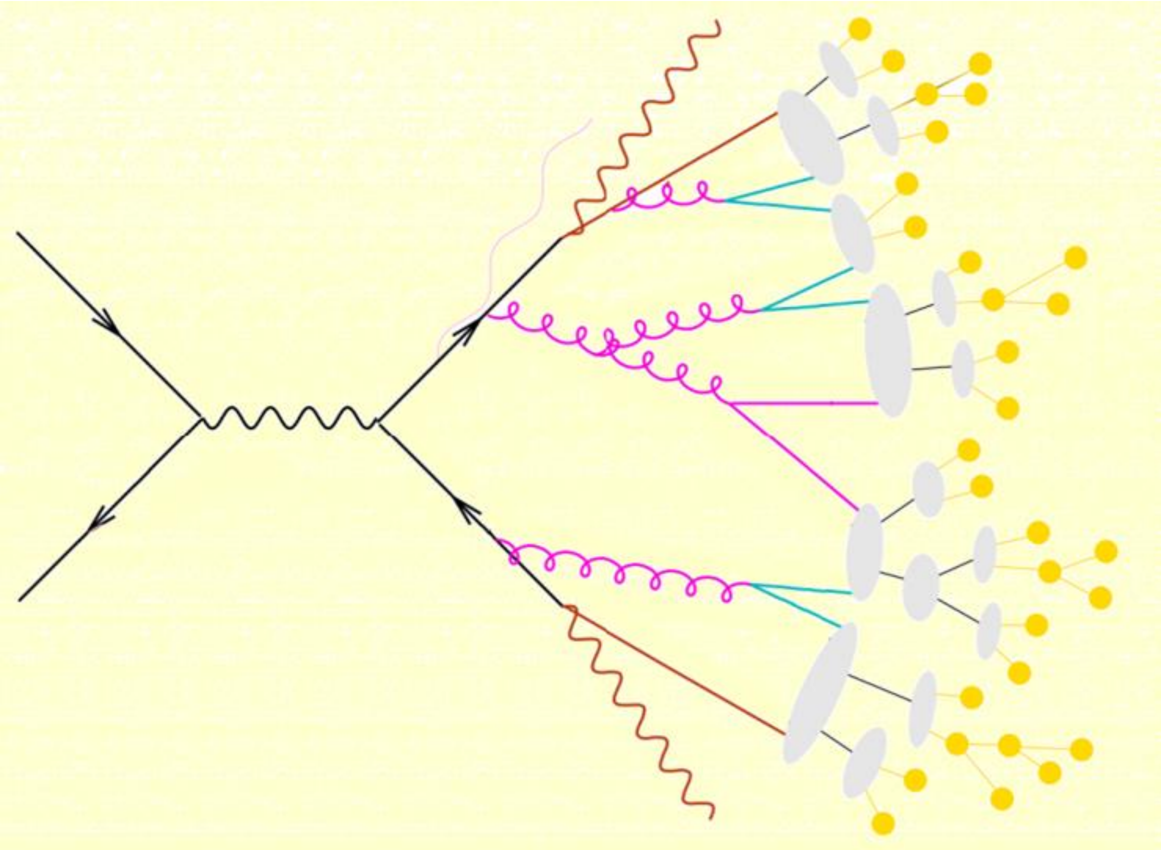
Events have the same weight (unweighted)

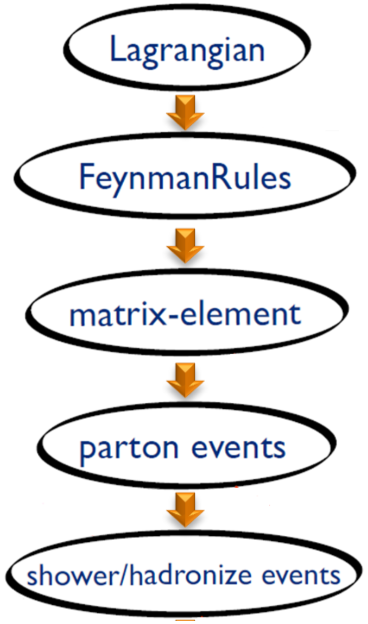
Events distributed as in nature

Pythia

[arXiv:2203.11601]

Pythia adds parton showers, multiparton interactions, hadronization and decay

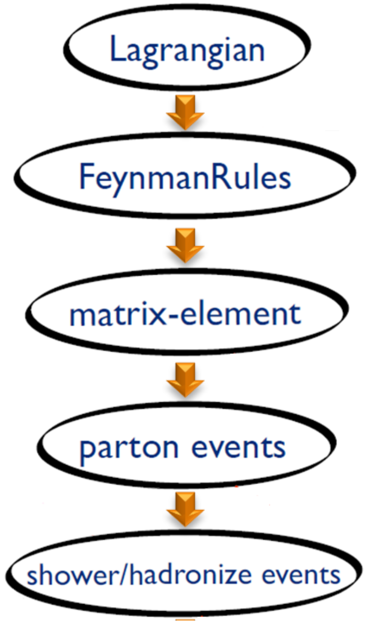




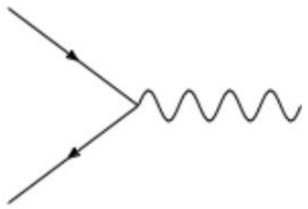
Pythia

[arXiv:2203.11601]

Matching to parton shower



PARTON EVENTS



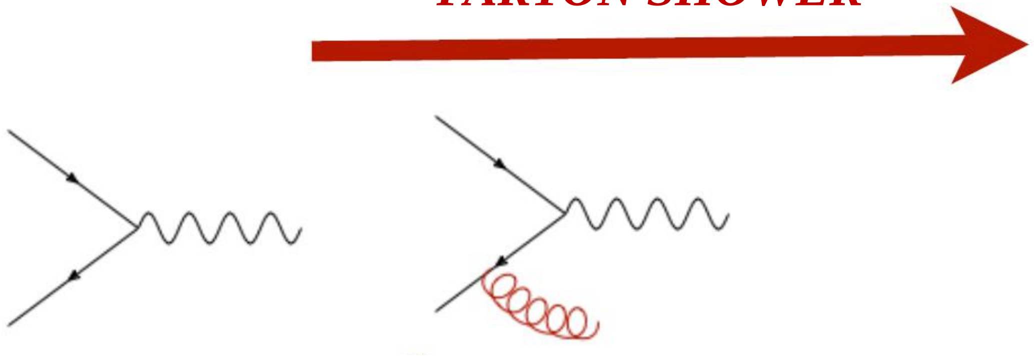
Pythia

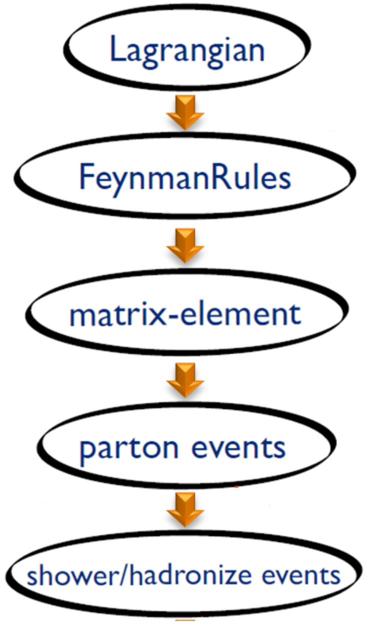
[arXiv:2203.11601]

Matching to parton shower

PARTON SHOWER

PARTON EVENTS

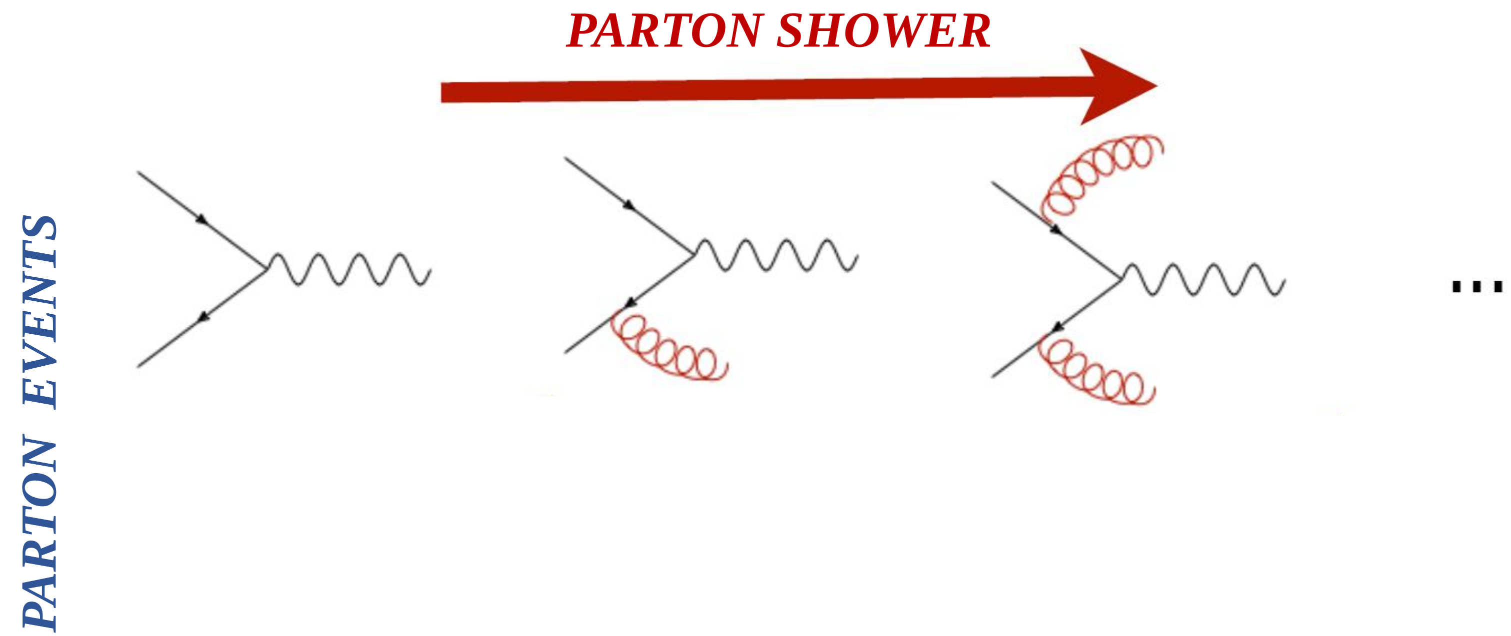


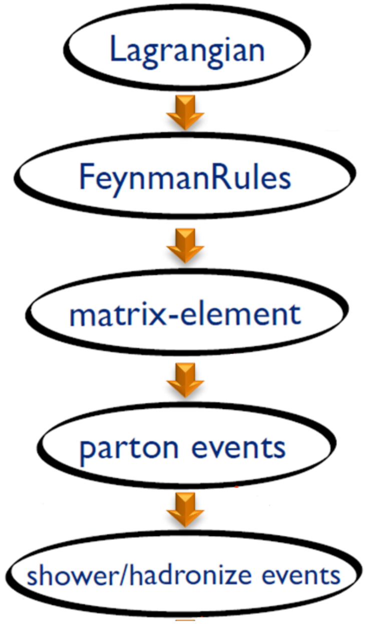


Pythia

[arXiv:2203.11601]

Matching to parton shower

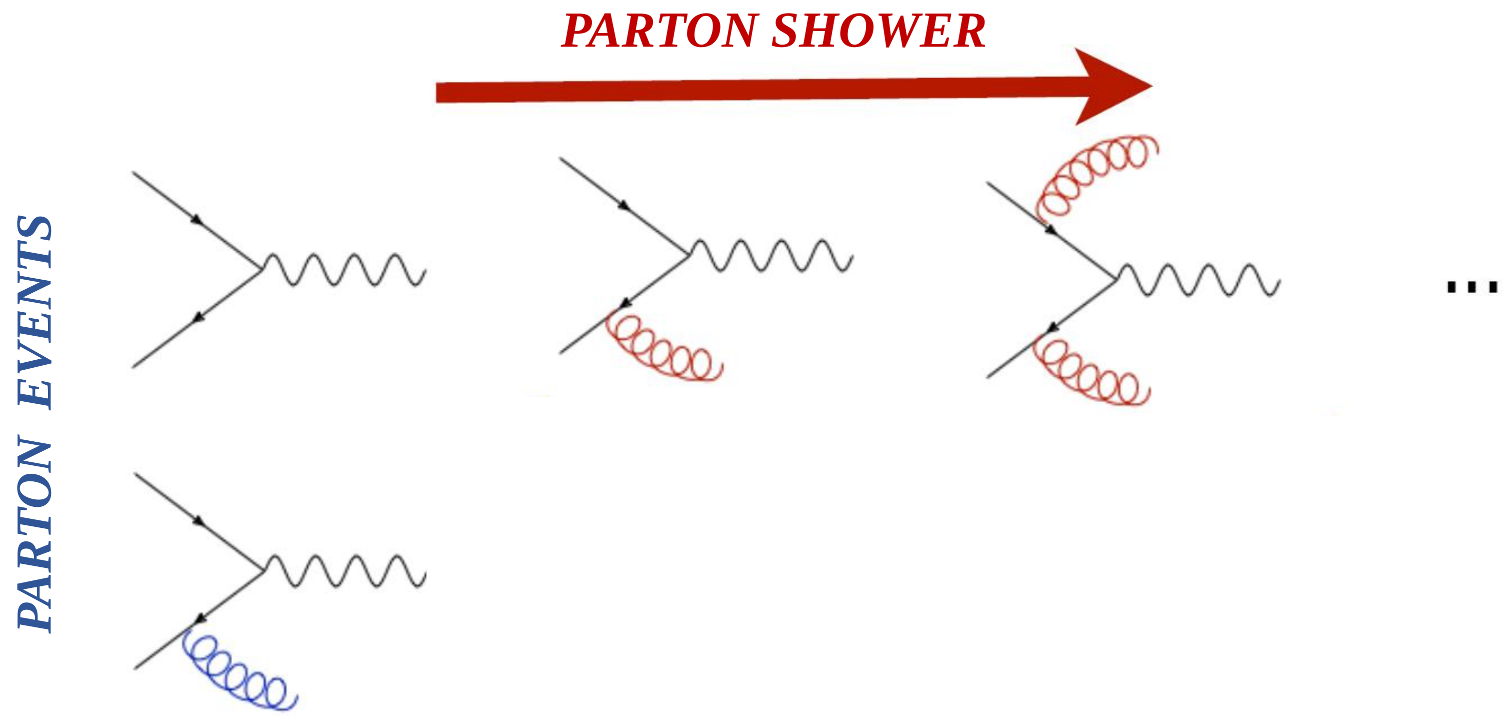


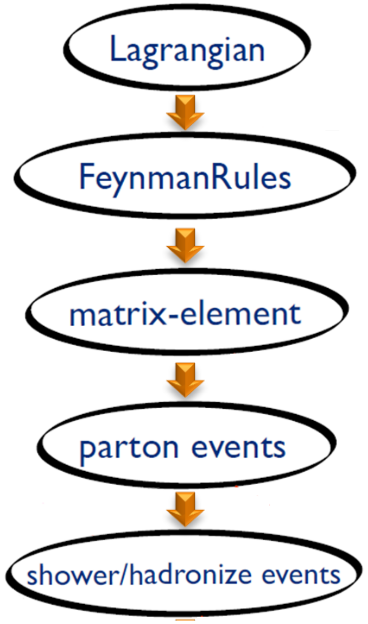


Pythia

[arXiv:2203.11601]

