Monte Carlo methods in HEP

Tomasz Wlodek

University of the Great State of Texas

What am I going to talk about:

- **What are MC methods?**
- **Why are they useful?**
- **MC in methods in HEP:**
  1. **Monte Carlo for event generation**
  2. **Monte Carlo for detector response simulation**

MC methods - introduction

- **Invented by Stanislaw Ulam, while working on Manhattan project**
- **When invented they were of purely academic interest, there were no computers in those days**
- **With the advent of computers became very useful**
- **Applied in many areas: science, financial world**
- ***The Mathematics of Financial Derivatives : A Student Introduction* *Paul Wilmott*, *Sam Howison*, *J. Dewynne***

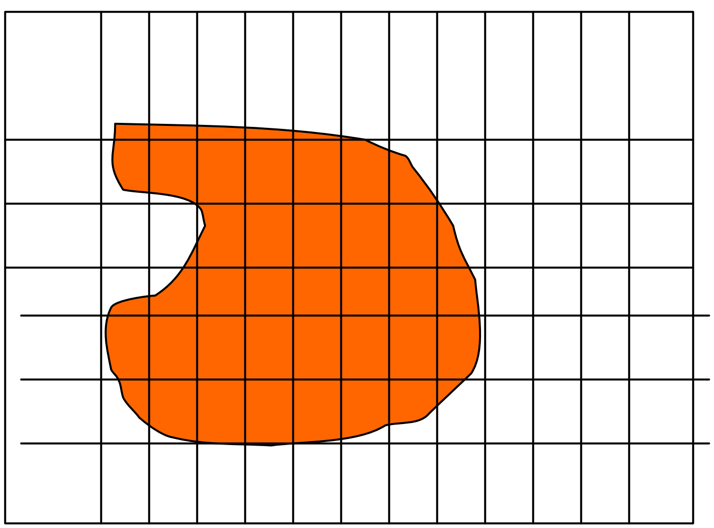
What are Monte Carlo Methods?

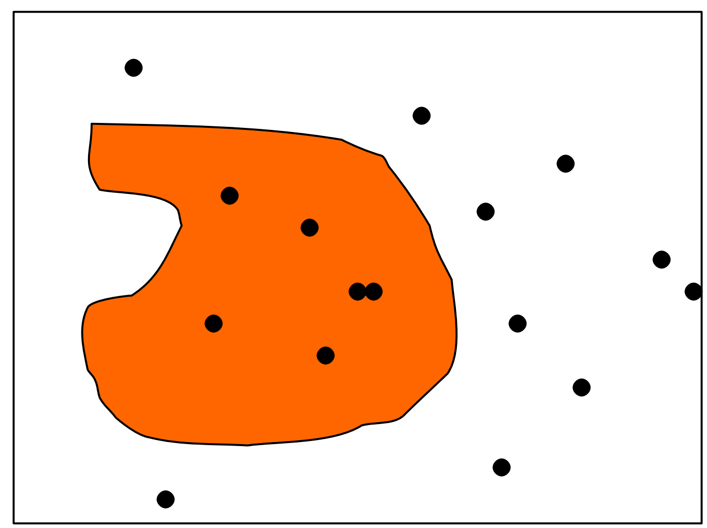
- **Monte Carlo methods are like pornography: no official definition exist, however those who have seen them know what they are about.**
- **General idea is: instead of performing long complex calculations, perform large number of "experiments" using random number generation and see what happens.**

