Пакет моделирования Geant4

А.С.Жемчугов
ОИЯИ
E-mail: zhemchugov@jinr.ru

2023

Цели моделирования

- При планировании эксперимента
  - Оптимизация конструкции детектора
  - Отладка алгоритмов реконструкции событий
  - Расчет ожидаемых значений сигнала и фоновых процессов. Оценка ожидаемой точности измерений
- При анализе данных
  - оптимизация процедуры анализа данных
  - определение аксептанса установки
  - определение вклада фоновых процессов
  - оценка систематических погрешностей
  - сравнение результатов анализа с теоретическими предсказаниями

Метод Монте-Карло

Метод Монте-Карло – это численный метод решения прикладных математических задач при помощи моделирования случайных величин и статистической оценки их характеристик.

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION

Number 247 SEPTEMBER 1949 *Volume 44*

THE MONTE CARLO METHOD

NICHOLAS METROPOLIS AND S. ULAM
Los Alamos Laboratory

We shall present here the motivation and a general description of a method dealing with a class of problems in mathematical physics. The method is, essentially, a statistical approach to the study of differential equations, or more generally, of integro-differential equations that occur in various branches of the natural sciences.

Метод появился при работе над Манхэттенским проектом:

- **S.M. Ulam, J. von Neumann, “On combination of stochastic and deterministic processes”. Bull. Amer. Math. Soc. 53 1120 (1947)**
- **S.M. Ulam, N. Metropolis, “The Monte-Carlo method”, J. Amer. Statist. Assoc. 1949 , 44 Vol 247, 335-341**

Как это работает

- Входные данные (объекты) стохастического процесса генерируются случайным образом согласно известной (например, теоретически предсказанной) функции плотности вероятности

- Детерминистически моделируются индивидуальные истории объектов

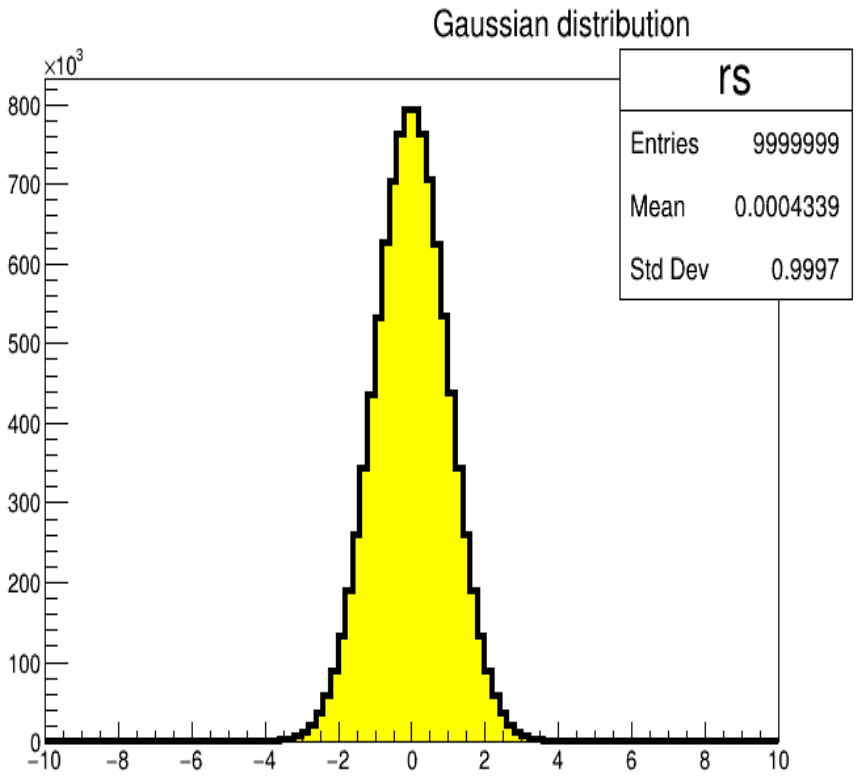
- Получившаяся в результате функция плотности вероятности описывает результат смоделированного стохастического процесса

Вспомним теорвер

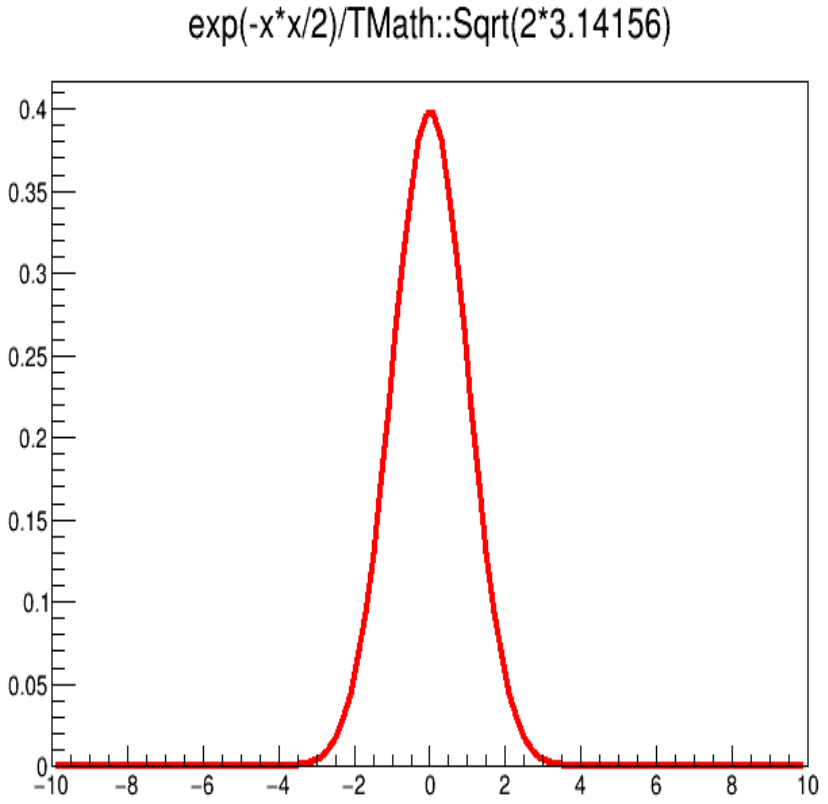
- **случайные величины**
  - дискретные
  - непрерывные
- **распределение** показывает частоту наступления различных событий **в выборке (sample)**
- **вероятность** это число от 0 до 1, которое показывает насколько возможен данный исход
- **функция плотности вероятности (p.d.f.)** показывает, как полная вероятность 1 распределена между **всеми** возможными исходами

Вспомним теорвер

Распределение



p.d.f.