Matching to parton shower

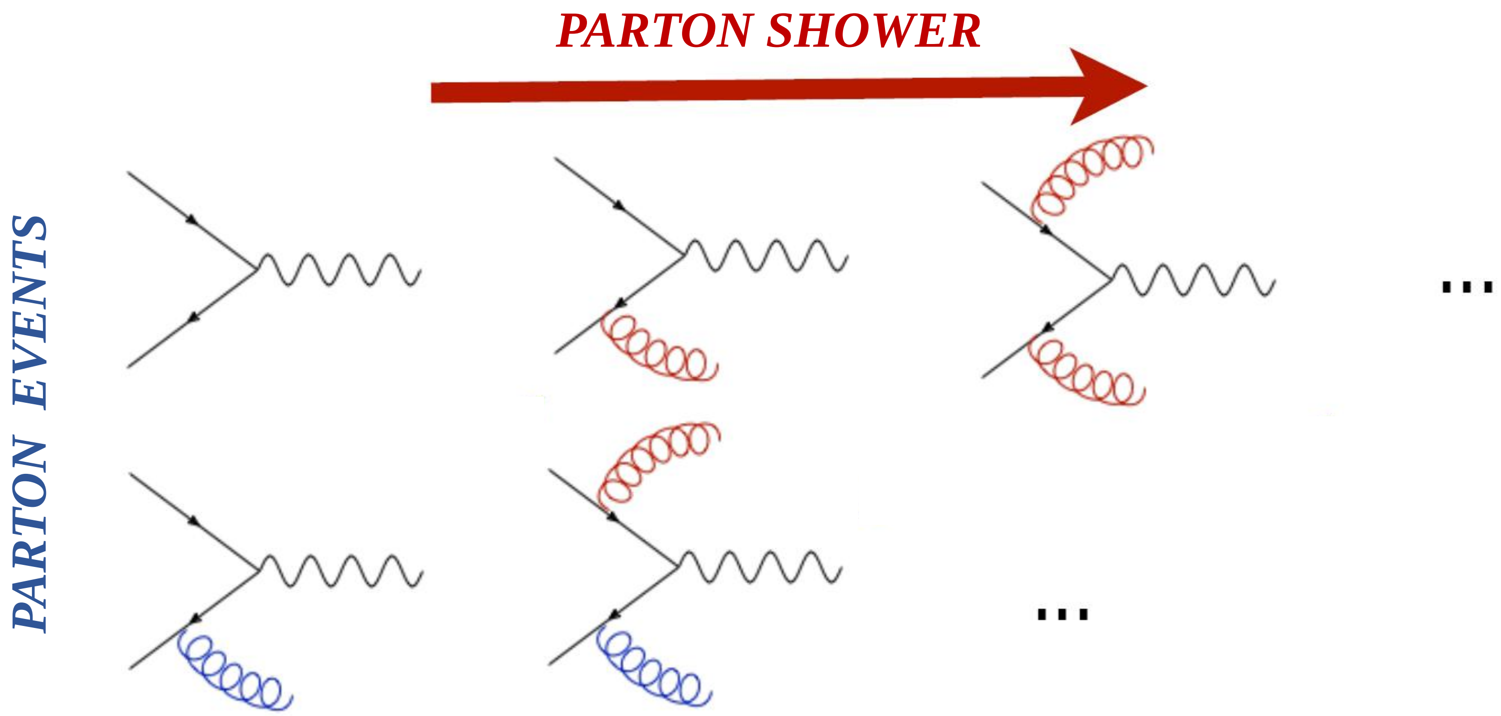


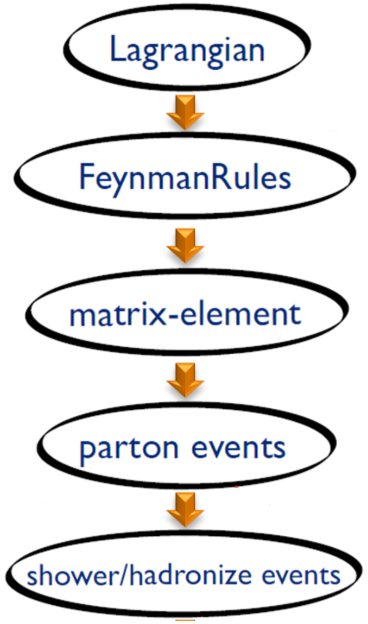


Pythia

[arXiv:2203.11601]

Matching to parton shower

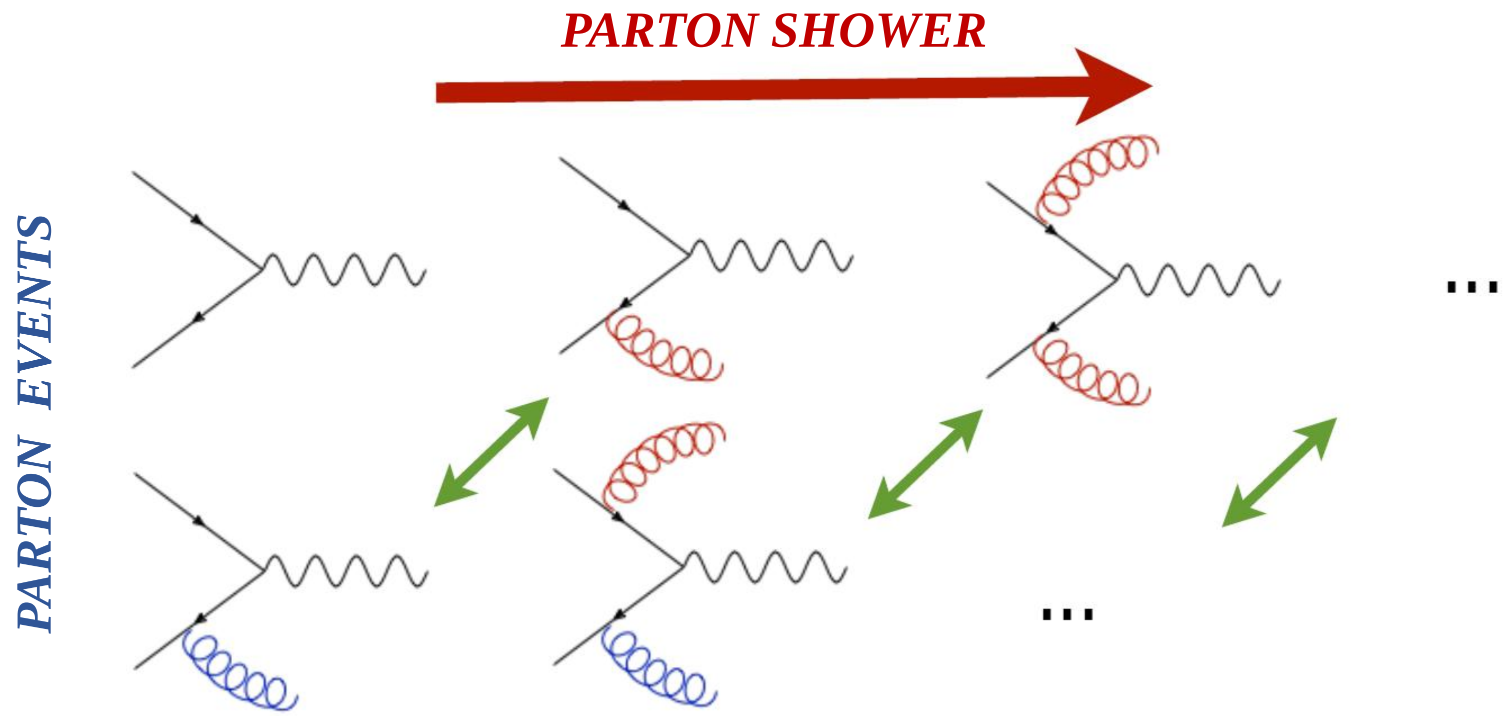


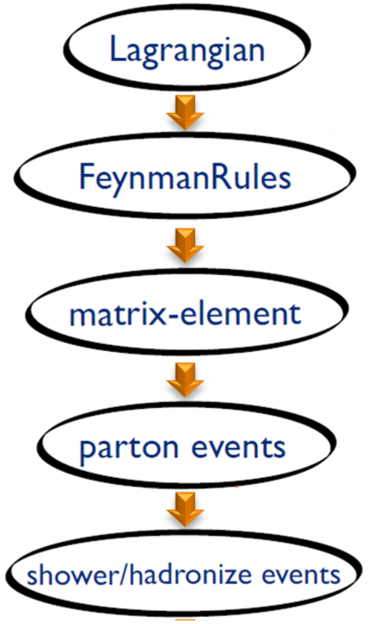


Pythia

[arXiv:2203.11601]

Matching to parton shower

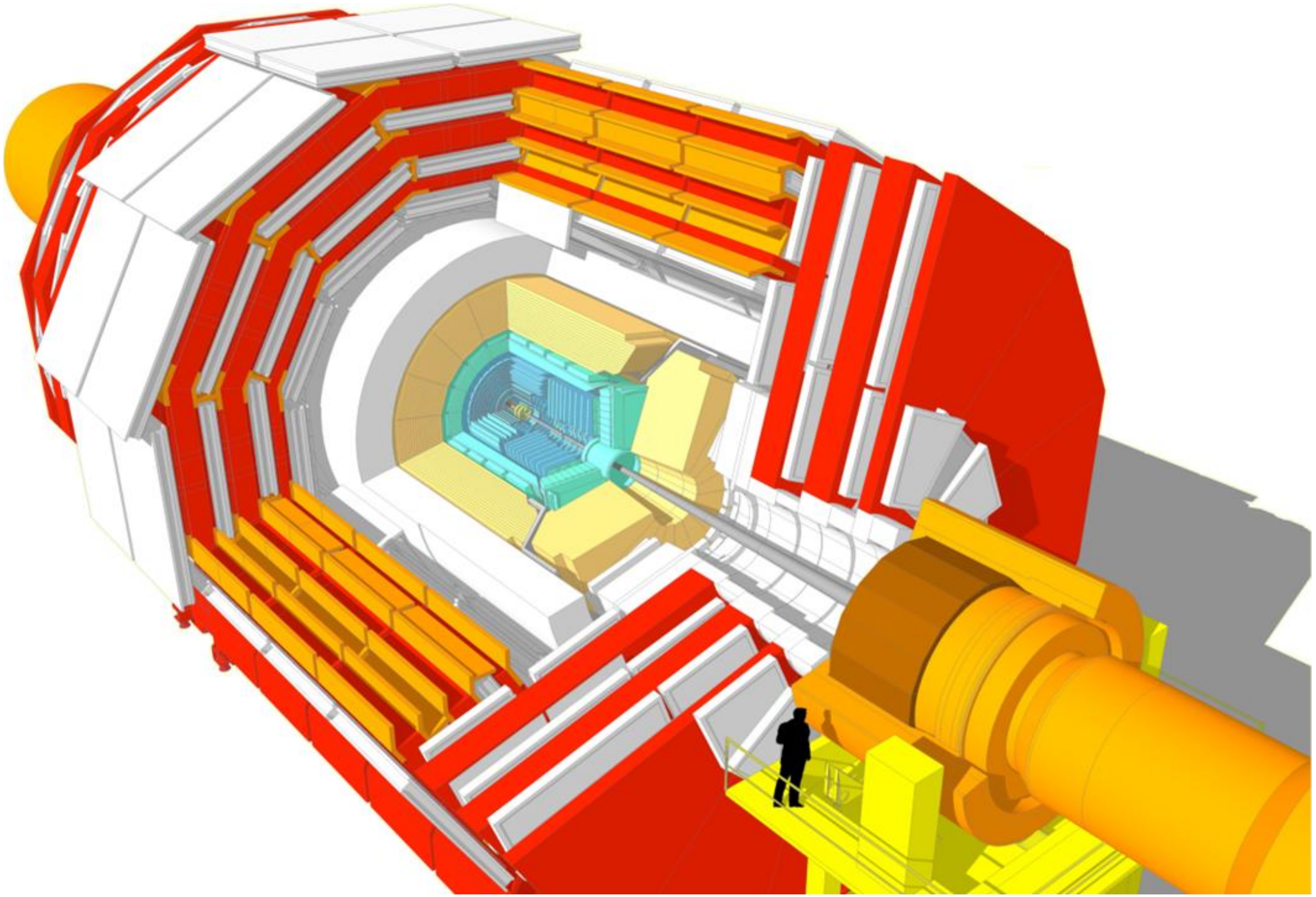


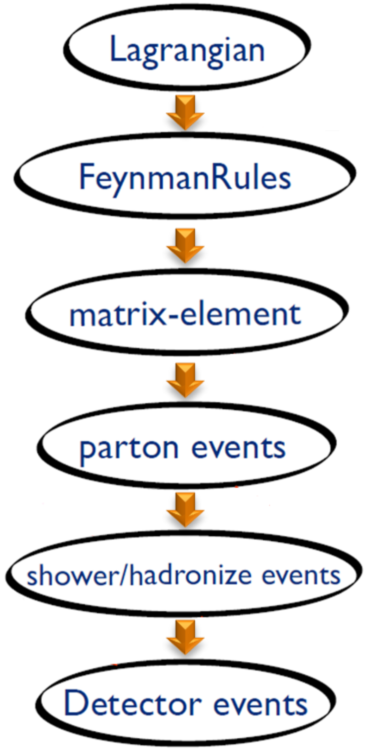


Double counting occurs!

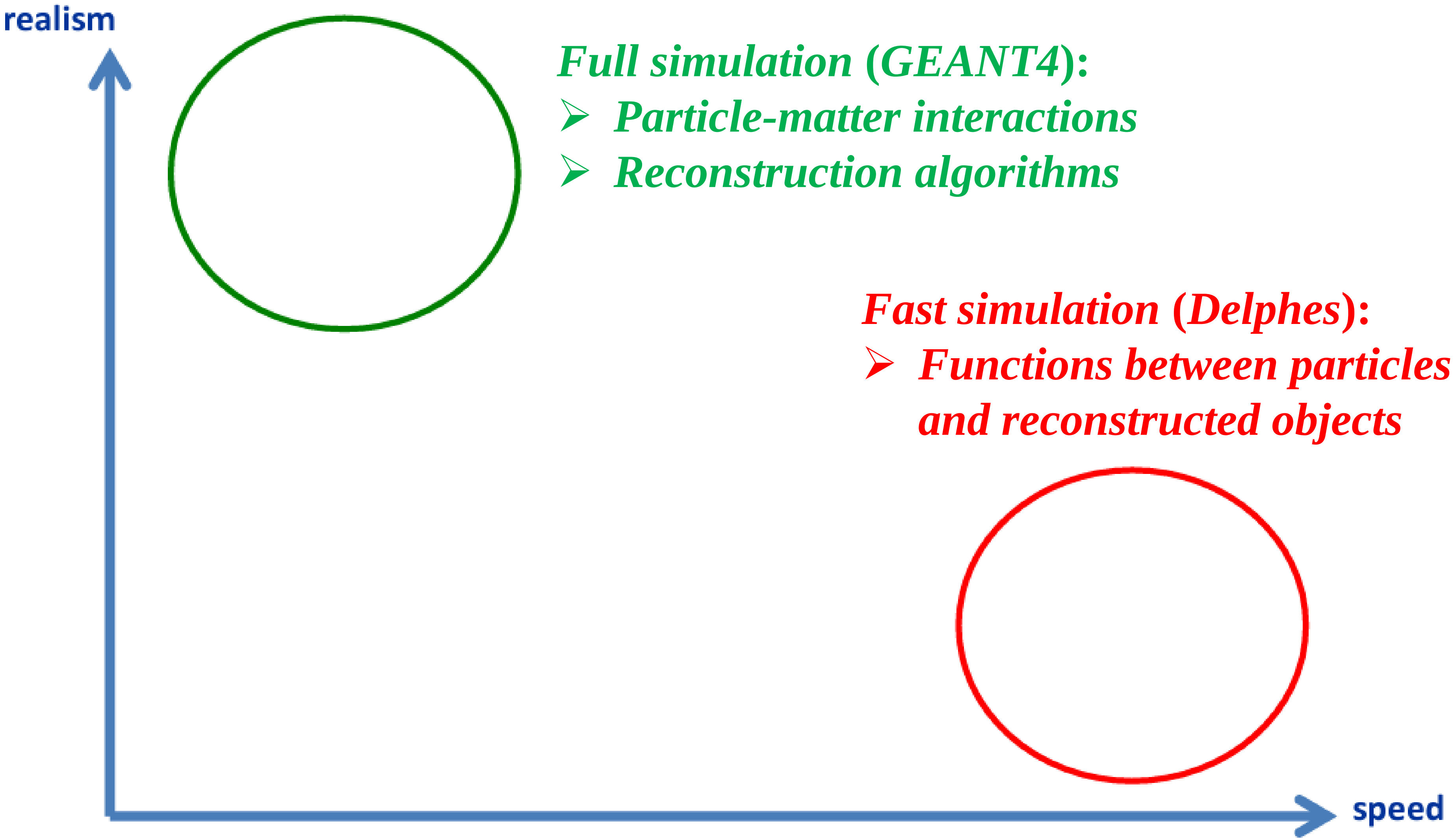
Possible solution: MLM merging

Detector simulation tools



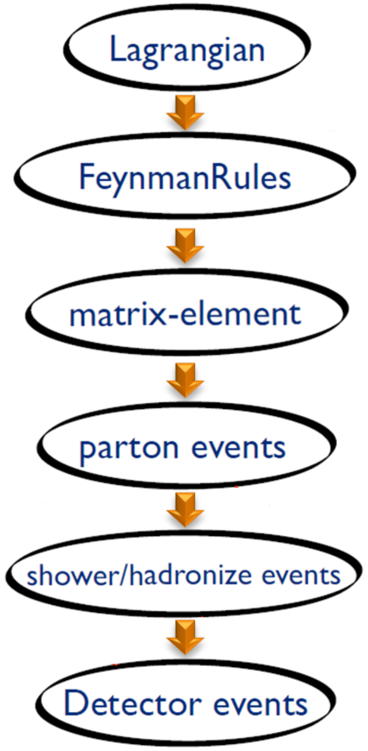


Detector simulation tools



Full simulation (GEANT4):
- *Particle-matter interactions*
- *Reconstruction algorithms*