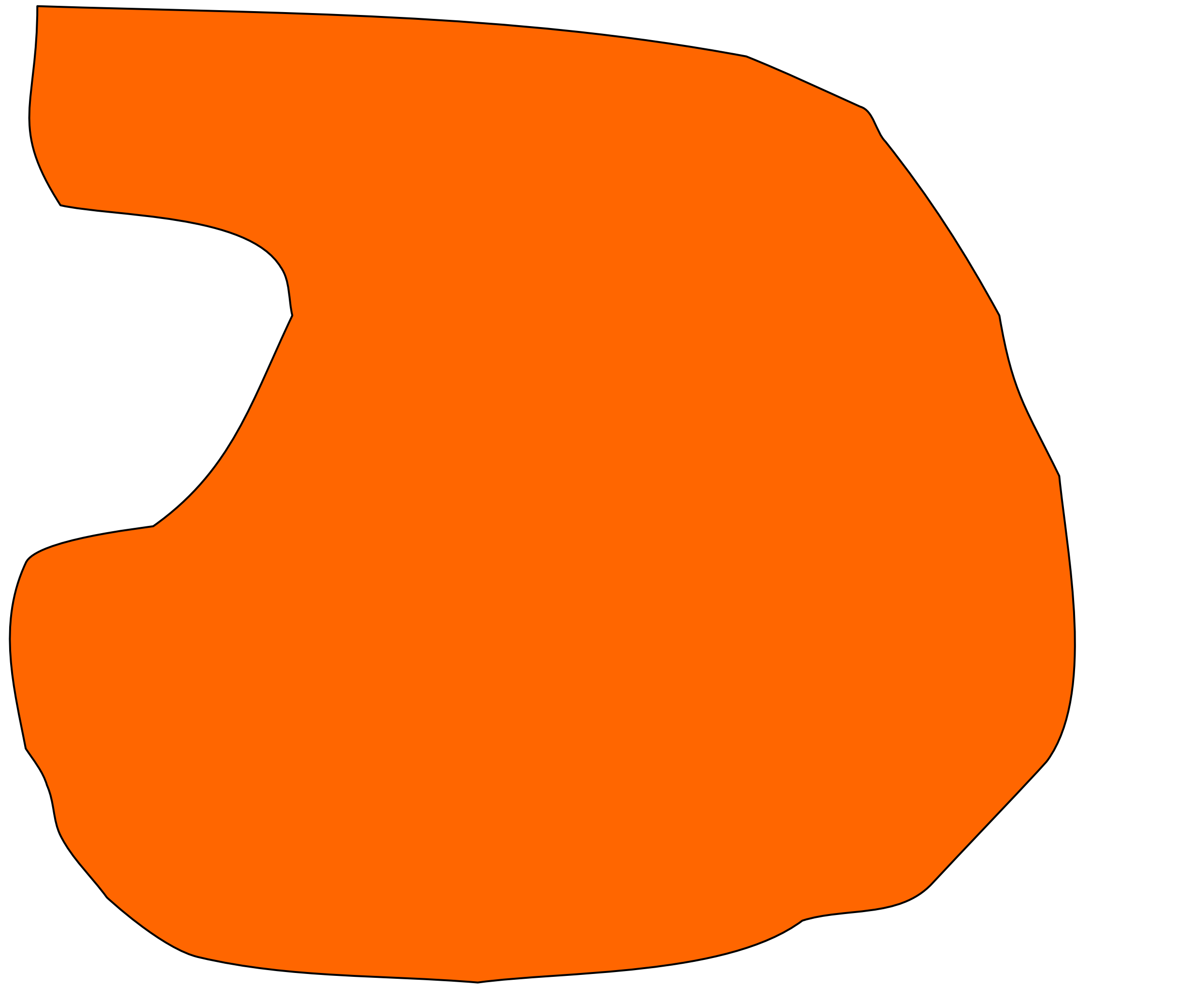
Simplest example: calculate area of a figure

- Cover the figure by a grid, calculate the number of grid cells which are inside and this gives you the area

- Shoot at random at the figure. Count the bullets that hit it. The area of then figure is
- S=(Nhit/Ntotal)*S(rectangle)







