QCD and Collider Physics II:
Monte Carlo Methods and event generators

H. Jung (DESY)

- Announcements
  What this lecture aims at, and what it does not ...

http://www-h1.desy.de/~jung/qcd_collider_physics_sose_2005

Outline of the lectures

- Part A

19.4.	Monte Carlo methods and event generators	H. Jung
26.4.	Accelerators	H. Mais
3.5.	PDF fit and error treatment	A. Glazov
10.5.	Heavy Quarks	A. Meyer
17.5.	Jets: algorithms and measurements	G. Grindhammer
24.5.	Diffraction	K. Borras

- Part B

31.5.	High Density Systems I	J. Bartels
7.6.	Pentecost / Whitsundays break	
14.6.	High Density Systems II	J. Bartels
21.6.	High Density Systems III	J. Bartels
28.6.	High Density Systems IV	NN
5.7.	High Density Systems V	NN
12.7.	High Density Systems VI	J. Bartels

Requests to you ...

- If things go wrong .. lecture is too easy... too trivial ... too complicated, too chaotic or too boring ...
- **PLEASE complain immediately !**
- PLEASE ask questions any time !

General literature

- Many new books are available in DESY library NEW ... ask at the desk there ...

- Statistische und numerische Methoden der Datenanalyse V. Blobel & E. Lohrmann
- STATISTICAL DATA ANALYSIS. *Glen Cowan*.
- Particle Data Book S. Eidelman et al., Physics Letters B592, 1 (2004) (http://pdg.lbl.gov/)
- Applications of pQCD R.D. Field Addison-Wesley 1989
- Collider Physics *V.D. Barger & R.J.N. Phillips* Addison-Wesley 1987
- Deep Inelastic Scattering. *R. Devenish & A. Cooper-Sarkar*, Oxford 2
- Handbook of pQCD *G. Sterman et al*
- Quarks and Leptons, *F. Halzen & A.D. Martin*, J.Wiley 1984
- QCD and collider physics *R.K. Ellis & W.J. Stirling & B.R. Webber* Cambridge 1996
- QCD: High energy experiments and theory G. Dissertori, I. Knowles, M. Schmelling Oxford 2003

Simulations in High Energy Physics

- Simulation:

Simulations in High Energy Physics

- Simulation:
  Oxford advanced dictionary: simulate = pretend to be
- Simulation: why ?

Simulations in High Energy Physics

- Simulation:
  - Oxford advanced dictionary: simulate = pretend to be
- Simulation: why ?
  - Can't we just calculate things ????
-

Simulations in High Energy Physics

- Simulation:
  Oxford advanced dictionary: simulate = pretend to be
- Simulation: why ?
  Can't we just calculate things ????
- Simulation: what ?

Simulations in High Energy Physics

- Simulation:
  Oxford advanced dictionary: simulate = pretend to be
- Simulation: why ?
  Can't we just calculate things ????
- Simulation: what ?
  Detector response
  Particle decays
  ep, e^+ e^-, *pp* interactions
  Economy
  Life

Simulations in High Energy Physics

- Simulation:
  - Oxford advanced dictionary: simulate = pretend to be
- Simulation: why ?
  - Can't we just calculate things ????
- Simulation: what ?
  - Detector response
  - Particle decays
  - *ep*, $e^+ e^-$, *pp* interactions
  - Economy
  - Life
- Simulation: How-to ?

Simulations in High Energy Physics

- Simulation:
  - Oxford advanced dictionary: simulate = pretend to be
- Simulation: why ?
  - Can't we just calculate things ????
- Simulation: what ?
  - Detector response
  - Particle decays
  - *ep*, $e^+ e^-$, *pp* interactions
  - Economy
  - Life
- Simulation: How-to ?
  - apply Monte Carlo technique:
  - solve complicated integrals
  - simulate complicated processes

Application in Economy

What is monte carlo simulation? montecarlo analysis? http://www.decisioneering.com/monte-carlo-simulation.html

DECISIONEERING
Crystal Ball Risk Analysis Software & Solutions

Search Site
Find

Risk Resources | Products | Training | Consulting | Support | How to Buy | News & Company

Information On
Six Sigma & DFSS
Industries & Applications
Getting Started on Our Site
Training Classes
Academic Program
User Conference

Quick Links
Shop
Download
Newsletters
Contact Us

Worldwide Offices
United States
United Kingdom
Germany

RISK ANALYSIS OVERVIEW

WHAT IS MONTE CARLO SIMULATION?

What do we mean by "simulation?"

When we use the word ***simulation***, we refer to any analytical method meant to imitate a real-life system, especially when other analyses are too mathematically complex or too difficult to reproduce.

Without the aid of simulation, a spreadsheet model will only reveal a single outcome, generally the most likely or average scenario. Spreadsheet risk analysis uses both a spreadsheet model and simulation to automatically analyze the effect of varying inputs on outputs of the modeled system.

> One type of spreadsheet simulation is **Monte Carlo simulation**, which randomly generates values for uncertain variables over and over to simulate a model.

Overview Start
What is Risk?
What is a Model?
Traditional Risk Analysis
Spreadsheet Risk Analysis
Monte Carlo Simulation
Analysis of Results
Benefits of Risk Analysis
Optimization
Time-series Forecasting

How did Monte Carlo simulation get its name?

Monte Carlo simulation was named for Monte Carlo, Monaco, where the primary attractions are casinos containing games of chance. Games of chance such as roulette wheels, dice, and slot machines, exhibit random behavior.

The random behavior in games of chance is similar to how Monte Carlo simulation selects variable values at random to simulate a model. When you roll a die, you know that either a 1, 2, 3, 4, 5, or 6 will come up, but you don't know which for any particular roll. It's the same with the variables that have a known range of values but an uncertain value for any particular time or event (e.g. interest rates, staffing needs, stock prices, inventory, phone calls per minute).

Application in Nuclear Waste ...

Applied Intelligence: The Use of Monte Carlo Simulation...http://www.applied-intelligence.co.uk/Papers/Supercon

Home | Company | Technologies | Clients | Projects | Links | Associates | Contact

Applied Intelligence

Business intelligence through knowledge technology

Case Study: The Use of Monte Carlo Simulation to Optimise the Supercompaction Process at the Waste Treatment Complex, Sellafield

First published in *Unicom seminar on AI and Optimisation in Process Control* (Heathrow) June 1996

ABSTRACT

Mathematical modelling and Monte Carlo simulation have been used to model the supercompaction process at WTC, BNFL Sellafield. A better understanding of the process was achieved, and the algorithm initially specified to select drums for compression was found to hav some surprising and undesirable effects. The application of statistical decision theory allowed the development and testing of improved algorithms, which should result in major operational cost savings.

Literature & References

- F. James *Rep. Prog. Phys., Vol 43, 1145 (1980)*
- *Glen Cowan* STATISTICAL DATA ANALYSIS. Clarendon, 1998.
- Particle Data Book S. Eidelman et al., Physics Letters B592, 1 (2004) section on: Mathematical Tools (http://pdg.lbl.gov/)
- Michael J. Hurben *Buffons Needle* (http://www.angelfire.com/wa/hurben/buff.html)
- J. Woller (Univ. of Nebraska-Lincoln) B*asics of Monte Carlo Simulations* (http://www.chem.unl.edu/zeng/joy/mclab/mcintro.html)
- Hardware Random Number Generators:
  A Fast and Compact Quantum Random Number Generator (http://arxiv.org/abs/quant-ph/9912118)
  *Quantum Random Number Generato*r (http://www.idquantique.com/products/quantis.htm)
  Hardware random number generator (http://en.wikipedia.org/wiki/)
- Monte Carlo Tutorals (http://www.cooper.edu/engineering/chemechem/MMC/tutor.html)
- History of Monte Carlo Method (http://www.geocities.com/CollegePark/Quad/2435/history.html)
- Google: search for Monte Carlo Simulations

Literature & References (cont'd)

- T. Sjostrand et al
  PYTHIA/JETSET manual - The Lund Monte Carlos
  http://www.thep.lu.se/tf2/staff/torbjorn/Pythia.html
- H. Jung
  RAPGAP manual
  http://www-h1.desy.de/~jung/rapgap.html
  CASCADE manual
  http://www-h1.desy.de/~jung/cascade.html
- V. Barger and R. J.N. Phillips
  Collider Physics
  Addison-Wesley Publishing Comp. (1987)
- R.K. Ellis, W.J. Stirling and B.R. Webber
  QCD and collider physics
  Cambridge University Press (1996)

Monte Carlo method

- Monte Carlo method
  - refers to any procedure that makes use of random numbers
  - uses probability statistics to solve the problem
- Monte Carlo methods are used in:
  - Simulation of natural phenomena
  - Simulation of experimental apparatus
  - Numerical analysis
- Random number:

Monte Carlo method

- Monte Carlo method
  - refers to any procedure that makes use of random numbers
  - uses probability statistics to solve the problem
- Monte Carlo methods are used in:
  - Simulation of natural phenomena
  - Simulation of experimental apparatus
  - Numerical analysis
- Random number:

  one of them is 3

Monte Carlo method

- Monte Carlo method
  - refers to any procedure that makes use of random numbers
  - uses probability statistics to solve the problem
- Monte Carlo methods are used in:
  - Simulation of natural phenomena
  - Simulation of experimental apparatus
  - Numerical analysis
- Random number:

  one of them is 3

  No such thing as a single random number

Monte Carlo method

- Monte Carlo method
  - refers to any procedure that makes use of random numbers
  - uses probability statistics to solve the problem
- Monte Carlo methods are used in:
  - Simulation of natural phenomena
  - Simulation of experimental apparatus
  - Numerical analysis
- Random number:

  one of them is 3

  No such thing as a single random number

  A sequence of random numbers is a set of numbers that have nothing to do with the other numbers in a sequence

Going out to Monte Carlo

- Obtain true Random Numbers from Casino in Monte Carlo
- Puhhh... Going out every night ...

Random Numbers

- In a uniform distribution of random numbers in [0,1] every number has the same chance of showing up
- Note that 0.000000001 is just as likely as 0.5

To obtain random numbers:

- Use some chaotic system like roulette, lotto, 6-49, ...
- Use a process, inherently random, like radioactive decay
- Tables of a few million truly random numbers exist .....

(.....until a few years ago.....)