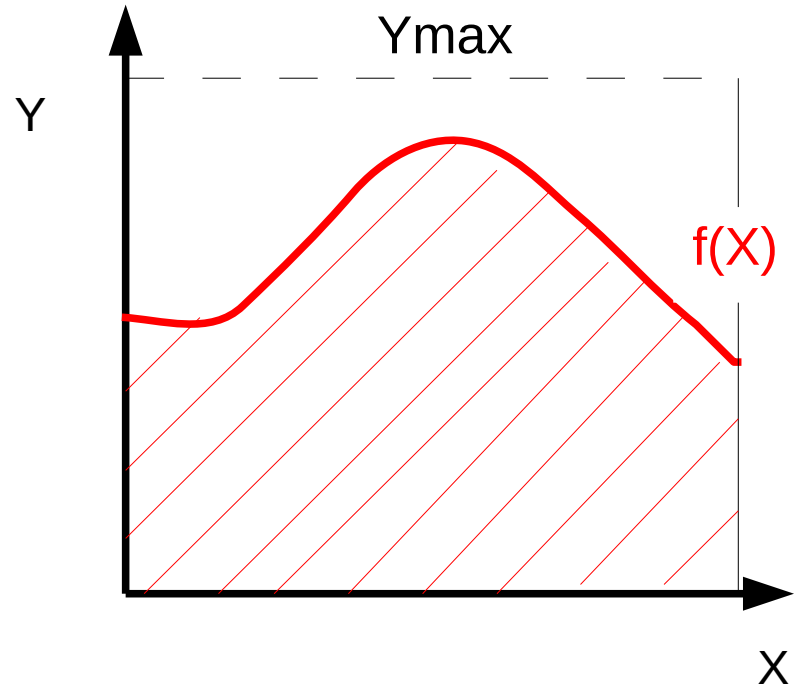
Как это работает

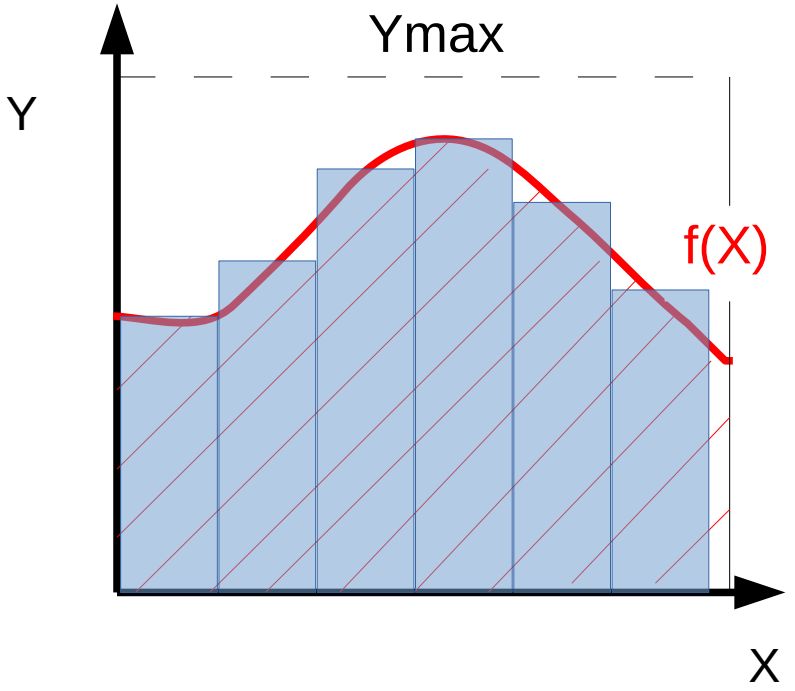
- Входные данные (объекты) стохастического процесса генерируются случайным образом согласно известной (например, теоретически предсказанной) функции плотности вероятности

- Детерминистически моделируются индивидуальные истории объектов

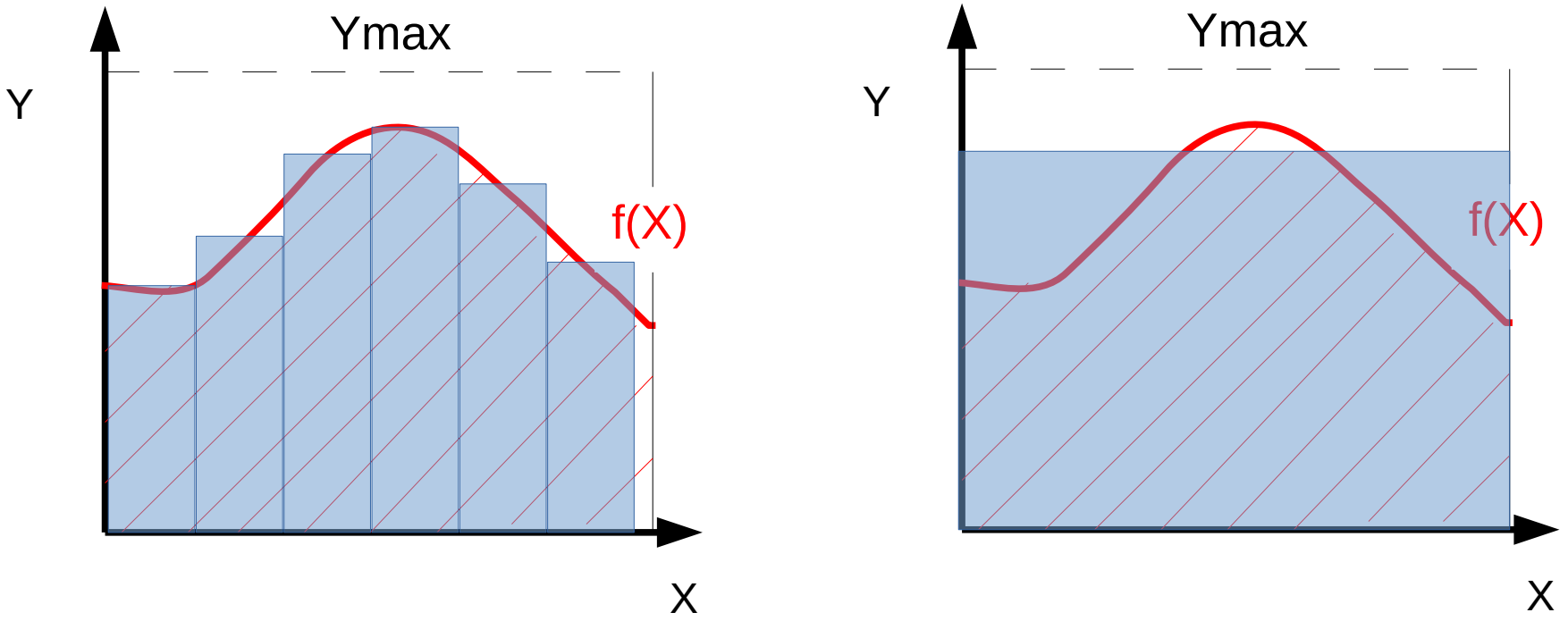
- Получившаяся в результате функция плотности вероятности описывает результат смоделированного стохастического процесса

- Математический смысл: эффективный способ вычисления многомерных интегралов со сложными пределами интегрирования
- Пример:

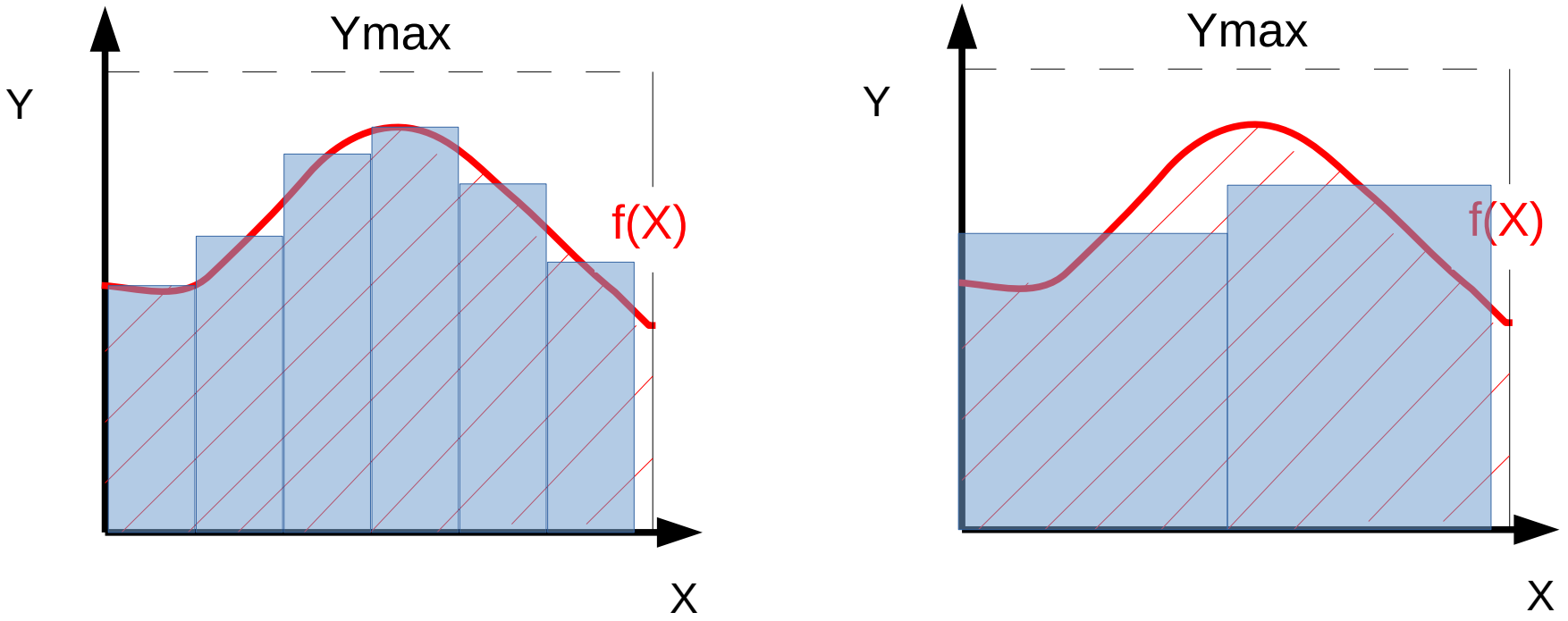




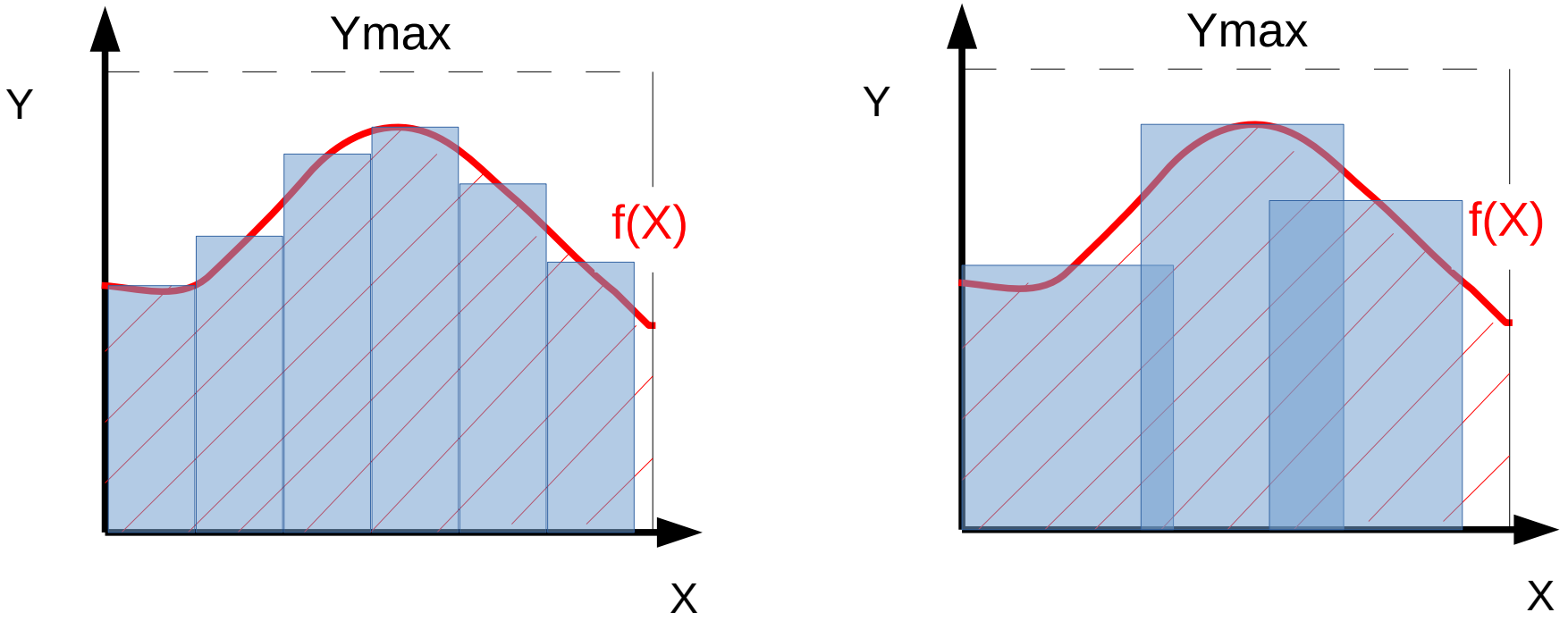
- Математический смысл: эффективный способ вычисления многомерных интегралов со сложными пределами интегрирования
- Пример:



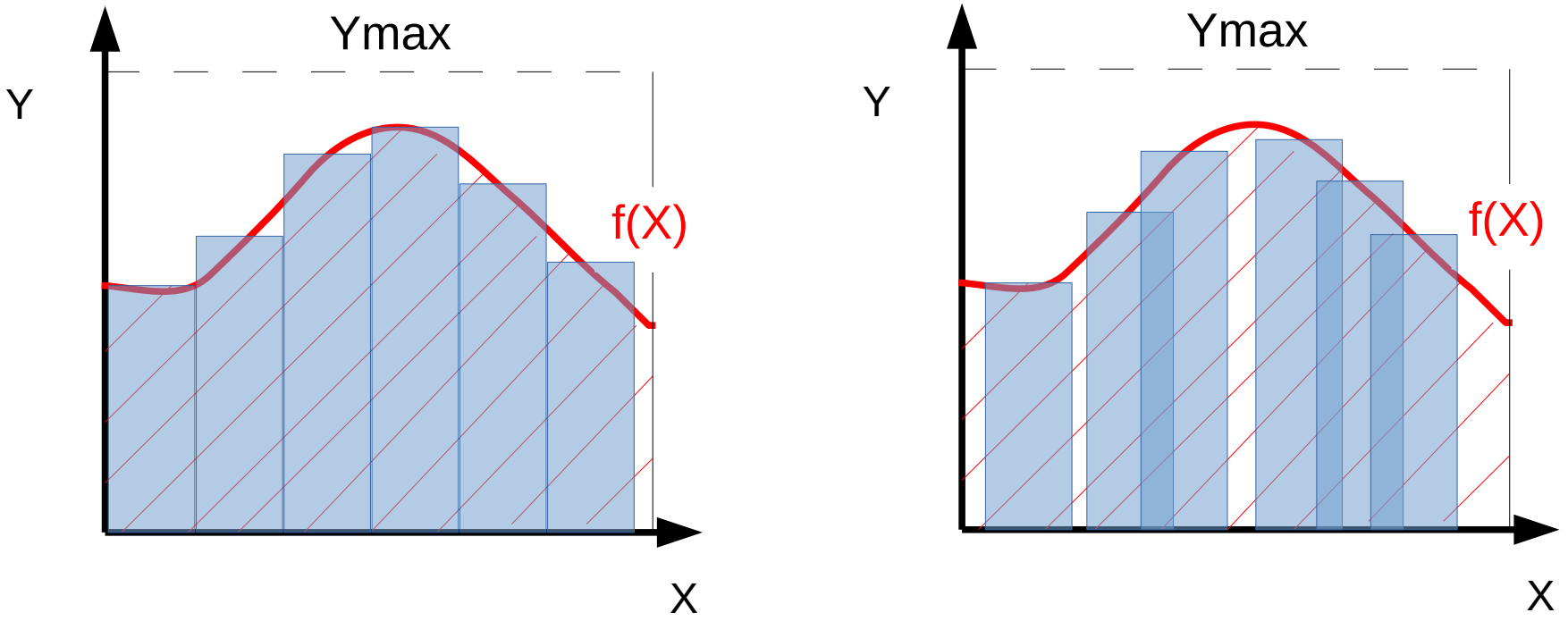
- Математический смысл: эффективный способ вычисления многомерных интегралов со сложными пределами интегрирования
- Пример:



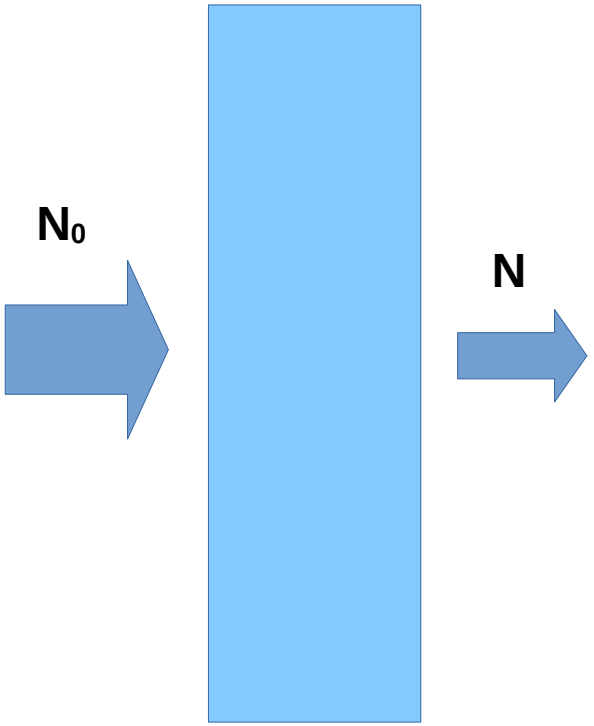
- Математический смысл: эффективный способ вычисления многомерных интегралов со сложными пределами интегрирования
- Пример:



- Математический смысл: эффективный способ вычисления многомерных интегралов со сложными пределами интегрирования
- Пример:



- Математический смысл: эффективный способ вычисления многомерных интегралов со сложными пределами интегрирования
- Пример:



Поглощение излучения в веществе:

$dN = -\mu N\, dx$ $\qquad$ $N = N_0 \exp(-\mu x)$

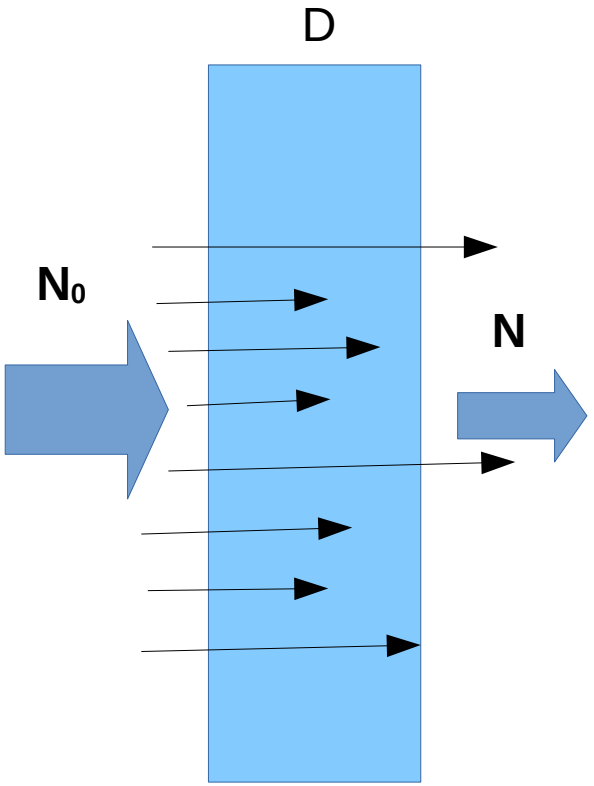
Если μ зависит от (x,y,z,E) $\qquad$ **??**

Если N зависит от (E, $\bar{r}$) $\qquad$ **???**

Если учесть рассеяние $\qquad$ **????**

Если μ нельзя задать аналитически ?????

Решение методом Монте-Карло

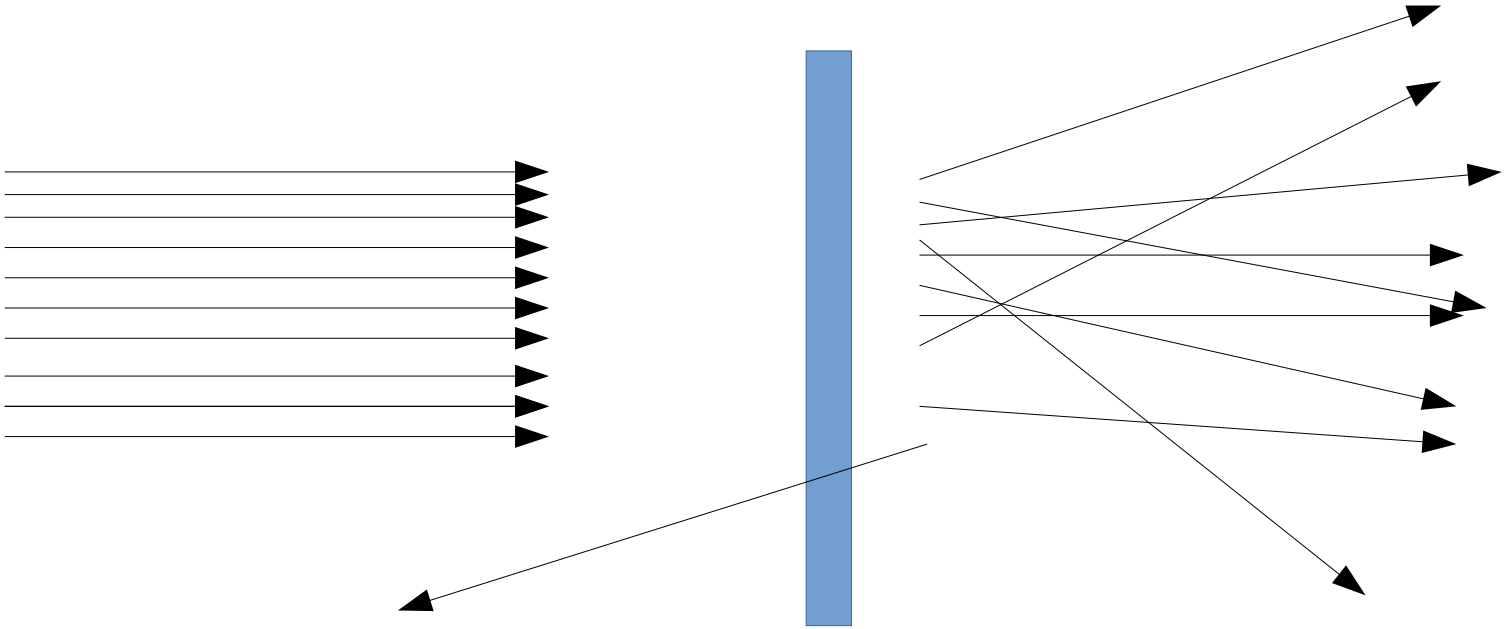


- По определению полного сечения, вероятность взаимодействия на пути dx равна σdx
- Рассмотрим пробег частицы в веществе как случайную величину, распределенную с плотностью вероятности $p(x)$:

$$p(x)dx = \{1 - \int_0^x p(y)dy\}\ \sigma dx$$

- Решение: $p(x) = \sigma \exp(-\sigma x)$
- Разыграем N траекторий частиц, для каждой вычислим пробег x согласно известной $p(x)$
- Если $x>D$, частица прошла через вещество

Еще пример: опыт Резерфорда



$$\frac{d\sigma}{d\Omega} = \left(\frac{Z_1 Z_2 e^2}{2mv^2}\right)^2 \frac{1}{\sin^4 \frac{\Theta}{2}}$$

~ вероятность рассеяния на угол Θ

Шаги моделирования:

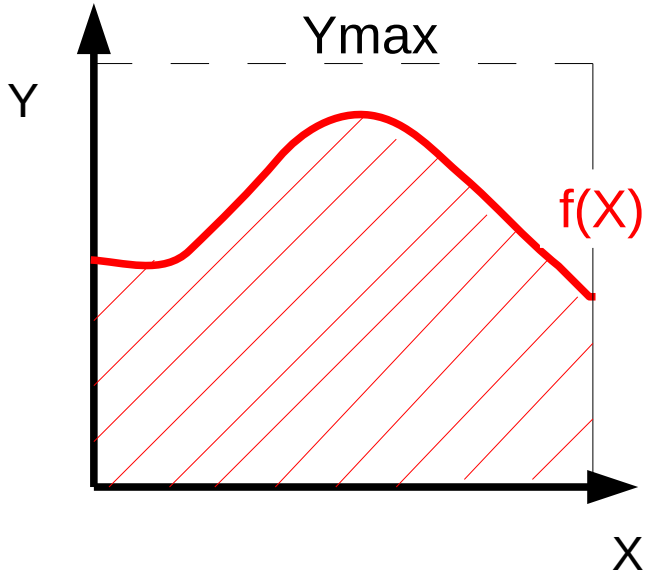
1. Генерируется пучок из N частиц
2. Для каждой частицы вычисляется случайный угол рассеяния Θ в соответствии с известным распределением и заданной скоростью частиц
3. Для каждой частицы вычисляется случайный азимутальный угол φ, который разыгрывается равномерно в интервале [0,2π]
4. Рассеяние частиц смоделировано

Генераторы псевдослучайных чисел

- Программы, генерирующие последовательность чисел, похожих на случайные.
- Последовательность однозначно определяется начальным значением (seed) и полностью повторяется через определенный период
- Основные алгоритмы
  - RANLUX (алгоритм Marsaglia & Zaman, см. M. Lüscher, Comp. Phys. Comm. 79 (1994) 100) — период $5*10^{171}$
  - RANECU (алгоритм L'Ecuyer см. l'Ecuyer, Commun. ACM 31(1988) 742) — период $2*10^{18}$
  - RANMAR (алгоритм Marsaglia-Zaman-Tsang, см. G. Marsaglia, A. Zaman and W.-W. Tsang, Stat. Prob. Lett. 9 (1990) 35.) — период $2^{144}=2*10^{43}$
  - Mersenne Twister (M. Matsumoto and T. Nishimura, ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, January 1998, pp 3–30) - период $(2^{19937}-1)$, медленнее RANECU и намного быстрее RANLUX

Как смоделировать произвольное распределение?