Fast simulation (Delphes):
- *Functions between particles and reconstructed objects*

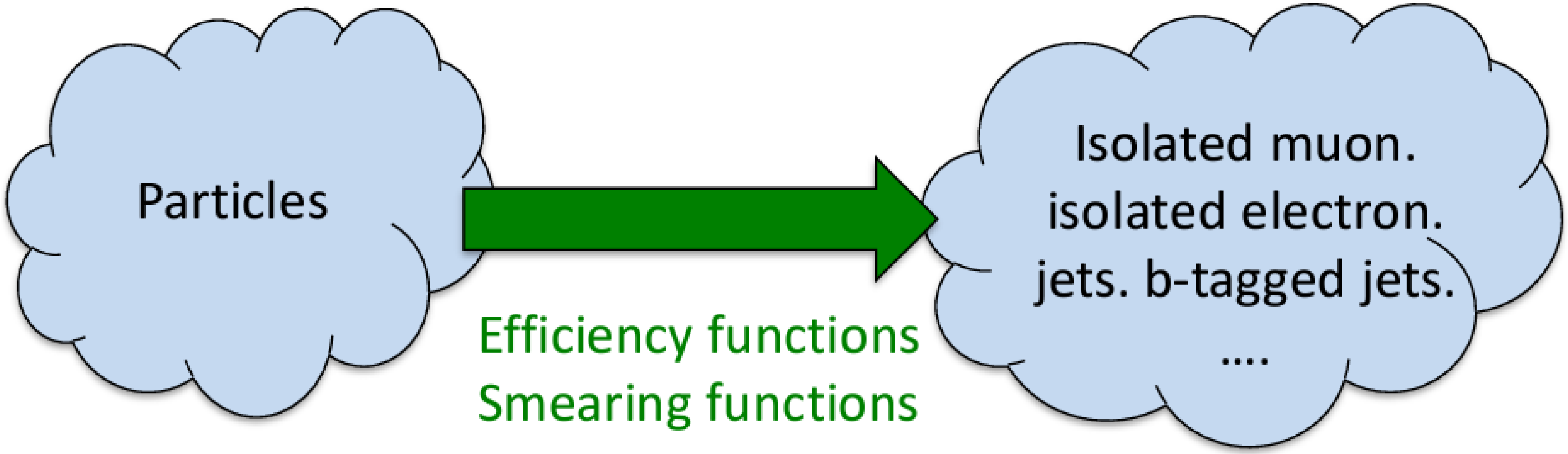


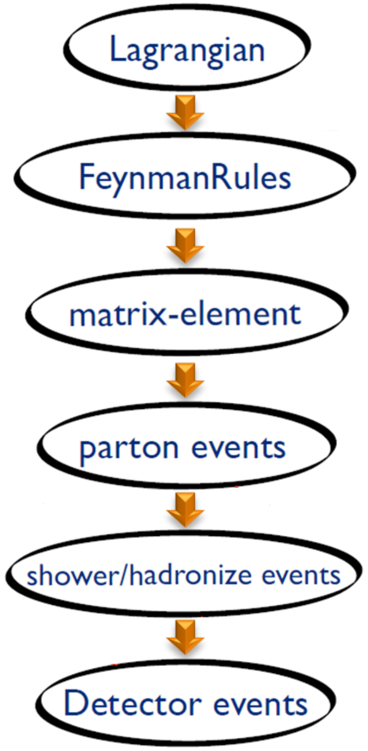
Delphes

[arXiv:1307.6346]

Fast detector simulation for phenomenological studies

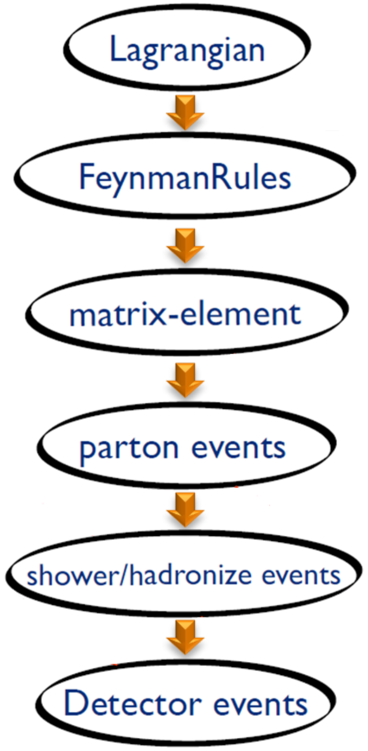
Based on the parameterization of the detector response





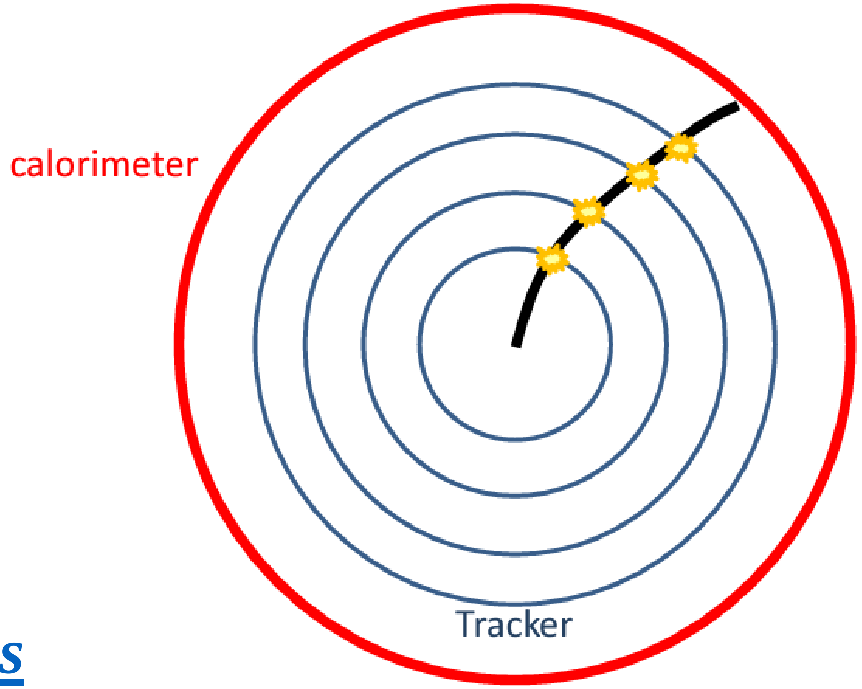
Delphes

[arXiv:1307.6346]



Fast detector simulation for phenomenological studies

Based on the parameterization of the detector response



See how Delphes works

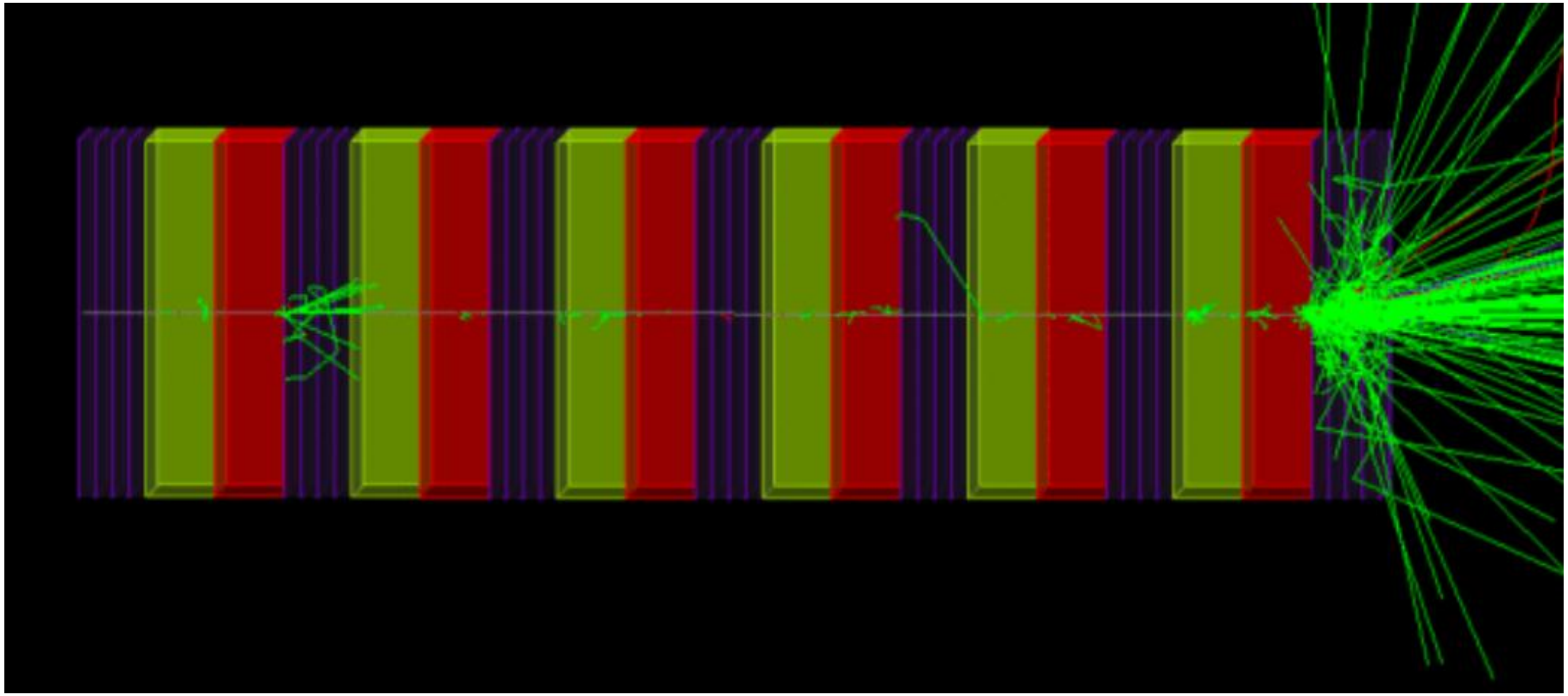
No real tracking in Delphes

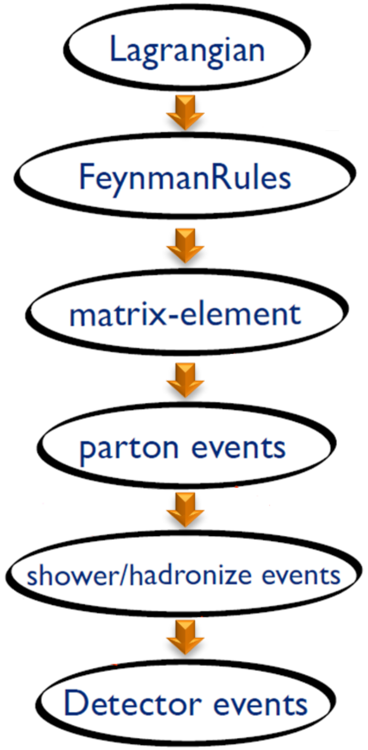
GEANT

[0.1109/TNS.2006.869826]

GEometry ANd Tracking

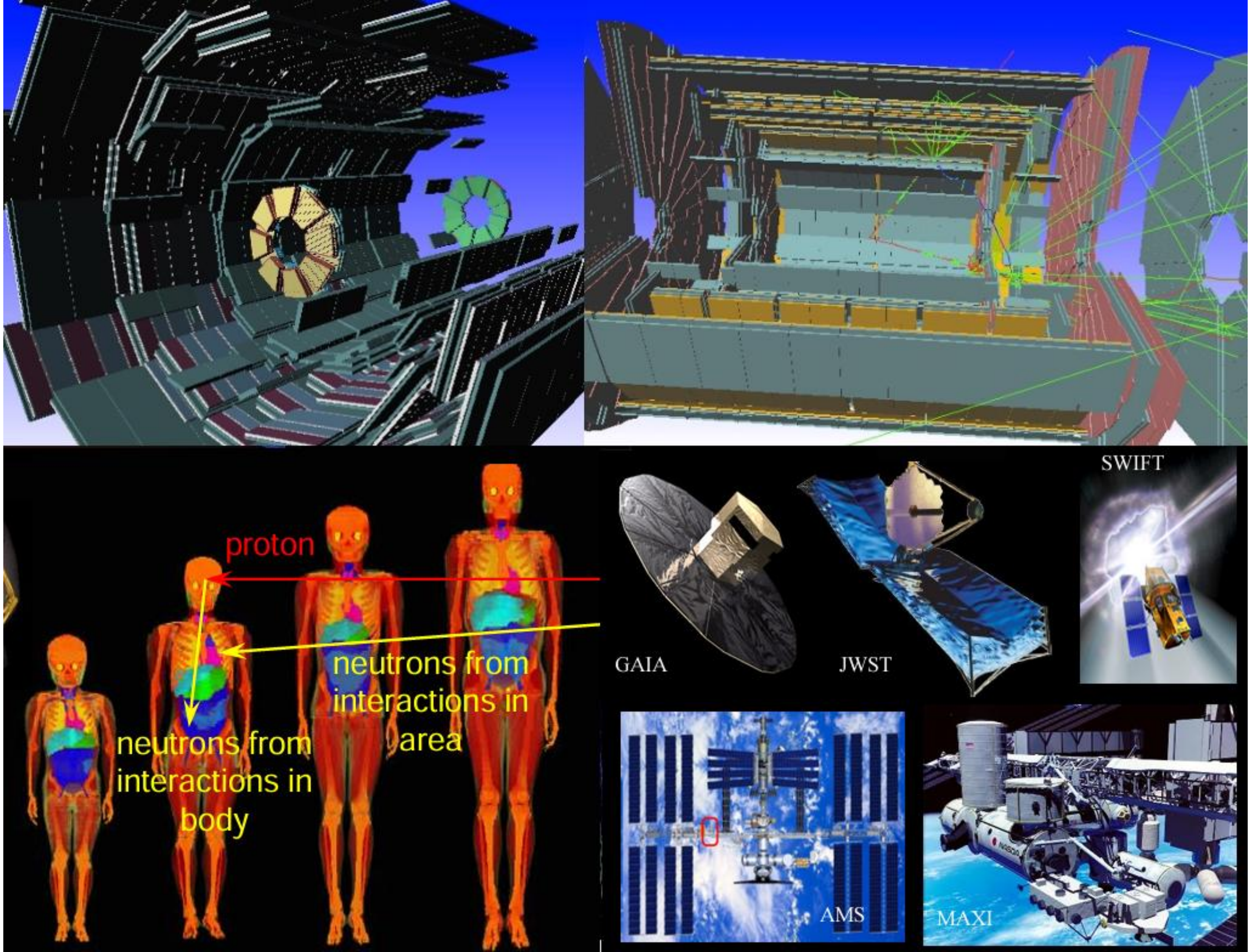
A Monte Carlo software toolkit
to simulate the passage of particles through matter