BUT not enough for most applications

- Hooking up a random machine to a computer is NOT toooooo good, as it leads to irreproducible results, making debugging difficult....
- **Develop Pseudo Random Number generators !!!!**

Random Numbers

- In a uniform distribution of random numbers in [0,1] every number has the same chance of showing up
- Not that 0.000000001 is just as likely as 0.5

To obtain random numbers:

- Use some chaotic system like roulette, lotto, 6-49, ...
- Use a process, inherently random, like radioactive decay
- Tables of a few million truly random numbers exist .....
  BUT not enough for most applications
- Hooking up a random machine to a computer is NOT toooooo good, as it leads to irreproducible results, making debugging difficult....
- **Develop Pseudo Random Number generators !!!!**

Pseudo means: Oxford Advanced Dict.: **False**
Quasi means: Oxford Advanced Dict.: **almost**
BUT here the meaning is different

Pseudo Random Numbers

Pseudo Random Numbers

- are a sequence of numbers generated by a computer algorithm, usually uniform in the range [0,1]
- more precisely: algo's generate integers between *0* and *M*, and then $r_n = I_n / M$
- A very early example: Middles Square (John van Neumann, 1946):
  generate a sequence, start with a number of 10 digits, square it, then take the middle 10 digits from the answer, as the next number etc.:
  $5772156649^2 = 33317792380594909291$
  Hmmmm, sequence is not random, since each number is determined from the previous, but it appears to be random
- this algorithm has problems .....
  BUT a more complex algo does not necessarily lead to better random sequences ....
  Better us an algo that is well understood

Quasi Random Numbers

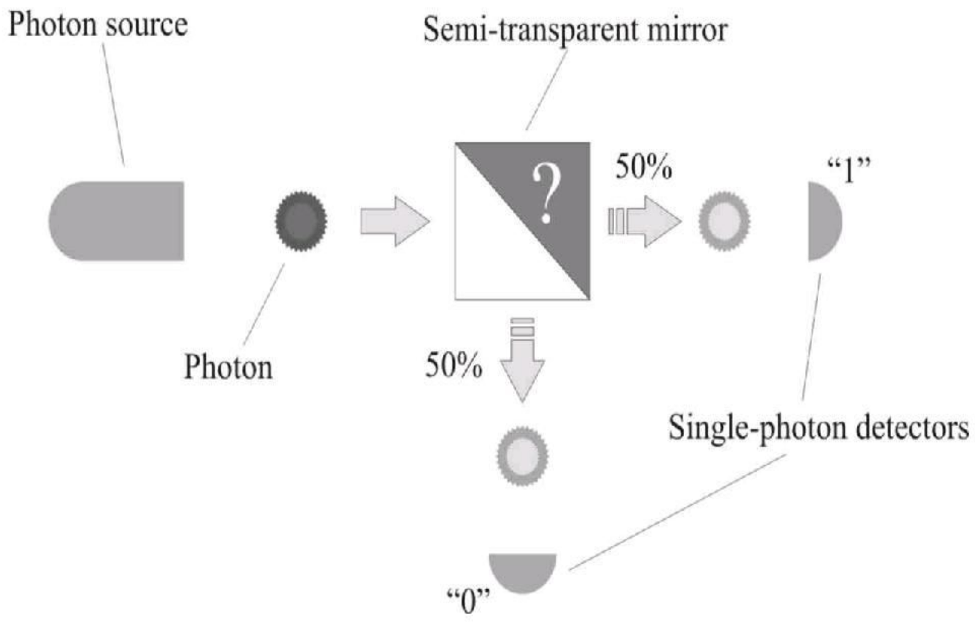
- mathematical randomness is not attainable in computer generated random numbers
- more important: assure that the “random” sequence has the necessary properties to produce a desired result ... (hmmmm !!!)
  - examples:
    - in multidimensional integration, each multi-dim point is considered independently of the others, and the order in which they appear plays no role !
    - degree of fluctuations about uniformity: in many cases a “super-uniform” distribution is more desirable than a truly random distribution with uniform probability density !
- use of Quasi Random Numbers might lead to faster convergence of the integration .... but needs to be checked carefully ...

Important in
Monte Carlo integrations

True Random Numbers

- Random numbers from classical physics: coin tossing
  evolution of such a system can be predicted, once the initial condition is known... however it is a chaotic process ... extremely sensitive to initial conditions.
- Truly Random numbers used for
  - Cryptography
    Confidentiality
    Authentication
  - Scientific Calculation
  - Lotteries and gambling
    do not allow to increase chance of winning by having a bias .... too bad

- Random numbers from quantum physics: intrinsic random photons on a semi-transparent mirror



- Available and tested in MC generator by last years summer student
- Generator is however very slow...

Pseudo Random Numbers

Pseudo Random Numbers

- are a sequence of numbers generated by a computer algorithm, usually uniform in the range [0,1]
- more precisely: algo's generate integers between *0* and *M*, and then $r_n = I_n / M$
- A very early example: Middles Square (John van Neumann, 1946):
  generate a sequence, start with a number of 10 digits, square it, then take the middle 10 digits from the answer, as the next number etc.:
  $5772156649^2 = 33317792380594909291$
  Hmmmm, sequence is not random, since each number is determined from the previous, but it appears to be random
- this algorithm has problems .....
  BUT a more complex algo does not necessarily lead to better random sequences ....
  Better us an algo that is well understood

From now on assume:
we have good random number generator

Simulating Radioactive Decay

- radioactive decay is a truly random process
- $dN = - N \alpha\, dt$ i.e. $N = N_0\, e^{-\alpha t}$
- probability of decay is constant ... independent of the age of the nuclei: probability that nucleus undergoes radioactive decay in time Δt is p:
  $p = \alpha\, \Delta t$ (for $\alpha\, \Delta t \ll 1$)
- Problem:
  consider a system initially having N_0 unstable nuclei.
  How does the number of parent nuclei, N, change with time ?
- Algorithm:

```
LOOP from t=0 to t, step Δt
  LOOP over each remaining parent nucleus
      Decide if nucleus decays:
      IF ( random # < αΔt ) then
             reduce number of parents by 1
      ENDIF
  END LOOP over nuclei
  Plot or record N vrs t
END LOOP over time
END
```

The first simulation: radioactive decay

- implement algo into a small program
- show results after 3000 sec for:

  $N_0 = 100,\ \alpha = 0.01\ s^{-1}$

  $\Delta t = 1s$

  $N_0 = 5000,\ a = 0.03\ s^{-1}$

  $\Delta t = 1s$
- algo:

```
   alpha1 = 0.01
   N01 = 100
   deltat = 1
   do I=1,300
      it = it + 1
      do j = 1, N01
         x = RN1
         fr = deltat*alpha1
         if(x.lt.fr) then
c    reduce number of parents N01
            N01 = N01 - 1
         endif
c    fill for each time it number N01
       call hfill(400,real(it+0.3),0,1.) !
       enddo
```

The first simulation: radioactive decay

- implement algo into a small program
- show results after 3000 sec for:

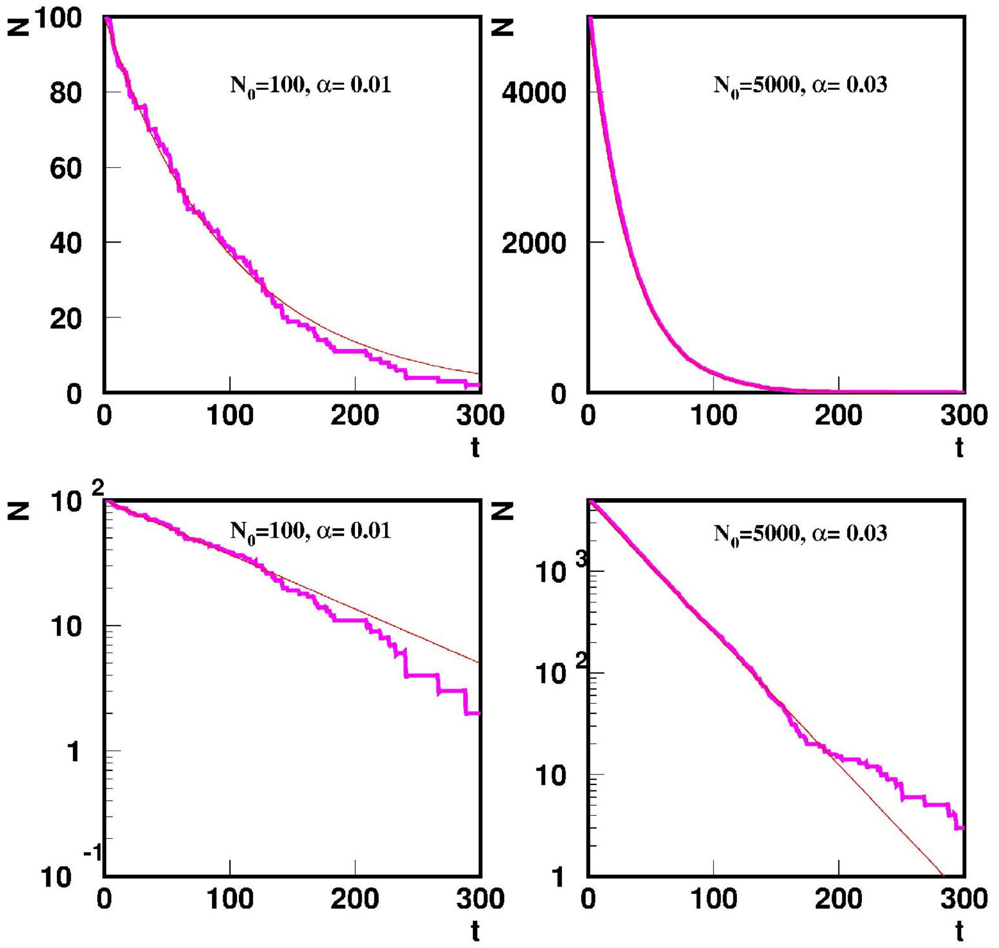
  $N_0 = 100, \alpha = 0.01\ s^{-1}$

  $\Delta t = 1s$

  $N_0 = 5000, a = 0.03\ s^{-1}$

  $\Delta t = 1s$

- MC experiment does not excatly reproduce theory ....
- results from MC experiment show statistical fluctuations ...
- ....as expected ....



Monte Carlo technique: basics

- **Law of large numbers**

chose *N* numbers u_i *randomly*, with probability density uniform in *[a,b]*, evaluate *$f(u_i)$ for each u_i* :

$$\frac{1}{N}\sum_{i=1}^{N} f(u_i) \rightarrow \frac{1}{b-a}\int_a^b f(u)du$$

for large enough N Monte Carlo estimate of integral converges to correct answer.

- Convergence

is given with a certain probability ...

THIS is a mathematically serious and precise statement !!!!

Expectation values and variance

- Expectation value (defined as the average or mean value of function *f*):

$$E[f] = \int f(u) dG(u) = \left(\frac{1}{b-a} \int_a^b f(u) du \right) = \frac{1}{N} \sum_{i=1}^{N} f(u_i)$$

for uniformly distributed *u* in *[a,b]* *then* $dG(u) = du/(b-a)$

- rules for expectation values:

$$E[cx+y] = cE[x] + E[y]$$

- Variance

$$V[f] = \int (f - E[f])^2 \, dG = \left(\frac{1}{b-a} \int_a^b (f(u) - E[f])^2 \, du \right)$$

- rules for variance:

if x,y uncorrelated: $V[cx+y] = c^2 V[x] + V[y]$

if x,y correlated

$$V[cx+y] = c^2 V[x] + V[y] + 2cE\left[(y - E[y]) (x - E[x]) \right]$$

Central Limit Theorem

- Central Limit Theorem
  for large N the sum of independent random variables is always normally (Gaussian) distributed:

$$f(x) = \frac{1}{s\sqrt{2\pi}} \exp\left[-\frac{(x-a)^2}{2s^2}\right]$$

- example: take sum of uniformly distributed random numbers:

$$R_n = \sum_{i=1}^{n} R_i$$
$$E[R_1] = \int u du = 1/2,$$
$$V[R_1] = \int (u-1/2)^2 du = 1/12$$
$$E[R_n] = n/2$$
$$V[R_n] = n/12$$

Central Limit Theorem

- Central Limit Theorem
  for large N the sum of independent random variables is always normally (Gaussian) distributed:

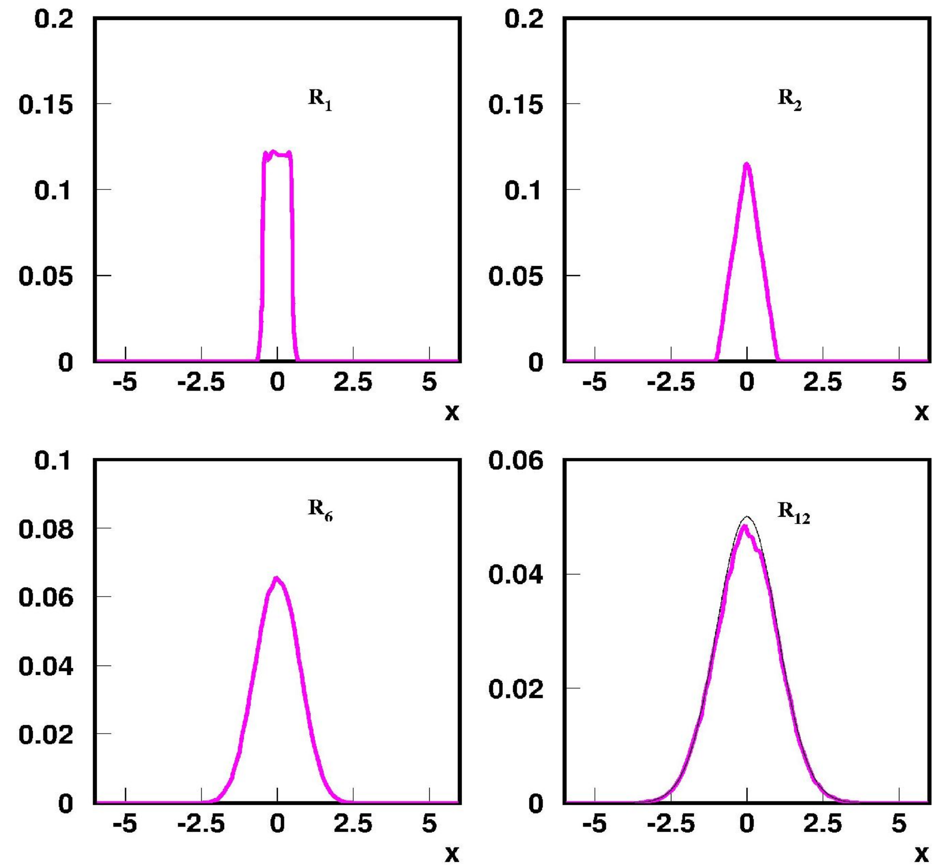
$$f(x) = \frac{1}{s\sqrt{2\pi}} \exp\left[-\frac{(x-a)^2}{2s^2}\right]$$

- example: take sum of uniformly distributed random numbers:

$$R_n = \sum_{i=1}^{n} R_i$$
$$E[R_1] = \int u du = 1/2,$$
$$V[R_1] = \int (u-1/2)^2 du = 1/12$$
$$E[R_n] = n/2$$
$$V[R_n] = n/12$$

- for Gaussian with mean=0 and variance=1, take for *n=12*:

$$\frac{R_n - n/2}{n/12} \to N(0,1)$$



Resumee: Monte Carlo technique

- **Law of large numbers**

$$\frac{1}{N}\sum_{i=1}^{N} f(u_i) \rightarrow \frac{1}{b-a}\int_a^b f(u)du$$

MC estimate converges to true integral

- **Central limit theorem**

MC estimate is asymptotically normally distributed
it approaches a Gaussian density

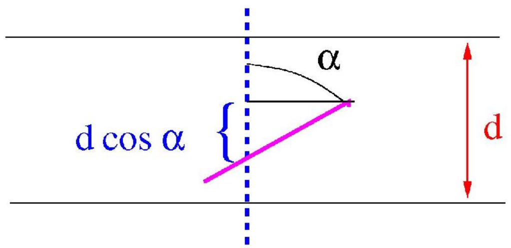
$$\sigma = \frac{\sqrt{V[f]}}{\sqrt{N}} \sim \frac{1}{\sqrt{N}}$$

with effective variance *V(f)*
decrease σ: reduce *V(f)* or increase *N*

- advantages for n-dimensional integral ...

i.e. phase space integrals *2* → *n* processes
is where other approaches tend to fail

Monte Carlo: Buffons Needle - Hit & Miss



- Buffons needle (Buffon 1777)
  pattern of parallel lines with distance d,
  randomly throw needle with length *d* onto stripes,
  count hit, when needle crosses strip
  count miss, if not
- probability for hit is:

$$\frac{d\cos(\alpha)}{d} = \cos(\alpha)$$

all angles are equally likely:

$$\frac{\int_0^{\pi/2}\cos(\alpha)d\alpha}{\pi/2} = \frac{2}{\pi}$$

http://www.angelfire.com/wa/hurben/buff.html

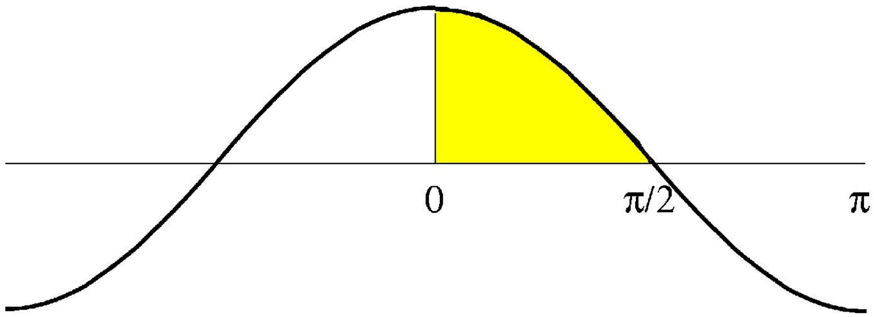
```
loop over ntrials
   x=RN(1) * d
   alpha = RN(2) *3.1415 * 2
   y = d * abs(cos(alpha))
   if((x+y).gt. d) nhit = nhit + 1
endloop
write ' pi = ', 2*ntrial/nhit
```

trials	π	error
100	2.9850	0.2374
1000	3.2733	0.0749
10000	3.1645	0.0237
100000	3.1483	0.0075
1000000	3.1401	0.0024
10000000	3.1422	0.0008

Buffons Needle: Crude Monte Carlo

- Buffons needle (Buffon 1777) is essentially integration of

$$\int_0^{\pi/2} \cos(\alpha) d\alpha$$



- apply Law of large numbers:

$$\frac{1}{N}\sum_{i=1}^{N} f(u_i) \rightarrow \frac{1}{b-a}\int_a^b f(u)du$$

- compare Hit & Miss with Integration

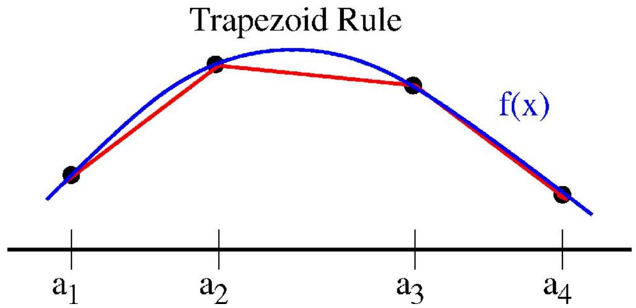
trials	π(hit&miss)	π(integral)
100	3.27869	3.12265
1000	3.36700	3.11833
10000	3.14218	3.15129
100000	3.13087	3.13416
1000000	3.14127	3.14337
10000000	3.14154	3.14168
100000000	3.12174	3.14156

- 1st example of true Monte Carlo experiment
- equivalence of integration and MC event generation

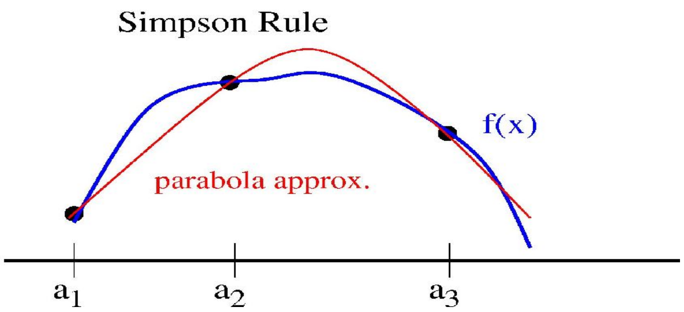
Integration: Monte Carlo versus others

One dimensional quadrature

$$I = \int f(x)dx = \sum_{i=1}^{n} w_i f(x_i)$$



- Monte Carlo: Hit & Miss
  w =1 and x_i chosen randomly
- Trapezoidal Rule:
  approximate integral in sub-interval by area of trapezoid below (above) curve
- Simpson quadrature
  approximate by parabola
- Gauss quadrature
  approximate by higher order polynomial



method	err (1d)	error
MC	$n^{-1/2}$	$n^{-1/2}$
Trapez	n^{-2}	$n^{-2/d}$
Simpson	n^{-4}	$n^{-4/d}$
Gauss	n^{-2m+1}	$n^{-(2m-1)/d}$

MC method: advantage of hit & miss

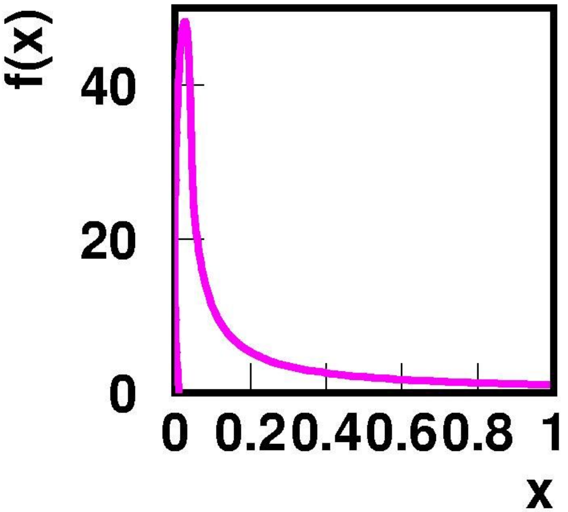
- integration ☛ weighting events
  large fluctuations from large weights
  weights also to errors applied
  difficult to apply further hadronisation
- real events all have weight = 1 !!!
- Hit & Miss method:

MC for function $f(x)$:
get random number:
$R1$ in $(0,1)$ and $R2$ in $(0,1)$
calculate $x = R1$
reject event if: $f_x < f_{max} R2$

MC method: advantage of hit & miss

- integration ☛ weighting events
  large fluctuations from large weights
  weights also to errors applied
  difficult to apply further hadronisation
- real events all have weight = 1 !!!
- Hit & Miss method:

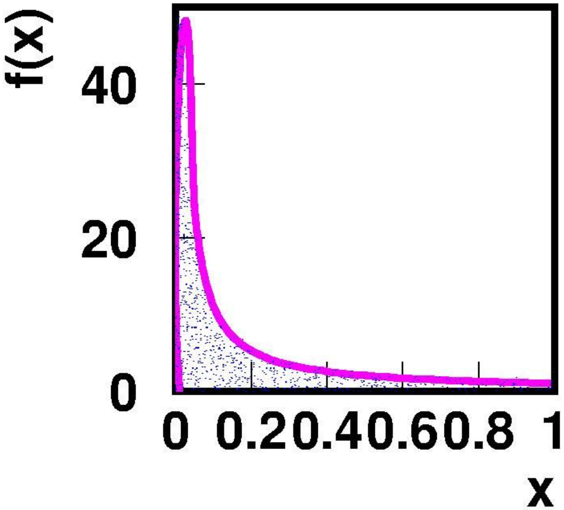
MC for function $f(x)$:
get random number:
$R1$ in $(0,1)$ and $R2$ in $(0,1)$
calculate $x = R1$
reject event if: $f_x < f_{max} R2$



MC method: advantage of hit & miss

- integration ☛ weighting events
  large fluctuations from large weights
  weights also to errors applied
  difficult to apply further hadronisation
- real events all have weight = 1 !!!
- Hit & Miss method:

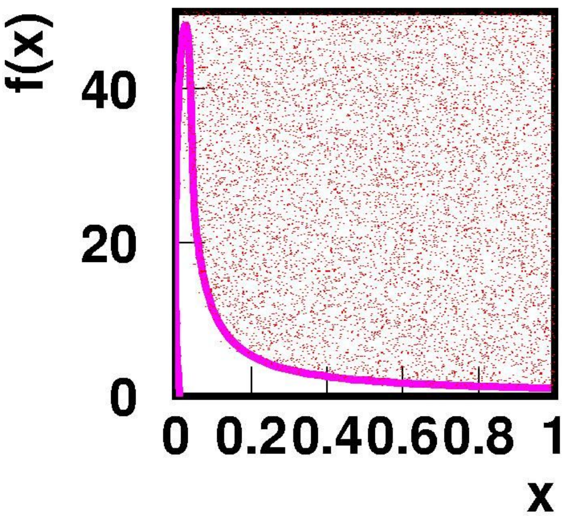
MC for function *f(x)*:
get random number:
R1 in *(0,1)* and *R2* in *(0,1)*
calculate $x = R1$
reject event if: $f_x < f_{max} R2$



MC method: advantage of hit & miss

- integration ☛ weighting events
  - large fluctuations from large weights
  - weights also to errors applied
  - difficult to apply further hadronisation
- real events all have weight = 1 !!!
- Hit & Miss method:

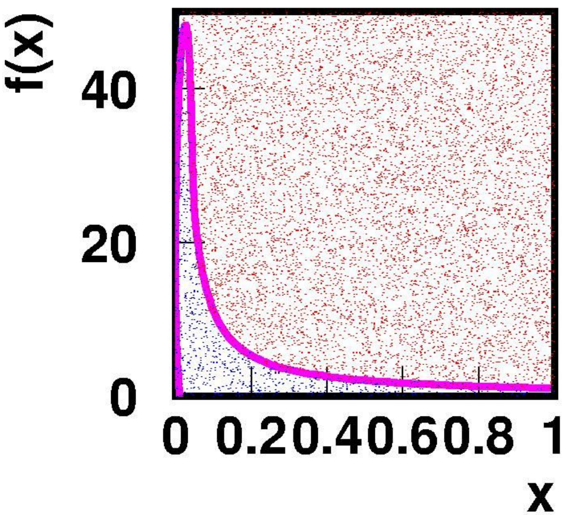
MC for function *f(x)*:
get random number:
R1 in *(0,1)* and *R2* in *(0,1)*
calculate $x = R1$
reject event if: $f_x < f_{max} R2$



MC method: advantage of hit & miss

- integration ☛ weighting events
  large fluctuations from large weights
  weights also to errors applied
  difficult to apply further hadronisation
- real events all have weight = 1 !!!
- Hit & Miss method:

MC for function *f(x)*:
get random number:
R1 in *(0,1)* and *R2* in *(0,1)*
calculate $x = R1$
reject event if: $f_x < f_{max}\, R2$



- BUT: Hit & Miss method inefficient for peaked *f(x)*

MC method: do even better ...

- Importance sampling

MC for function f(x)

approximate *f(x)* ~ *g(x)*

with *g(x)* > *f(x)* simple and integrable

generate x according to *g(x)*:

$$\int_{x_{min}}^{x} g(x')dx' = R1 \int_{x_{min}}^{x_{max}} g(x')dx'$$

example: $f(x) = 1/x^{0.7}$

$$g(x) = 1/x$$

$$x = x_{min} \cdot \left(\frac{x_{max}}{x_{min}}\right)^{R1}$$

reject event if: *f(x)* < *g(x) R2*

MC method: do even better ...

- Importance sampling

MC for function f(x)

approximate *f(x)* ~ *g(x)*

with *g(x)* > *f(x)* simple and integrable

generate x according to *g(x)*:

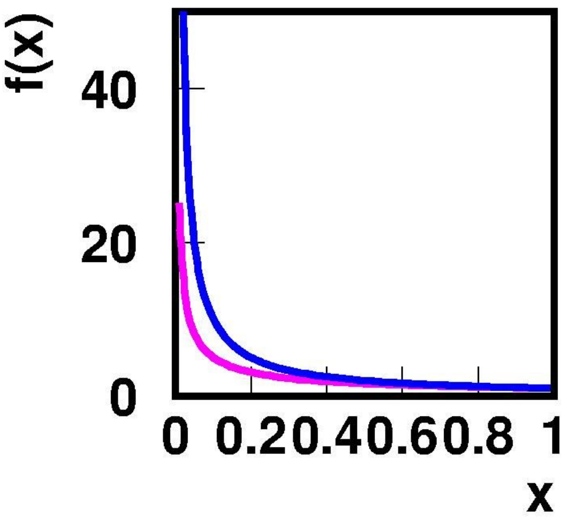
$$\int_{x_{min}}^{x} g(x')dx' = R1 \int_{x_{min}}^{x_{max}} g(x')dx'$$

example: $f(x) = 1/x^{0.7}$

$$g(x) = 1/x$$

$$x = x_{min} \cdot \left(\frac{x_{max}}{x_{min}}\right)^{R1}$$

reject event if: *f(x)* < *g(x)* *R2*



MC method: do even better ...

- Importance sampling

MC for function f(x)

approximate *f(x)* ~ *g(x)*

with *g(x)* > *f(x)* simple and integrable

generate x according to *g(x)*:

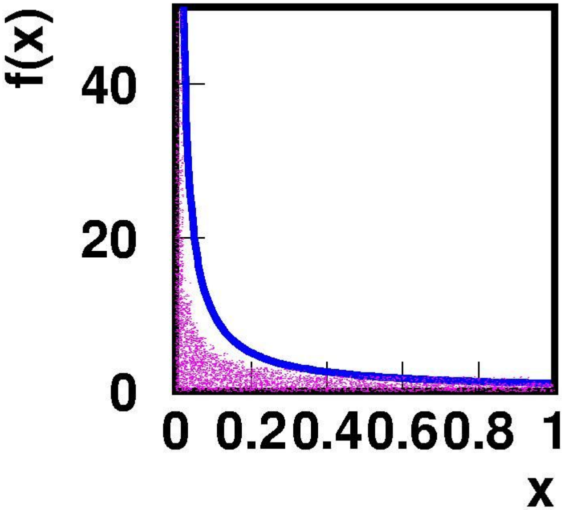
$$\int_{x_{min}}^{x} g(x')dx' = R1 \int_{x_{min}}^{x_{max}} g(x')dx'$$

example: $f(x) = 1/x^{0.7}$

$$g(x) = 1/x$$

$$x = x_{min} \cdot \left(\frac{x_{max}}{x_{min}}\right)^{R1}$$

reject event if: *f(x)* < *g(x) R2*



MC method: do even better ...

- Importance sampling

MC for function f(x)

approximate *f(x)* ~ *g(x)*

with *g(x)* > *f(x)* simple and integrable

generate x according to *g(x)*:

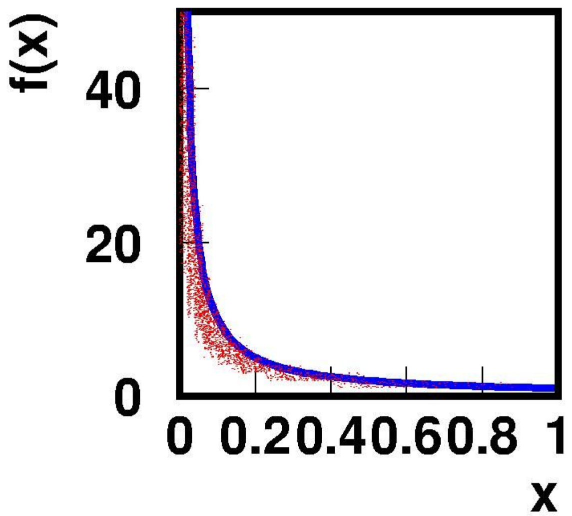
$$\int_{x_{min}}^{x} g(x')dx' = R1 \int_{x_{min}}^{x_{max}} g(x')dx'$$

example: $f(x) = 1/x^{0.7}$

$$g(x) = 1/x$$

$$x = x_{min} \cdot \left(\frac{x_{max}}{x_{min}}\right)^{R1}$$

reject event if: *f(x)* < *g(x) R2*



MC method: do even better ...

- Importance sampling

MC for function f(x)

approximate *f(x)* ~ *g(x)*

with *g(x)* > *f(x)* simple and integrable

generate x according to *g(x)*:

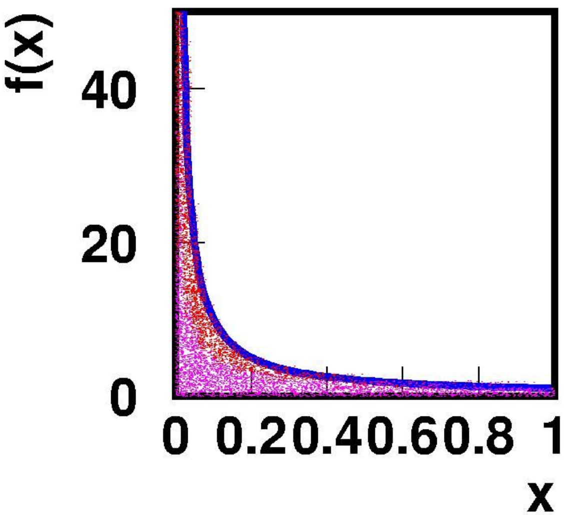
$$\int_{x_{min}}^{x} g(x')dx' = R1 \int_{x_{min}}^{x_{max}} g(x')dx'$$

example: $f(x) = 1/x^{0.7}$

$$g(x) = 1/x$$

$$x = x_{min} \cdot \left(\frac{x_{max}}{x_{min}}\right)^{R1}$$

reject event if: *f(x)* < *g(x) R2*



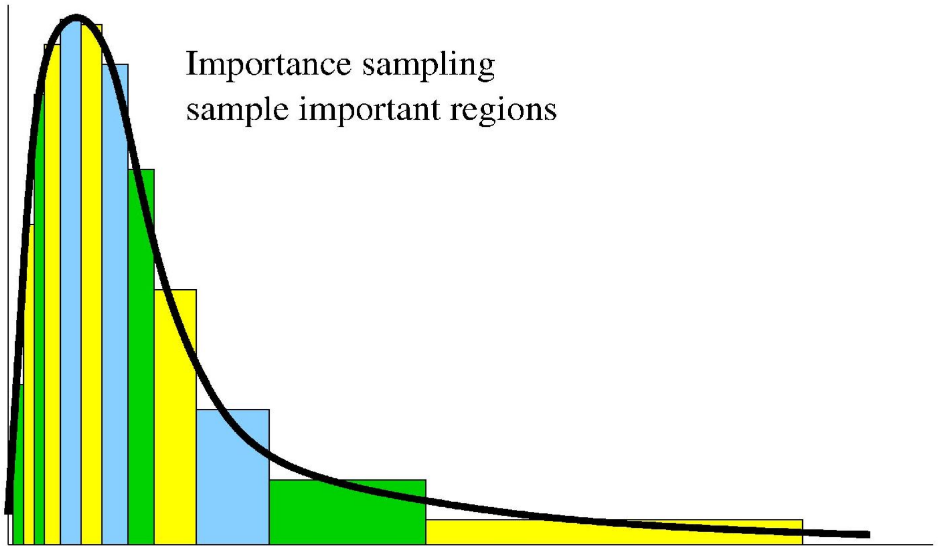
- WOW !!! very efficient even for peaked f(x)

Importance Sampling

- MC calculations most efficient for small weight fluctuations:
  $f(x)dx \rightarrow f(x)\, dG(x)/g(x)$
- chose point according to g(x) instead of uniformly
- f is divided by $g(x) = dG(x)/dx$
- generate x according to:

$$R \int_a^b g(x')dx' = \int_a^x g(x')dx'$$

- relevant variance is now $V(f/g)$:
  small if $g(x) \sim f(x)$
- how-to get $g(x)$
  (1) $g(x)$ is probability: $g(x) > 0$ and $\int dG(x) = 1$
  (2) integral $\int dG(x)$ is known analytically
  (3) $G(x)$ can be inverted (solved for x)
  (4) $f(x)/g(x)$ is nearly constant, so that $V(f/g)$ is small compared to $V(f)$