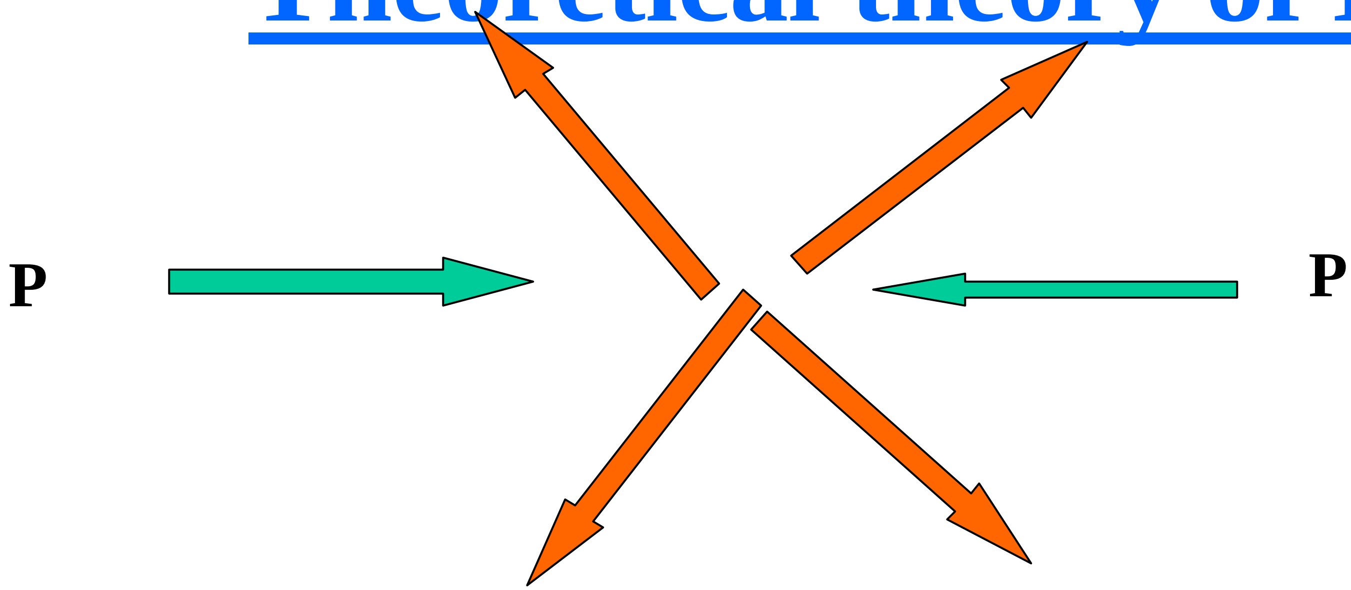
Now, let us do High Energy Physics

- **In HEP we use 2 types of MC methods:**
- **MC for event generation and calculation of process cross sections**
- **MC simulation of detectors**
- **Let us start with event generation**

Event generation and cross section calculations using MC

- **Plan:**
- **Theoretical theory**
- **Practical theory**
- **Practice**
- **Real life**

Theoretical theory of MC



Two particles of 4-momenta P1 and P2 collide.
N final state particles of final states p1,p2,…,pn fly out.
What is the probability that the final states 4-momenta will be in some part of available phase space C?

Reminder from basic probablility theory:

- **Suppose that random variable x has distribution g(x)**
- **What us the probability that x will be between a and b**?

$$P(a < x < b) = \int_a^b g(x)dx$$

Reminder from Field Theory

- **A particle production/decay can be described by matrix element of that particular process**
- **I will not talk here how to calculate matrix elements for particle physics processes. (Read *Itzykson&Zuber, Quantum Field Theory*)**
- **Module of matrix element squared is a measure of probability density for a particular process**

What is the probability of a reaction?

$$s \propto \iint |M(p_1,\ldots,p_n)|^2 \prod dp_i$$

$$\prod d(p_i^2 - m_i^2) d(\sum p_i - E_{tot})$$

This is an integral over 3n-4 dimensional surface in 4n dimensional space

To calculate it we have to parameterize the surface by a 3n-4 cube

$$s \propto \iint |M|^2 \, J(p(x)) \prod dx_i$$

Problem: This is pure theoretical theory

If n (number of final state particles) is high (n>few) this becomes an integral with several dimensions: forget about evaluating is by standard methods.

Here is where MC helps: how can we calculate integrals?

