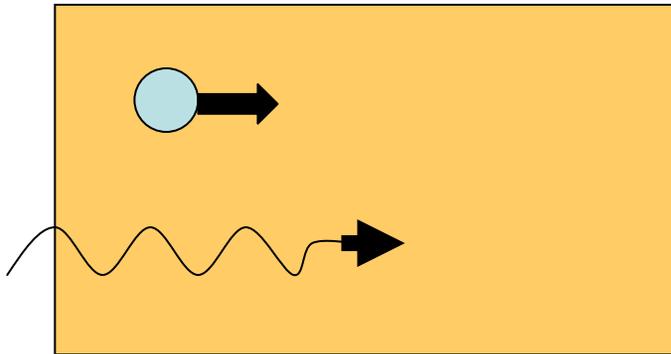


Monte Carlo ゼミ 1

2006/11/16

相互作用の確率過程 1

- 素粒子・原子核の相互作用を考えたとき、非常に大くの確率過程が存在します。
- 例えば粒子が物質中を飛んでいるとしましょう。



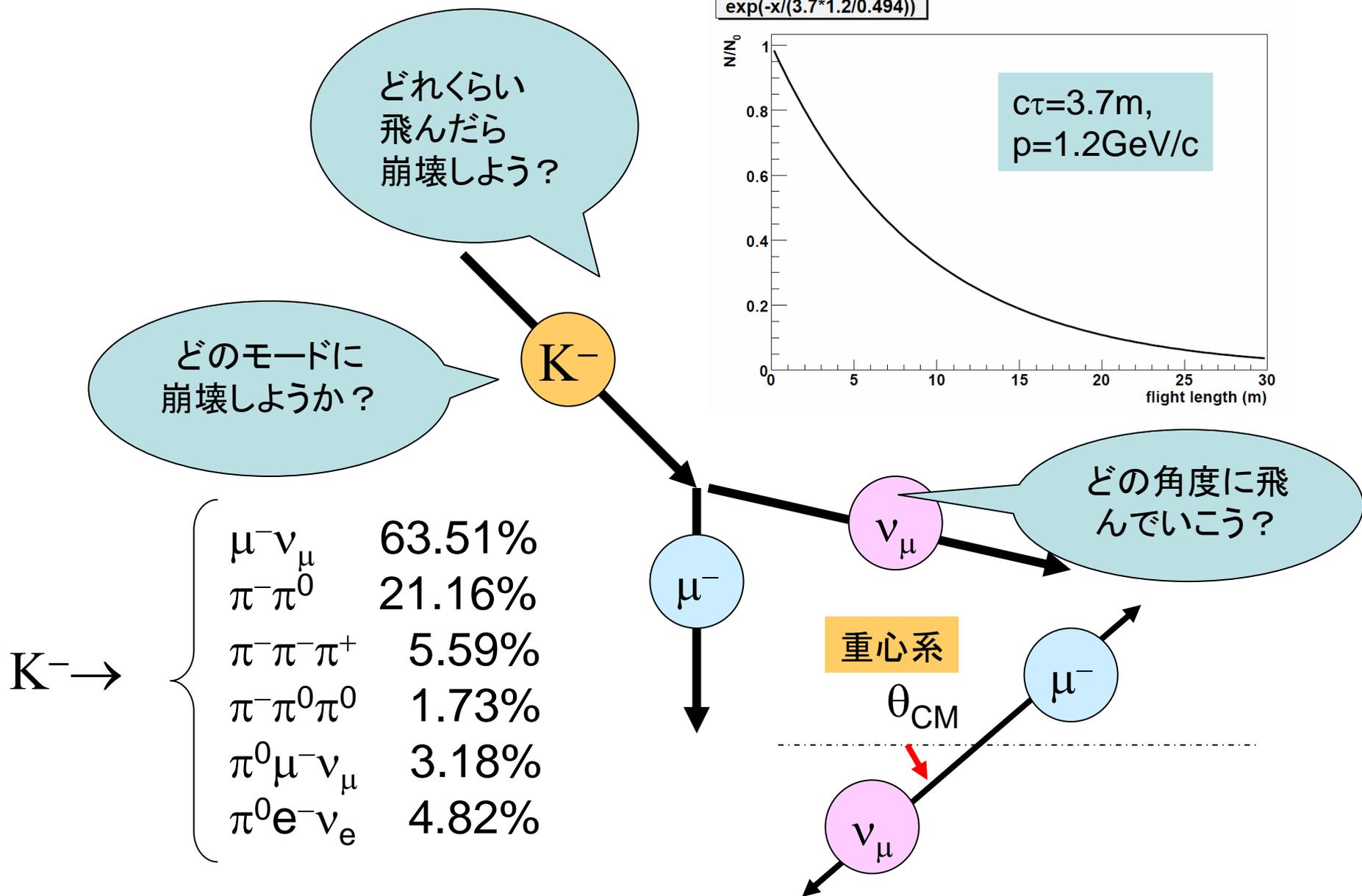