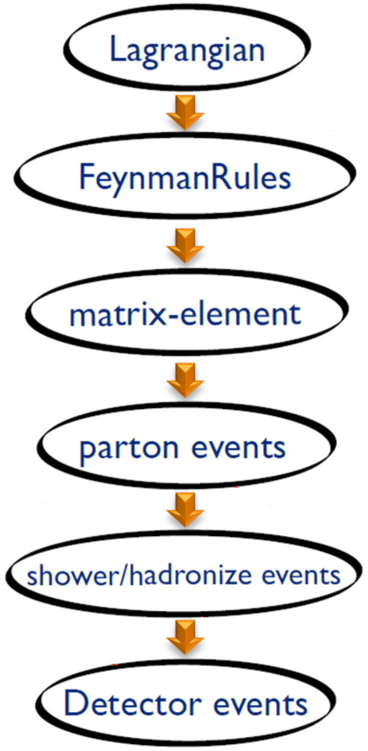




Widely used

A gift from particle physics



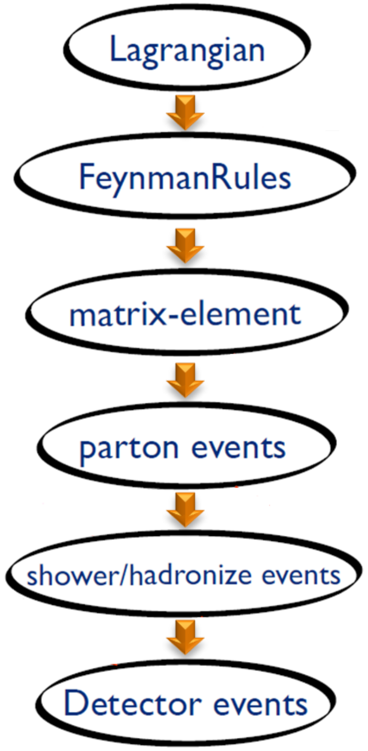


GEANT

[0.1109/TNS.2006.869826]

Particle-Matter interaction

Digitization

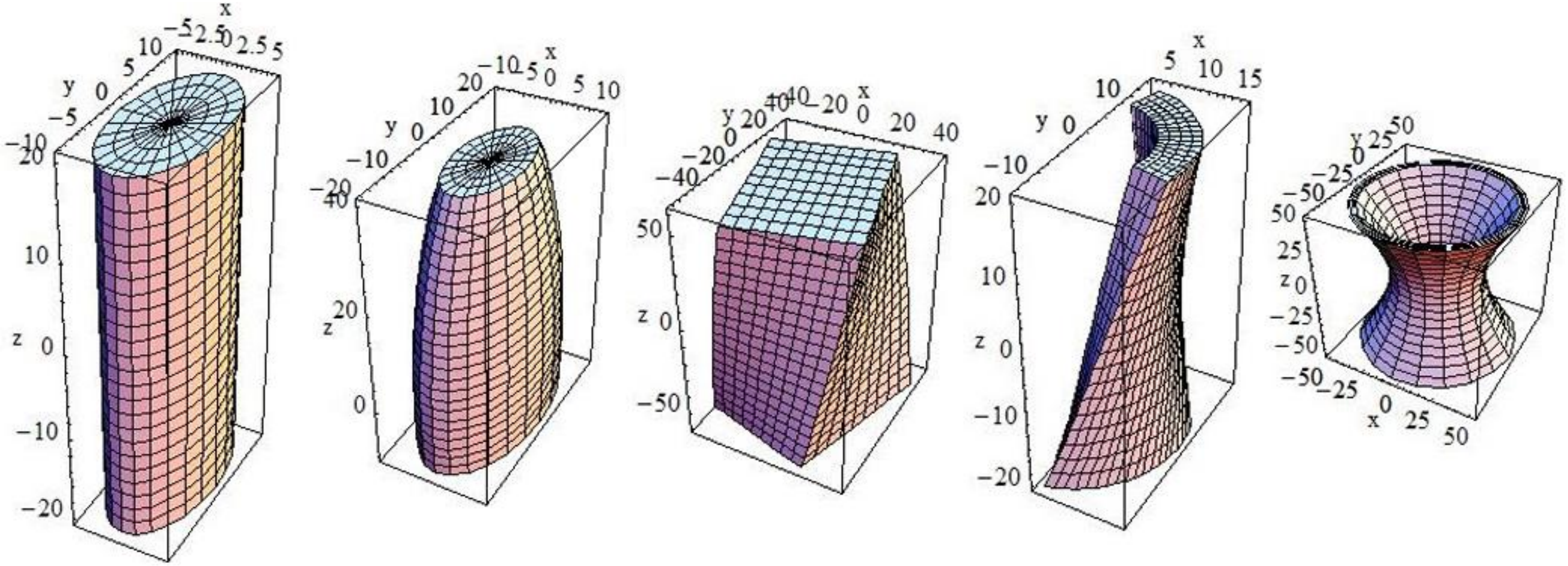


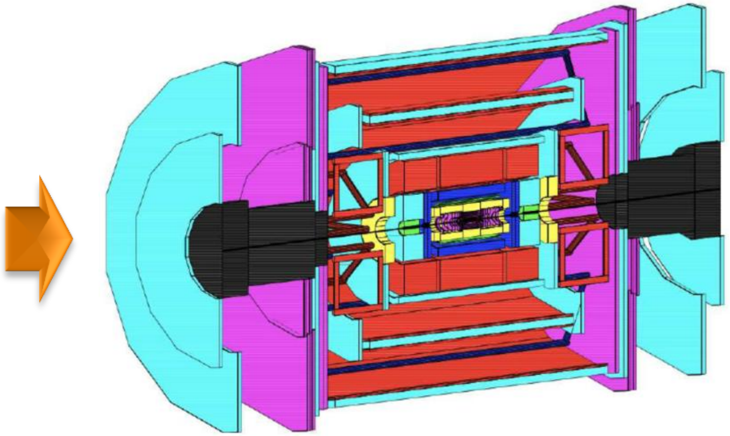
GEANT

[0.1109/TNS.2006.869826]

Particle-Matter interaction

- ***Describe the geometry and the material of the detector***





GEANT

[0.1109/TNS.2006.869826]

Particle-Matter interaction

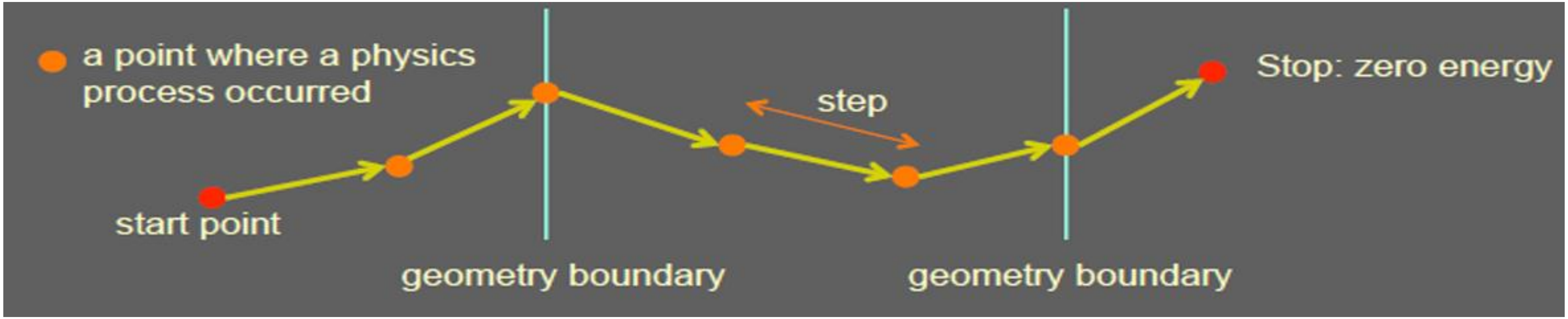
- ***Describe the geometry and the material of the detector***
- ***Treat a particle at a time***

GEANT

[0.1109/TNS.2006.869826]

Particle-Matter interaction

- ➢ *Describe the geometry and the material of the detector*
- ➢ *Treat a particle at a time*
- ➢ *Trajectory of the particle is split in steps (finite displacements)*

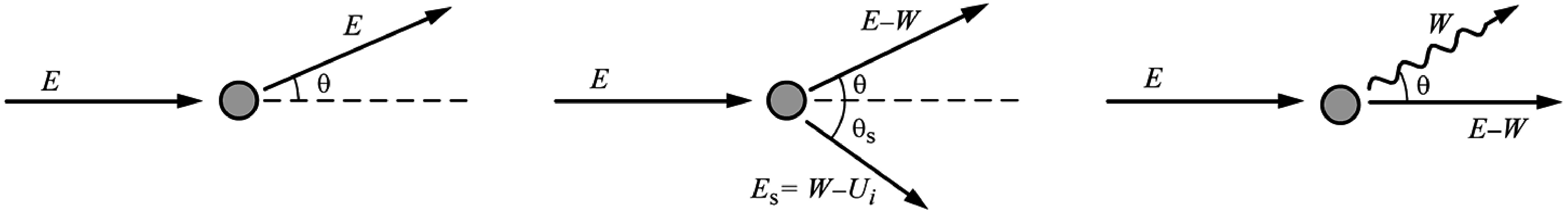


GEANT

[0.1109/TNS.2006.869826]

Particle-Matter interaction

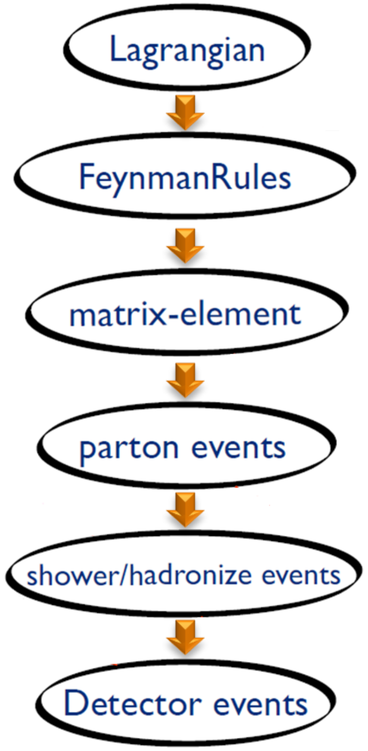
- ***Describe the geometry and the material of the detector***
- ***Treat a particle at a time***
- ***Trajectory of the particle is split in steps (finite displacements)***
- ***Simulate the physics along a step and at the end of each step***



Elastic scattering | Inelastic scattering | Bremsstrahlung emission

GEANT

[0.1109/TNS.2006.869826]



Particle-Matter interaction

- *Describe the geometry and the material of the detector*
- *Treat a particle at a time*
- *Trajectory of the particle is split in steps (finite displacements)*
- *Simulate the physics along a step and at the end of each step*

Digitization

- *convert the energy deposit into electric signal*

GEANT

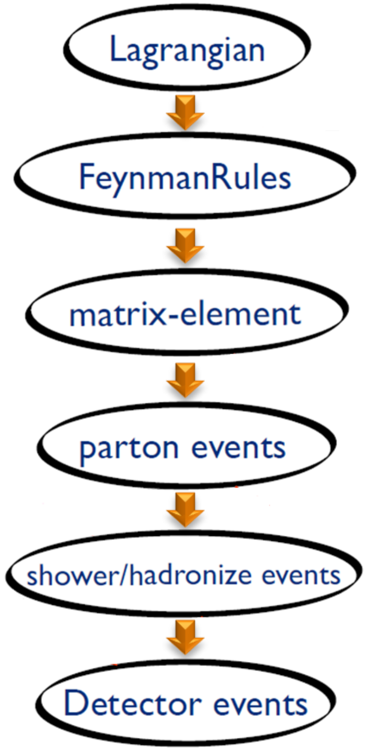
[0.1109/TNS.2006.869826]

Particle-Matter interaction

- *Describe the geometry and the material of the detector*
- *Treat a particle at a time*
- *Trajectory of the particle is split in steps (finite displacements)*
- *Simulate the physics along a step and at the end of each step*

Digitization

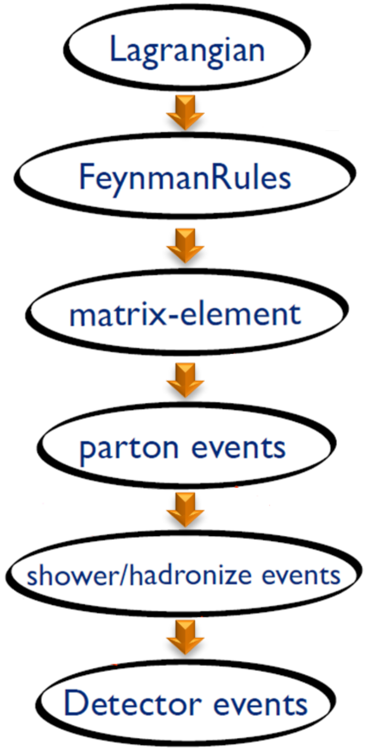
- *convert the energy deposit into electric signal*
- *Identify the sensitive part of the detector*



GEANT

[0.1109/TNS.2006.869826]

GEANT4 A SIMULATION TOOLKIT



Particle-Matter interaction

- ***Describe the geometry and the material of the detector***
- ***Treat a particle at a time***
- ***Trajectory of the particle is split in steps (finite displacements)***
- ***Simulate the physics along a step and at the end of each step***

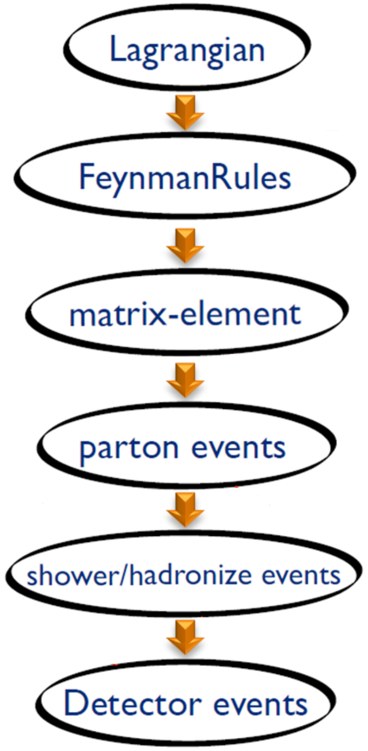
Digitization

- ***convert the energy deposit into electric signal***
- ***Identify the sensitive part of the detector***
- ***Modelize detector answer***

GEANT

[0.1109/TNS.2006.869826]