- Suppose you need to evaluate $\int_a^b f(x)\,dx$
- You can approximate it by

$$\sum_{i=0}^{n} f(a+i\cdot\Delta x)\Delta x,\ \Delta x=(b-a)/n$$

Or you can generate random numbers in the interval (a,b)

$$\frac{b-a}{n}\sum_{n} f(\chi_n)$$

Ok, here we are: let us generate randomly <u>uniformly</u> 4 momenta of final state particles p1,…,pn

$$s \propto \sum |M(p_1,\ldots,p_n)|^2$$

If you do not know how to generate the particles uniformly You have to take into account their weight

$$s \propto \sum |M(p_1,\ldots,p_n)|^2 \, J(p_1,\ldots,p_n)$$

How to compute cross sections: theoretical practice:

- **Take matrix elements**
- **Generate randomly and uniformly the momenta of particles in available phase space**
- **Calculate the average value of matrix element squared for those momenta**
- **That’s about it (give or take few details)**
- **The only problem: how to generate final state momenta uniformly?**

How to generate final state momenta uniformly?

- One way: parameterize the final state phase space yourself. It takes some patience and algebra skills, or
- Use an existing program which can generate the phase space distribution for you. (For example code RAMBO by Ronald Kleiss from Amsterdam).

How does a MC generator look like? (practical theory)

- CROSS=0.
- NTRY=100000000
- DO I=1,NTRY
- CALL RAMBO(N,M1,M2,…,Mn,P1,…,PN)
- WEIGHT= MATREL(P1,…,PN)
- CROSS=CROSS+WEIGHT
- IF (RND.GT.WEIGHT/WMAX) PRINT P1,…,PN
- END DO
- PRINT “CROSS SECTION=“,CROSS/NTRY

Now you are experts! You can write a MC generator yourself!

- **Please note that using MC to calculate the cross section does not become much more complicated if the number of particles becomes higher!**
- **The program can print out 4 momenta of final state particles – with the energy, angle distributions corresponding to the momenta of particles produced in real life!**
- **We can then track the particles produced in interaction point through the detector!**

MC event generation: Practical practice, or real life:

- **In real life things are more complex than that and the computer codes are much more elaborate**
- **Number of final states particles is not constant**
- **Phase space is not uniform, there are plenty of resonances**
- **Better leave the job of writing MC generator for the professionals!**

MC generators – real life:

- **We (ie HEP experimentalists) do not write MC generators ourselves**
- **We hire theoreticians to do this for us**
- **Not all theoreticians know how to do it (most have no clue)**
- **In the world there are only a few theoretical groups which specialize in providing us with MC event generators for various applications**
- **We trust in their programs**

LUND University, Sweden

- Mecca of MC generator authors
- Their main products are JETSET and PYTHIA generators
- JETSET simulates jet production and fragmentation and is part of PYTHIA
- PYTHIA – general purpose program which can simulate almost any particle interaction you can imagine

INP, Krakow, Poland

- **A group of theoreticians, which wrote plenty of MC software**
- **Main products:**
- **KORALB,KORALZ,KORALW: simulation of electron-positron anihilations at b,Z and W resonances**
- **TAUOLA: simulation of tau lepton decays**
- **PHOTOS: Initial state radiation from electrons**
- **YFS: Initial and final state radiation in electron-position annihilation to leptons**

Other generators, from different groups:

- **HERWIG – hadronic interactions**
- **SUSYGEN – super symmetric particle production**
- **EXCALIBUR – used to simulate some hadron production, but the author was hired by Shell so the code is not supported anymore**
- **QQ – quark production at low energy electron-positron annihilations**
- **ISAJET – general purpose**

How to use those generators?

- Read manual (can be obtained from CERN)
- Fill some configuration cards
- Run the executable
- Hope for the best
- You should understand what you are doing before you do something.