荷電粒子だったら微小距離 Δx 通ったとき

- 崩壊する？崩壊するならどのモードに出る？
- エネルギー損失はどれくらいする？
- 多重散乱で進行方向はどうなる？
- 反応を起こす？

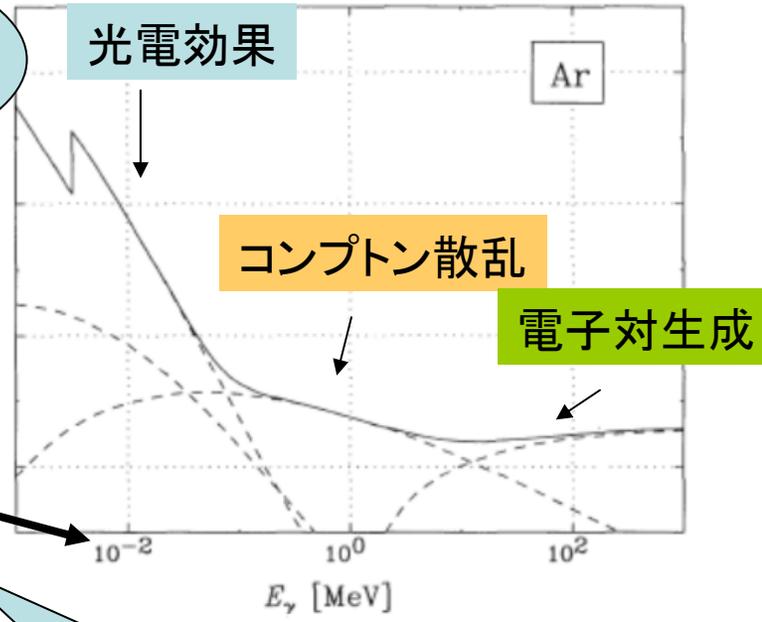
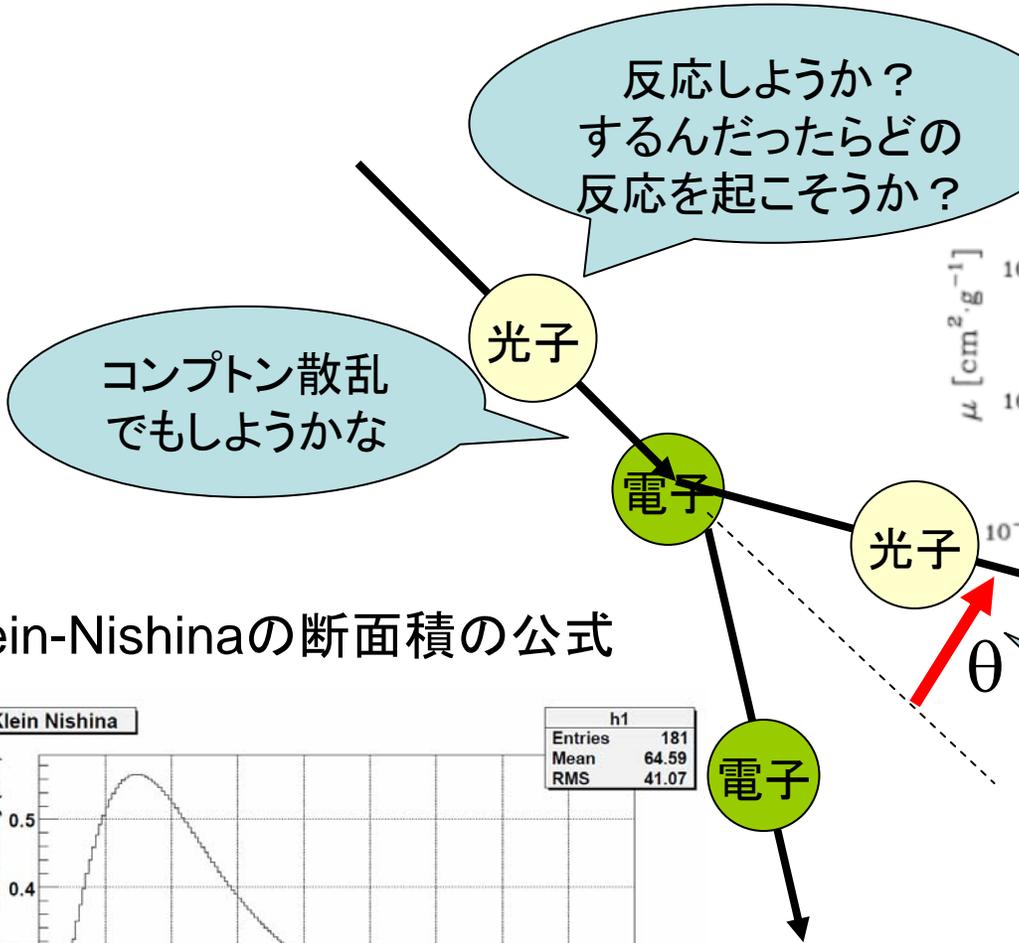
例えば光だったら

