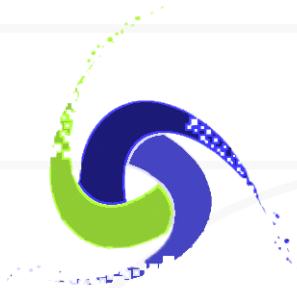
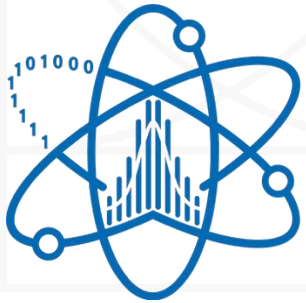
Национальный
исследовательский
**Томский
государственный
университет**



Е.В. Черняев

Генерация случайных точек равномерно распределенных на криволинейных поверхностях. Примеры использования в Geant4.



**Лаборатория
анализа данных
физики высоких энергий**

Томского
государственного
университета

Содержание

- Введение: где используется генерация случайных точек на поверхности
- Генераторы псевдо-случайных чисел
  - Семейство генераторов Xorshift
- Специализированные алгоритмы
  - Генерация случайных точек на поверхности плоских фигур
  - Генерация случайных точек на поверхности основных 3-х мерных поверхностей
- Генерация случайных точек на криволинейных поверхностях методом «методом отбрасывания»
  - Примеры для эллипса, параболоида, гиперболоид и параболического гиперболоида
- Использование генерации случайных точек на поверхностях в Geant4

Использование генерации случайных точек на поверхности

Генерация случайных точек на криволинейных поверхностях находит применение в ряде научных, инженерных и прикладных задач:

1. Компьютерная графика и визуализация

- **Рендеринг и освещение**: для глобального освещения, рассеянного отражения и трассировки лучей.
- **Текстурирование**: размещение деталей (например, капель воды, пятен грязи) на 3D-объектах.

2. Моделирование и симуляции

- **Физические симуляции**: моделирование осаждения частиц, капель, пыли на поверхности.
- **Аэродинамика/гидродинамика**: создание сеток или контрольных точек на поверхности тела в потоке.

3. Компьютерное зрение и робототехника

- **Задачи распознавания и реконструкции**: сэмплинг точек на реконструированных поверхностях для сравнения, регистрации и анализа формы.
- **Планирование движений**: оценка поверхностей препятствий и объектов для планирования траекторий.

4. 3D-печать

- **Контроль качества**: случайное выборочное сканирование точек на поверхности напечатанного объекта.

5. Машинное обучение и анализ данных

- **Генерация обучающих данных**: случайные точки на поверхностях объектов для обучения нейросетей

6. Геоинформатика и моделирование природных объектов

Генераторы псевдослучайных чисел

Почему псевдо?

- Для того, чтобы сгенерировать координаты случайной точки, нам нужно иметь генератор случайных чисел.
- Такие генераторы случайных чисел чаще называют **генераторами псевдослучайных чисел** (PRNG – pseudorandom number generators). Причины:
  - Генератор — это **детерминированный алгоритм**, который по заданному начальному состоянию (seed) производит последовательность чисел.
  - Если использовать один и тот же seed, получится **одинаковая последовательность** (повторяемость важна для отладки и воспроизводимости экспериментов).
- Разработано множество алгоритмов генерации псевдослучайных чисел. Зачастую такие алгоритмы используют очень специальный математический аппарат. Например, основной генератор MIXMAX используемый в Geant4 (период 10^{4389}) содержит следующее описание:

*The **MIXMAX generator** is a family of pseudorandom number generators (PRNG) and is based on Anosov C-systems (Anosov diffeomorphism) and Kolmogorov K-systems (Kolmogorov automorphism).*

Генератор псевдослучайных чисел Xorshift

- **Xorshift** — это семейство генераторов псевдослучайных чисел, предложенное Джорджем Марсальей (George Marsaglia) в 2003 году. Это чрезвычайно быстрый генератор, он основан на очень простых операциях:
  - **XOR** (побитовое исключающее ИЛИ, ^)
  - **сдвиг** (влево/вправо)
- Код на языке C/C++ для 32 битного генератора:

A	B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

```
uint32_t xorshift32()
{
 static uint32_t y = 2463534242;
 uint32_t x = y;
 x ^= x << 13;
 x ^= x >> 17;
 x ^= x << 5;
 return y = x;
}

double rand() // xorshift32()/2^32
{
 return xorshift32()/4294967296.;
}
```

- Период данного генератора равен $2^{32} - 1 = 4294967295$, т. е. генератор перебирает все 32-битные целые положительные числа без повторений.
- В дальнейшем мы будет считать, что у нас есть функция rand(), которая возвращает случайные числа равномерно распределенные от 0 до 1.

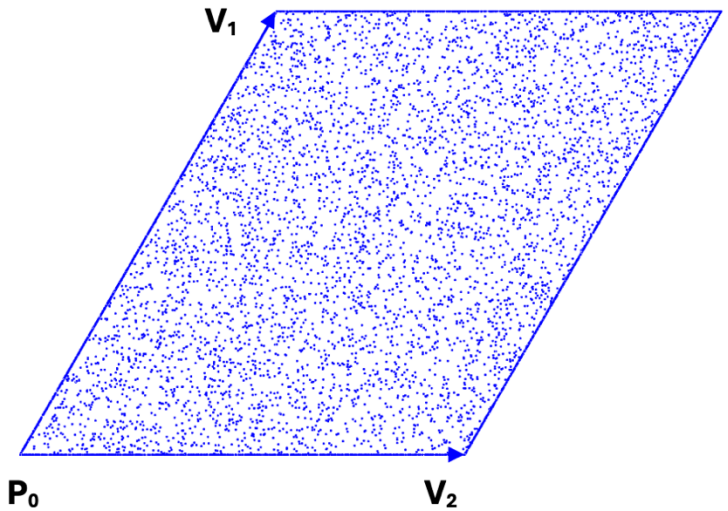
Генерация случайных точек на плоских фигурах

Параллелограмм

- Будем считать, что параллелограмм задан точкой $\mathbf{P_0}$ и двумя векторами $\mathbf{V_1}$ и $\mathbf{V_2}$
- Алгоритм генерации случайной точки внутри параллелограмма очевиден

```
u = rand()
v = rand()
p = p0 + v1*u + v2*v
```

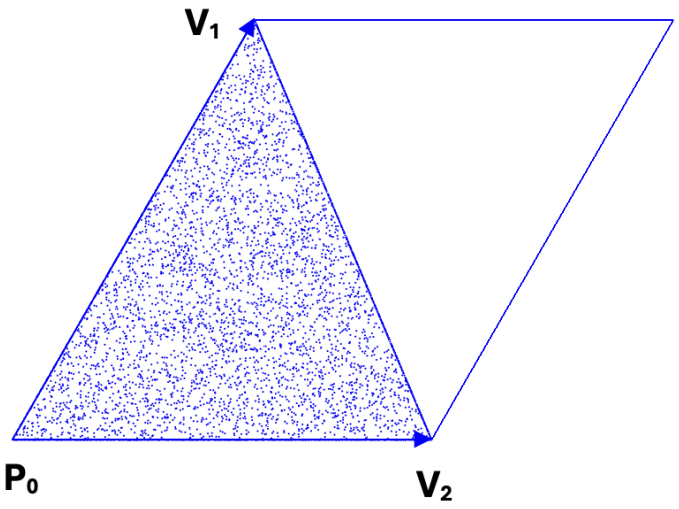
u, v – простые переменные,
p, p0, v1, v2 – объекты класса вектор