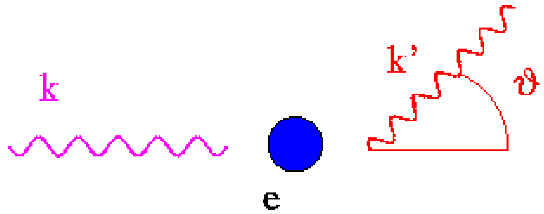
Applications in High Energy Physics

- Simulation of detector response
- Apply MC method to e^+e^-
- what about hadronsiation
- what about QCD radiation
- going even further: initial state radiation
- how-to do a DIS Monte Carlo event generator
- some examples

Application of MC method: Compton scattering

- Compton scattering (O. Klein, Y. Nishima, Z. Physik, 52, 853 (1929))
  energy of the final photon *k'*:

$$k' = \frac{k}{1 + (k/m)(1 - \cos\theta)}$$



- Differential cross section

$$\frac{d\sigma}{d\Omega} = \frac{\alpha_{em}^2}{2m^2}\left(\frac{k'}{k}\right)^2\left(\frac{k'}{k} + \frac{k}{k'} - \sin^2\theta\right)$$

- angular distribution of the photon is:

$$\sigma(\theta,\phi)d\theta d\phi = \frac{\alpha_{em}^2}{2m^2}\left(\left(\frac{k'}{k}\right)^3 + \left(\frac{k}{k'}\right) - \left(\frac{k'}{k}\right)^2\sin^2\theta\right)\sin\theta d\theta d\phi$$

- generate azimuthal ϕ independently: $\phi = 2\pi R_1$

Application of MC method: Compton scattering

- to generate θ, use approximation
  for $k \gg m$, x-section peaked at small angles
  (using $u=(1-\cos\theta)$):

$$\sigma^a d\theta d\phi = \frac{\alpha_{em}^2}{2m^2}\left(\frac{k'}{k}\right)\sin\theta d\theta d\phi$$

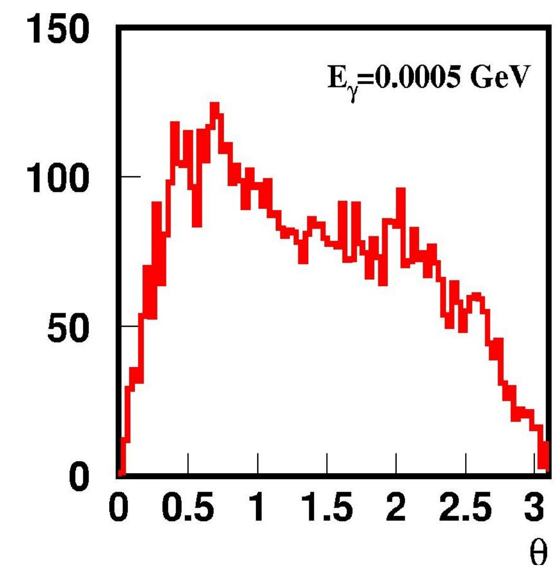
using $k' = \frac{k}{1+(k/m)(1-\cos\theta)}$

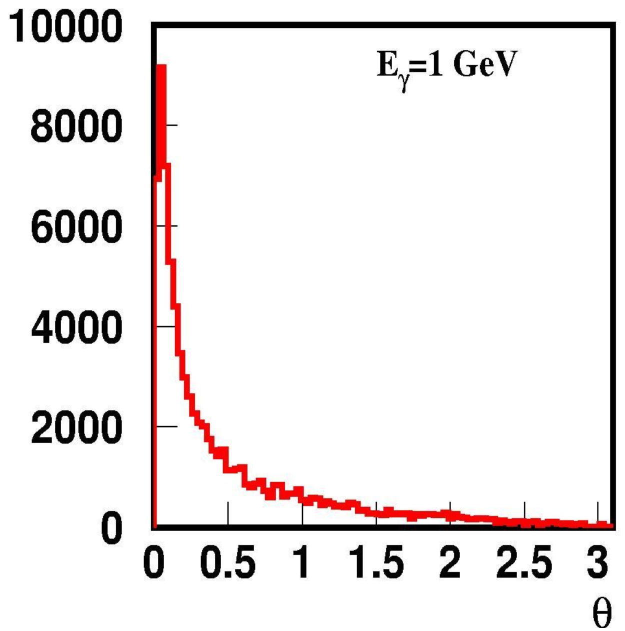
$$\sigma^a d\theta d\phi = \frac{\alpha_{em}^2}{2m^2}\left(1+\frac{k}{m}u\right)^{-1} du d\phi$$

- use:

$$R_2 \int_0^2 \left(1+\frac{k}{m}u'\right)^{-1} du' = \int_0^u \left(1+\frac{k}{m}u'\right)^{-1} du'$$

- generate u with $u = \frac{m}{k}\left[\left(1+2\frac{k}{m}\right)^{R_2} - 1\right]$
- weight by: $\frac{\sigma}{\sigma^a}$





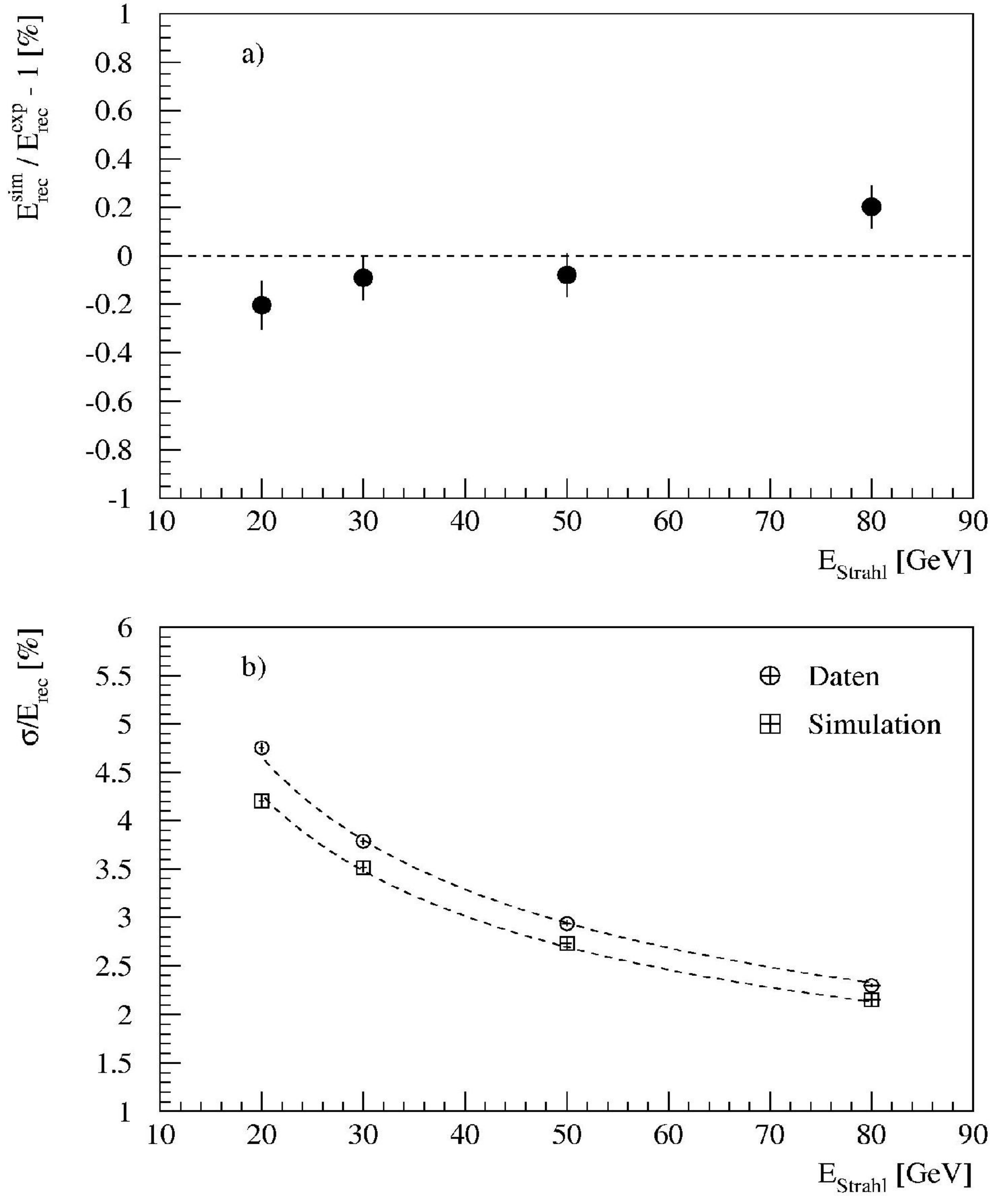
Application of MC method: photon transport in matter

Program for Compton scattering and similar programs for photo-effect and pair-creation build program that simulates interactions of photons with matter

- Algorithm
  - break path into small pieces
  - in each step, decide whether interaction (and which) takes place, given the total cross section for each possible interaction
  - from mean free path length, decide where interaction takes place
  - simulate interaction: give photon new energy and angle, or produce $e^{+}e^{-}$ pair, etc ...
  - continue path with new parameters
- such program exist
  - EGS (SLAC)
  - GEANT (CERN)
- Detector simulation with programs for particle transport in matter
  - to study detector design
  - to obtain a detailed simulation of the detector response
  - to estimate efficiencies, bias, etc...

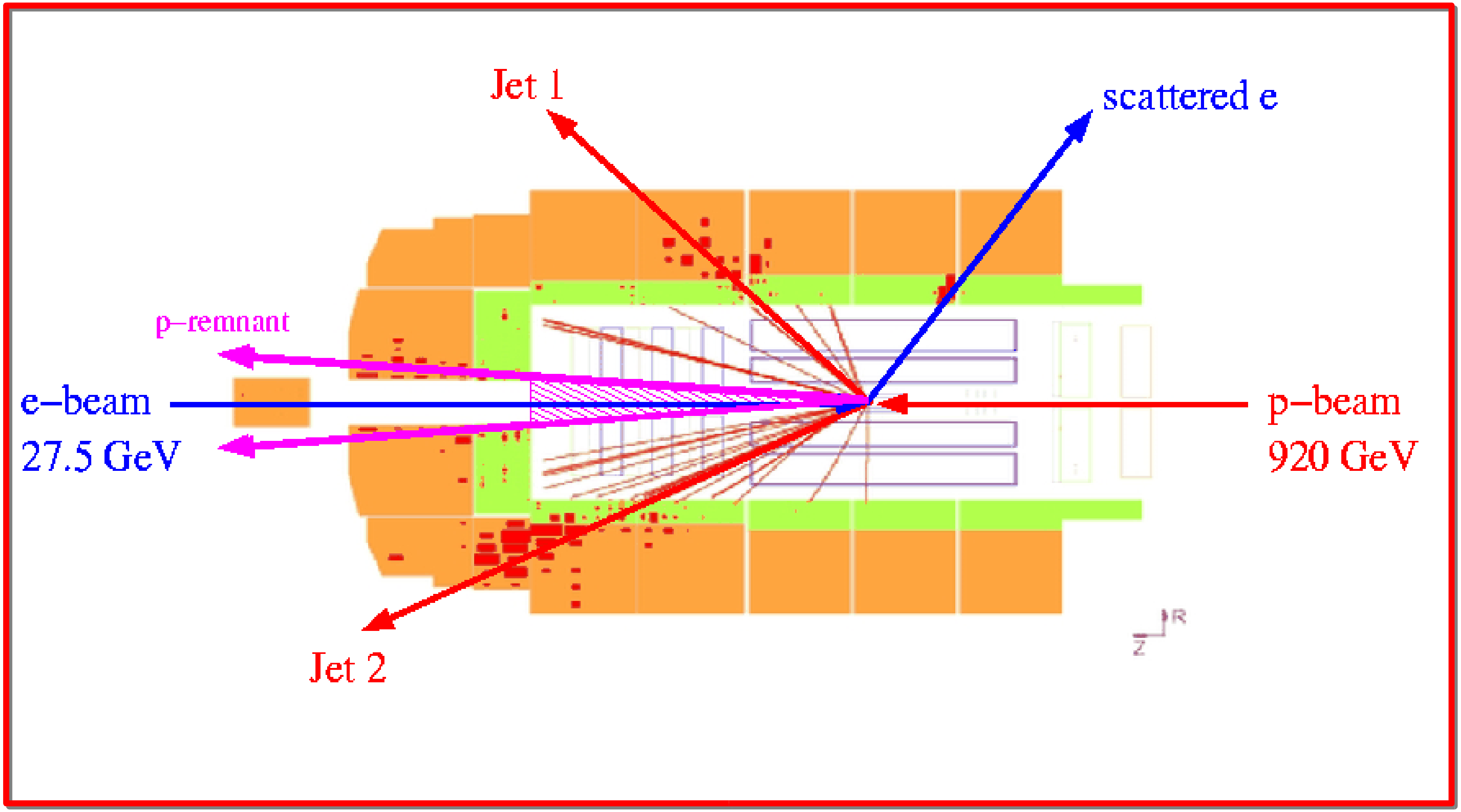
Application of Simulation: Calibration of H1 Calorimeter