- 相互作用をする？
- 相互作用をするならどれを起こす？
 - 光電効果？
 - コンプトン散乱？
 - 電子対生成？
- コンプトンなら電子はどの角度に出る？

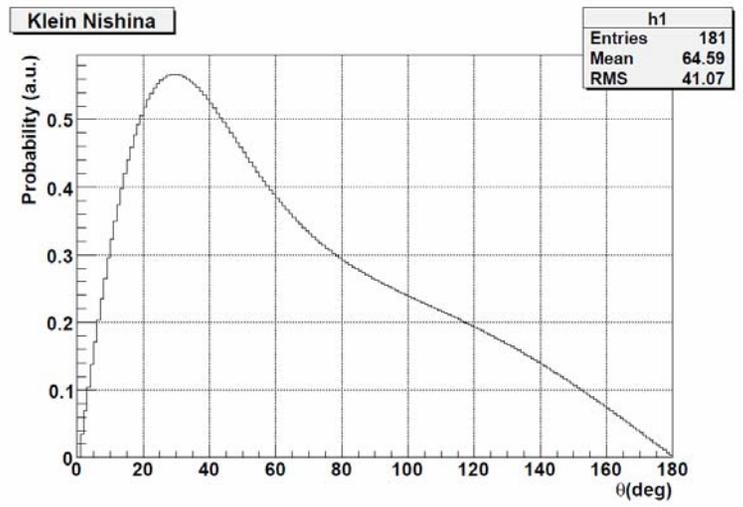
相互作用の確率過程 2



相互作用の確率過程 3



Klein-Nishinaの断面積の公式



どの角度に散乱してどれくらい電子にエネルギーを渡そうか？

物理プロセスとMonte Carlo simulation

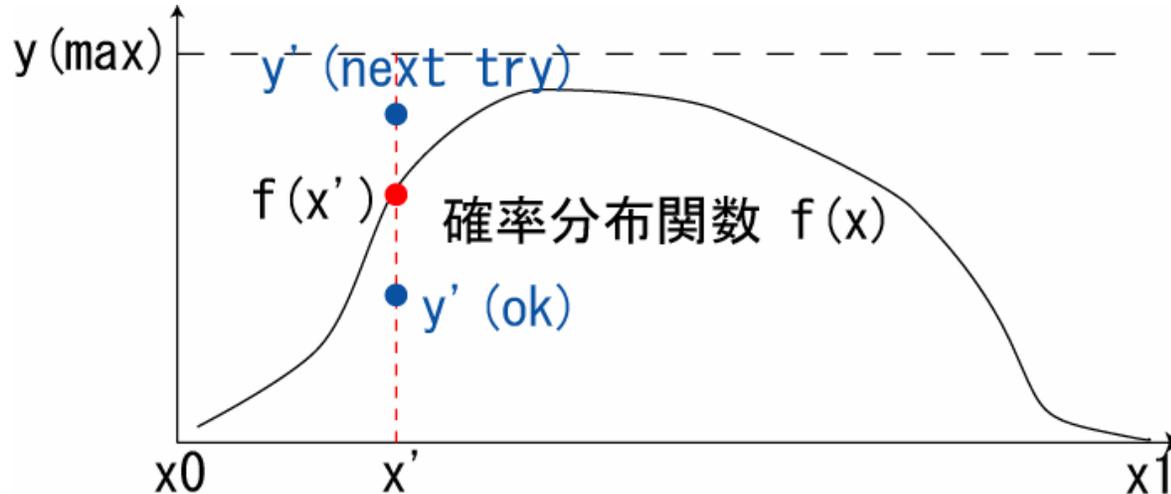
- こうして見てきたように、様々な物理プロセスは**確率分布に従って**発生します。
- このような現象を記述するプロセスが良く知られていたとしても全体を直接計算することは難しい
- このようなときに威力を発揮するのがモンテカルロ法である。モンテカルロ法は**個々のプロセスの起こる確率**、あるいは**出現確率の比**によって**乱数を発生させ**、多数回の試行を行うことによって、実際の現象をシミュレートするものである。

乱数の発生

- 様々な乱数発生関数が用意されています
 - `stdlib.h` c言語の標準関数にも
 - `int rand()` → 0~`RAND_MAX`までの整数値を一様に生成
 - `(double)rand()/((double)RAND_MAX+1.0)`で0から1まで一様に発生することができます。
- 僕らがよく使うライブラリーとしては
 - ROOT → TRandom classのUniform(double x1, x2)
 - <http://root.cern.ch/root/html/TRandom.html> 参照
 - GEANT4, CLHEP → RandFlat class
 - `RandFlat::shoot()`
 - http://lcgapp.cern.ch/doxygen/CLHEP/CLHEP_2_0_2_2/doxygen/html/class_c_l_h_e_p_1_1_hep_random.html 参照

