

Geant 4

A Short Course on Geant4 Simulation Toolkit

Introduction

<http://cern.ch/geant4>

The full set of lecture notes of this Geant4 Course is available at
<http://www.ge.infn.it/geant4/events/nss2004/geant4course.html>

Main Subjects of This Lecture

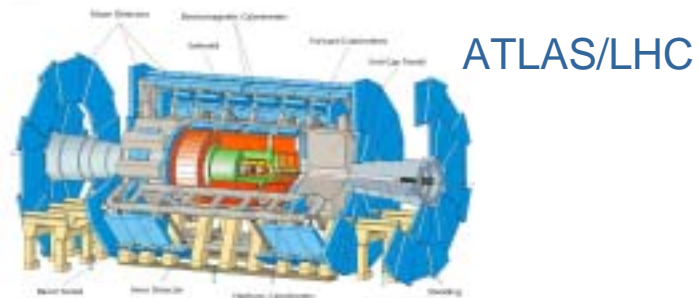
- You will hear details of the toolkit after my talk:
- The main subjects of this lecture:
 - *Brief overview of basic concepts in the Monte Carlo simulation of particle interactions with matter*
 - *Geant4 vision: scope, fundamental concepts*

What is Geant4?

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

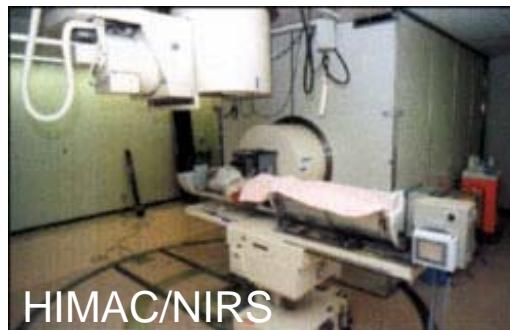
■ It is for detector simulation of research in

- High energy physics
- Nuclear physics
- Cosmic ray physics



■ It is also for application in

- Space science
- Radiological science
- Radiation background calculation
- etc



Detector Simulation - General

- General characteristics of a particle detector simulation program:
 - ✓ You specify the *geometry of a detector*.
 - ✓ Then the program automatically *transports* the particle you injected to the detector by simulating the *particle interactions in matter* based on the Monte Carlo method.

The heart of the simulation

→ *The Monte Carlo method to simulate the particle interactions in matter*

Chapter 1

Basic concepts in the Monte Carlo simulation of particle interactions with matter.

What is Monte Carlo Method? - 1

- A method to search for solutions to mathematical problem using a statistical sampling with random numbers.
- This method was developed by *Stanislaw Ulam* while he committed the hydrogen bomb project at Los Alamos Laboratory after World War II.

- ***Although the method is applied these days to a wide spectrum of problems, it is worth to know that it was developed by a mathematician who tried to solve a physics problem in hydrodynamics of radiation.***



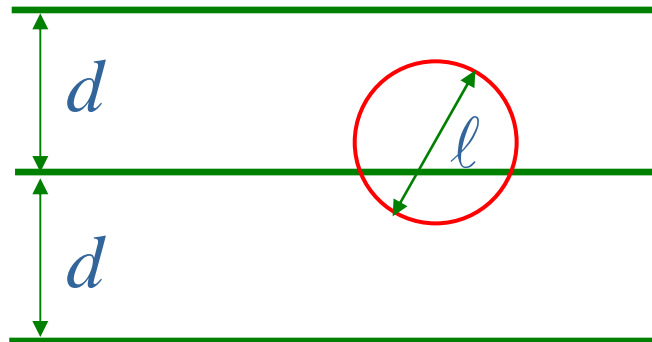
Stanislaw Ulam 1909~1984

What is Monte Carlo Method? - 2

- Historical example of the MC method is **Buffon's needle**
 - Throw a needle randomly on a sheet on which parallel lines with an equal distance are drawn.
 - Counts the number of throwing which makes the needle crossing the parallel lines.



Georges Buffon
(1707 ~ 1788)



For $d = l$

$$N_{throw} / N_{cross} \sim \pi / 2$$

You can get π by random throws.