- Равномерные распределения случайных чисел получаются при помощи программ - генераторов псевдослучайных чисел
- Метод выбраковки (“Hit-and-miss”) - наиболее простой способ моделирования произвольного распределения f(x) при помощи равномерно распределенных случайных чисел



1. Выбирается случайное X в интервале [0, Xmax]
2. Выбирается случайное Y в интервале [0, Ymax]
3. X принимается при условии Y< f(X)

Программы для моделирования

Существует большое количество уже написанных программ и программных пакетов для моделирования физических процессов с участием элементарных частиц с помощью метода Монте-Карло:

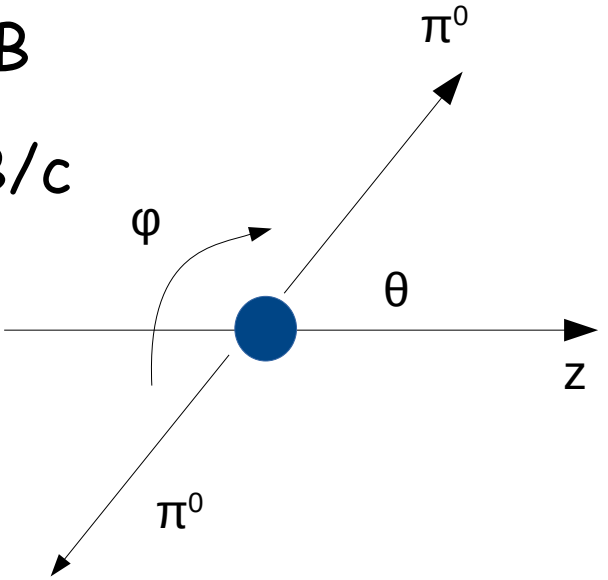
- Генераторы событий
- Специализированные программы
- Универсальные программы (Geant4, FLUKA, MCNP, MARS)

Генераторы событий

- Позволяют моделировать наблюдаемые величины (импульс, энергия, точка рождения ...) на основе теоретических предсказаний.
- Искажение наблюдаемых величин, связанное с прохождением частиц через вещество, не учитывается
- Генератор событий - это «мост» между теоретиками и экспериментаторами.
- Генераторов существует великое множество. Только на LHC применяются несколько десятков генераторов. Самый известный, пожалуй, PYTHIA.
- Давайте сделаем свой генератор?

Простой генератор: распад каона $K^0 \to 2\pi^0$

- Масса каона 498 МэВ, масса пиона 135 МэВ
- Для определенности, импульс каона 5 ГэВ/с
- Перейдем в СЦМ:
  - $E_\pi = M_K/2$
  - $P_\pi = sqrt(E_\pi^2 - M_\pi^2)$



- Разыграем направление одного из пионов (изотропно)
  - $\cos(\theta)$ = RNDM[-1,1]
  - φ = RNDM [0, 2π]
- Зададим направление второго пиона противоположно первому
- Перейдем в лабораторную систему при помощи преобразования Лоренца

Как это выглядит на C++

Пример kaon.C на http://uc-moodle.jinr.ru

```
{

  double mkaon = 498; // MeV
  double mpion = 135; // MeV
  double pkaon = 5000; // MeV/c
  double beta = pkaon/sqrt(pkaon*pkaon+mkaon*mkaon);

  int i;
  for (i=0; i < 100000 ; i++) // 100000 events
   {
     // calculate kinematics
     double epion = mkaon/2.;
     double ppion = sqrt(epion*epion - mpion*mpion);
     double costheta = 2*gRandom->Rndm() - 1; // [-1,1]
     double sintheta = sqrt(1-costheta*costheta);
     double phi = gRandom->Rndm()*2*3.14159;  // [0,2pi]

     // calculate momentum of the 1st pion
     double px1 = ppion*sintheta*cos(phi);
     double py1 = ppion*sintheta*sin(phi);
     double pz1 = ppion*costheta;

     // calculate momentum of the 2nd pion
     double px2 = -px1;
     double py2 = -py1;
     double pz2 = -pz1;

     // make 4-vectors
     TLorentzVector pion1 (px1, py1, pz1, epion);
     TLorentzVector pion2 (px2, py2, pz2, epion);
     // Lorentz boost
     pion1.Boost(0., 0., beta);
     pion2.Boost(0., 0., beta);
   }

}
```

Добавим сохранение в дерево ROOT

```
{
 double mkaon = 498; // MeV
 double mpion = 135; // MeV
 double pkaon = 5000; // MeV/c
 double beta = pkaon/sqrt(pkaon*pkaon+mkaon*mkaon);
 int i;
 for (i=0; i < 100000 ; i++) // 100000 events
  {
    // calculate kinematics
    double epion = mkaon/2.;
    double ppion = sqrt(epion*epion - mpion*mpion);
    double costheta = 2*gRandom->Rndm() - 1; // [-1,1]
    double sintheta = sqrt(1-costheta*costheta);
    double phi = gRandom->Rndm()*2*3.14159;  // [0,2pi]

    // calculate momentum of the 1st pion
    double px1 = ppion*sintheta*cos(phi);
    double py1 = ppion*sintheta*sin(phi);
    double pz1 = ppion*costheta;

    // calculate momentum of the 2nd pion
    double px2 = -px1;
    double py2 = -py1;
    double pz2 = -pz1;

    // make 4-vectors
    TLorentzVector pion1 (px1, py1, pz1, epion);
    TLorentzVector pion2 (px2, py2, pz2, epion);
    // Lorentz boost
    pion1.Boost(0., 0., beta);
    pion2.Boost(0., 0., beta);

  }
}
```

```
// open ROOT file
TFile fout("kaon.root","RECREATE");
TTree* tree = new TTree("kaon","kaon");

 // initialize ROOT tree
double p1[4], p2[4];
tree->Branch("p1",p1,"p1[4]/D");
tree->Branch("p2",p2,"p2[4]/D");
```

```
// Fill the tree
p1[0] = pion1.Px();  p2[0] = pion2.Px();
p1[1] = pion1.Py();  p2[1] = pion2.Py();
p1[2] = pion1.Pz();  p2[2] = pion2.Pz();
p1[3] = pion1.E();   p2[3] = pion2.E();
tree->Fill();
```

```
// Save file
tree->Write();
fout.Close();
```

Задача повышенной сложности: смоделируйте распад на три частицы?