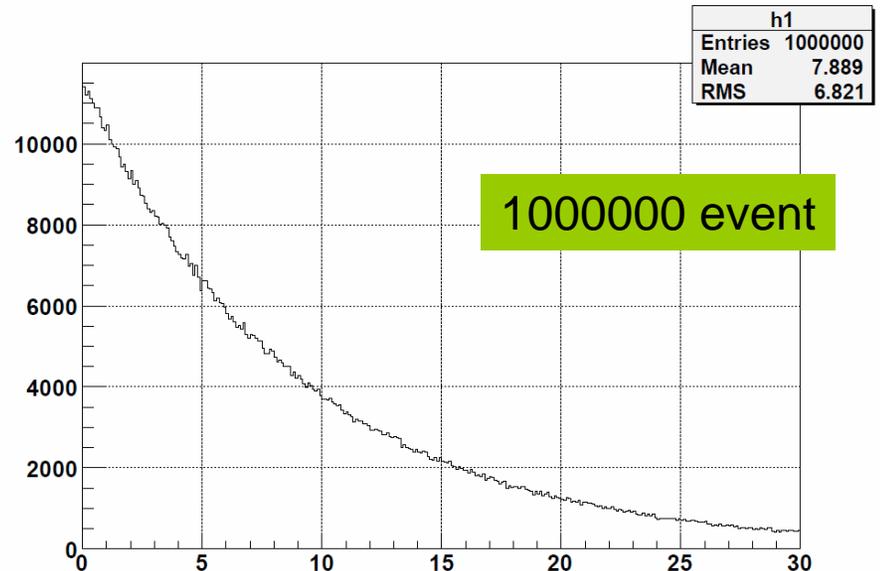
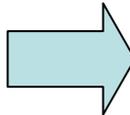
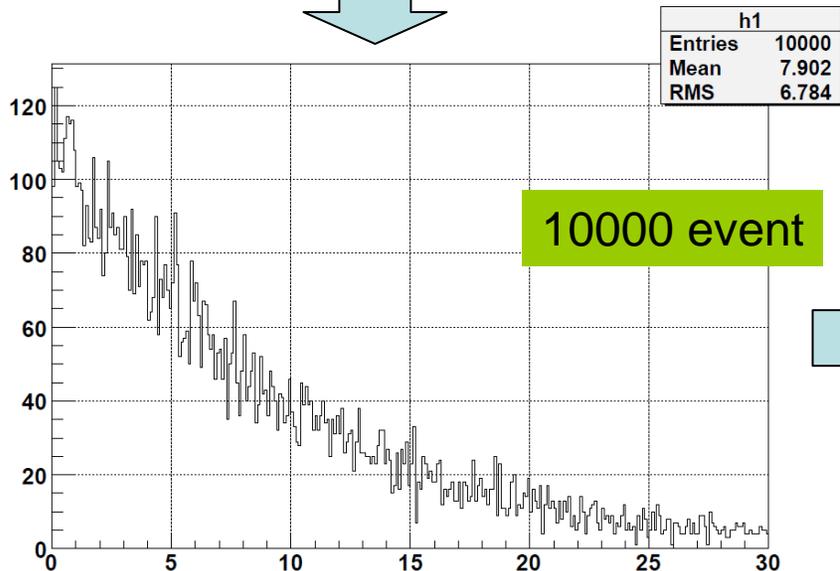
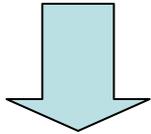
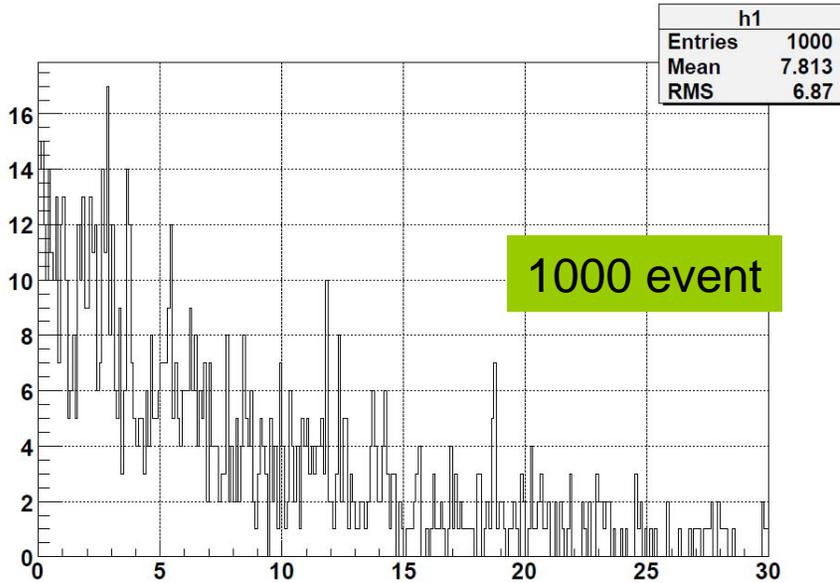
分布に従う乱数の発生

- 一様乱数から確率分布にしたがう乱数へ



1. まず $x_0 \sim x_1$ までの間で一様な乱数である値 x' を決める。
2. $0 \sim y(\max)$ まで一様な乱数である値 y' を決める。
3. $y' < f(x')$ ならこの x' を返してあげればOK。 $y' > f(x')$ ならばまた1.から繰り返す。