Homework for students

- **Go to CERN www site**
- **Download PYTHIA manual (you will need it for your thesis so it is worthwhile to have it)**
- **Read the story in preface to PYTHIA manual explaining why PYTHIA generator is called PYTHIA.**

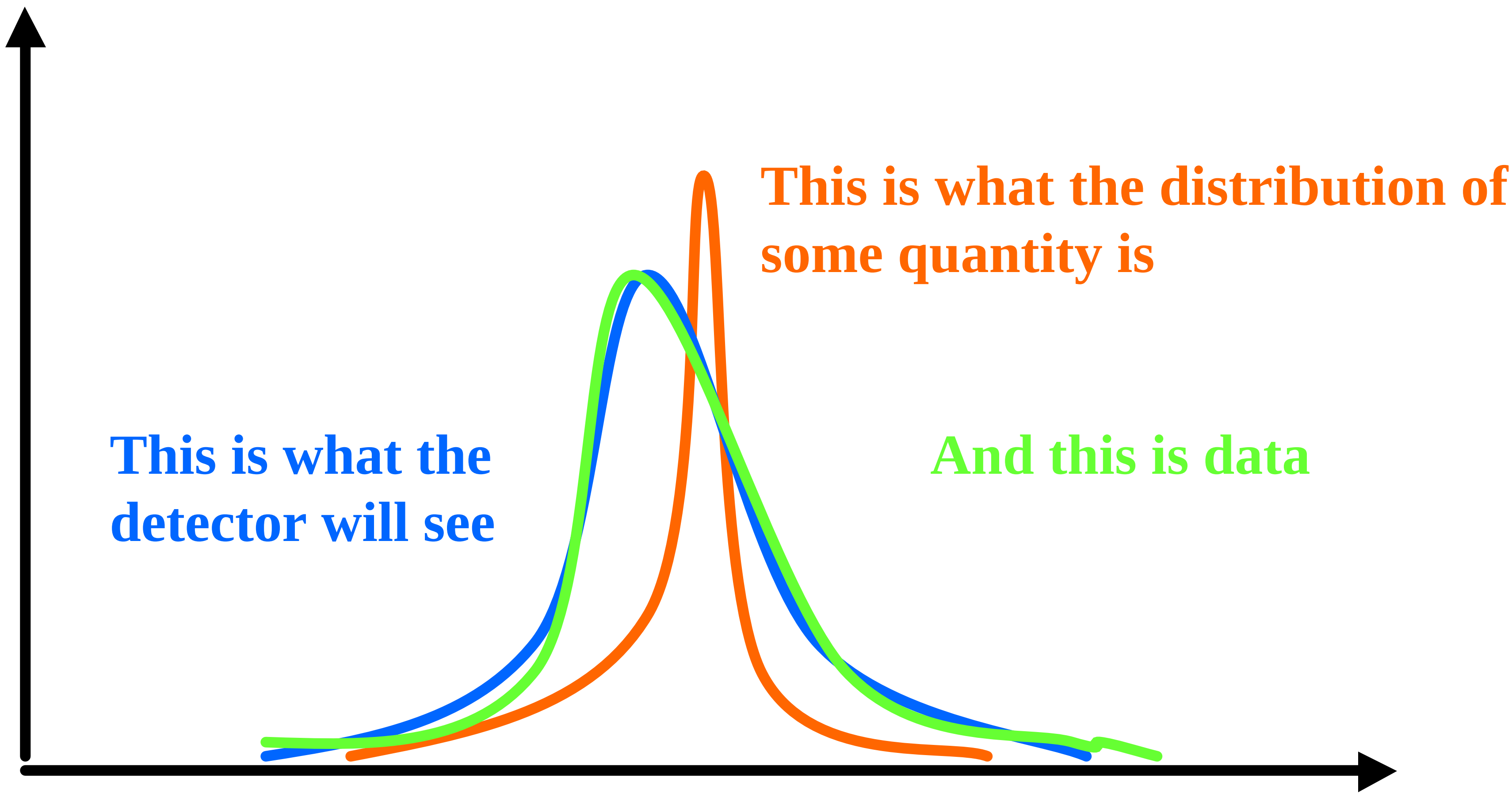
MC event generation - summary

- **Theory:** integrate matrix element over phase space
- **Practice:** use Monte Carlo method for this
- **Real life:** take an off-shelf program provided by theoreticians

PART 2: Monte Carlo for detector simulation

- **Particle production generators tell us what the total cross section for a particular process is**
- **They also deliver four-vectors of momenta of particles produced in interaction**
- **Now we need to know how our detector will see those particles →Monte Carlo detector simulation**

Why we need detector simulation



Detector simulation

- Particle production generators provide input to detector simulation program
- Detector simulation tracks the particles through detector material simulating their interaction with material
- Then it simulates the detector response and produces output “What the detector should see”

Particle tracking

- Take particle of four momentum P
- Calculate, assuming known laws of motion, what its position and four momentum will be after short time dt
- Find out in what kind of environment it is now (vacuum? Gas? Iron? …)
- Depending on where it is now – what can happen to this particle?