- simulated energy response in calorimeter
  using GEANT package includir
  full detector geometry and material information
- test beam measurement of ener response
- test of understanding detector performance
- nice agreement within ~ 3%
- difference due to dead material i front of detector



J. Spiekermann, diploma 1994

MC event: hadron and detector level



$\sqrt{s} \sim 318$ GeV → $x \sim 7.\ 10^{-5}$ at $Q^2 = 4$ GeV^2

From experiment to measurement

take data

run MC generator

detailed detector simulation

compare detector level response: data with MC

define visible x - section in kinematic variables

calculate factor C_{corr} to correct from detector to hadron level

$$\frac{d\sigma^{data}_{had}}{dx} = \frac{d\sigma^{data}_{det}}{dx} C_{corr} \text{ with } C_{corr} = \frac{\frac{d\sigma^{MC}_{had}}{dx}}{\frac{d\sigma^{MC}_{det}}{dx}}$$

visible x-section on hadron level

All measurements rely on proper MC's !!!

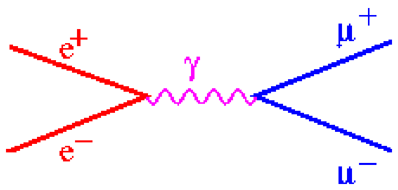
MC generators - different applications ...

- calculate x-section of various processes ☛ complicated integrals
- multi - differential, in any variable

- MC simulation of detector response
  - input: hadron level events - output: detector level events
    - Calorimeter ADC hits
    - Tracker hits
    - ....
  - need knowledge of particle passage through matter, x-section ...
  - need knowledge of actual detector
  - x-section on parton level
- multipurpose MC event generators:
  - x-section on parton level
  - including multi-parton (initial & final state) radiation
  - remnant treatment (proton remnant, electron remnant)
  - hadronisation/fragmentation (more than simple fragmentation functions...)
- fixed order parton level ...... theorists like it
  - integration of multidimensional phase space

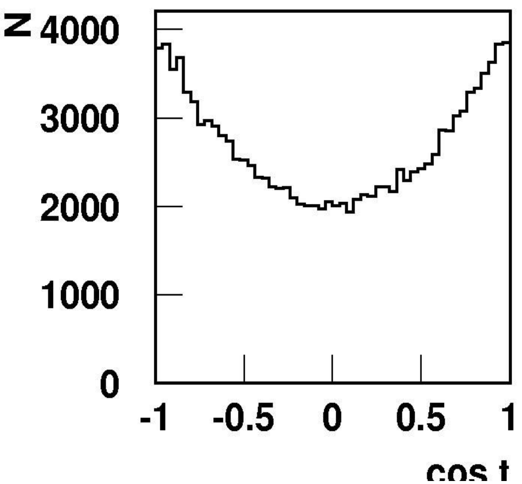
Constructing a MC for e^+e^-: the simple case

- process: $e^+e^- \rightarrow \mu^+ \mu^-$



- $$\frac{d\sigma}{d\cos\theta d\phi} = \frac{\alpha_{em}^2}{4s}\left(1+\cos^2\theta\right)$$
- goal: generate 4-momenta of μ's, need *cm* energy *s*, *cos* θ, ϕ

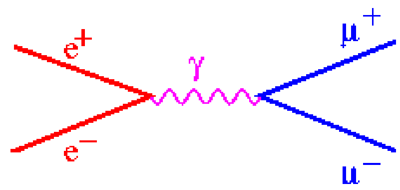
after 100000 events



random number R1(0,1) $\phi = 2\,\pi$ R1
random number R2(0,1) cos θ = -1 + 2 R2

- for every R1, R2 use weight with $\frac{d\sigma}{d\cos\theta\, d\phi}$
- repeat many times

Constructing a MC for e^+e^-: the simple case

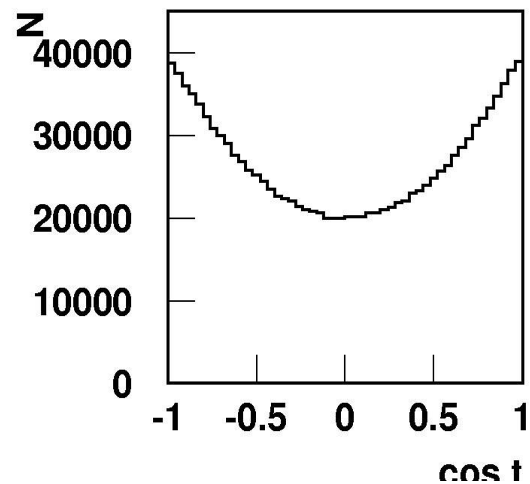


- process: $e^+e^- \rightarrow \mu^+ \mu^-$

- $$\frac{d\sigma}{d\cos\theta d\phi} = \frac{\alpha_{em}^2}{4s}\left(1+\cos^2\theta\right)$$

- goal: generate 4-momenta of μ's,
  need *cm* energy *s*, *cos* θ, ϕ

after 10^6 events



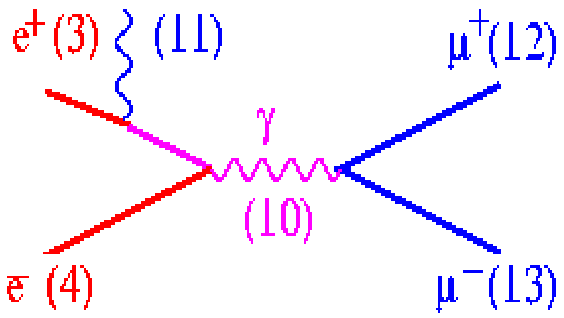
random number R1(0,1) $\phi = 2\pi$ R1
random number R2(0,1) cos θ = -1 + 2 R2

- for every R1, R2 use weight with $\frac{d\sigma}{d\cos\theta\, d\phi}$
- repeat many times

Example event: $e^+e^- \to \mu^+ \mu^-$

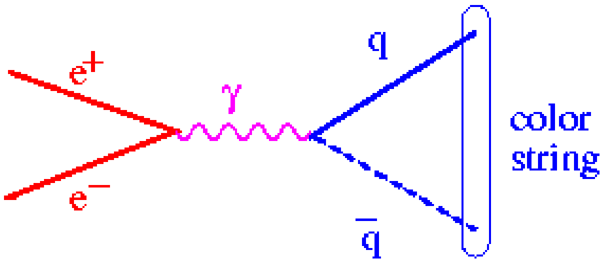
- example from PYTHIA: Event listing

I	particle/jet	KS	KF	orig	p_x	p_y	p_z	E	m
1	!e+!	21	-11	0	0.000	0.000	30.000	30.000	0.001
2	!e-!	21	11	0	0.000	0.000	-30.000	30.000	0.001
3	!e+!	21	-11	1	0.000	0.000	30.000	30.000	0.000
4	!e-!	21	11	2	0.000	0.000	-30.000	30.000	0.000
5	!e+!	21	-11	3	0.143	0.040	26.460	26.460	0.000
6	!e-!	21	11	4	0.000	0.000	-29.998	29.998	0.000
7	!Z0!	21	23	0	0.143	0.040	-3.539	56.458	56.347
8	!mu-!	21	13	7	-9.510	1.741	24.722	26.546	0.106
9	!mu+!	21	-13	7	9.653	-1.700	-28.261	29.913	0.106
10	(Z0)	11	23	7	0.143	0.040	-3.539	56.458	56.347
11	gamma	1	22	3	-0.143	-0.040	3.539	3.542	0.000
12	mu-	1	13	8	-9.510	1.741	24.722	26.546	0.106
13	mu+	1	-13	9	9.653	-1.700	-28.261	29.913	0.106
		sum:	0.00		0.000	0.000	0.000	60.000	60.000



- technicalities/advantages
  - ➔ can work in any frame
  - ➔ Lorentz-boost 4-vectors back and forth
  - ➔ can calculate any kinematic variable
  - ➔ history of event process

Constructing a MC for $e^+e^- \to q\bar{q}$



- process $e^+e^- \to q\bar{q}$
- $$\frac{d\sigma}{d\cos\theta d\phi} = \frac{\alpha_{em}^2}{4s}\left(1+\cos^2\theta\right)$$
- generate scattering as for $e^+e^- \to \mu^+\mu^-$
- BUT what about fragmentation/hadronization ???
- use concept of local parton-hadron duality

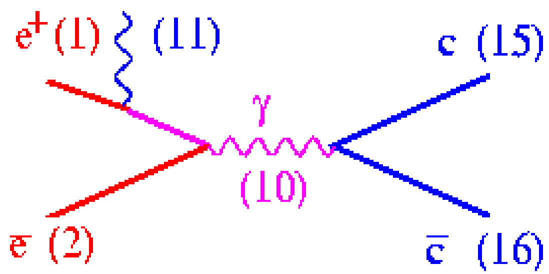
linear confinement potential: $V(r) \sim -1/r + \kappa r$
with $\kappa \sim 1$ GeV/fm
qq connected via color flux tube of transverse size of hadrons (~1 fm)
color tube: uniform along its length → linearly rising potential