Пакет GEANT4

Исходный код GEANT4

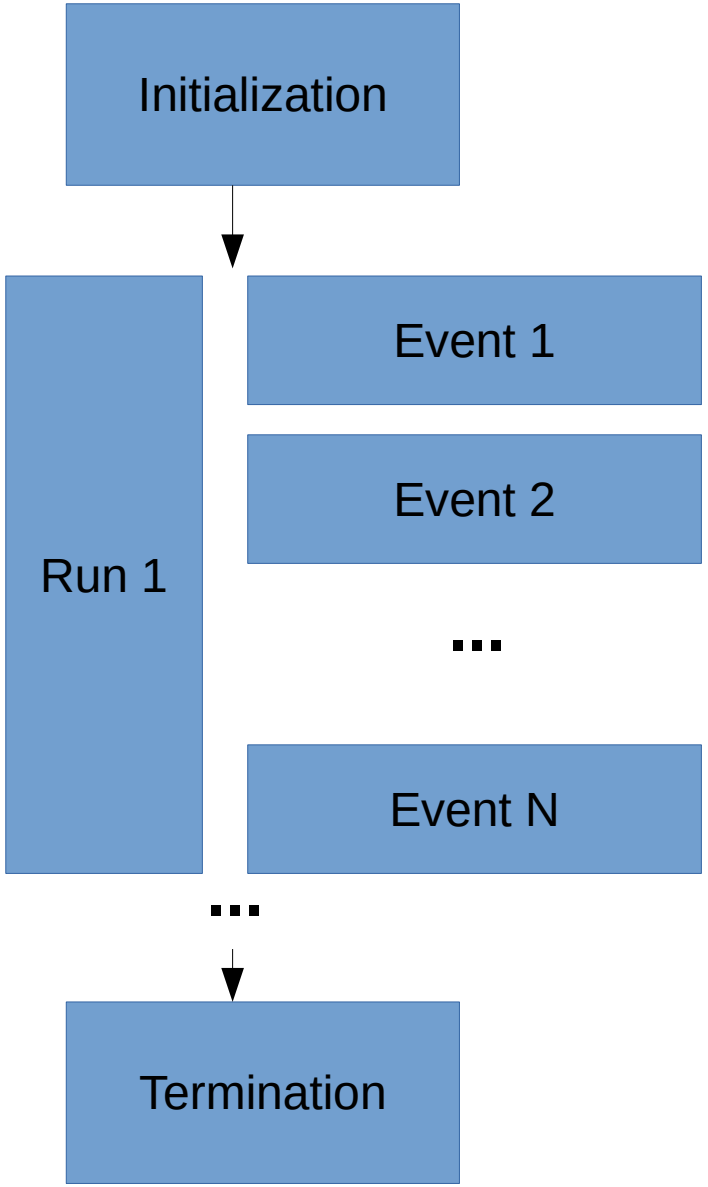
- GEANT4 разрабатывается международной коллаборацией GEANT4
- Это открытое ПО, которое свободно распространяется по лицензии GEANT4
- Исходный код, сборки и документация доступны на http://cern.ch/geant4
- Написан на C++
- Собирается и работает на Linux, Windows and MacOS
- Ссылки:

S. Agostinelli et al., Geant4: a simulation toolkit, NIM A 506 (2003) 250-303

J. Allison et al., Geant4 developments and applications, IEEE Trans. Nucl. Sci. 53 No. 1 (2006) 270-278

- GEANT4 - это набор инструментов. Вы должны сами написать программу моделирования, используя методы и библиотеки, имеющиеся в пакете GEANT4 .
- GEANT4 содержит мощный инструментарий для описания геометрии и материалов детектора
- GEANT4 содержит библиотеки для моделирования практически всех известных взаимодействий элементарных частиц с веществом
- GEANT4 имеет развитый интерфейс для взаимродействия с пользователем, управления циклом моделирования и средствами визуализации.

Цикл моделирования



Необходимо описать:

- Распределение вещества в детекторе и поля
- Генератор первичной вершины
- Список физических процессов, учитываемых в моделировании

Кроме того, по желанию:

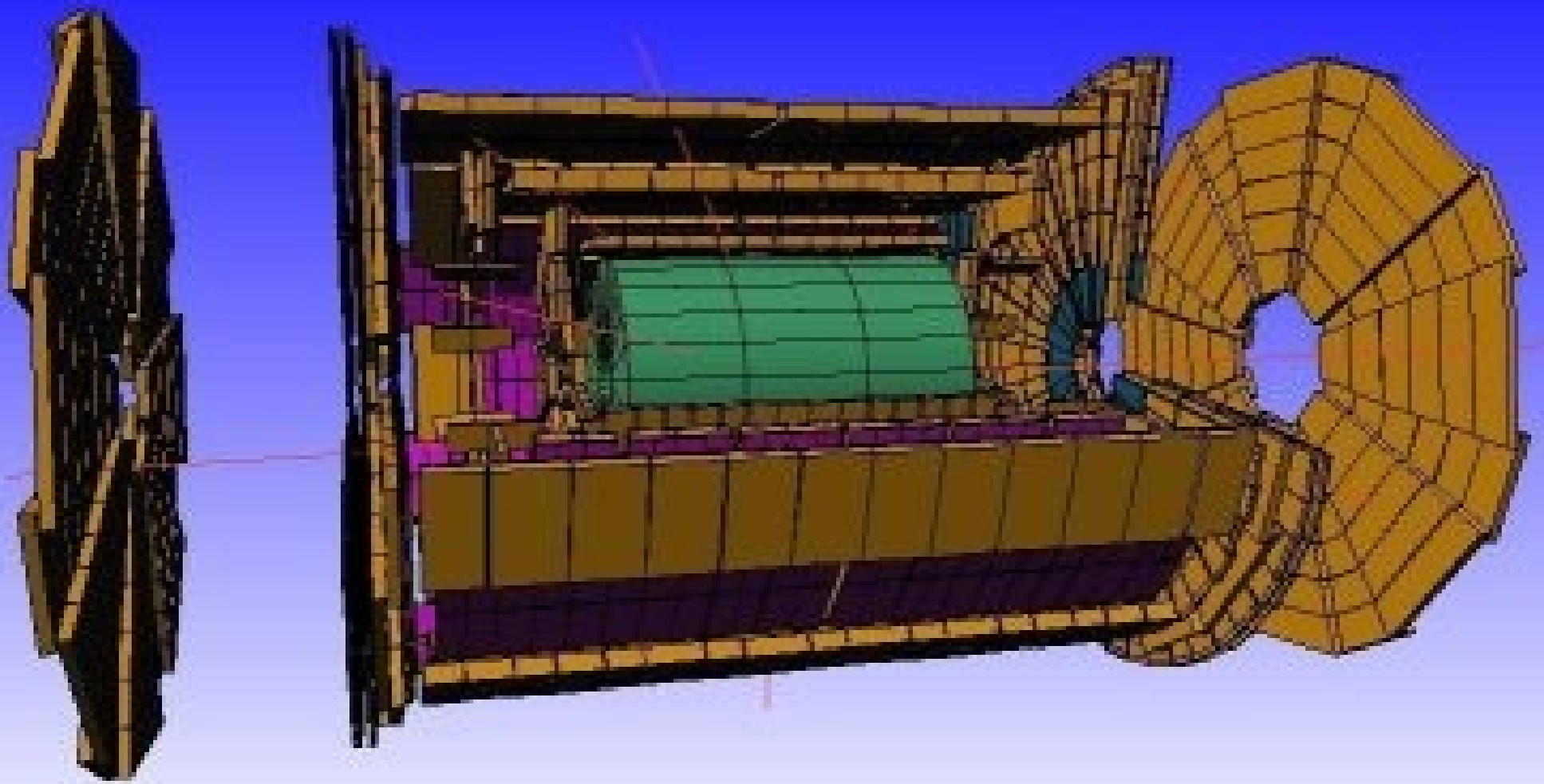
- Чувствительные элементы и способы моделирования отклика
- Способы визуализации
- Графический интерфейс
- Пользовательские расширения

Простая программа моделирования

```
#include "G4RunManager.hh"
#include "G4UImanager.hh"
#include "DetectorConstruction.hh"
#include "PhysicsList.hh"
#include "PrimaryGeneratorAction.hh"

int main()
{
  // construct the default run manager
  G4RunManager* runManager = new G4RunManager;

  // set mandatory initialization classes
  runManager->SetUserInitialization(new DetectorConstruction);
  runManager->SetUserInitialization(new PhysicsList);

  // set mandatory user action class
  runManager->SetUserAction(new PrimaryGeneratorAction);

  // initialize G4 kernel
  runManager->initialize();

  // start a run
  int numberOfEvent = 3;
  runManager->BeamOn(numberOfEvent);

  // job termination
  delete runManager;
  return 0;

}
```