MC Simulation of Particle Interactions with Matter



- ***The exponential law:***

$P(x)$: *probability of not having an interaction after a distance x*

$w\,dx$: *probability of having an interaction between x and x+dx*

depends on material and physical process (refers to w)

$$P(x+dx) = P(x)(1-w\,dx)$$

$$P(x) = e^{-wx} \qquad P_{int}(x) = 1 - e^{-wx}$$

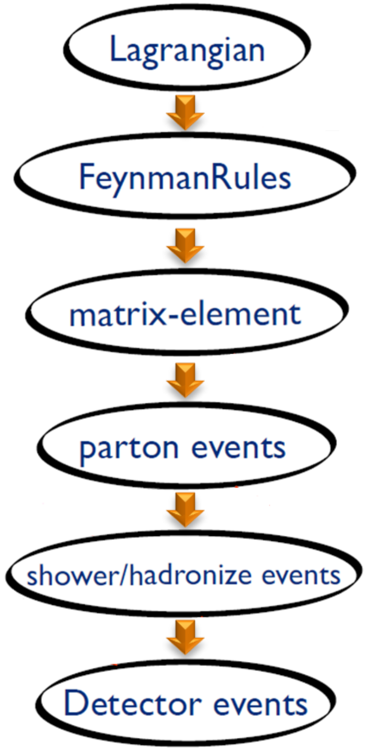
Now use the inverse method to generate an interaction:

$P_{int} = \alpha$: *uniform random number of [0,1]* → $xw = -\ln(1-\alpha)$

GEANT

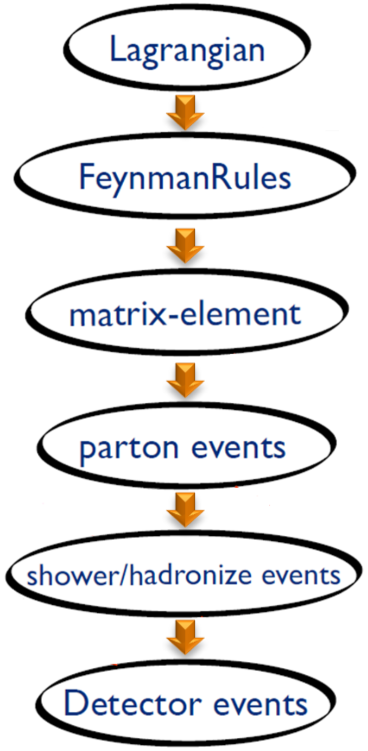
[0.1109/TNS.2006.869826]

Particle Transportation: How to Determine a Step



GEANT

[0.1109/TNS.2006.869826]



Particle Transportation: How to Determine a Step

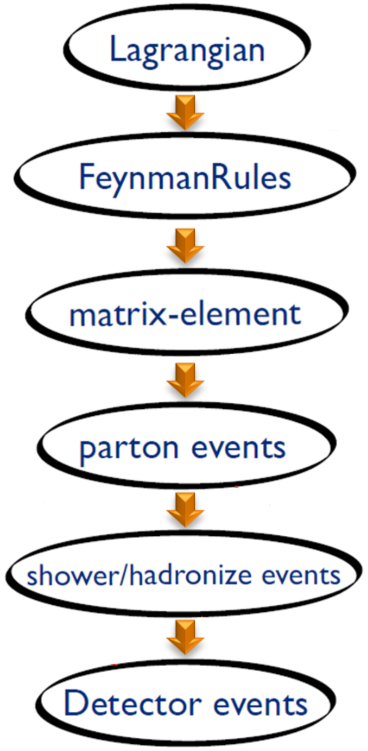
1) *Evaluate x using α,w for each physical process independently* ($xw = -\ln(1-\alpha)$)

GEANT

[0.1109/TNS.2006.869826]

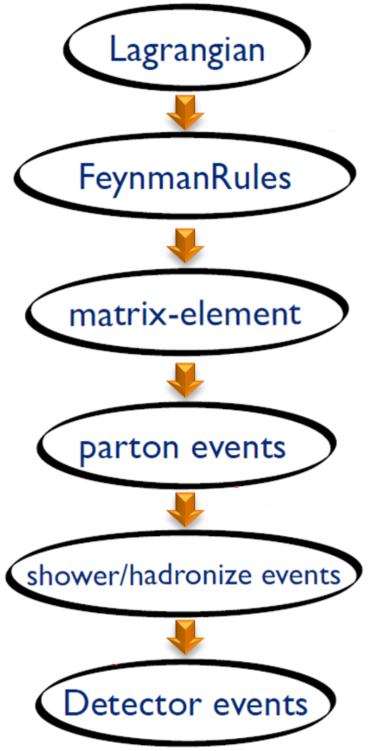
Particle Transportation: How to Determine a Step

1) *Evaluate x using α,w for each physical process independently (xw = - ln(1-α))*
2) *Compare: process with minimum x determines the step length*



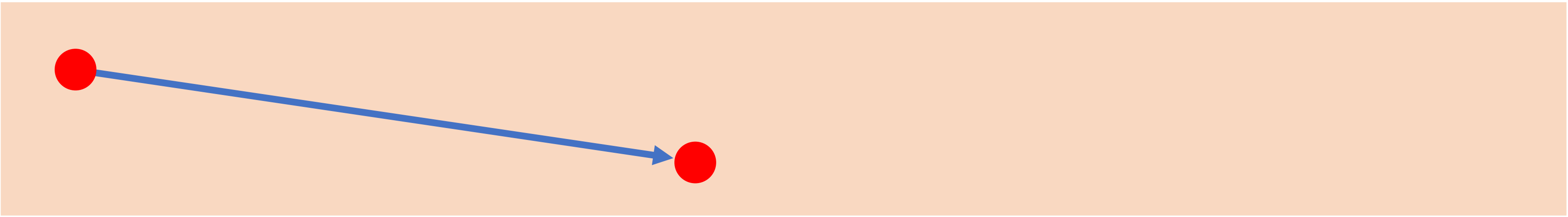
GEANT

[0.1109/TNS.2006.869826]



Particle Transportation: How to Determine a Step

1) *Evaluate **x** using **α,w** for each physical process independently* ($xw = -\ln(1-\alpha)$)
2) *Compare: process with **minimum x** determines the step length*
3) *Transport particle for the determined step*

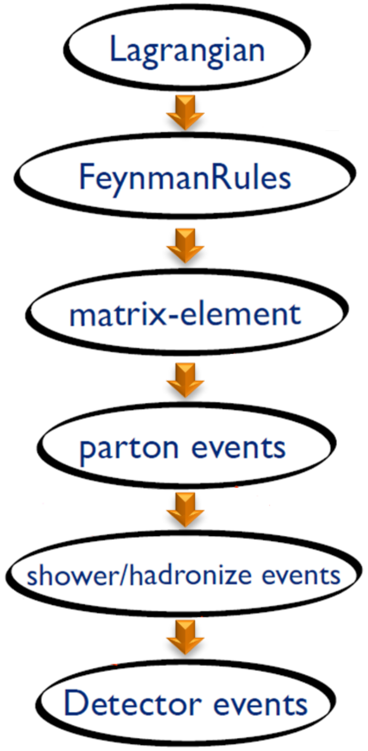


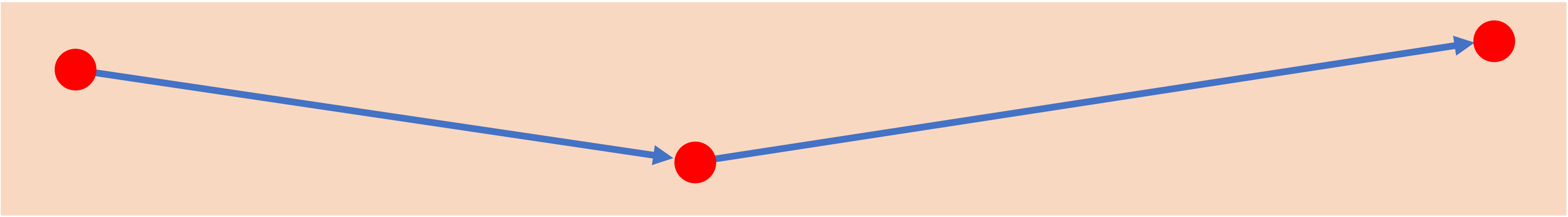
GEANT

[0.1109/TNS.2006.869826]

Particle Transportation: How to Determine a Step

1) *Evaluate x using α,w for each physical process independently ($xw = -\ln(1-\alpha)$)*
2) *Compare: process with minimum x determines the step length*
3) *Transport particle for the determined step*
4) *If the particle is still alive after the interaction, do the sampling again and continue transportation*



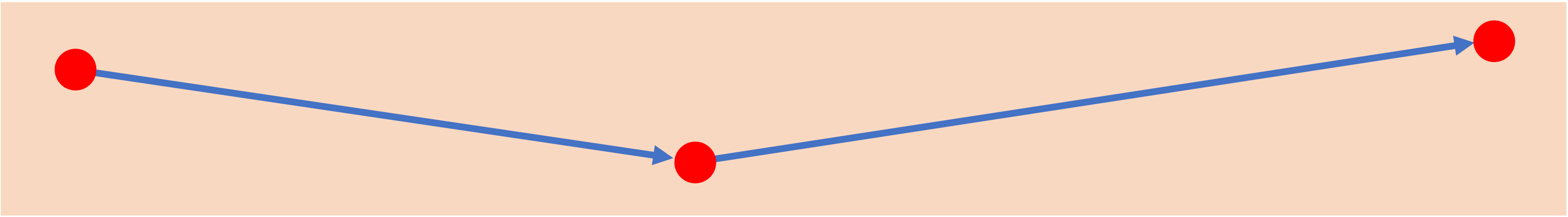


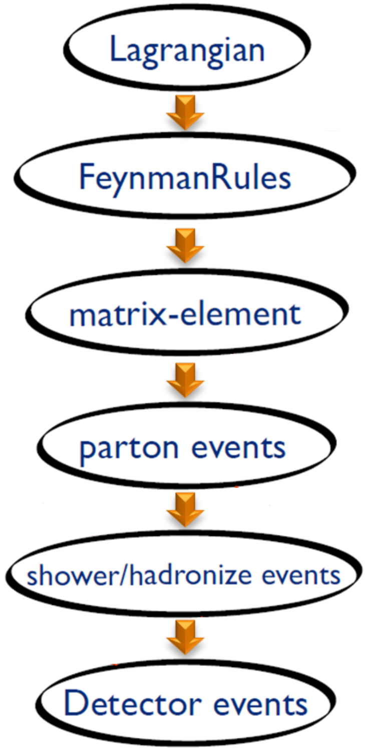
GEANT

[0.1109/TNS.2006.869826]

Particle Transportation: How to Determine a Step

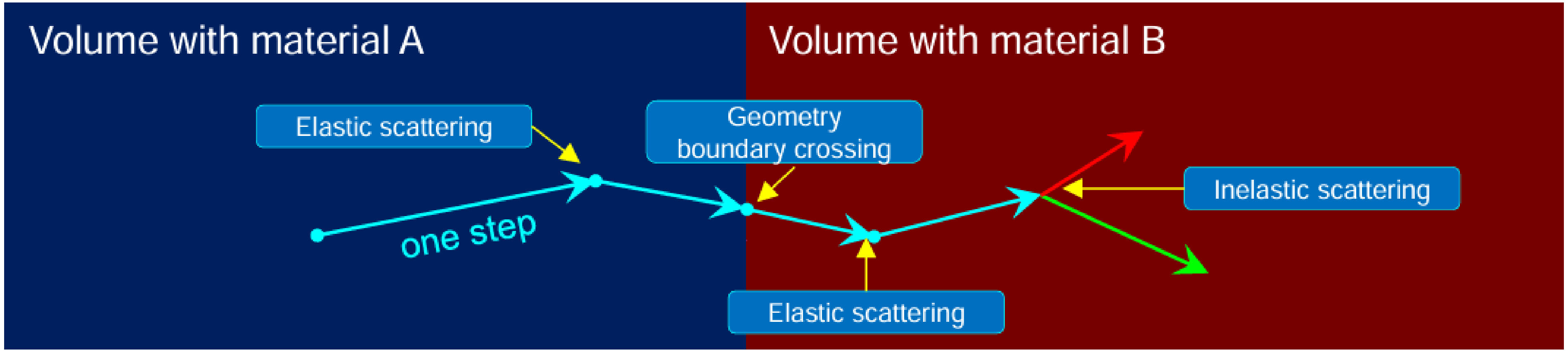
1) *Evaluate x using α,w for each physical process independently (xw = - ln(1-α))*
2) *Compare: process with **minimum x** determines the step length*
3) *Transport particle for the determined step*
4) *If the particle is still alive after the interaction, do the **sampling again and continue transportation***
5) *If the particle disappears after the interaction, then the **transportation is terminated***

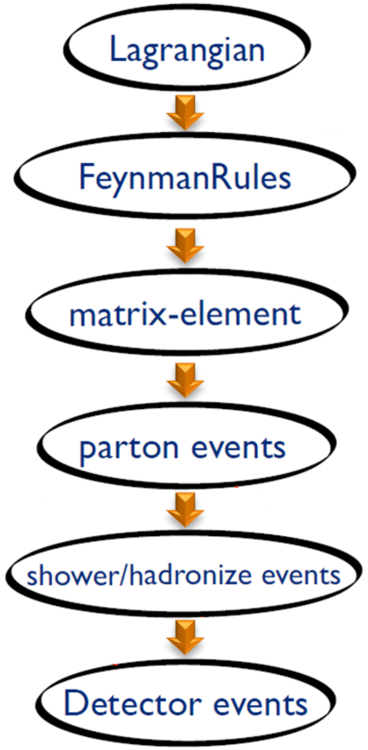




GEANT

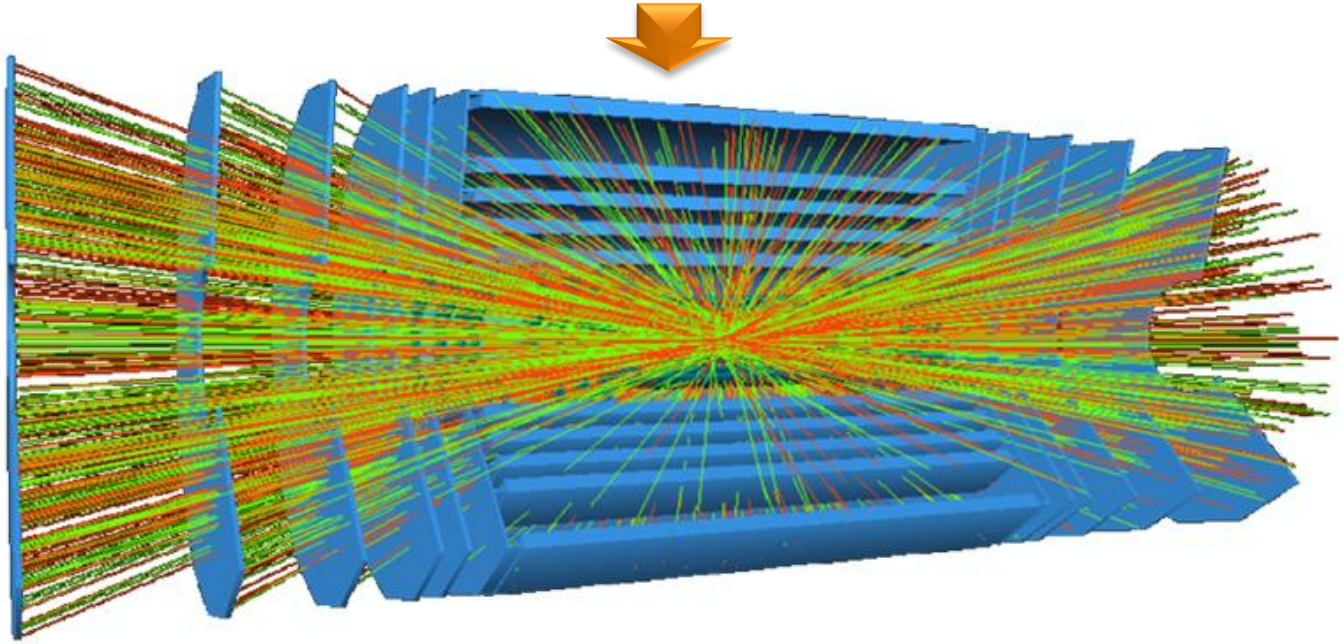
[0.1109/TNS.2006.869826]