Assume that the particle is electron in gas. It can:

- **With probability p1 : ionize the gas, loose some momentum, produce N secondary electrons with momenta P1,…**
- **Do nothing with probability 1-p1**
- **Generate random number r.**
- **if r<p1**

  Generate momenta of secondary electrons, add them to your list of particles to be tracked, reduce the momentum of initial electron

If the particle is photon, it can:

- **Convert and produce electron-positron pair with probability p1**
- **Compton scatter with probability p2**
- **Ionize the matter with probability p3**
- **Generate random number r**
- **If r<p1, convert the electron**
- **If p1<r<p1+p2 generate Compton electron, reduce photon momentum**
- **If p1+p2<r<p1+p2+p3 ionize the matter….**

If the particle is hadron …

- **Simulate its interaction with matter, produce hadronic showers, add them to your list of particles,**

- **And so on! Continue until all your particles leave the detector area, decay or are stopped in material!**

Problem:

- **A single particle (electron for example) can produce a shower of millions of bazylions of secondary particles.**
- **Do I need to track all of them, one by one?**
- **Yes!**
- **This is why particle tracking through detector is VERY CPU intensive.**

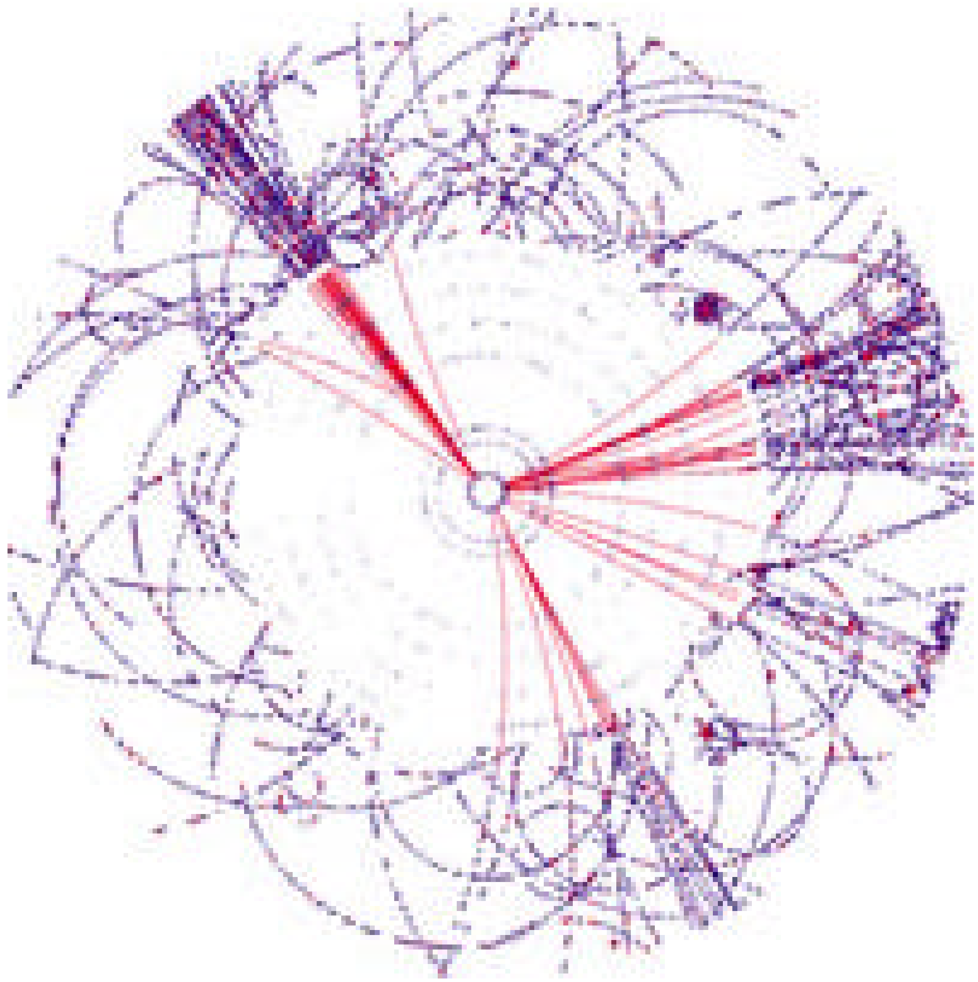
Fast detector simulation

- Particle tracking is very slow
- Which is why people often use “fast” and “full” detector simulation