Треугольник

- Треугольник, как и параллелограмм, задается точкой $\mathbf{P_0}$ и двумя векторами $\mathbf{V_1}$ и $\mathbf{V_2}$,
- Алгоритм генерации случайной точки внутри треугольника:

```
u = rand()
v = rand()
if (u + v > 1)
{
  u = 1 – u
  v = 1 – v
}
p = p0 + v1*u + v2*v
```



- Идея алгоритма проста. Если к треугольнику добавить другой, полученный отражением исходного треугольника относительно стороны противоположной точке p0, то у нас получится параллелограмм. Мы сначала генерируем случайную точку в параллелограмме, и если она оказалась во втором треугольнике (u + v > 1) отражаем ее в исходный треугольник (u = 1 – u, v = 1 – v)

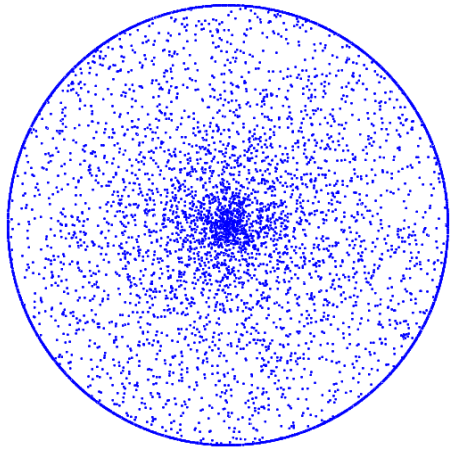
Круг

- Параметрическое представление круга:

  $x = r\cos\varphi$
  $y = r\sin\varphi$
  где $0 \le r \le R, 0 \le \varphi < 2\pi$

- Если мы будем генерировать случайные r и φ как в нижеследующем псевдокоде, то у нас явно получится неравномерное распределение точек внутри круга:

```
r = R*rand()
phi = 2*pi*rand()
x = r*cos(phi)
y = r*sin(phi)
```



Круг (продолжение)

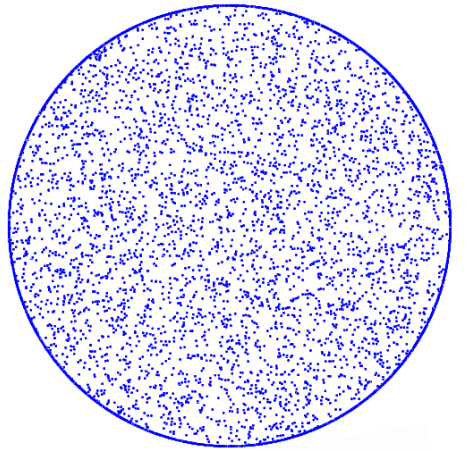
- Попробуем догадаться, какие переменные нужно использовать, чтобы получилось равномерное распределение случайных точек.
- Будем понимать под **элементом фигуры** ее часть, которая меняется при изменении параметров, но при этом **продолжает оставаться внутри исходной фигуры**. При определенных значениях параметров элемент фигуры будет занимать всю ее площадь. Если точки распределены равномерно по площади фигуры, то их количество в элементе фигуры будет прямо пропорционально площади этого элемента.

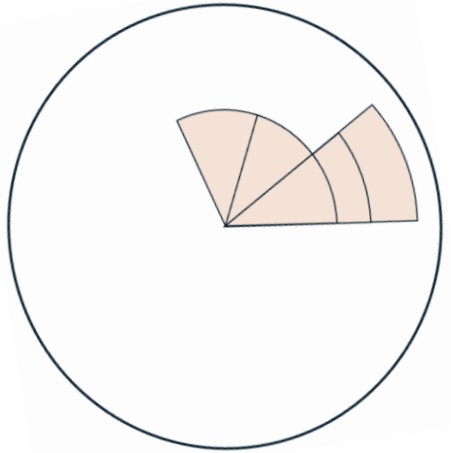
  Если в формуле площади элемента фигуры выделить переменные от которых **площадь зависит линейно**, то станет ясно значения каких переменных надо будет генерировать.
- В случае круга его элементом является сектор. Формулу площади сектора:

  $S_{sector} = (\varphi)\ (r^2)$,

  где $0 \leq r^2 \leq R^2, 0 \leq \varphi < 2\pi$

  При $\varphi = 2\pi, r = R$ сектор превращается в исходный круг.
  Ясно, что вместо r нам надо разыгрывать значение r^2





- Алгоритм генерации точки:

```
r2 = R*R*rand()
phi = 2*pi*rand()
r = sqrt(r2)
x = r*cos(phi)
y = r*sin(phi)
```

Альтернативный алгоритм для треугольника

- Найдем формулу площади части треугольника, из которой станет ясно какие переменные надо использовать для генерирования равномерного распределения точек. Формула

  $S_\Delta = \frac{1}{2}ah$, где $0 \le a \le A$ – длина основания, $0 \le h \le H$ – длина высоты

  в таком виде нам явно не подходит, т.к. если мы зафиксируем одну из переменных, вторая не сможет меняться в нужных нам пределах.

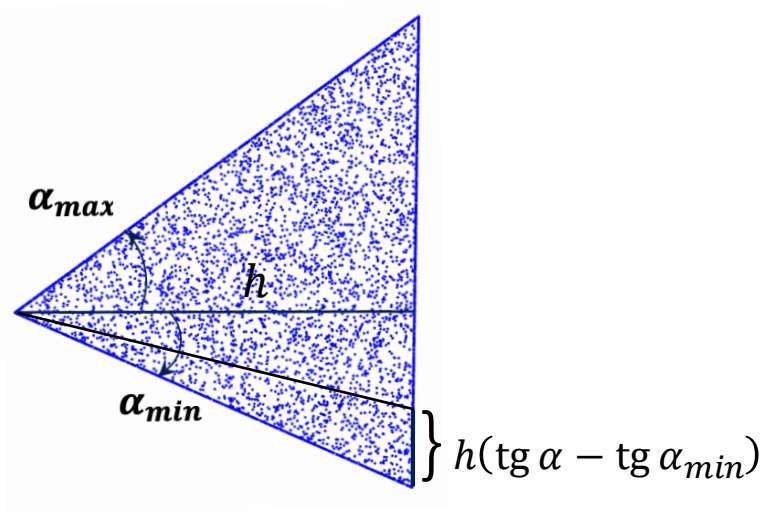
- Перепишем эту формулу в другом виде:

  $S_\Delta = \frac{1}{2}(h^2)(\mathrm{tg}\,\alpha - \mathrm{tg}\,\alpha_{min})$, где $\alpha_{min} \le \alpha \le \alpha_{max}$, соответственно

  $0 \le h^2 \le H^2$, $0 \le (\mathrm{tg}\,\alpha - \mathrm{tg}\,\alpha_{min}) \le (\mathrm{tg}\,\alpha_{max} - \mathrm{tg}\,\alpha_{min})$

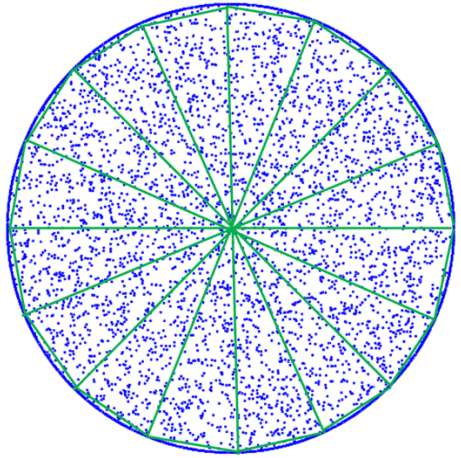
- Теперь понятно, что надо разыгрывать значения h^2 и $(\mathrm{tg}\,\alpha - \mathrm{tg}\,\alpha_{min})$. Соответственно алгоритм для треугольника будет выглядеть следующим образом:

```
h2 = H*H*rand()
dy = (tan(amax)-tan(amin))*rand()
x = sqrt(h)
y = x*(tan(amin) + dy)
```