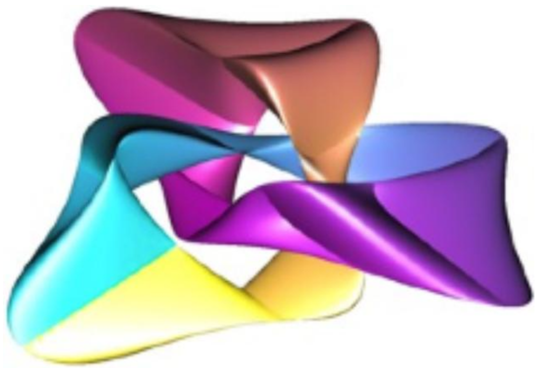
Then modelize the detector response using a digitizer

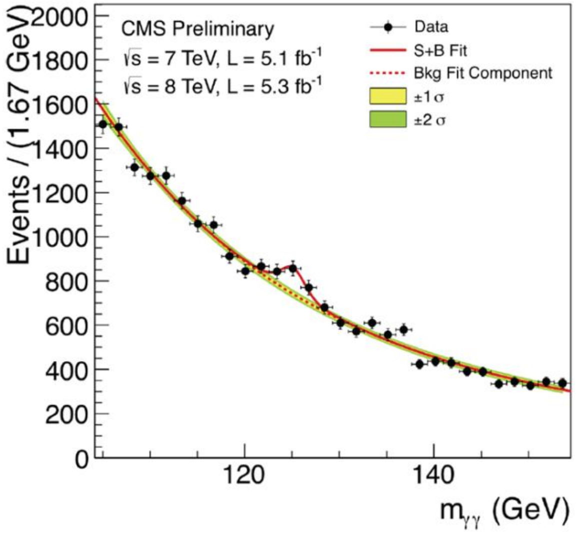
GEANT is done

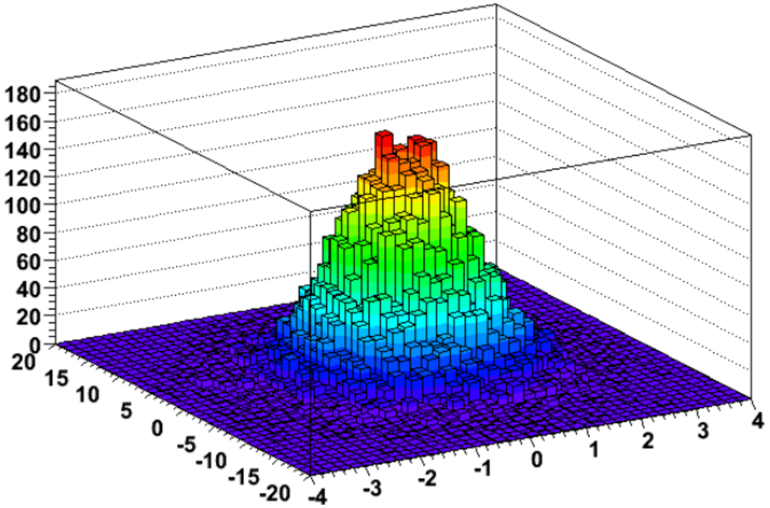


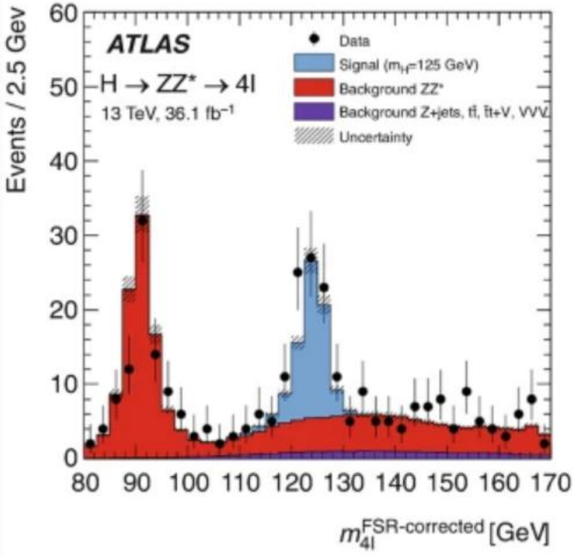
ROOT

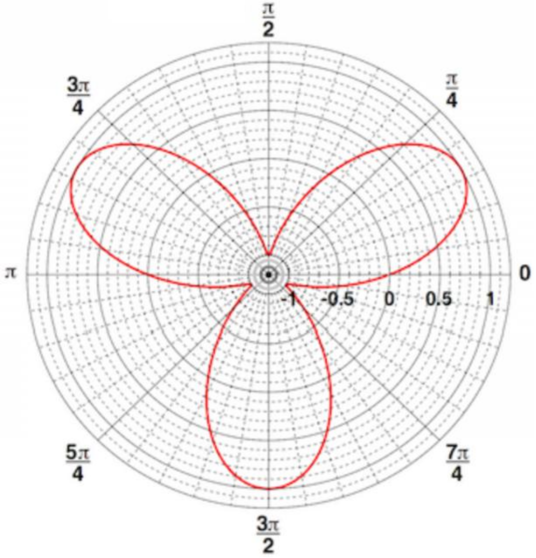
[10.1016/S0168-9002(97)00048-X]

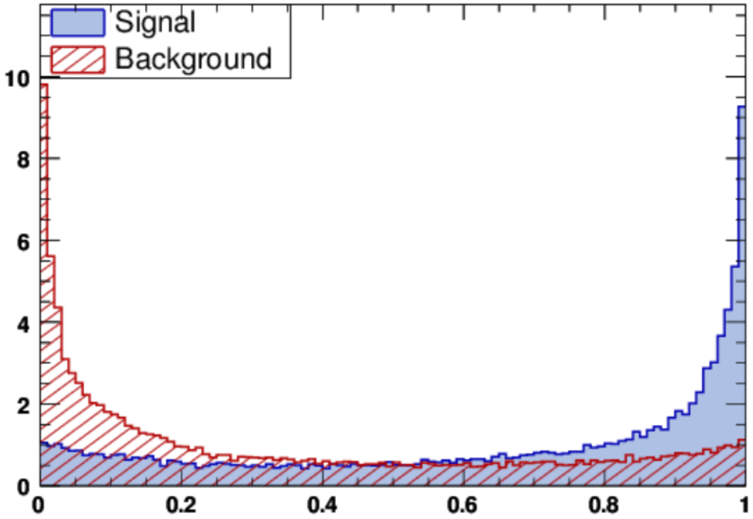










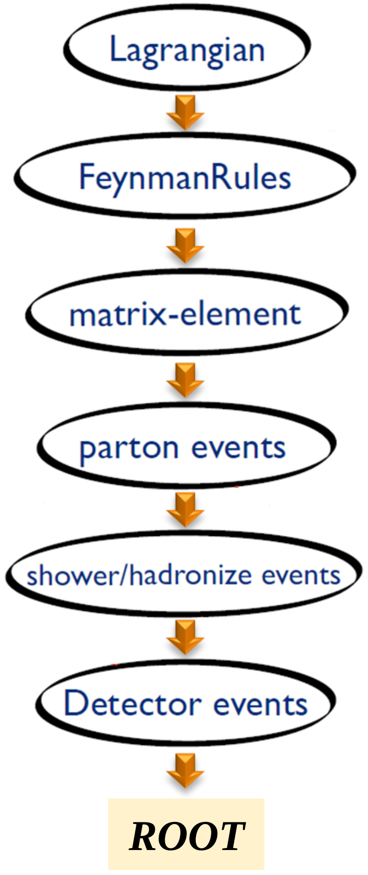


Lagrangian
↓
FeynmanRules
↓
matrix-element
↓
parton events
↓
shower/hadronize events
↓
Detector events
↓
ROOT

ROOT

[10.1016/S0168-9002(97)00048-X]

- ROOT is written in C++ and is designed for
  - Data processing
  - Data analysis
  - Data visualization
  - Data storage
- Widely used in High Energy Physics and other sciences/industry
- Can be used for petabytes/year rates of data
- Provides Python Bindings
- I/O: row-wise, column-wise storage of any C++ object

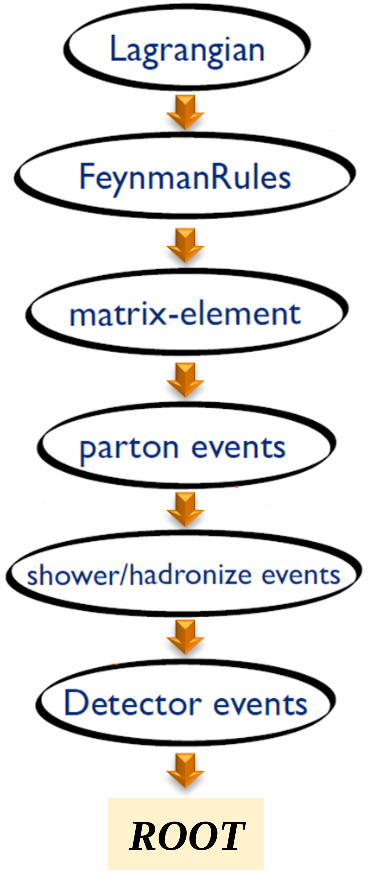


ROOT

[10.1016/S0168-9002(97)00048-X]

- ***Modes of work:***
  - *Interactive (ROOT prompt with CLING interpreter)*
    - *interpretted C++ commands*
    - *Macros : interpretted or (Just In Time) compiled*
  - *As compilable C++ code : using Root libraries*

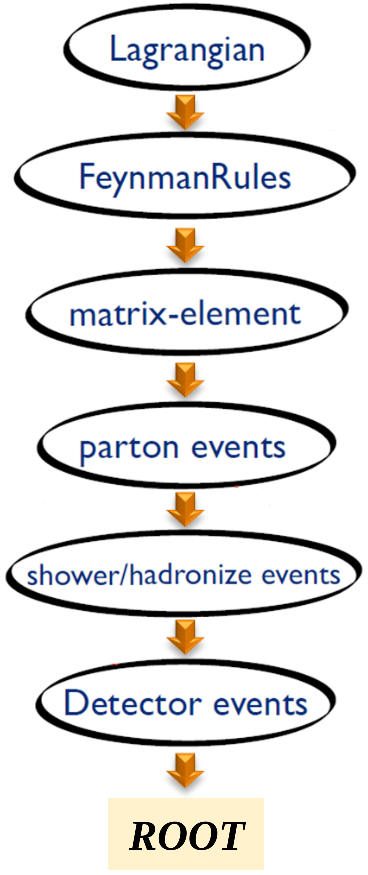
```
[haghighat@SuperMicro10 SR1]$ root
   ------------------------------------------------------------------
  | Welcome to ROOT 6.22/06                        https://root.cern |
  | (c) 1995-2020, The ROOT Team; conception: R. Brun, F. Rademakers |
  | Built for linuxx8664gcc on Nov 27 2020, 15:14:08                 |
  | From tags/v6-22-06@v6-22-06                                      |
  | Try '.help', '.demo', '.license', '.credits', '.quit'/'.q'       |
   ------------------------------------------------------------------

root [0] 
```



ROOT

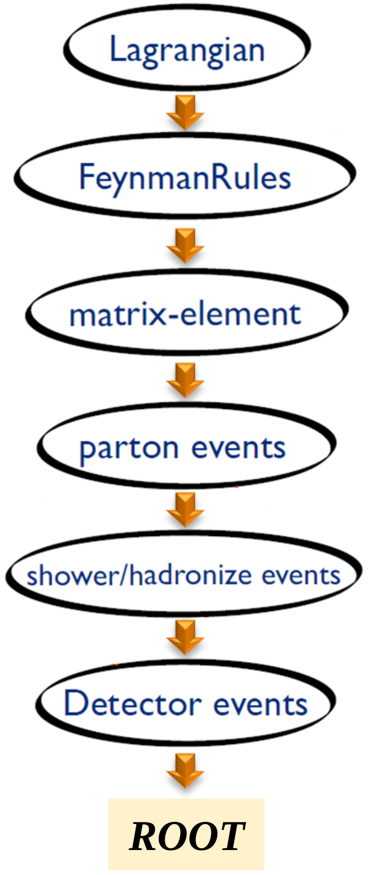
[10.1016/S0168-9002(97)00048-X]

❖ ***Examples of what ROOT provides:***



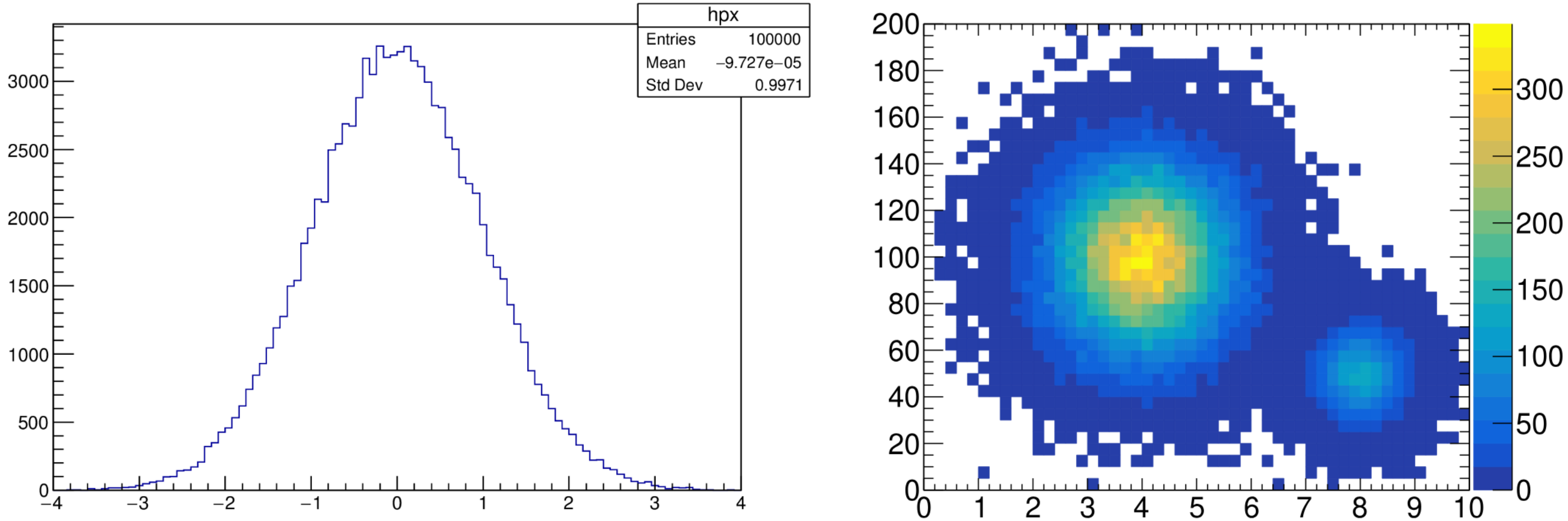
ROOT

[10.1016/S0168-9002(97)00048-X]



- ***Examples of what ROOT provides:***
  - ***Histograms, graphs, trees, ntuples: TH1,TGraph,TTree,TNtuple***