- Fast simulation is fast, but it gives an approximate description of the detector, it can be used at preliminary stages of analysis
- For final analysis full simulation should be used. (At least in theory)

Detector simulation - practice

- We (HEP experimentalists) do not write detector simulation programs ourselves
- In CERN there is a dedicated group which does this for us.
- They develop GEANT code for simulation of interaction of radiation with matter.
- GEANT started in 1950’s, evolves since
- GEANT 3.* is used now, GEANT 4.* is next.

GEANT event



GEANT for detector simulation

- **You can configure GEANT to describe any particle detector you would like**
- **Just modify some data cards which describe your detector and you can have it simulate your machine.**
- **A small problem: it takes a couple of man-years to convert GEANT into a program which simulates a particular detector.**
- **All particle physics experiments use GEANT based detector simulation**

GEANT for detector simulation

Experiment	Det. Simulation
OPAL	GOPAL
DELPHI	DELSIM
CLEO	CLEOG
D0	D0GSTAR

Event simulation: practice

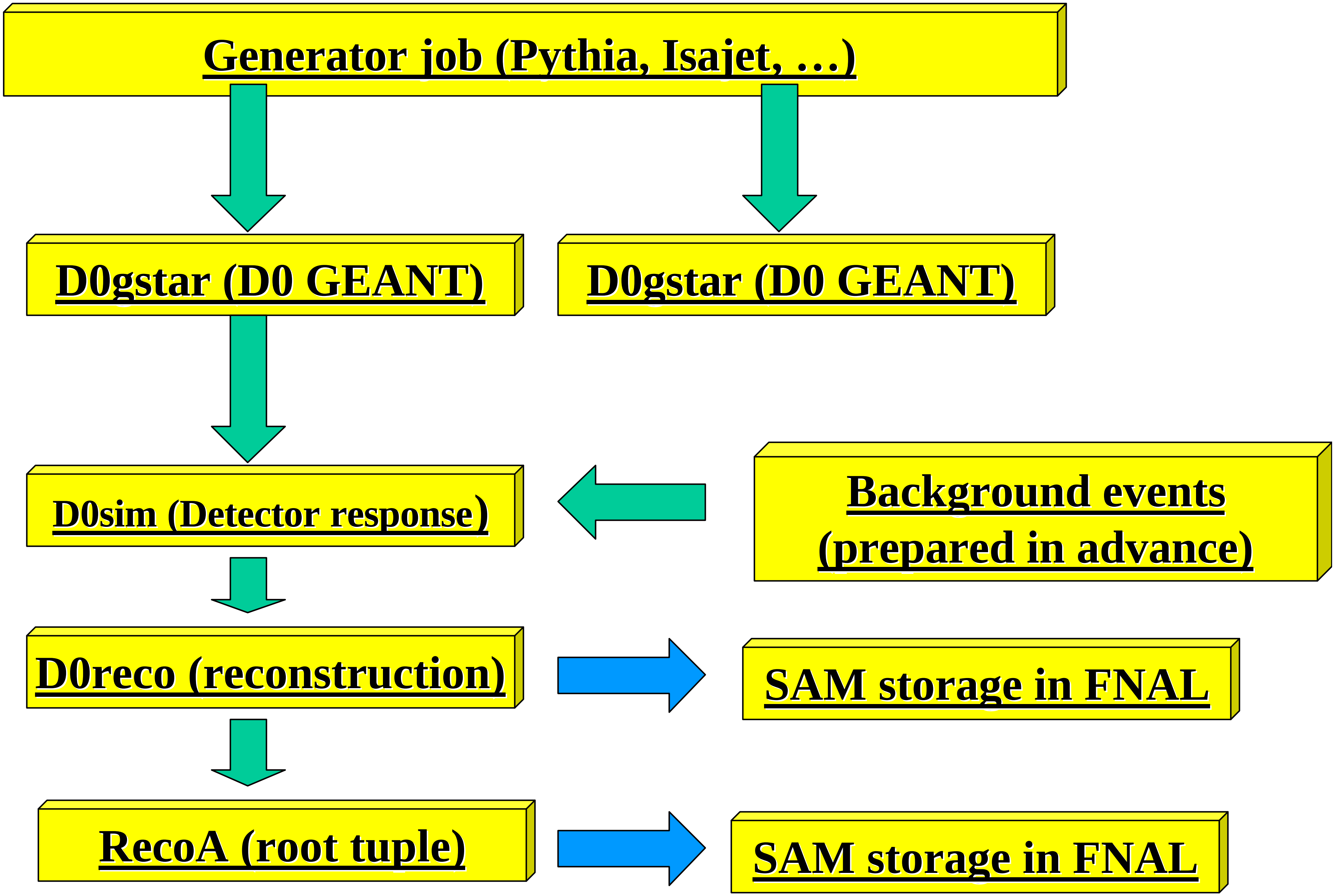
- We run event generators (PYTHIA,…) and produce particles at interaction point
- Then we pass those particles through detector simulation program
- We get response from the detector in the same format as real data
- We compare Monte Carlo simulated data with real data