MC Simulation of Particle Interactions with Matter - 1

■ Basic concept : The exponential law

$P(x)$: probability of not having an interaction after a distance x
 $w dx$: probability to having an interaction between x and $x+dx$

$$w = N \cdot \sigma$$

N : Number of target particles per unit volume

σ : Interaction cross section

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



Probability of no-interaction in dx
Probability of no-interaction up to x

$$\therefore P(x) = \exp(-w x) \quad (P(0) = 1)$$

← **Probability distribution function**

= Exponential distribution

Ref: W.R.Leo, "Techniques for Nuclear and Particle Physics Experiments"

IEEE NSS-MIC 2004

MC Simulation of Particle Interactions with Matter - 2

■ Generation of interactions

The probability of interaction, $P_{int}(x)$, between $x \sim x+dx$ is

$$P_{int}(x)dx = P(x)w dx$$

$P_{int}(x)$: **Probability Density Function (PDF)**

The **cumulative distribution function (CDF)** is

$$\begin{aligned} \int P_{int}(x)dx &= \int P(x)w dx = \int w \exp(-wx) dx \\ &= 1 - \exp(-wx) \end{aligned}$$

Then you can generate an interaction using the inverse method:

$$\eta = 1 - \exp(-wx)$$

η : **Uniform random number of [0,1]**

$$x = -\ln(1 - \eta) / w$$

MC Simulation of Particle Interactions with Matter - 3

- Generation of interactions in heterogeneous matter

$$x = -\ln(1 - \eta) / w$$

' x ' has the dimension of '**length**' and depends on material. Therefore the sampling depends on material.

However, the following sampling is independent of material:

$$x w = -\ln(1 - \eta)$$

Therefore we introduce the '**mean free path**' λ as

$$\lambda = \int x P(x) dx / \int P(x) dx = 1 / w$$

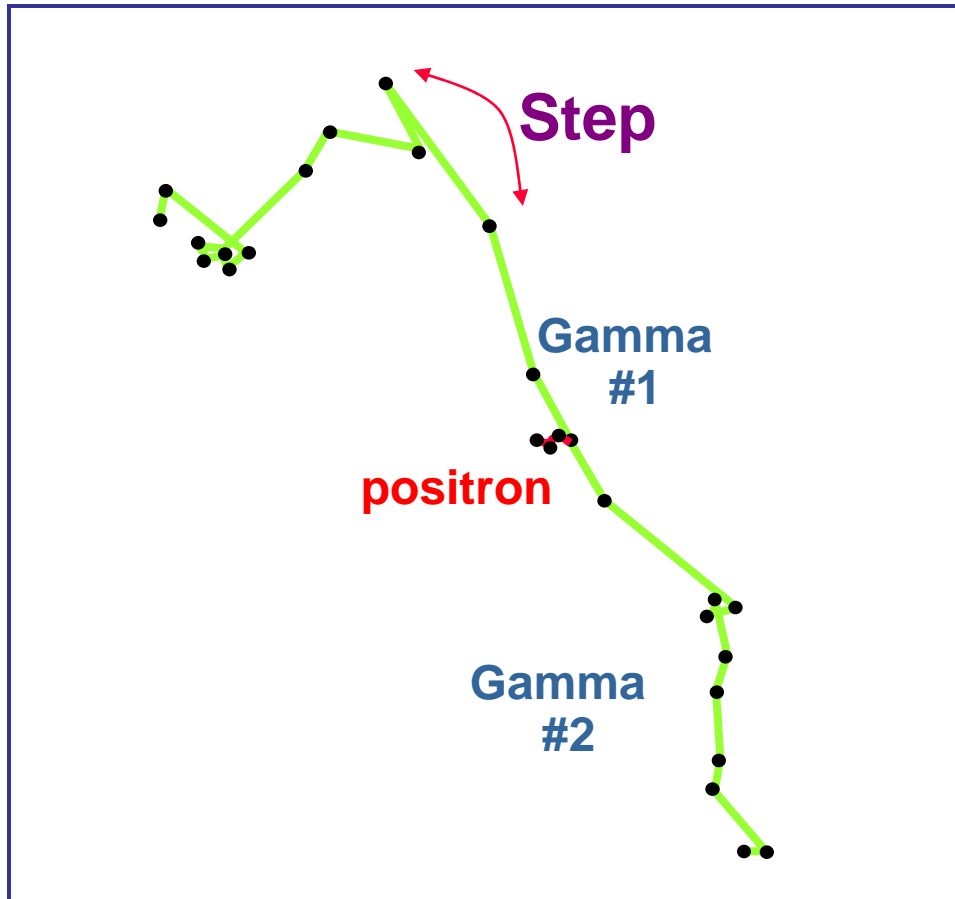
Then we can sample *in the material independent way* by measuring the length in the unit of λ .