例えばexpの形の分布だったら



- 試行の数を増やしていくと滑らかな分布になっていく
- 作り方の参考に

練習として

- $-1 < x < 1$ の間で $f(x) = 1/2(1+x)$ の分布に従う乱数を発生させてヒストグラムにつめる。
- Poisson分布に従う乱数を発生させてみたりする。平均値は何でもOK
- 三次元空間に一様に発生させる (θ, ϕ)
- Klein-Nishinaの微分断面積にしたがって θ を発生させる

$$\frac{d\sigma}{d\Omega} = \frac{r_e^2}{2} \frac{1}{[1 + \gamma(1 - \cos \theta)]^2} \left(1 + \cos^2 \theta + \frac{\gamma^2 (1 - \cos \theta)^2}{1 + \gamma(1 - \cos \theta)} \right)$$

$$\gamma = E_\gamma / m_e c^2$$

と、つらつら書いてきたけど

- みんなはプログラムは大丈夫??
- 最終的にはGEANT4を少し身に付けてもらう予定です。
- ってことはちょっと知っという方がいい言語としては
 - C言語 (やっぱりまずこっちから)
 - C++言語 (GEANT4はこれで書きます。)
- まあ恐れることなかれ。
- はっきり言って慣れです。

業界人だったら持っときたいこの2つのプログラム (良い練習に)

- Kinematicsのプログラム

- 配布したメモを参考に

- 質量 $M1$ で運動量 $P1$ の粒子が質量 $M2$ で運動量 $P2$ の粒子にぶつかって質量 $M3$ の粒子が角度 θ へ、質量 $M4$ の粒子が角度 ϕ の方向へでました。それぞれの運動量は？ 重心系での角度は？