Описание геометрии детектора

- В отличие от САПР, для физического моделирования помимо формы элементов конструкции нужно знать распределение вещества и задать физические факторы (электрические и магнитные поля, температуру, оптические свойства ...)
- Любое вещество в GEANT4 задается как смесь атомов, используя Z, A и плотность. Можно задавать разные фазовые состояния (газ, жидкость, твердое тело). Можно задавать любые электрические и магнитные поля, в том числе изменяющиеся во времени.
- Для описания геометрии применятся 'принцип матрешки' (сложный для разработчиков пакета, но удобный для пользователей):
  - все элементы конструкции разбиваются на простые объемы (кубики, цилиндры, пирамиды и т.д.)
  - для описания структуры внутренние (дочерние) объемы содержатся во внешних (материнских) объемах и положение их задается в локальной системе координат (Инкапсуляция!) Пересечение объемов не допускается.
  - существует самый большой объем (мир или экспериментальный зал), который содержит все элементы геометрии
- Каждый объем описывается через цепочку трех объектов C++: 'Solid (форма) → Logical Volume (форма+вещество) → Placement (объем+положение в пространстве)'



ATLAS

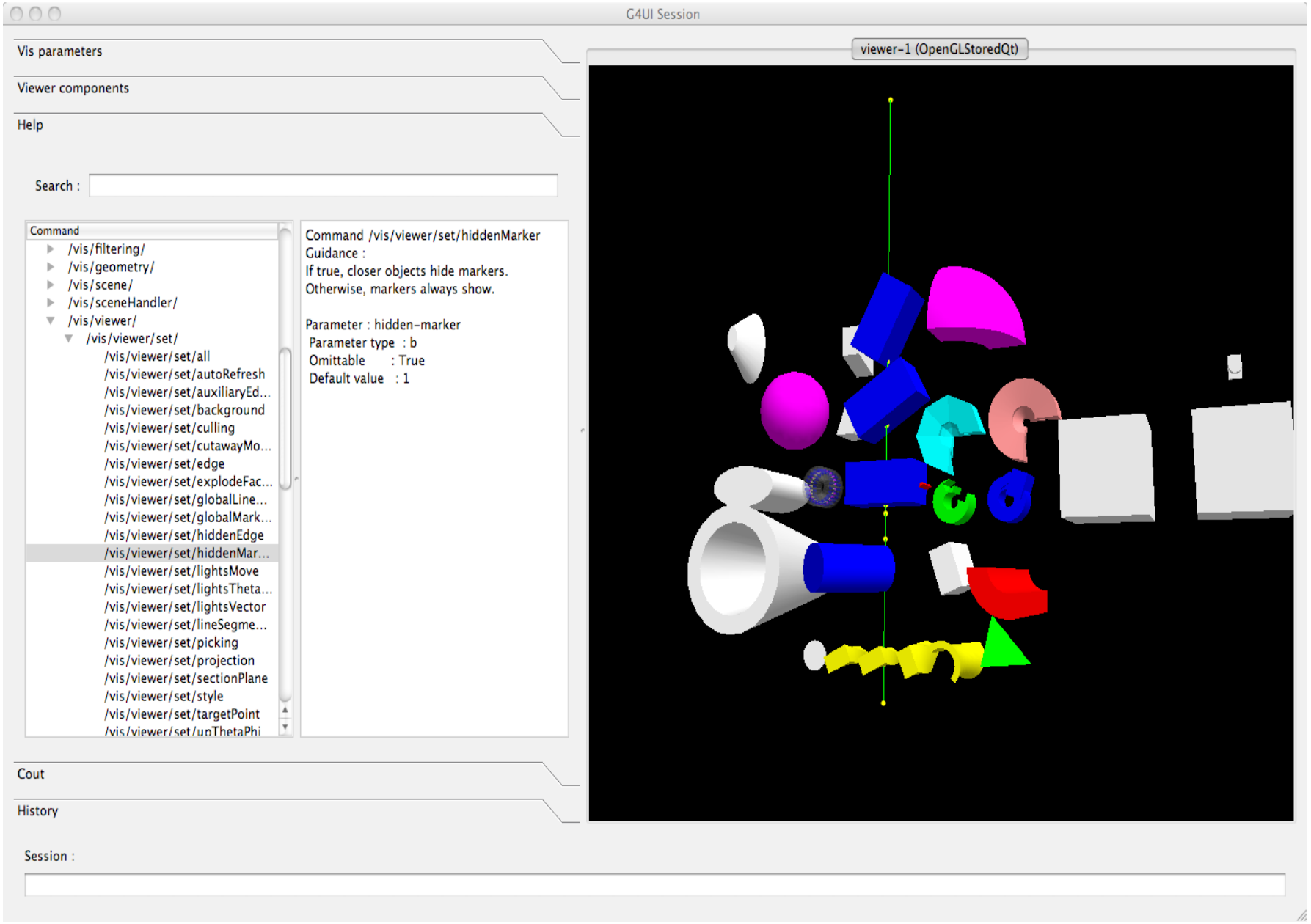
Интерфейс пользователя

- **пакетный режим**

- **интерактивный режим:**

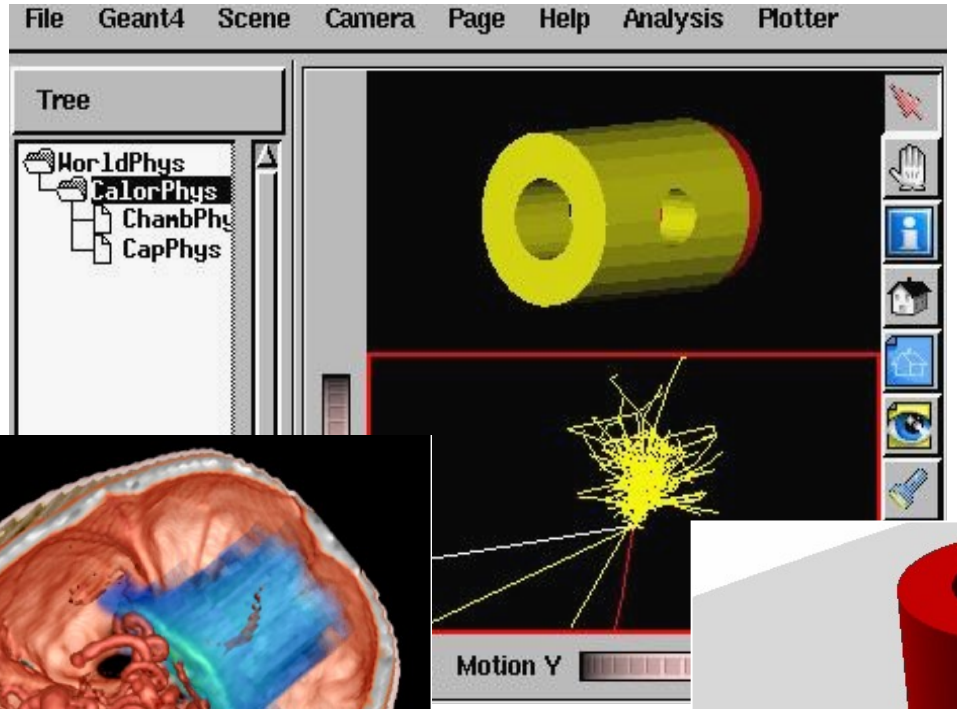
```
jemtchou: ~/geant4/bin/Linux-g++$ ./prog01
*************************************************************
 Geant4 version Name: geant4-08-00-patch-01    (10-February-2006)
                      Copyright : Geant4 Collaboration
                      Reference : NIM A 506 (2003), 250-303
                            WWW : http://cern.ch/geant4
*************************************************************
Idle>
Idle> /tracking/verbose 1
Idle> /control/execute run01.mac
```