Histograms

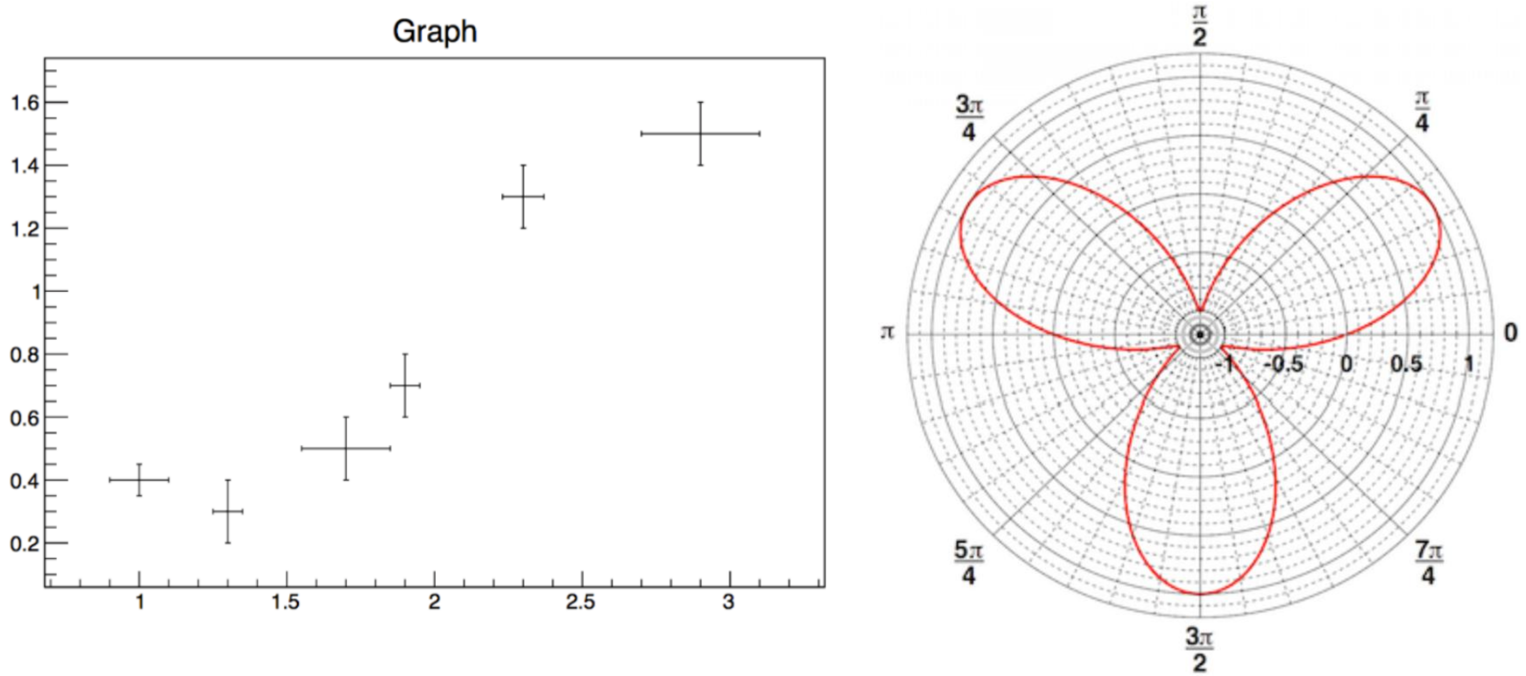


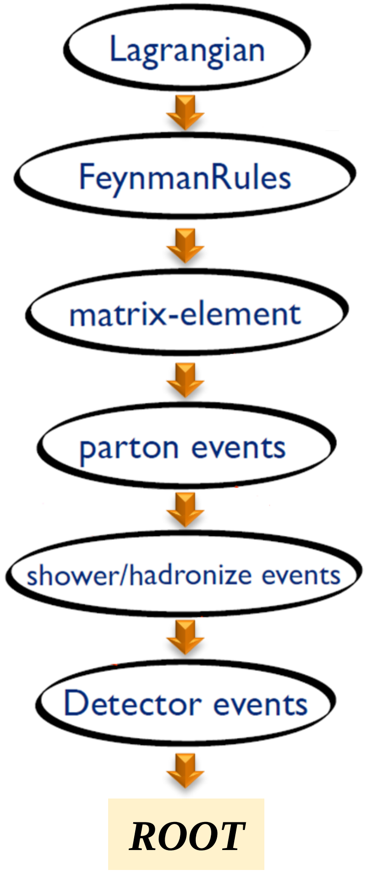
ROOT

[10.1016/S0168-9002(97)00048-X]

- ***Examples of what ROOT provides:***
  - ***Histograms, graphs, trees, ntuples: TH1, TGraph, TTree, TNtuple***

Graphs





ROOT

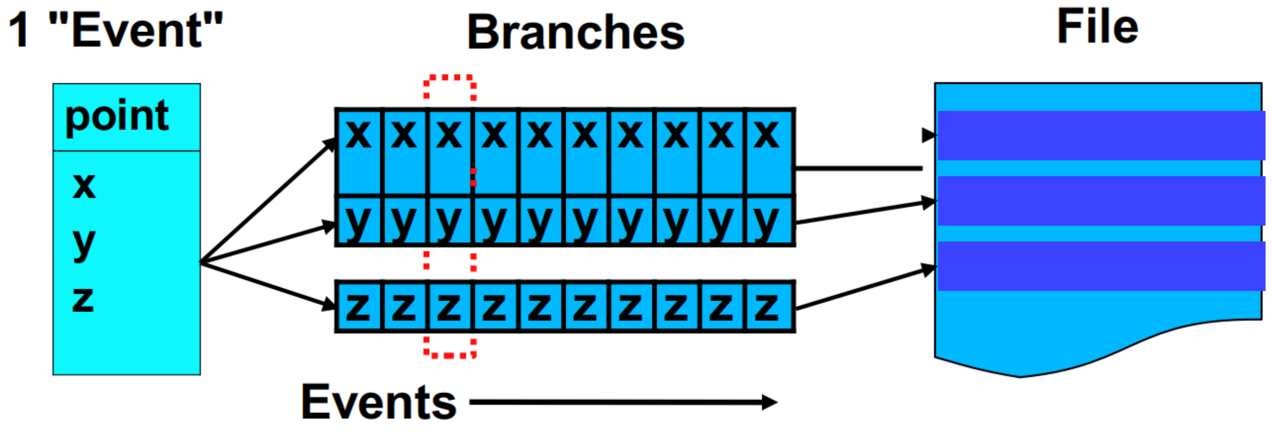
[10.1016/S0168-9002(97)00048-X]

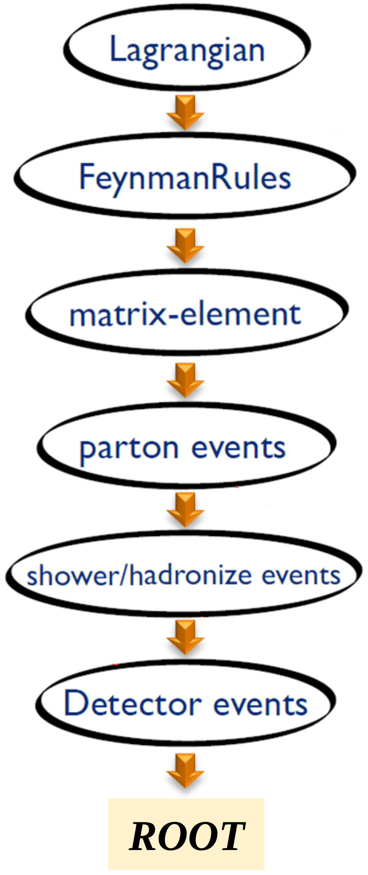
❖ *Examples of what ROOT provides:*

➢ *Histograms, graphs, trees, ntuples: TH1,TGraph,TTree,TNtuple*

Trees

- *Data structure provided to store large quantities of objects*
- *Organized in branches, each one holding objects*
- *Organized in independent events, e.g. collision events*
- *Efficient disk space usage, optimized I/O runtime*

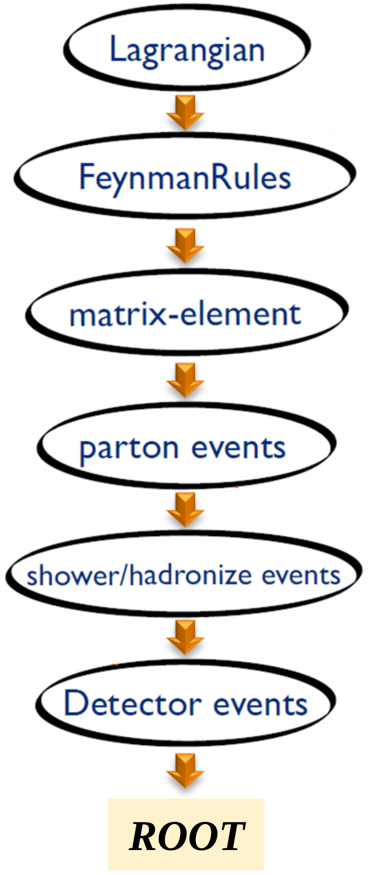




ROOT

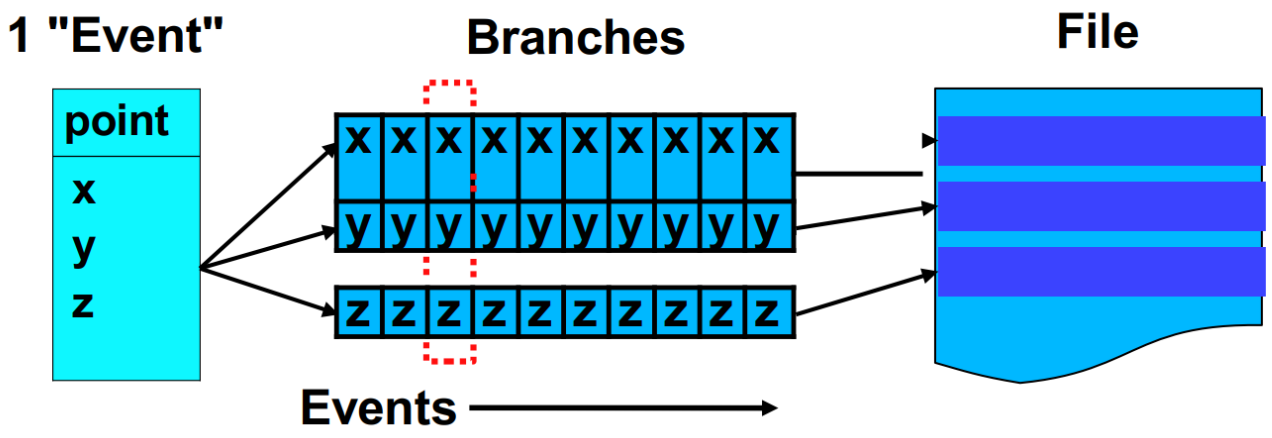
[10.1016/S0168-9002(97)00048-X]

- ***Examples of what ROOT provides:***
  - ***Histograms, graphs, trees, ntuples: TH1, TGraph, TTree, TNtuple***



Ntuples

A simplified version of the TTree: store only floating point numbers

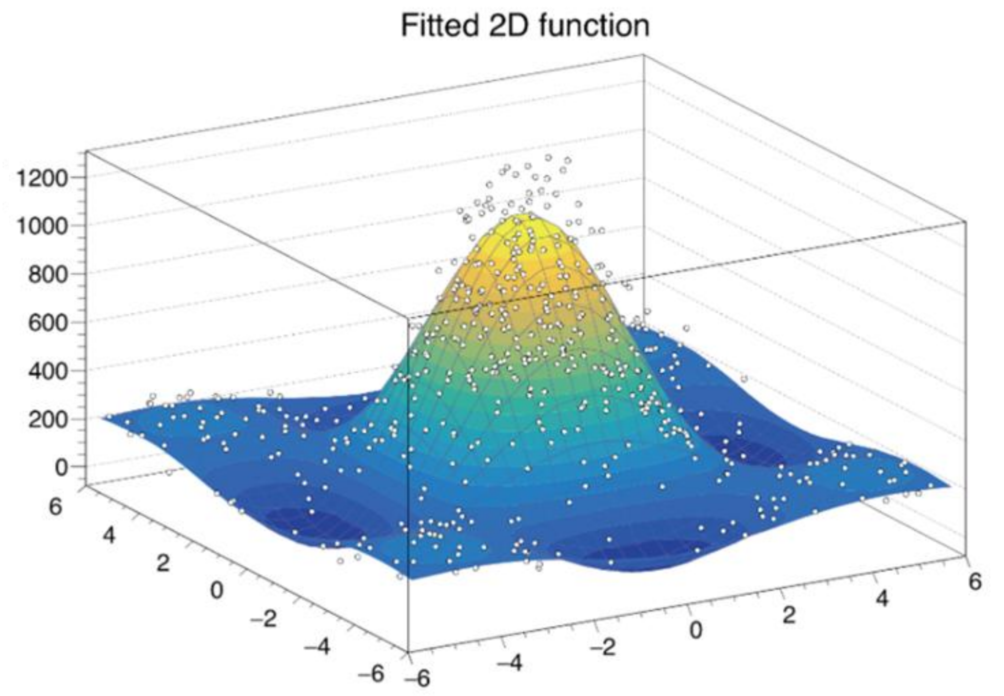


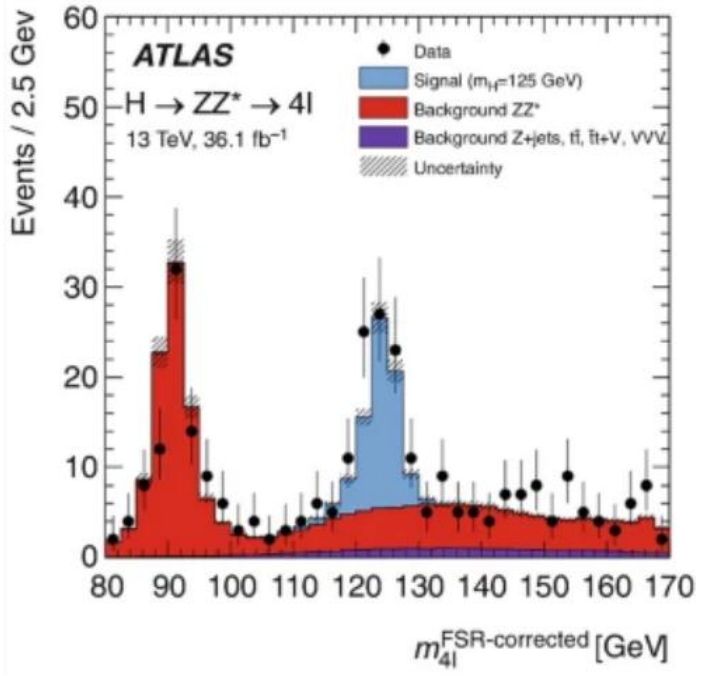
ROOT

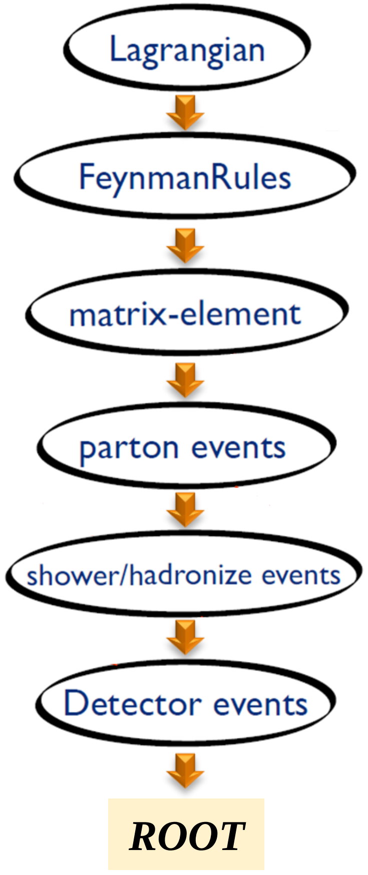
[10.1016/S0168-9002(97)00048-X]

❖ ***Examples of what ROOT provides:***

- ***Histograms, graphs, trees, ntuples: TH1, TGraph, TTree, TNtuple***
- ***Statistical tools: RooFit / RooStats***