- エネルギーロスを計算するプログラム

- Bethe-Blochの式を用いて計算

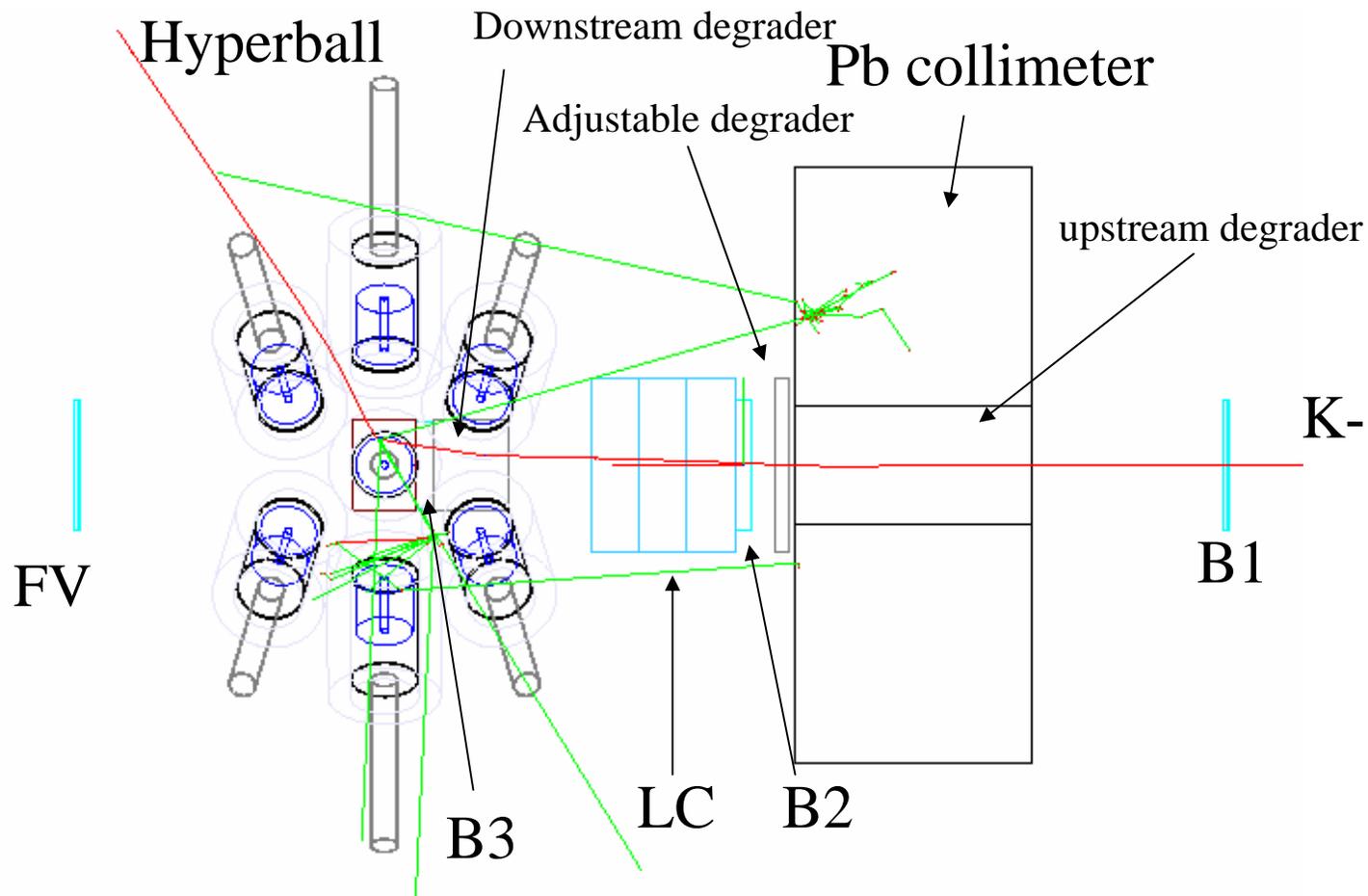
- 速度 β の粒子が物質(密度 $d(\text{g/cm}^3)$, 厚さ $t(\text{cm})$)を通過した。落とすエネルギーはどれくらい。
- 微小区間 Δx の間でのエネルギー損失を計算し、エネルギーの損失を計算し、粒子の β を計算しなおし、また微小区間でエネルギー損失を計算して、、、というのを物質を通り過ぎるまで繰り返す。

GEANTって何？

- CERNで主に開発された素粒子・原子核実験のシミュレーション用に開発されたツール
- 粒子を発生させ、様々な物理プロセスにしたがって粒子をtransportする。
 - 磁場や電場の中での粒子の運動
 - 電磁相互作用
 - ハドロン相互作用
 - 崩壊
- 検出器を配置して、その中でのエネルギー損失や、粒子の通過した場所などを調べることができる。

やってみると

- なんだか、やっぱりかっこいい。
- 物理やってる気がしてくる。
- 実験や物理を楽しむひとつのツールですね。



でも、よくよく考えてみると

- 粒子を発生させ、様々な物理プロセスにしたがって粒子をtransportする。
 - 磁場や電場の中での粒子の運動
 - 微分方程式を自分で解けば粒子の軌跡は計算できる
 - 電磁相互作用
 - 教科書に載っているプロセスで自分で計算できる
 - ハドロン相互作用
 - これも反応断面積が分かればあとは運動学を解くだけ
 - 崩壊
 - これも角度分布が分かればあとは運動学を解くだけ

GEANTは便利なツールだけれども、それを用いた結果が妥当であるかどうかをチェックできる目を身につけないといけなない。
GEANTのツールを組み合わせる使う我々はすぐにミスしてしまうから。

なんで、GEANTをやる前に

- 自分の手作りプログラムで、いくつかシミュレーションをしてみましょう。
- 今、考えてるのは
 - 611keVの γ 線に対する応答関数
 - 必要な知識は
 - コンプトン散乱と光電効果の断面積
 - 検出器の分解能
 - コンプトンの場合、電子がどの角度に散乱されるか
 - でつくれるはずだよね。
- まあ、ちょっと考えといてみてください。