→ Lund string fragmentation

Example event $e^+e^- \to q\bar{q}$

- example from PYTHIA Monte Carlo generator including hadronization

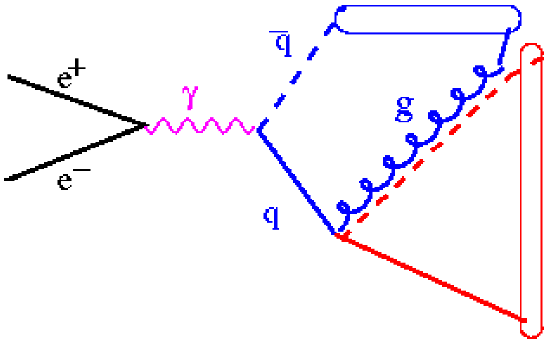
I	particle/jet		KS	KF	orig	p_x	p_y	p_z	E	m
1	!e+!		21	-11	0	0.000	0.000	30.000	30.000	0.001
2	!e-!		21	11	0	0.000	0.000	-30.000	30.000	0.001
5	!e+!		21	-11	3	0.018	0.040	0.702	0.703	0.000
6	!e-!		21	11	4	0.000	0.000	-29.998	29.998	0.000
10	(Z0)		11	23	7	0.018	0.040	-29.297	30.701	9.180
11	gamma		1	22	1	-0.018	-0.040	29.298	29.298	0.000
15	(c)	A	12	4	10	-1.950	-3.529	-19.752	20.215	1.500
16	(cbar)	V	11	-4	10	1.967	3.569	-9.545	10.486	1.500
17	(string)		11	92	15	0.018	0.040	-29.297	30.701	9.180
18	(D0)		11	421	17	-0.455	-1.495	-9.002	9.325	1.865
19	(omega)		11	223	17	-0.300	-0.076	-3.228	3.338	0.793
20	pi+		1	211	17	-0.168	-0.172	-0.861	0.904	0.140
21	(rho-)		11	-213	17	-0.114	-0.513	-4.992	5.106	0.932
22	(omega)		11	223	17	-0.173	0.118	-2.022	2.180	0.789
23	pi+		1	211	17	0.226	0.925	-2.593	2.766	0.140
24	(D*-)		11	-413	17	1.001	1.253	-6.599	7.082	2.010
25	e+		1	-11	18	-0.191	0.241	-1.261	1.297	0.001
26	nu_e		1	12	18	-0.154	-0.789	-4.174	4.250	0.000
....										
....										
....										
53	pi-		1	-211	47	0.318	-0.061	-1.293	1.340	0.140
			sum:	0.00		0.000	0.000	0.000	60.000	60.000



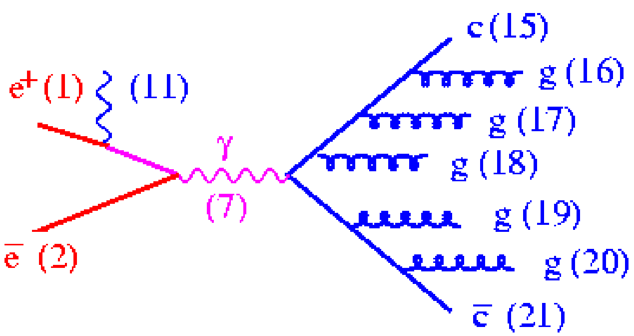
- apply fragmentation directly to parton
- all covered by hadronization .... soft
- **where is QCD ???**

Doing things better: $e^+e^- \to qqg$

- process $e^+e^- \to qqg$
- full matrix element calculation
- watch out color flow !!!
- gluons act as kicks on strings



I	particle/jet		KS	KF	orig	p_x	p_y	p_z	E	m
1	!e+!		21	-11	0	0.000	0.000	30.000	30.000	0.001
2	!e-!		21	11	0	0.000	0.000	-30.000	30.000	0.001
5	!e+!		21	-11	1	0.000	0.000	29.699	29.699	0.000
6	!e-!		21	11	2	-1.319	-1.236	-26.950	27.011	0.000
7	!Z0!		21	23	0	-1.319	-1.236	2.748	56.710	56.614
8	!c!		21	4	7	-15.986	16.072	18.293	29.167	1.500
9	!cbar!		21	-4	7	14.667	-17.308	-15.545	27.542	1.500
11	gamma		1	22	2	1.320	1.236	-2.744	3.286	0.000
15	(c)	A	12	4	8	-11.291	11.550	13.219	20.926	1.500
16	(g)	I	12	21	8	-3.992	3.139	4.805	6.991	0.000
17	(g)	I	12	21	8	-0.279	0.951	0.179	1.007	0.000
18	(g)	I	12	21	8	0.122	-0.178	-0.505	0.550	0.000
19	(g)	I	12	21	9	0.128	-0.237	0.146	0.307	0.000
20	(g)	I	12	21	9	-0.093	-0.746	-0.364	0.835	0.000
21	(g)	I	12	21	9	8.331	-6.743	-6.396	12.482	0.000
22	(cbar)	V	11	-4	9	5.754	-8.971	-8.335	13.613	1.500

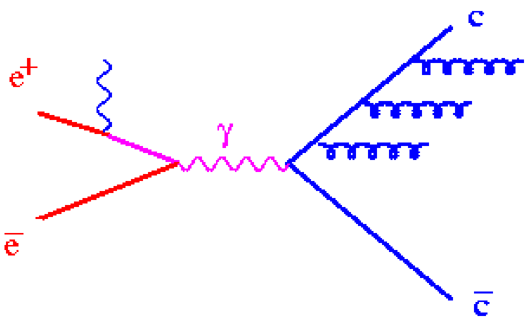


- more large p_t emissions
- not all covered by fixed order calculations
- doing much better needed

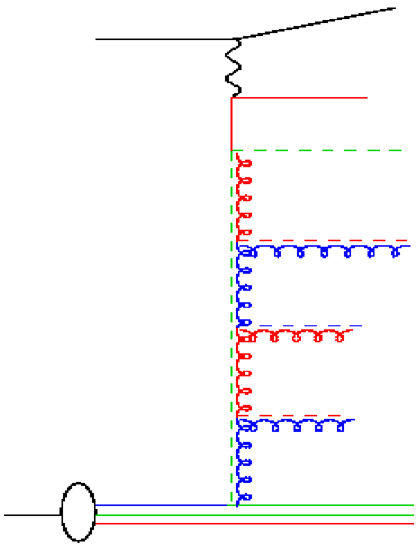
➔ parton shower approach

Approximations to higher orders: parton showers

- Approximation to higher orders.....



- fragmentation functions



- parton density functions

- since alphas is not small, higher orders contributions are important
- Approximations:

  DGLAP (Dokshitzer, Gribov, Lipatov, Altarelli, Parisi)

  BFKL (Balitski, Fadin, Kuraev, Lipatov)

  CCFM (Catani, Ciafaloni, Fiorani, Marchesini)

DGLAP equation

- differential form $$q\frac{\partial}{\partial q} f(x,q) = \int \frac{dz}{z}\frac{\alpha_s}{2\pi} P_+(z)\, f\left(\frac{x}{z}, q\right)$$

- modified differential form using *"Sudakov form factor"*

$$\Delta_s(q_0, q) = \exp\left(-\bar{\alpha}_s \int \frac{dz}{z} \int_{q_0}^{q} \frac{dq'}{q'} \tilde{P}(z)\right)$$

$$q\frac{\partial}{\partial q}\frac{f(x,q)}{\Delta_s(q,q_0)} = \int \frac{dz}{z}\frac{\alpha_s}{2\pi}\frac{\tilde{P}(z)}{\Delta_s(q,q_0)} f\left(\frac{x}{z}, q\right)$$

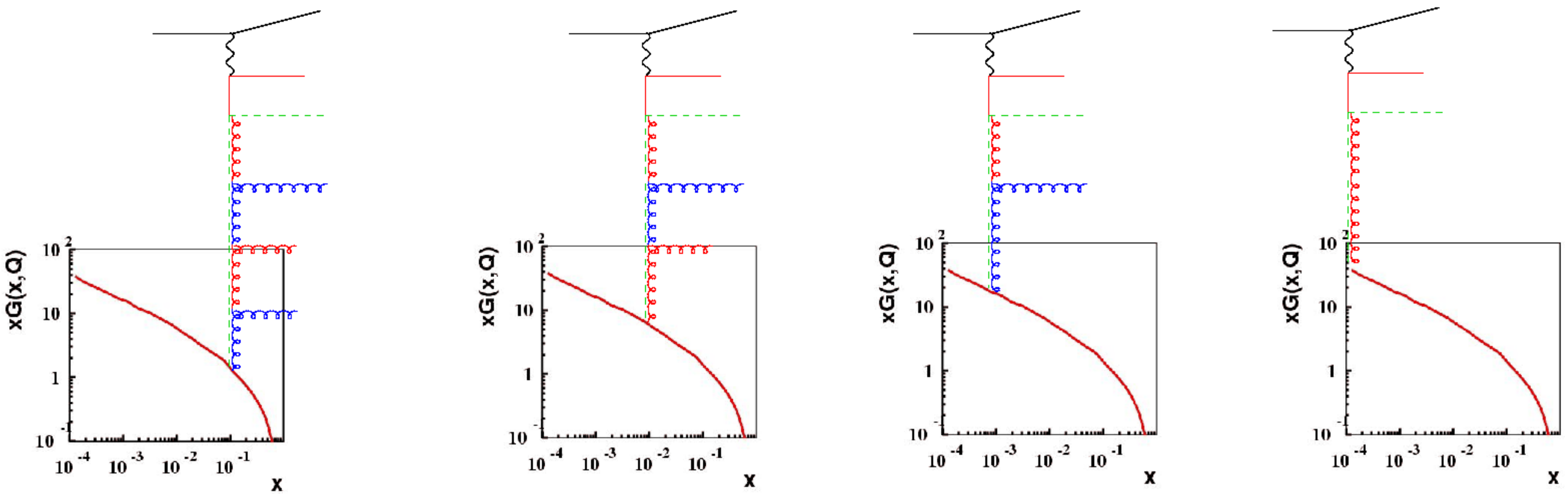
- **integral form**

$$f(x,q) = f_0(x,q)\Delta_s(q,q_0) + \int \frac{dz}{z}\int \frac{dq'}{q'} \cdot \Delta_s(q', q_0)\tilde{P}(z) f\left(\frac{x}{z}, q\right)$$

- no-branching probability form q_0 to q

DGLAP evolution equation

- for fixed x and Q^2 chains with different branchings contribute
- iterative procedure, spacelike parton showering



- $$f(x,Q) = f_0(x,q_0)\Delta_s(Q,q_0) + \sum_{k=1}^{\infty} f_k(x_k,q_k)$$

Parton Shower

- Evolution equation with Sudakov form factor recovers exactly evolution equation (with $_+$ prescription)
- Sudakov form factor particularly suited form Monte Carlo approach
- Sudakov form factor
  - ➔ gives probability for no-branching between q_0 and q
  - ➔ sums virtual contributions to all orders (via unitarity)
    - ➔ virtual (parton loop) and
    - ➔ real (non-resolvable) parton emissions
- need to specify scale of hard process (matrix element) $Q \sim p_t$
- need to specify cutoff scale $Q_0 \sim 1$ GeV

The DIS process ep → epX

- cross section $\frac{d\sigma(ep\to e'X)}{dy\,dQ^2} = \frac{4\pi\alpha^2}{yQ^4}\left(\left(1-y+\frac{y^2}{2}\right)F_2^p(x,Q^2) - \frac{y^2}{2}F_L^p(x,Q^2)\right)$

  with $F_2^p(x,Q^2) = \sum_f e_f^2\left(xq_f(x,Q^2) + x\bar{q}_f(x,Q^2)\right)$

- generate y with $g(y)=1/y$, and Q^2 with $g(Q^2)=1/Q^2$:

$$y = y_{min}\left(\frac{y_{max}}{y_{min}}\right)^{R_1}$$

$$Q^2 = Q^2_{min}\left(\frac{Q^2_{max}}{Q^2_{min}}\right)^{R_2}$$

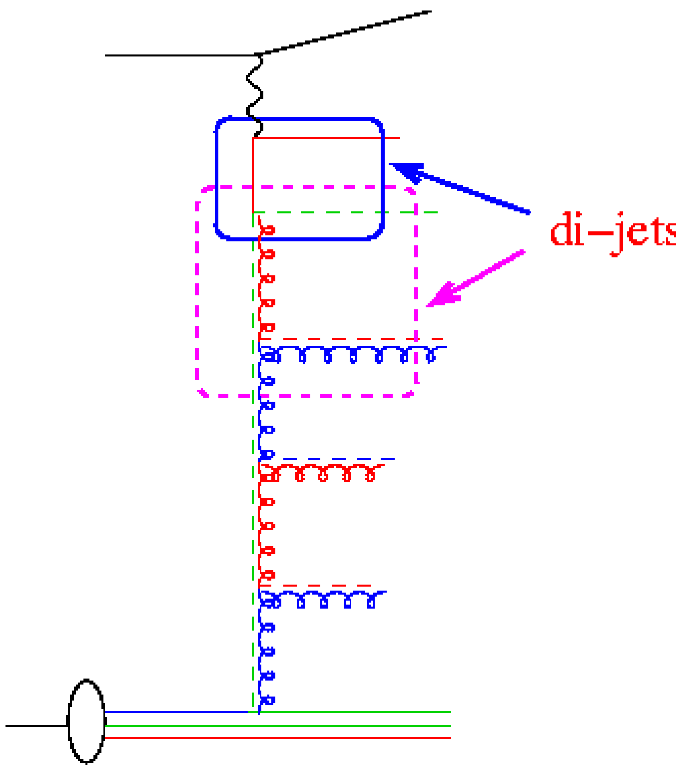
- calculate x-section with:

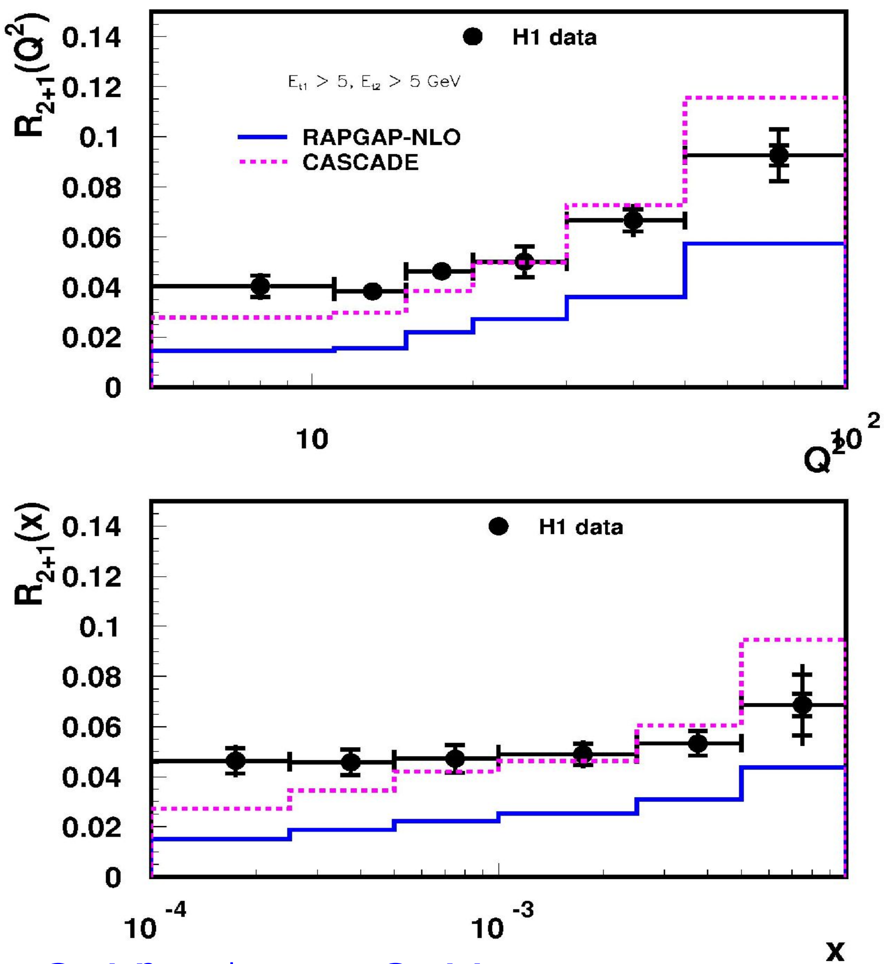
$$\sigma(ep\to e'X) = \frac{1}{N}\sum_{i=1}^{N}\frac{\frac{d\sigma}{dy_i\,dQ_i^2}}{g(y_i)g(Q_i^2)}\int g(y)dy\int g(Q^2)dQ^2$$

$$\sigma(ep\to e'X) = \frac{1}{N}\sum_{i=1}^{N} y_i Q_i^2\frac{d\sigma}{dy_i\,dQ_i^2}\log\left(\frac{y_{max}}{y_{min}}\right)\log\left(\frac{Q^2_{max}}{Q^2_{min}}\right)$$

- calculate 4-momenta of scattered electron and virtual photon

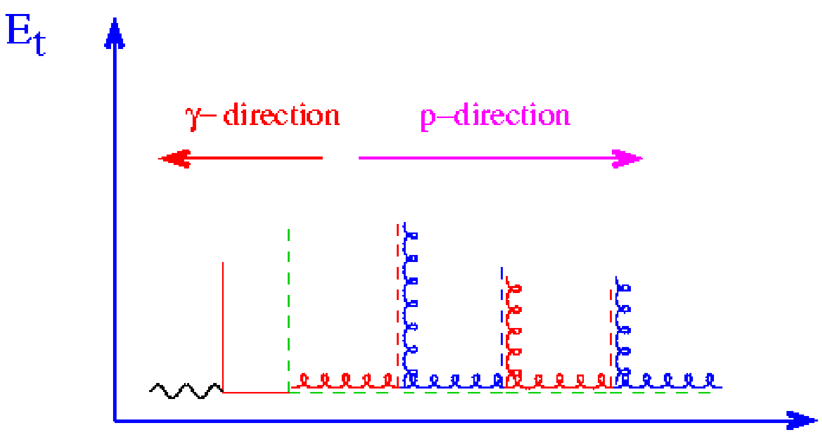
Hadronic final state: Di-jet rates

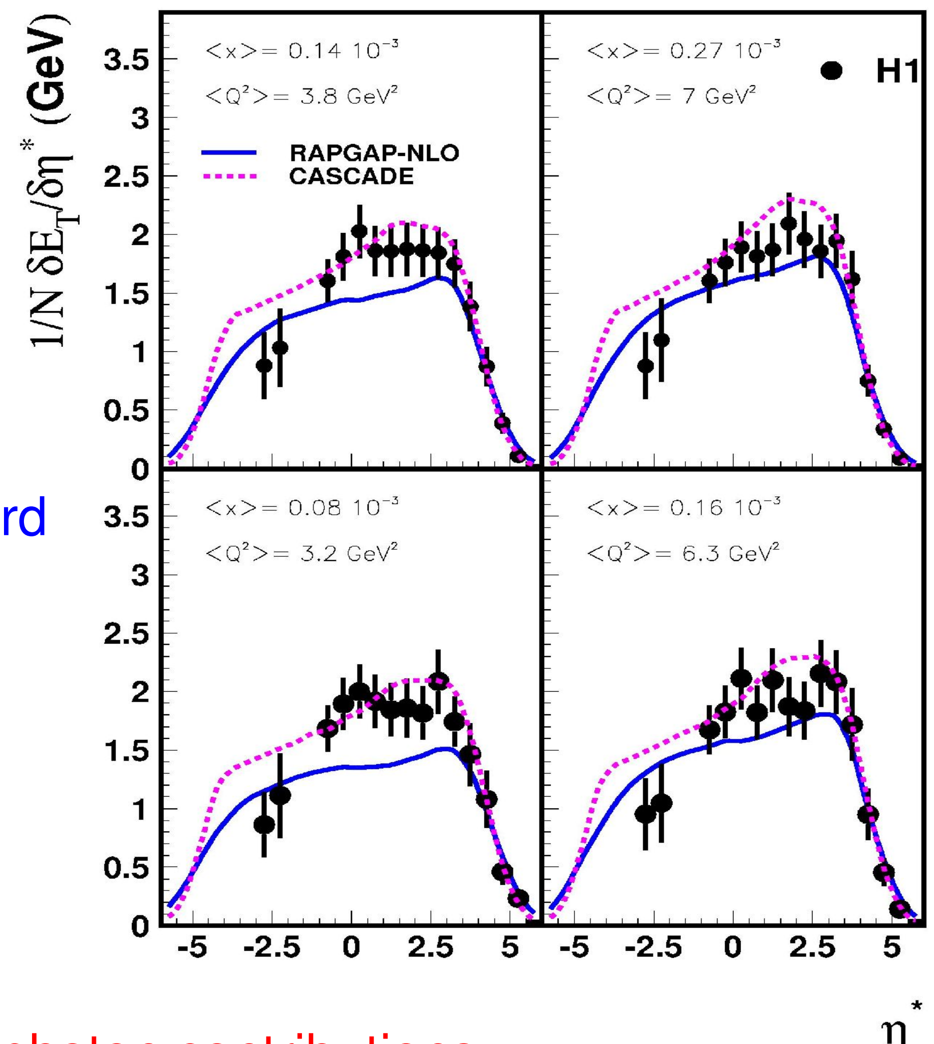




- (2+remnant) jets in DIS for $Q^2 > 5\ GeV^2$, $p_t^{jets} > 5$ GeV
- $\mathcal{O}(\alpha_s)$ processes not enough
  - ➔ need $\mathcal{O}(\alpha_s^2)$ or resolved virtual photon contributions
  - ➔ or something new ???

Hadronic final state: Energy flow





- Et flow in DIS at small x and forward angle (p-direction):

➔ $\mathcal{O}(\alpha_s)$ processes not enough

- need $\mathcal{O}(\alpha_s^2)$ or resolved virtual photon contributions
- or something new ???

Conclusion

- Monte Carlo event generators are needed to calculate multi-parton cross sections
- Monte Carlo method is a well defined procedure
- hadronization is needed to compare with measurements
- parton shower (leading log) approach is needed, hadronziation not enough
- MC approach extended from simple e+e- processes to
  - ep processes
  - pp processes
  - and heavy Ion processes
- proper Monte Carlos are essential for any measurement

Monte Carlo event generators contain all our physics knowledge !!!!!

List of available MC program

- HERA Monte Carlo workshop: www.desy.de/~heramc
- ARIADNE
  A program for simulation of QCD cascades implementing the color dipole model
- AROMA
  Heavy quark production in boson-gluon fusion using full electroweak LO cross-sections (with quark masses) in ep collisions, DIS and photoproduction. Parton showers and Lund hadronization gives full events.
- CASCADE
  is a full hadron level Monte Carlo generator for ep and $p\bar{p}$scattering at small x build according to the CCFM evolution equation. It is applicable in ep to photoproduction and DIS, and for heavy quark production as well as inelastic J/ψ.
- HERWIG
  General purpose generator for Hadron Emission Reactions With Interfering Gluons; based on matrix elements, parton showers including color coherence within and between jets, and a cluster model for hadronization.
- JETSET
  The Lund string model for hadronization of parton systems.
- LDCMC
  A program which implements the Linked Dipole Chain (LDC) model for deeply inelastic scattering within the framework of ARIADNE. The LDC model is a reformulation of the CCFM model.

List of available MC program

- LEPTO

  Deep inelastic lepton-nucleon scattering based on LO electroweak cross sections (incl. lepton polarization), first order QCD matrix elements, parton showers and Lund hadronization giving complete events. Soft color interaction model gives rapidity gap events.

- PHOJET

  Multi-particle production in high energy hadron-hadron, photon-hadron, and photon-photon interactions (hadron = proton, antiproton, neutron, or pion).

- POMPYT

  Diffractive hard scattering in $p\bar{p}$, $\gamma-p$ and ep-collisions, based on pomeron flux and pomeron parton densities (several options included). Also pion exchange is included. Parton showers and Lund hadronization to give complete events.

- PYTHIA

  General purpose generator for e^+e^-, $p\bar{p}$ and ep-interactions, based on LO matrix elements, parton showers and Lund hadronization.

- RAPGAP

  A full Monte Carlo suited to describe Deep Inelastic Scattering, including diffractive DIS and LO direct and resolved processes. Also applicable for γ-production and partially for $p\bar{p}$ scattering.