Альтернативные алгоритмы для круга

- Для генерации случайных точек внутри круга может быть использована генерация случайных точек внутри треугольника:

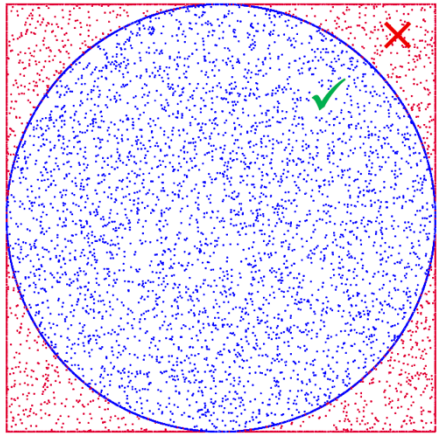


```
u = rand()
v = rand()
phi = 2*pi*rand()
if (u + v > 1)
{
  u = 1 – u
  v = 1 – v
}
r = R*(u + v)
x = r*cos(phi)
y = r*sin(phi)
```

- Мы также можем генерировать точки внутри окружающего круг квадрата, а затем проверять лежит ли точка внутри круга или нет:

```
do
{
  u = 2*rand() – 1
  v = 2*rand() – 1
} while (u*u + v*v > 1)
x = R*u
y = R*v
```

Это самый быстрый алгоритм для круга!



Эллипс

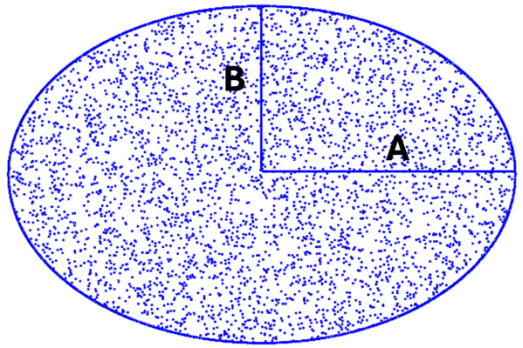
- Параметрическое представление эллипса:

  $x = a\cos\varphi$
  $y = b\sin\varphi$

  где $0 \le a \le A, 0 \le b \le B, 0 \le \varphi < 2\pi$

- Чтобы сгенерировать случайную точку внутри эллипса, мы сначала с помощью любого алгоритма генерируем точку внутри единичного круга, а затем масштабируем полученные координаты в соответствии с размером полуосей эллипса (“растягиваем” круг) :

```
do
{
  u = 2*rand() – 1
  v = 2*rand() – 1
} while (u*u + v*v > 1)
x = A*u
y = B*v
```



Равномерное растягивание/сжимание поверхности в любом направлении не нарушает равномерности распределения случайных точек.

Генерация случайных точек на криволинейных поверхностях

Цилиндр и Конус

- Генерация случайных точек на поверхности цилиндра и конуса достаточно проста, т.к. цилиндр можно развернуть в прямоугольник, а конус можно развернуть в сектор круга.

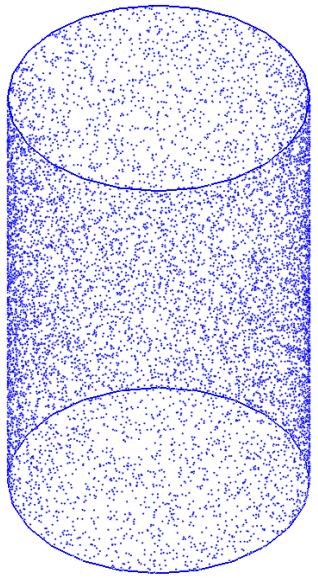
- Параметрическое представление поверхности цилиндра:

  $x = R\cos\varphi$

  $y = R\sin\varphi$

  $z = h$

  где $0 \le h \le H, 0 \le \varphi \le 2\pi$



- Алгоритм генерации случайной точки

```
h = H*rand()
phi = 2*pi*rand()
x = R*cos(phi)
y = R*sin(phi)
z = h
```

- Параметрическое представление поверхности конуса:

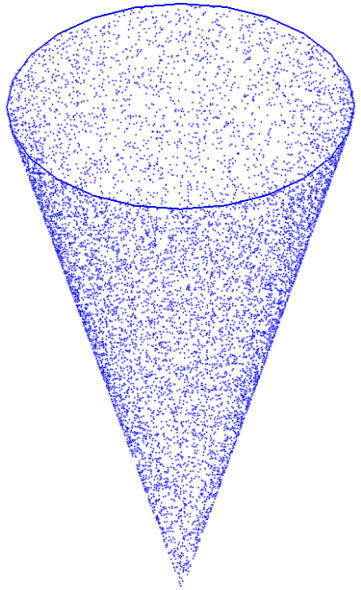
  $x = R\frac{h}{H}\cos\varphi$

  $y = R\frac{h}{H}\sin\varphi$

  $z = h$

  где $0 \le h \le H, 0 \le \varphi \le 2\pi,$

  R – радиус при $z = H$



- Алгоритм генерации случайной точки

```
h2 = H*H*rand()
phi = 2*pi*rand()
h = sqrt(h2)
rho = R*h/H
x = rho*cos(phi)
y = rho*sin(phi)
z = h
```

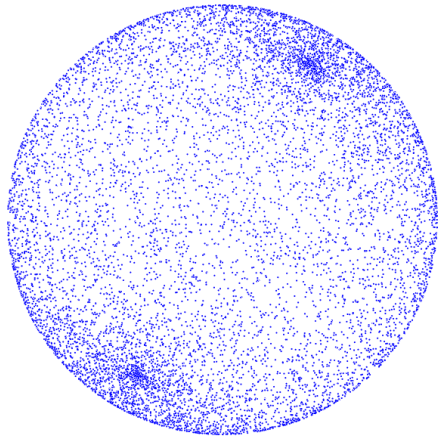
Сфера

- Стандартное параметрическое представление сферы:

$$x = R\sin\theta\cos\varphi$$
$$y = R\sin\theta\sin\varphi$$
$$z = R\cos\theta$$

где $0 \le \theta \le \pi, 0 \le \varphi < 2\pi$

- Если генерировать случайные точки разыгрывая θ и φ, то распределение точек будет неравномерным:



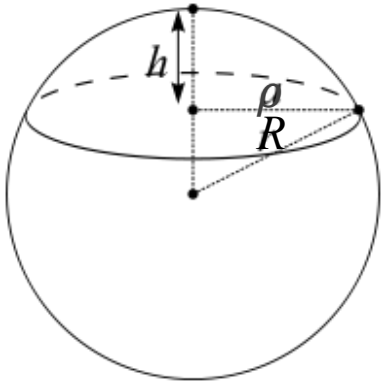
- Чтобы найти альтернативное представление вспомним как вычисляется площадь сферического слоя (spherical cap):

$$S_{cap} = 2\pi Rh = R(2\pi)(h)$$

- Это соответствует следующему параметрическому представлению:

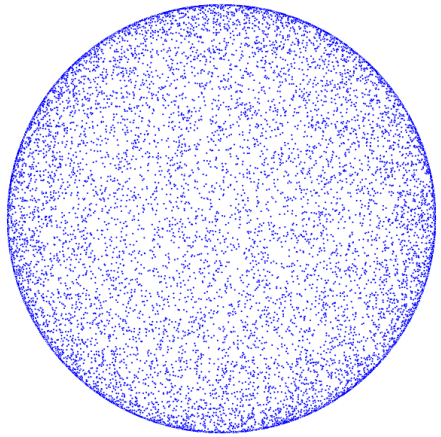
$$x = R\sqrt{1-u^2}\cos\varphi$$
$$y = R\sqrt{1-u^2}\sin\varphi$$
$$z = Ru$$