$$\underline{x / \lambda = -\ln(1 - \eta)}$$

← **Number of Mean Free Path (NMFP)**

Particle Transportation - Introduction

- A particle is transported in the *stepwise* manner.



Example:

**Annihilation of
the 8MeV
positron in
water**

Particle Transportation: How to Determine a Step - 1

- 1 At the beginning of a step, the **NMFP** (Number of Mean Free Path) for each physics process, which is associated to the particle, is sampled by the *material independent way*.

Example

The positron has the following physics processes. For each of these processes, assigns **NMFP** by the exponential law of interactions.

■ Bremsstrahlung

$$\text{NMFP} = N_{\text{brem}}$$

■ Ionization

$$\text{NMFP} = N_{\text{ion}}$$

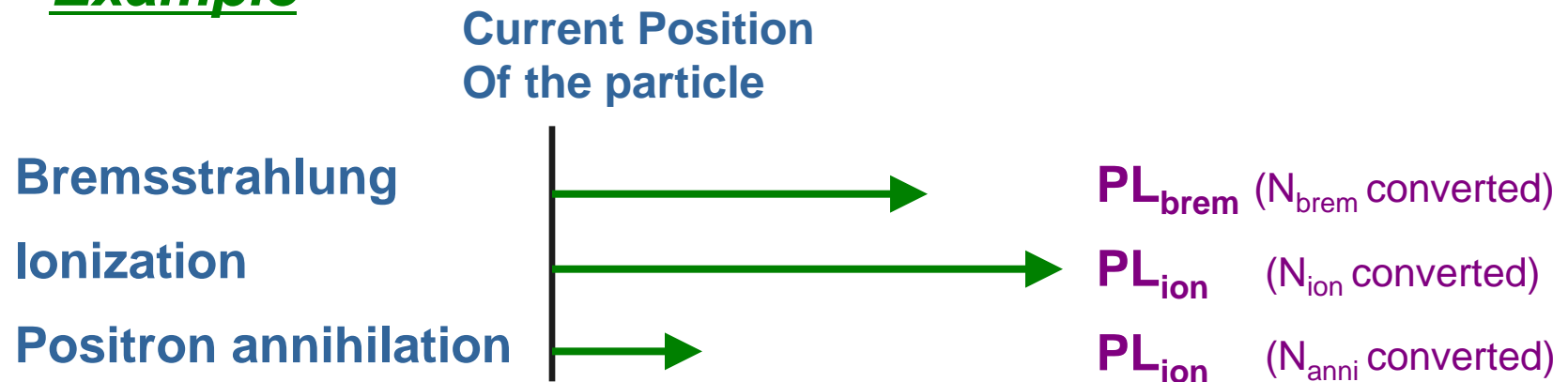
■ Positron annihilation

$$\text{NMFP} = N_{\text{anni}}$$

Particle Transportation: How to Determine a Step - 2

- Using the cross-section in the material where the particle is currently in, converts the each NMFP to the physical length (PL):

Example



- The process which has the minimum **PL** determines the step length.

→ **‘Positron annihilation’** in the above example.

Particle Transportation - continued

- ④ Transports the particle for the determined step.
- ⑤ If the particle is still alive after the interaction, do the sampling again for all **NMFPs**, and continue the transportation.
- ⑥ If the particle disappears after the interaction, then the transportation is terminated.