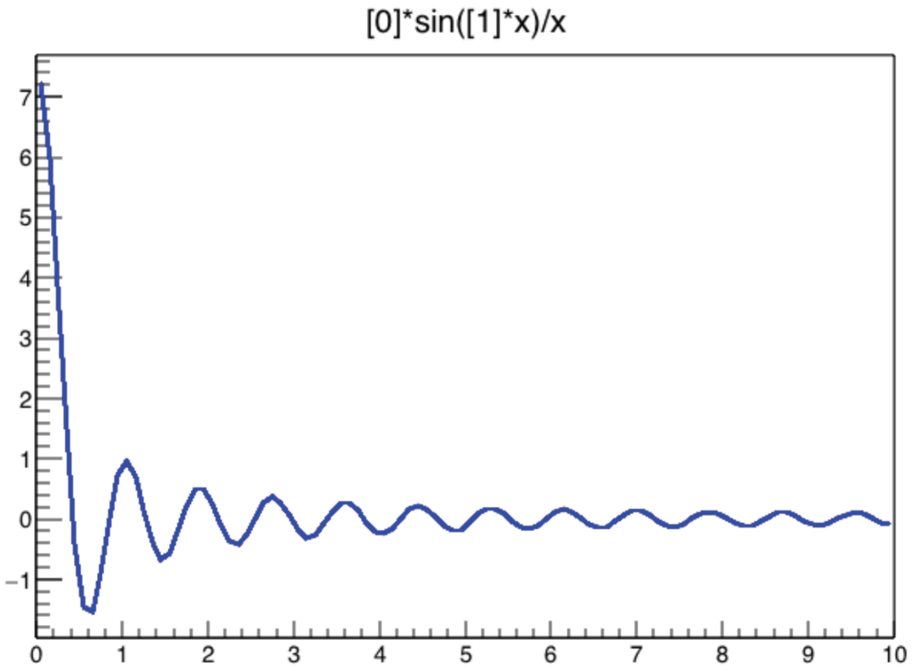


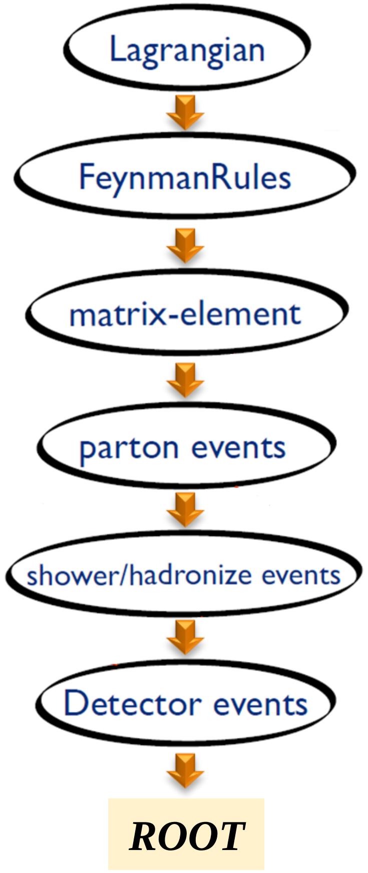


ROOT

[10.1016/S0168-9002(97)00048-X]

- ***Examples of what ROOT provides:***
  - ***Histograms, graphs, trees, ntuples: TH1, TGraph, TTree, TNtuple***
  - ***Statistical tools: RooFit / RooStats***
  - ***A rich collection of functions (also user-defined functions: TF1)***



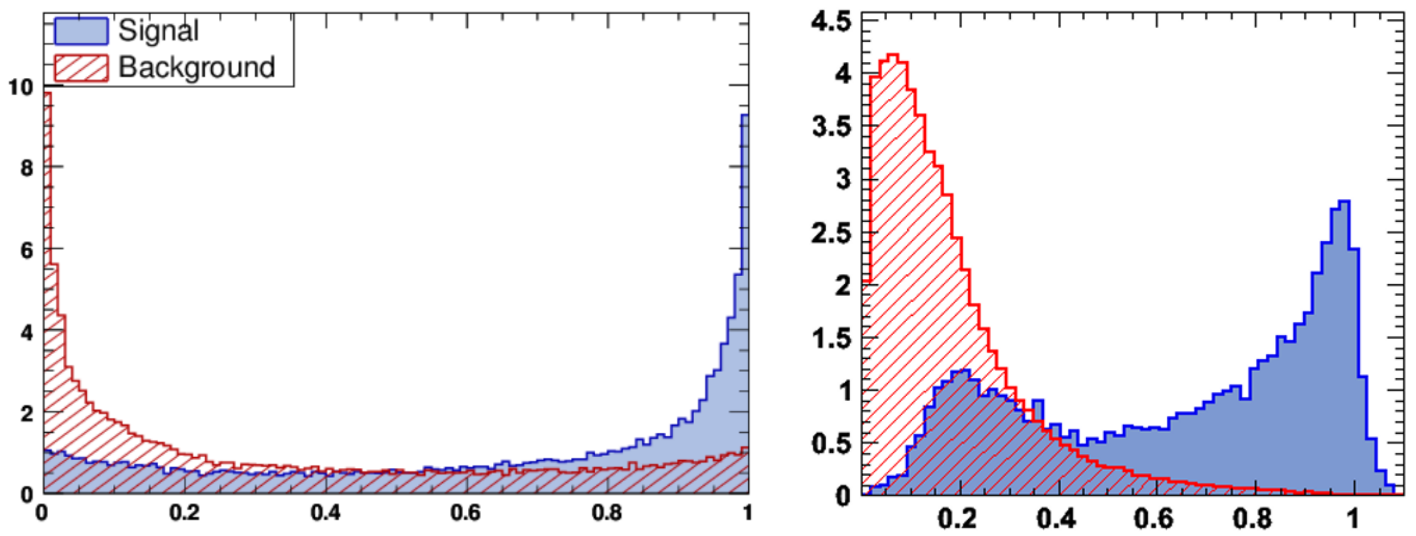


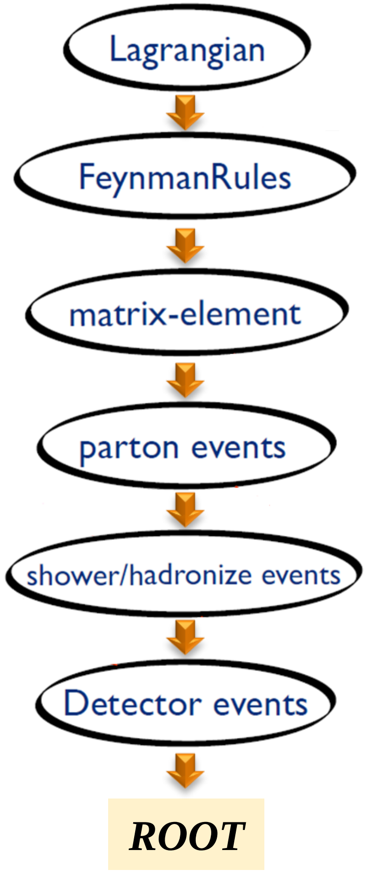
ROOT

[10.1016/S0168-9002(97)00048-X]

❖ ***Examples of what ROOT provides:***

- ***Histograms, graphs, trees, ntuples: TH1,TGraph,TTree,TNtuple***
- ***Statistical tools: RooFit/RooStats***
- ***A rich collection of functions (also user-defined functions: TF1)***
- ***Multivariate Analysis: TMVA (e.g. Boosted decision trees, neural networks)***

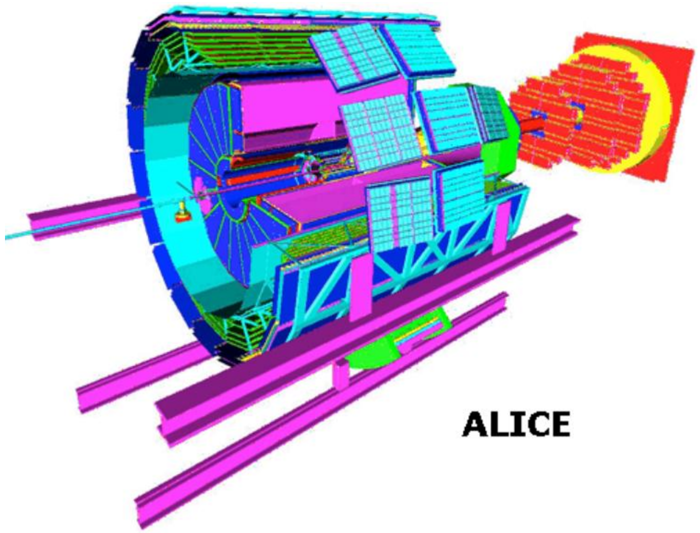


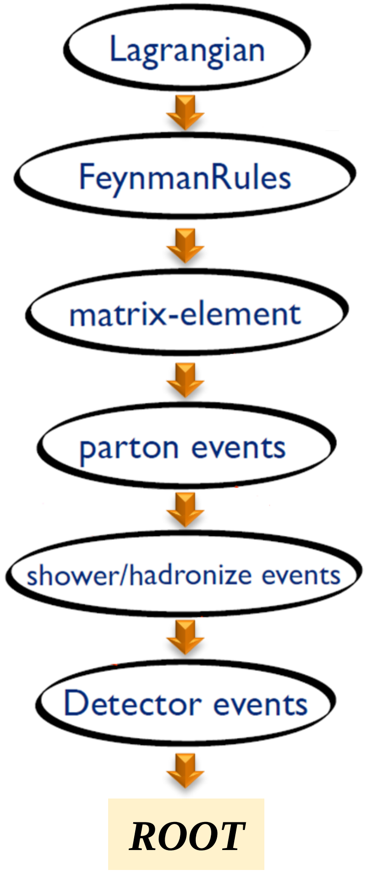


ROOT

[10.1016/S0168-9002(97)00048-X]

- ***Examples of what ROOT provides:***
  - ***Histograms, graphs, trees, ntuples: TH1, TGraph, TTree, TNtuple***
  - ***Statistical tools: RooFit/RooStats***
  - ***A rich collection of functions (also user-defined functions: TF1)***
  - ***Multivariate Analysis: TMVA (e.g. Boosted decision trees, neural networks)***
  - ***Geometry Toolkit: represent geometries as complex as detectors***



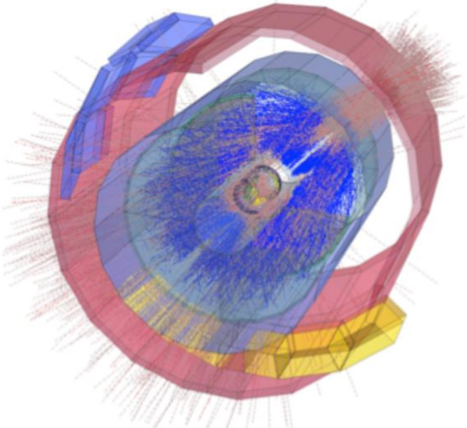


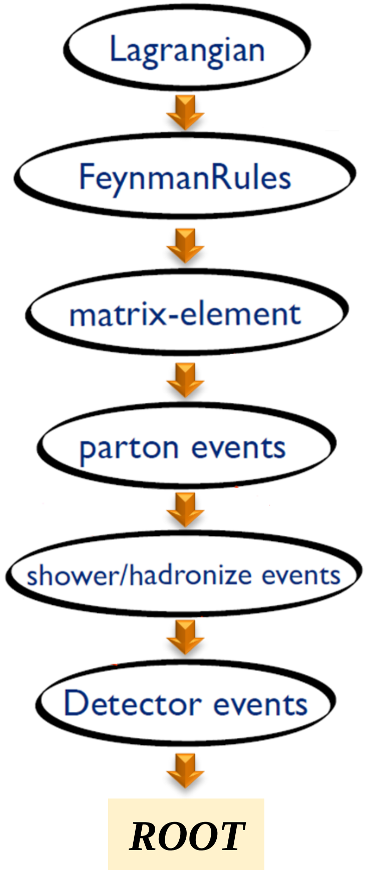
ROOT

[10.1016/S0168-9002(97)00048-X]

❖ ***Examples of what ROOT provides:***

- ***Histograms, graphs, trees, ntuples: TH1, TGraph, TTree, TNtuple***
- ***Statistical tools: RooFit/RooStats***
- ***A rich collection of functions (also user-defined functions: TF1)***
- ***Multivariate Analysis: TMVA (e.g. Boosted decision trees, neural networks)***
- ***Geometry Toolkit: represent geometries as complex as detectors***
- ***Event Display (EVE): visualize particles collisions in detectors***



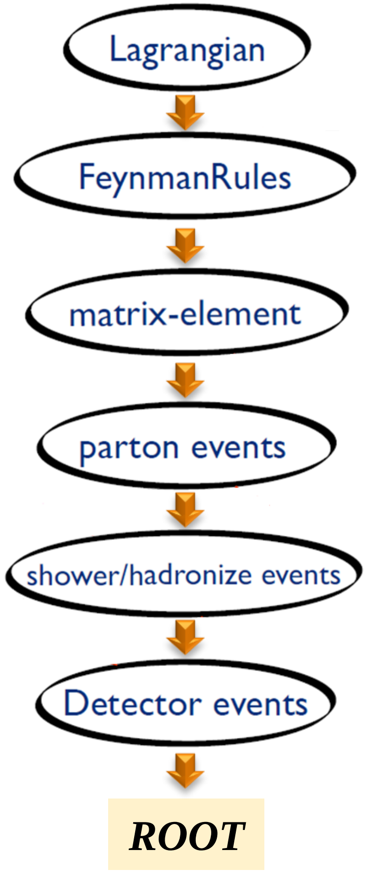


ROOT

[10.1016/S0168-9002(97)00048-X]

❖ ***Examples of what ROOT provides:***

- ***Histograms, graphs, trees, ntuples: TH1,TGraph,TTree,TNtuple***
- ***Statistical tools: RooFit/RooStats***
- ***A rich collection of functions (also user-defined functions: TF1)***
- ***Multivariate Analysis: TMVA (e.g. Boosted decision trees, neural networks)***
- ***Geometry Toolkit: represent geometries as complex as detectors***
- ***Event Display (EVE): visualize particles collisions in detectors***
- ***PyROOT: bindings to interface to Python***

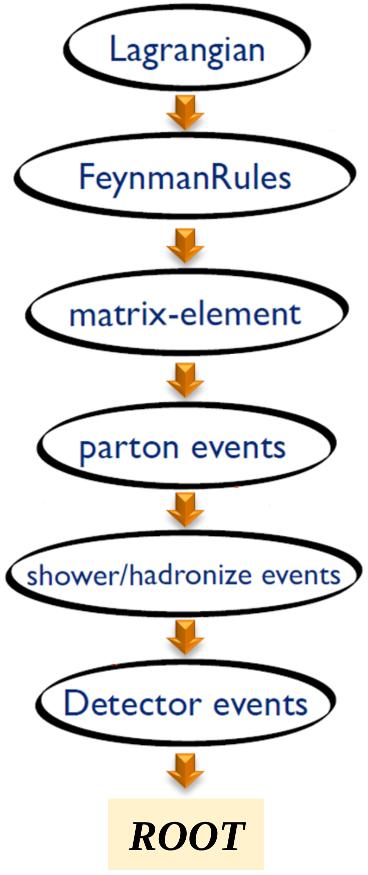


ROOT

[10.1016/S0168-9002(97)00048-X]

❖ *Examples of what ROOT provides:*

- ➢ *Histograms, graphs, trees, ntuples: TH1,TGraph,TTree,TNtuple*
- ➢ *Statistical tools: RooFit/RooStats*
- ➢ *A rich collection of functions (also user-defined functions: TF1)*
- ➢ *Multivariate Analysis: TMVA (e.g. Boosted decision trees, neural networks)*
- ➢ *Geometry Toolkit: represent geometries as complex as detectors*
- ➢ *Event Display (EVE): visualize particles collisions in detectors*
- ➢ *PyROOT: bindings to interface to Python*
- ➢ *PROOF: parallel analysis facility*
  - *Run in parallel on a large number of computers*
  - *Proof-lite: use multiple cores to run on a desktop machine*



We will see how to use the packages in the hands on session

Thank you