где $-1 \le u \le 1, 0 \le \varphi < 2\pi$



- Алгоритм для генерирования точки:

```
u = 2*rand() - 1
phi = 2*pi*rand()
rho = sqrt(1 – u*u)
x = R*rho*cos(phi)
y = R*rho*sin(phi)
z = R*u
```



Rejection sampling
Выборка с отбрасыванием

Генерация случайных точек «методом отбрасывания»

Выборка с отбрасывание (rejection sampling) - это общий метод позволяющий генерировать случайные точек равномерно распределенные по поверхности заданной в параметрическом виде:

$$\boldsymbol{r}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}$$

где $(u,v) \in D$, в нашем случае $u_{min} \leq u \leq u_{max}$, $v_{min} \leq v \leq v_{max}$

Площадь поверхности для такой поверхности вычисляется интегрированием элемента поверхности $dS(u,v)$ по области D:

$$S = \iint_D dS(u,v)\, dudv$$

Элемент поверхности вычисляется как модуль векторного произведения частных производных:

$$dS(u,v) = \left\| \frac{\partial \boldsymbol{r}}{\partial u} \times \frac{\partial \boldsymbol{r}}{\partial v} \right\|$$

Алгоритм генерации случайной точки выглядит следующим образом:

0. Находим максимальное значение dS_{max} в области D
1. Разыгрываем случайные параметры (u,v)
2. Вычисляем $dS(u,v)$
3. Генерируем случайное число $0 \leq \xi \leq dS_{max}$
4. Если $\xi \leq dS(u,v)$, то принимаем точку $\boldsymbol{r}(u,v)$, иначе повторяем шаги 1 – 4

Этот метод называется **методом отбрасывания** (rejection sampling) и обеспечивает равномерное покрытие поверхности по площади

Эллипсоид

- Параметрическое представление эллипсоида:

$$x = a\sqrt{1-u^2}\cos v$$
$$y = b\sqrt{1-u^2}\sin v$$
$$z = c\,u$$

где a, b, c – полуоси эллипсоида, $-1 \le u \le 1, 0 \le v < 2\pi$

- Векторное произведение частных производных (нормаль) в точке с параметрами (u, v) находится также как детерминант следующей матрицы:

$$\boldsymbol{n}(u,v) = \frac{\partial \boldsymbol{r}}{\partial u} \times \frac{\partial \boldsymbol{r}}{\partial v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -\frac{au\cos v}{\sqrt{1-u^2}} & -\frac{bu\sin v}{\sqrt{1-u^2}} & c \\ -a\sqrt{1-u^2}\sin v & b\sqrt{1-u^2}\cos v & 0 \end{vmatrix}$$

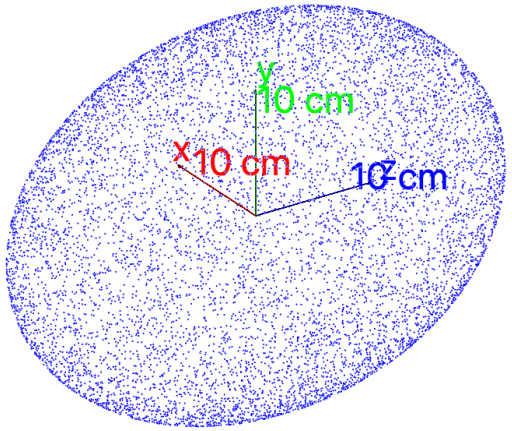
$$\boldsymbol{n}(u,v) = -(\boldsymbol{i}\, bc\sqrt{1-u^2}\cos v + \mathbf{j}\, ac\sqrt{1-u^2}\sin v + \mathbf{k}\, abu)$$

- Элемент поверхности равен длине нормали:

$$dS(u,v) = \sqrt{b^2c^2(1-u^2)\cos^2 v + a^2c^2(1-u^2)\sin^2 v + a^2b^2u^2}$$

- Максимальное значение элемента поверхности:

$$dS_{max} = \max(ab, bc, ac)$$



- Алгоритм генерации точки:

```
// generate point (xs,ys,zs) on unit sphere and
// check its acceptance
ds_max = max(a*b, b*c, a*c)
aa = a*a
bb = b*b
cc = c*c
do {
  u = 2*rand() – 1
  v = 2*pi*rand()
  rho = sqrt(1 – u*u)
  xs = rho*cos(v)
  ys = rho*sin(v)
  zs = u
  ds = sqrt(bb*cc*xs*xs + aa*cc*ys*ys + aa*bb*zs*zs)
} while (ds_max*rand() > ds)
x = a*xs
y = b*ys
z = c*zs
```

Тор (тороид)

- Параметрическое представление тора:

  $x = (R + a\cos v)\cos u$
  $y = (R + a\cos v)\sin u$
  $z = a\sin v$
  где R – радиус тора, a – радиус образующей окружности,
  $0 \le u \le 2\pi,\ 0 \le v < 2\pi$

- Элемент поверхности:

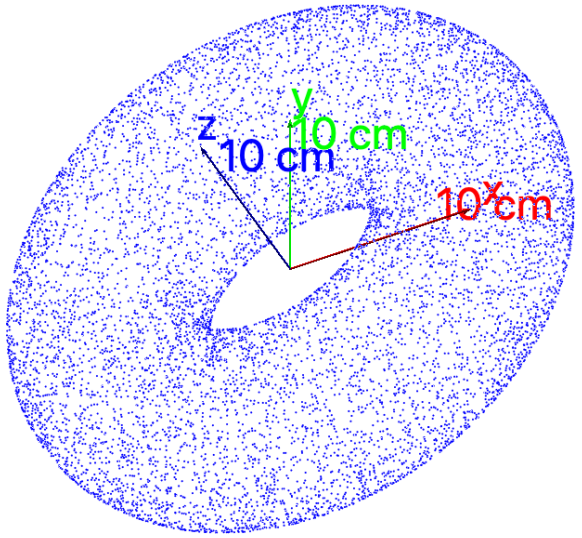
$$\boldsymbol{n}(u,v) = \frac{\partial \boldsymbol{r}}{\partial u} \times \frac{\partial \boldsymbol{r}}{\partial v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -(R + a\cos v)\sin u & (R + a\cos v)\cos u & 0 \\ -a\ \sin v\cos u & -a\ \sin v\sin u & a\cos v \end{vmatrix}$$

$$dS(u,v) = a(R + a\cos v)$$

$$dS_{max} = a(R + a)$$

- Алгоритм генерации точки

```
ds_max = R + a
do {
  v = 2*pi*rand()
  ds = R + a*cos(v)
} while (ds_max*rand() > ds)
u = 2*pi*rand()
x = ds*cos(u)
y = ds*sin(u)
z = a*sin(v)
```



Параболоид

- Параметрическое представление параболоида:

  $x = R\sqrt{u}\cos v$
  $y = R\sqrt{u}\sin v$
  $z = h\,u$
  где h – высота, R – радиус при $z = h$, $0 \le u \le 1$, $0 \le v < 2\pi$