Monte Carlo production is done at industrial scale!

- Users in experiment need to have their processes simulated
- They submit requests, then dedicated computing centers simulate them
- We produce events continuously, millions of them
- The resulting Monte Carlo data is stored in experiment databases for future analysis.

One of the Monte Carlo production centers for D0 experiment is located here, at UTA

D0 Monte Carlo production chain



Monte Carlo in action

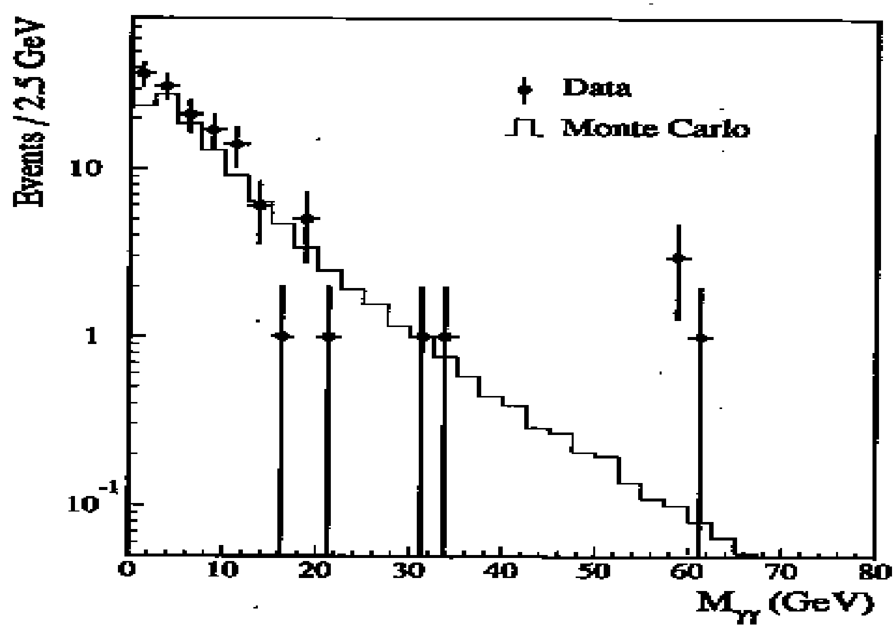


Figure 6

20