Chapter 2

Geant4 vision:

Scope and fundamental concepts

What Geant4 Can Do for You?

- Transports a particle step-by-step by taking into account the interactions with materials and external electro-magnetic field until the particle
 - **loses its kinetic energy to zero,**
 - **disappears by an interaction,**
 - **comes to the end of the simulation volume (end of the world).**
- Provides a way the user intervenes the transportation process and grabs the simulation results
 - **at the beginning and end of transportation,**
 - **at the end of each stepping in a transportation,**
 - **at the time when the particle going into the sensitive volume of the detector,**
 - **etc.**

← *These are called “User Actions”.*

What You Have to Do for Geant4?

- Three indispensable information you have to prepare:
 - Geometrical information of the detector
 - Choice of physics processes
 - Kinematical information of particles which go into the detector

- Auxiliary you have to prepare:
 - Magnetic and electric field
 - Actions you want to take when you intervene the particle transportation
 - Actions you want to take when a particle goes into a sensitive volume of the detector
 - etc.

Tools for Input Preparation

Geant4 provides standard tools to help you to prepare input information.

- Multiple choices to describe the detector geometry
 - Combining basic geometry elements (box, cylinder, trapezoid, etc)
 - Representation by surface planes
 - Representation by boolean operation, etc
- Standard way to define materials in the detector
 - A large collection of examples to define various materials
- A set of wide variety of particles
 - Standard elementary particles (electron, muon, proton,.....)
 - Unstable particles (resonances, quarks, ...)
 - Ions
 - Exotic particles (geantino, charged geantino)

Choice of Physics Processes

Geant4 provides a wide variety of physics models of particle interactions with matter you can select.

■ Category of physics processes

- Standard electromagnetic processes
- Low energy electromagnetic processes
- Hadronic processes

■ How to use physics processes

- A rich samples of *Physics List* provided with example applications.
- Recommended *Physics List* (educated guess) for hadronic.

Tools to Help Your Simulation

■ User interface

- **Interactive mode with character terminal or GUI**
- **Batch mode**

■ Visualization

- **Trajectories of a particle and its all 2ndary**
- **Detector geometry**

■ Debugging

- **Controllable verbose outputs from the kernel during the transportation**
- **Errors in the geometry definition, etc**

■ Data analysis

Minimum Software Knowledge to Use Geant4

■ C++

- Geant4 is purely implemented in C++, therefore a basic knowledge of C++ is mandatory.
- C++ is a complex language, therefore you are not required to be a C++ expert

■ Unix/Linux

- Unix/Linux is a standard working environment for Geant4, therefore a minimum knowledge/experience is required
 - How to use basic unix command (cp, mv, rm,
 - How to “make” a C++ program.
- Windows?
 - You can use Visual C++
 - Though still you need some knowledge of Unix (cygwin) for installation

Chapter 3

Additional Information of Geant4

Brief History of Geant4

■ Pre-R&D Phase

1993

- Study of OO redesigning of GEANT3 both at CERN and KEK

■ R&D Phase

Dec. 1994

- Submitted a R&D proposal to CERN – *the birth of Geant4!*

Dec. 1995

- The basic design and a prototype implementation completed
- The number of R&D members expanded to ~100 from 15 countries

1996~1998

- α -release, β -release

Dec. 1998

- Version1 released. The R&D phase finished

■ Geant4 Collaboration Phase

Dec. 1998

- The Geant4 collaboration based on “MoU” started

2004 *The 10th anniversary!*

How It Has Been Developed

- Development based totally on the object-oriented software technology
 - ← A pilot project (~10 years ago!) to move from the procedural to the object-oriented approach in HEP
- Benefit from experience and the algorithmic techniques accumulated in GEANT3.
 - ← Avoid to reinvent the wheel
 - ← Redesigned from scratch in OO
- Worldwide collaboration with distributed software development



Geant4 Collaboration Map



Stanford
Linear
Accelerator
Center