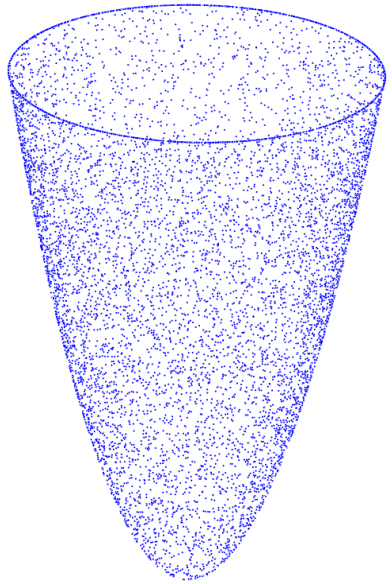
- Элемент поверхности:

$$\boldsymbol{n}(u,v) = \frac{\partial \boldsymbol{r}}{\partial u} \times \frac{\partial \boldsymbol{r}}{\partial v} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{R}{2\sqrt{u}}\cos v & \frac{R}{2\sqrt{u}}\sin v & h \\ -R\sqrt{u}\sin v & R\sqrt{u}\cos v & 0 \end{vmatrix}$$

$$dS = R\sqrt{h^2 u + \frac{R^2}{4}} \qquad dS_{max} = R\sqrt{h^2 + \frac{R^2}{4}}$$

- Алгоритм генерации точки

```
ds_max = sqrt(h*h + R*R/4)
do {
  u = rand()
  ds = sqrt(h*h*u + R*R/4)
} while (ds_max*rand() > ds)
v = 2*pi*rand()
x = R*sqrt(u)*cos(v)
y = R*sqrt(u)*sin(v)
z = h*u
```



Однополостный гиперболоид

- Параметрическое представление параболоида:

$$x = \sqrt{a^2 + (R^2 - a^2)u^2}\cos v$$
$$y = \sqrt{a^2 + (R^2 - a^2)u^2}\sin v$$
$$z = hu$$

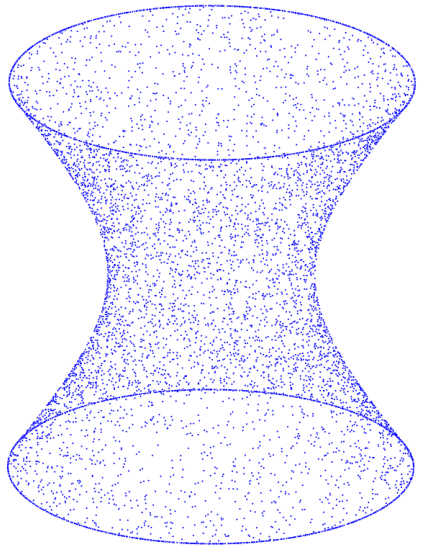
где h – полувысота, R – радиус при $z = h$, a – радиус при $z = 0$, $-1 \le u \le 1, 0 \le v < 2\pi$

- Элемент поверхности:

$$dS(u,v) = \sqrt{a^2h^2 + (R^2 - a^2)(R^2 - a^2 + h^2)u^2}$$
$$dS_{max} = \sqrt{a^2h^2 + (R^2 - a^2)(R^2 - a^2 + h^2)}$$

- Алгоритм генерации точки

```
aahh = a*a*h*h
RRaa = R*R – a*a
RRaahh = RRaa*(RRaa + h*h)
ds_max = sqrt(aahh + RRaahh)
do {
  u = 2*rand() - 1
  ds = sqrt(aahh + RRaahh*u*u)
} while (ds_max*rand() > ds)
v = 2*pi*rand()
rho = sqrt(a*a + RRaa*u*u)
x = rho*cos(v)
y = rho*sin(v)
z = h*u
```



Использование генерации случайных точек на поверхностях в Geant4

Что такое Geant4

Geant4 (Geometry and Tracking) — это программный инструментарий (фреймворк), предназначенный для моделирования прохождения частиц через вещество.

Geant4 поддерживается и развивается международной коллаборацией, в которую в настоящее время входят 156 человек имеющих статус член коллаборации и 35 контрибьюторов из различных научных организаций и компаний со всего мира. В том числе 9 человек из 5 институтов Российской Федерации.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
Физический институт имени П.Н.Лебедева
РОССИЙСКОЙ АКАДЕМИИ НАУК
Ф И А Н

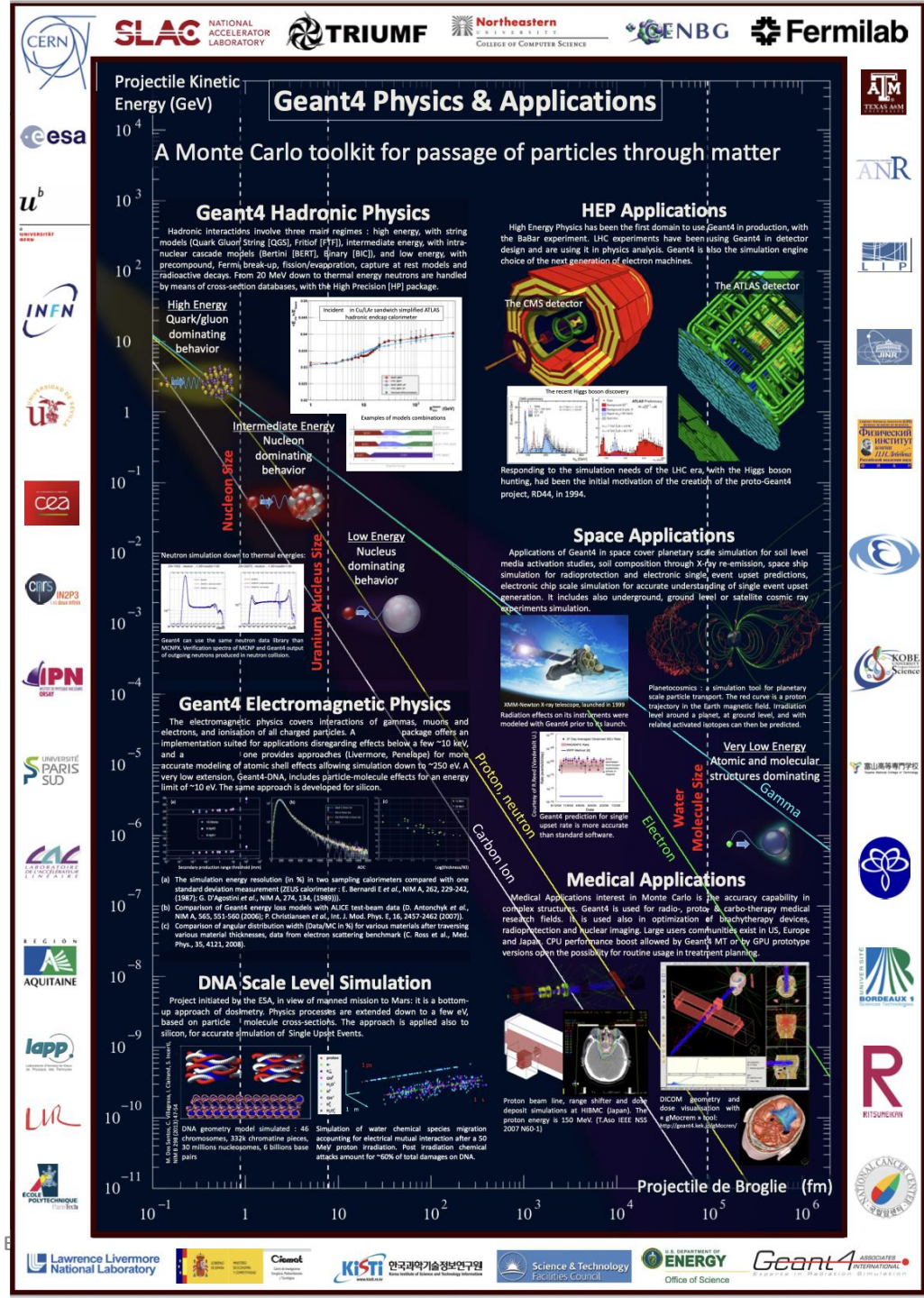
НИЯУ МИФИ

ОИЯИ

ИФВЭ
ИНЕР

Национальный исследовательский Томский государственный университет

Geant4 широко применяется в физике высоких энергий, ядерной физике, космических исследованиях, медицинской физике и инженерных задачах.