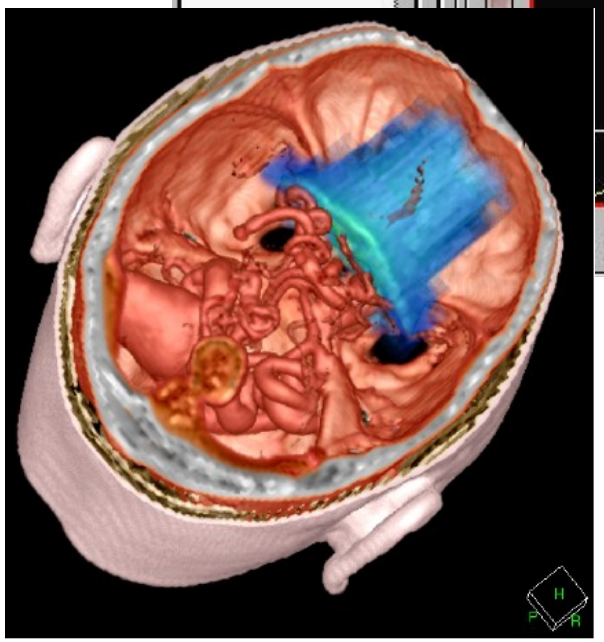
- **графический интерфейс**

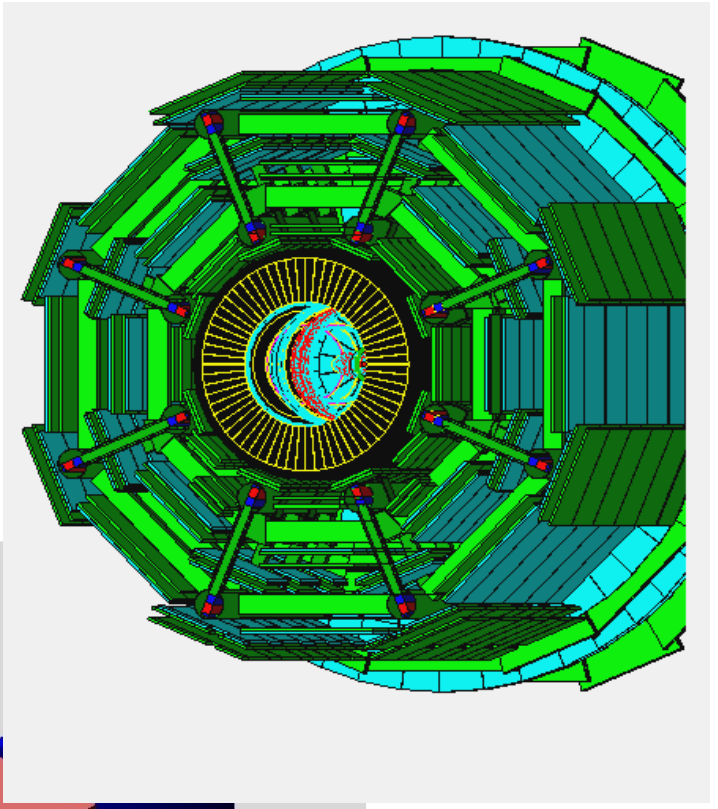

G4UI Session
Vis parameters
Viewer components
Help
Search :
Command
/vis/filtering/
/vis/geometry/
/vis/scene/
/vis/sceneHandler/
/vis/viewer/
/vis/viewer/set/
/vis/viewer/set/all
/vis/viewer/set/autoRefresh
/vis/viewer/set/auxiliaryEd...
/vis/viewer/set/background
/vis/viewer/set/culling
/vis/viewer/set/cutawayMo...
/vis/viewer/set/edge
/vis/viewer/set/explodeFac...
/vis/viewer/set/globalLine...
/vis/viewer/set/globalMark...
/vis/viewer/set/hiddenEdge
/vis/viewer/set/hiddenMar...
/vis/viewer/set/lightsMove
/vis/viewer/set/lightsTheta...
/vis/viewer/set/lightsVector
/vis/viewer/set/lineSegme...
/vis/viewer/set/picking
/vis/viewer/set/projection
/vis/viewer/set/sectionPlane
/vis/viewer/set/style
/vis/viewer/set/targetPoint
/vis/viewer/set/upThetaPhi
Command /vis/viewer/set/hiddenMarker
Guidance :
If true, closer objects hide markers.
Otherwise, markers always show.
Parameter : hidden-marker
Parameter type : b
Omittable : True
Default value : 1
Cout
History
Session :
viewer-1 (OpenGLStoredQt)

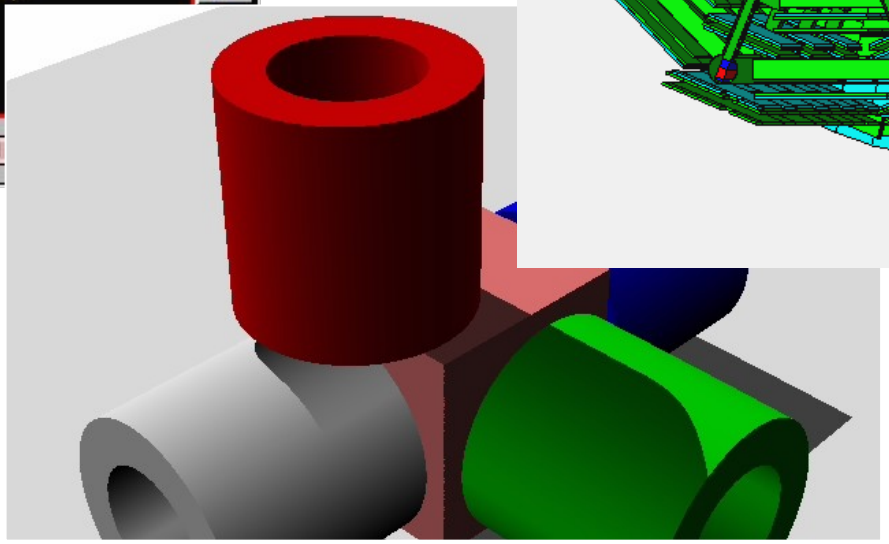
Визуализация

- Набор встроенных и внешних инструментов для визуализации: OpenGL, gMocren, JAS, HepRep, RayTracer, VRML, DAWN ...









Частицы и взаимодействия

Известные частицы

все стабильные и долгоживущие частицы, каоны, оптические фотоны, ядра и тяжелые ионы

Известные физические процессы

- ***Электромагнитные взаимодействия (0.1 keV – 10 TeV)***

ионизация, комптоновское рассеяние, многократное кулоновское рассеяние, тормозное излучение, рождение пар, фотоэффект, аннигиляция, фотоядерные и электроядерные процессы, синхротронное излучение, переходное излучение, излучение Вавилова-Черенкова, сцинтилляция, оптические процессы,...

- ***Адронные взаимодействия (0 eV – 100 TeV)***

упругое и неупругое рассеяние, захват, деление, снятие возбуждения ядер

- ***Транспорт частиц***
- ***Распады***
- ***Параметризация и «быстрое» моделирование***

Трекинг

- Частицы рождаются если их энергия выше порога рождения
- Новые частицы попадают в стек
- История частиц из стека моделируется до гибели частицы во взаимодействии, достижения нуля кинетической энергии или выхода из мира

Применение Geant4

- Физика высоких энергий
- Медицина
  - дозиметрия и расчет защит
  - КТ, ПЭТ, ОФЭКТ (проект GATE *http://www.opengatecollaboration.org*)
  - ядерная и радиационная терапия
  - радиобиология (проект Geant4-DNA *http://www.geant4-dna.org*)
- Исследование космоса *http://geant4.esa.int*
  - PLANETOCOSMICS is a simulation framework based on Geant4 that allows to compute the hadronic and electromagnetic interactions of cosmic rays with the Earth, Mars and Mercury.
  - MULASSIS: MUlti-LAyered Shielding SImulation Software (MULASSIS)
  - ESA ConeXpress: Radiation analysis for the ESA ConeXpress mission

Перспективы GEANT4

- GEANT4 имеет ряд преимуществ:
  - универсальное моделирование практически всех известных взаимодействий элементарных частиц с веществом
  - мощный инструментарий описания геометерии
  - преимущества объектно-ориентированного подхода: гибкость, модульность, возможность расширения функциональности пользователем
  - открытый код и минимум лицензионных ограничений
- Останется основным инструментом моделирования в физике высоких энергий по меньшей мере в течение 10-15 лет
- Начал широко применяться в ядерной физике и прикладных задачах (медицина, исследование космоса, материаловедение и т.д.)