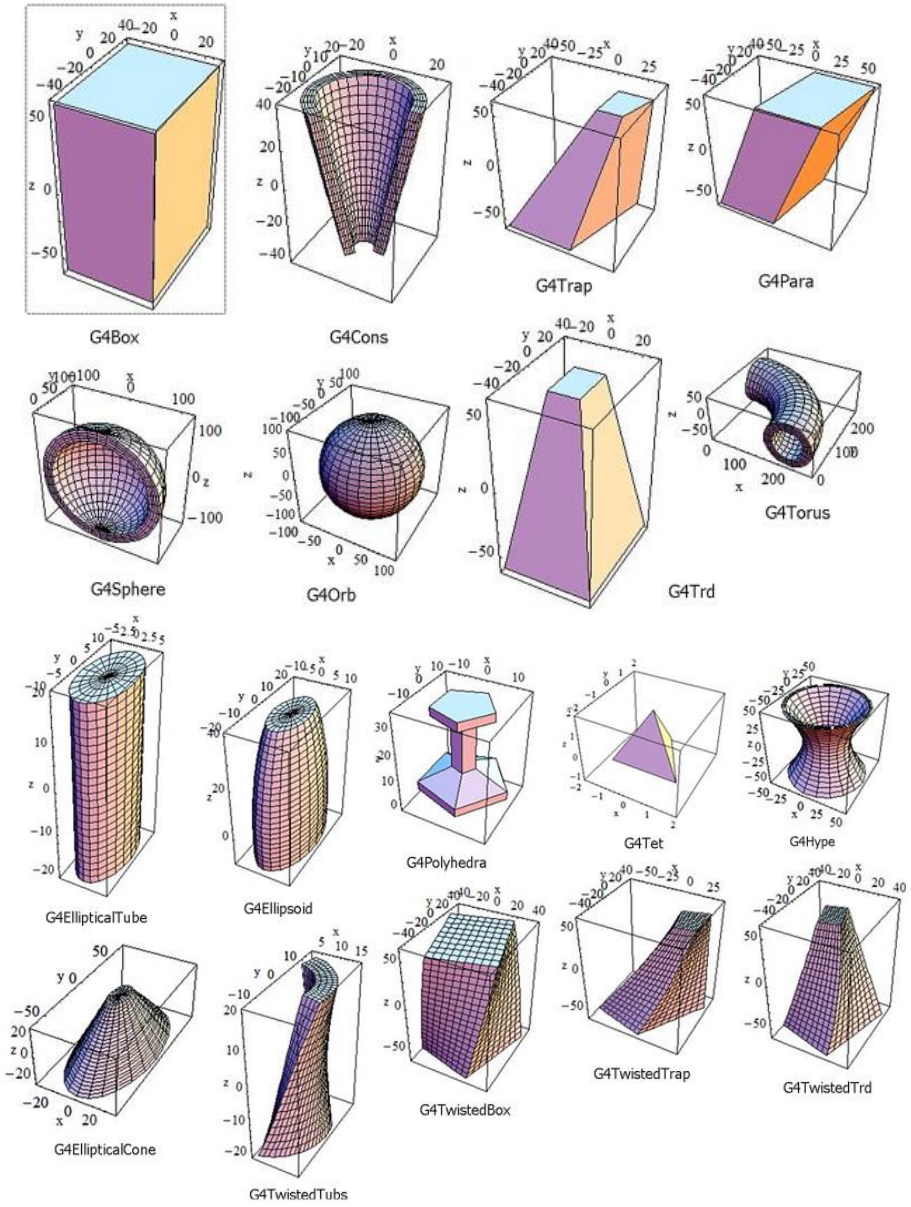
Геометрический пакет Geant4

Geant4 имеет развитый геометрический пакет. Есть богатый набор геометрических **примитивов**, из которых с помощью булевых операций (объединение, вычитание, пересечение) можно создавать более сложные формы. Таким образом, в Geant4 можно построить практически любую геометрию, комбинируя стандартные примитивы и булевы операции.

Каждый геометрический объект (**solid**) в Geant4 имеет набор обязательных функций для работы с ним, таких как, определение положения точки относительно данного объекта, определение примерного расстояния до границ объекта, определение точного расстояния до поверхности объекта вдоль заданного направления, вычисление объема и площади поверхности, и т.д.

Среди обязательных функций есть функция позволяющая генерировать случайные точки равномерно распределенные по поверхности объекта.

Далее мы рассмотрим как эта возможность используется в Geant4



GPS – General Particle Source

- В Geant4 под GPS понимается универсальный генератор частиц. Это продвинутый класс для задания начальных условий источника частиц. С его помощью можно настраивать:
  - Тип частицы (электрон, фотон, протон и т. д.).
  - Энергию (фиксированное значение, распределение, спектр).
  - Геометрию источника (точечный, сферический, цилиндрический, плоский диск, и т. д.).
  - Направление (изотропное распределение, в конусе, вдоль оси и др.).
  - Временные характеристики (например, импульсный источник).
- Источник частиц может плоским (квадрат, прямоугольник, круг, кольцо, эллипс) или объемным (сфера, эллипсоид, цилиндр, параллелепипед). В случае объемного источника частицы могут генерироваться как внутри источника, так и на его поверхности.

Оценка площади поверхности булевых тел

С помощью генерации случайных точек равномерно распределенных по поверхности можно оценить площадь поверхности составного тела, полученного в результате применения булевой операции (объединения, вычитания или пересечения) к исходным телам.

Для этого нужно сгенерировать $N = N_1 + N_2$ точек, где N_1 — это количество точек на первом теле, а N_2 — количество точек на втором теле. Количество точек на каждом из тел должно быть пропорционально площади поверхности этих тел:

$$\frac{N_1}{N_2} = \frac{S_1}{S_2}$$

Площадь поверхности результирующего тела с будет равна произведению суммы площадей исходных тел на отношение количества точек, оказавшихся на на поверхности результирующего тела, к общему количеству сгенерированных точек:

$$S_{result} = \frac{N_{surf}}{N_1 + N_2}(S_1 + S_2)$$

Относительная погрешность зависит от общего количества точек: $\delta X \approx \frac{1}{\sqrt{N}}$

Пример оценки площади поверхности составного тела

Для примера оценим площадь поверхности тела сконструированного путем объединение двух сфер и сравним полученный результат с площадью этого тела вычисленной аналитически.

Для простоты вычислений пусть обе сферы имеют одинаковый радиус $R = 10$ см и пусть расстояние между центрами сфер равно R. В таком случае площадь объединённой фигуры будет равна:

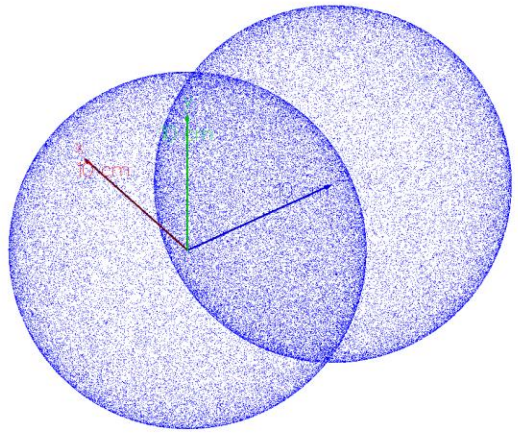
$$S = 6\pi R^2 = 1884.96 \text{ см}^2$$

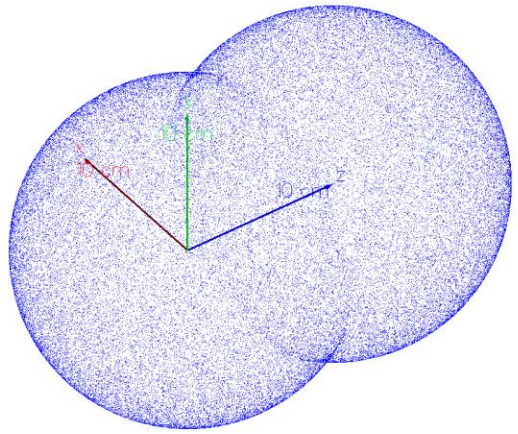
При генерации 10 тыс. точек, по 5 тыс. на каждой из сфер, у меня получилось:

$$S_{10к} = 1865.35 \pm 19 \text{ см}^2$$

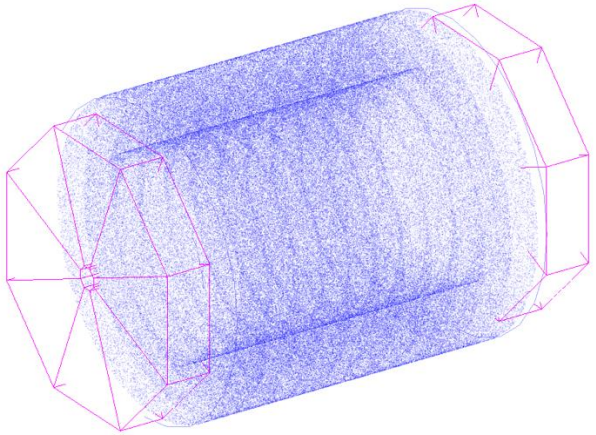
При генерации 100 тыс. точек:

$$S_{100к} = 1885.28 \pm 6 \text{ см}^2$$

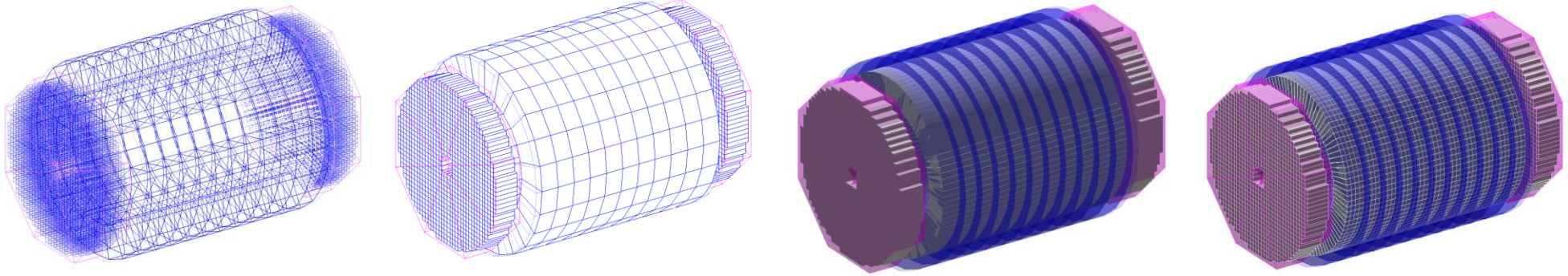




Визуализация тел в виде облака точек



- Geant4 предоставляет несколько возможностей для визуализации трехмерных тел. Любое тело может быть нарисовано:
  - В виде проволочного объекта без удаления невидимых линий;
  - Тоже самое с удалением невидимых линий;
  - С визуализацией поверхностей тела с учетом освещения (shading);
  - Тоже самое с наложением результата удаления невидимых линий (отрисовка ребер);
  - В виде облака точек случайный образом распределенным по поверхности тела;



- Изначально последняя опция была реализована только для отладки алгоритма генерации случайных точек на поверхности тел, но потом ее добавили в список стандартных возможностей.
- Визуализация составных (булевых) тел в виде облака точек происходит автоматически если есть возникают проблемы с генерацией полигонального представления этих тел.

Нахождение перекрытий в Geant4

Одним из важных применений генерации случайных точек на поверхности тел в Geant4 является нахождение перекрытий объектов в геометрическом описании моделируемой установки.

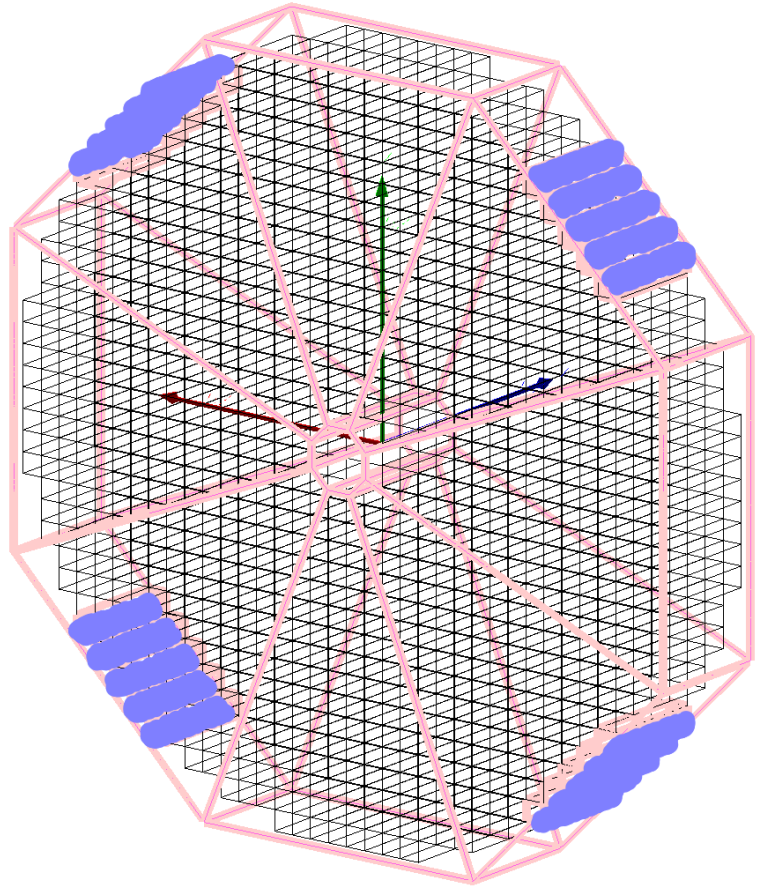
Перекрытия – это дефект в геометрии, они могут приводить к неправильному результату моделирования, а также создавать проблемы при построении трэков.

Идея алгоритма нахождения перекрытий достаточно проста. Чтобы проверить пересекаются два тела или нет, мы сначала генерируем набор точек на поверхности одного тела и проверяем не попадают какие либо из этих точек внутрь другого тела. Затем меняем тела местами, т. е. генерируем точки на поверхности второго тела и проверяем не оказались ли какие либо точки внутри первого тела.

Поскольку в Geant4 используется иерархическое описание геометрии, т.е. тела помещаются внутрь родительского объема, то существует два типа перекрытий:

- Тело пересекается с другим телом внутри родительского объема;
- Тело выходит за пределы своего родительского объема;

На рисунке мы как раз видим второй случай.



ECal endcap, NICA/SPD

Резюме

Основные моменты данного семинара:

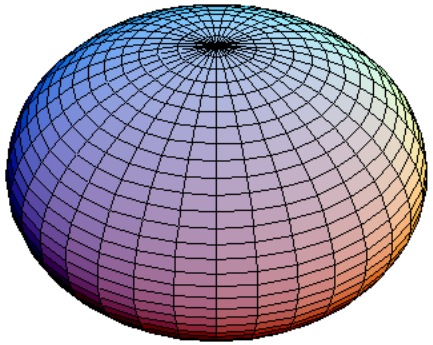
- Мы познакомились с простым и очень быстрым генератором случайных чисел Xorshift
- Были объяснены специализированные алгоритмы генерации случайных точек:
  - для плоских фигур: параллелограмма, треугольника, круга, эллипса
  - для основных криволинейных поверхностей: цилиндра, конуса, сферы
- Был подробно разобран алгоритм генерации равномерно распределенных случайных точек «методом отбрасывания» на примере эллипса, параболоида, гиперболоида и тора.
- Было рассказано как генерация случайный точек на поверхностях используется в Geant4: General Particle Source, оценка площади поверхности булевых тел, визуализация, обнаружение перекрытий.
- ✻ В приложении даны формулы вычисления площадей поверхностей второго порядка и связанных с ними объемов.

Приложение.
Формулы для поверхностей второго порядка.

Эллипсоид

- Уравнение в декартовых координатах:

  $\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$, где a, b, c – полуоси, $a \geq b \geq c$



- Объем:

  $$V=\frac{4}{3}\pi abc$$

- Площадь поверхности:

  $$S=2\pi c^2+\frac{2\pi ab}{\sin\phi}\left[F(\phi,k)\cos^2\phi+E(\phi,k)\sin^2\phi\right]$$

  где

  $F(\phi,k)$ – неполный эллиптический интеграл первого рода

  $E(\phi,k)$ – неполный эллиптический интеграл второго рода

  $$\phi=\arccos\frac{c}{a}$$

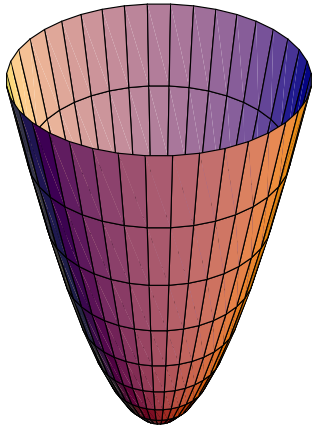
  $$k=\sqrt{\frac{a^2(b^2-c^2)}{b^2(a^2-c^2)}}$$

Параболоид

- Уравнение в декартовых координатах:

$$z = \frac{h}{a^2}(x^2 + y^2)$$

где a – радиус сечения при $z = h$



- Объем ограниченный плоскостью $z = h$:

$$V = \frac{1}{2}\pi a^2 h$$

- Площадь поверхности:

$$S = \frac{\pi a}{6h^2}\left[(a^2 + 4h^2)^{3/2} - a^3\right]$$

Однополостный гиперболоид

- Уравнение в декартовых координатах:

$$\frac{x^2}{a^2}+\frac{y^2}{a^2}-\frac{z^2}{c^2}=1$$

- Объем между $-\frac{h}{2}\le z\le+\frac{h}{2}$:

$$V=\frac{1}{3}\pi h(2a^2+R^2)$$

- Площадь поверхности:

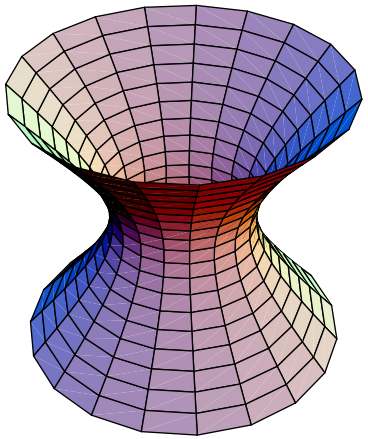
$$S=2\pi a\left[\frac{h\sqrt{[4c^4+(a^2+c^2)h^2]}}{4c^2}+\frac{c^2\,\mathrm{arsh}\left(\frac{h\sqrt{a^2+c^2}}{2c^2}\right)}{\sqrt{a^2+c^2}}\right]$$

где

a – радиус сечения при $z=0$

h – полная высота

R – радиус сечений при $z=\pm\frac{h}{2}$

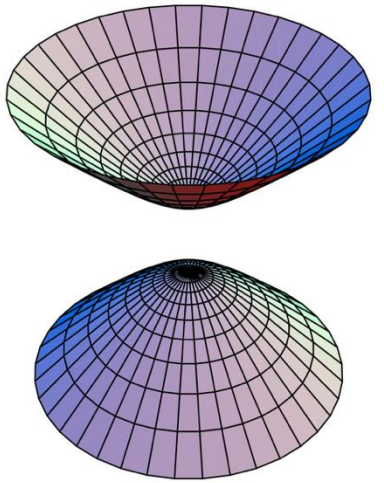


Двуполостный гиперболоид

- Уравнение в декартовых координатах:

$$\frac{z^2}{a^2} - \frac{r^2}{b^2} = 1$$

где a – половина расстояния между полостями, $r^2 = x^2 + y^2$



- Пусть h – высота полости, R – радиус сечения при $z = a + h$, тогда объем полости между $a \le z \le a + h$ будет равен:

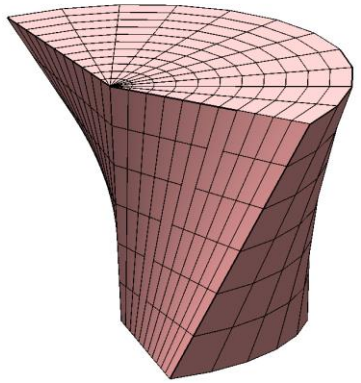
$$V = \frac{\pi\, h^2 b^2}{a^2}\left(a + \frac{h}{3}\right) = \frac{\pi\, R^2 h}{3}\left(1 + \frac{a}{2a + h}\right)$$

- Площадь поверхности полости:

$$S = \frac{\pi b}{a^2}(a+h)\sqrt{(a+h)^2(a^2+b^2) - a^4} - \pi b^2 - \frac{\pi b a^2}{\sqrt{a^2+b^2}} \ln\left(\frac{(a+h)\sqrt{a^2+b^2} + \sqrt{(a+h)^2(a^2+b^2) - a^4}}{a\sqrt{a^2+b^2} + ab}\right)$$

Скрученный сектор цилиндра. Гиперболический параболоид

- Скрученный сектор получается путем поворота оснований сектора цилиндра относительно друг друга. При этом внешняя цилиндрическая поверхность приобретает форму гиперболоида, а боковые грани приобретают форму гиперболического параболоида.
- На рисунке изображена половина скрученного сектора. Обозначим:
  φ – угол сектора
  ω – угол скручивания
  h – половина высоты цилиндра
  a – радиус скрученного сектора при $z = 0$
  r – радиус скрученного сектора при $z = h$ (радиус исходного цилиндра)



- Объем данной фигуры (сравните с объемом однополостного гиперболоида):

$$V = \frac{\varphi h}{6}\left(2a^2 + r^2\right)$$

- Площадь внешней поверхности (сравните с площадью поверхности однополостного гиперболоида):

$$S = \varphi a\left(h\sqrt{1 + k^2h^2} + \frac{1}{k}\sinh^{-1}(kh)\right), \text{ где } k = \frac{\sqrt{a^2+c^2}}{c^2}, c^2 = \frac{a^2h^2}{r^2-a^2}$$

- Площадь боковых граней (гиперболический параболоид):

$$S = \frac{ah}{3pq}\left(pqf + \frac{p(p^3+3)}{2}\operatorname{arth}\left(\frac{q}{f}\right) + \frac{q(q^3+3)}{2}\operatorname{arth}\left(\frac{p}{f}\right) + \operatorname{arctg}\left(\frac{f}{pq}\right) - \frac{\pi}{2}\right),$$

$$\text{где } p = \frac{r\sin\omega}{h}, q = \frac{r\sin\omega}{a}, f = \sqrt{p^2 + q^2 + 1}$$