

Гибридное моделирование эксперимента на коллайдере ВЭПП-6

Кирилл Чиликин

Институт ядерной физики имени Г.И. Будкера СО РАН, Новосибирск

Совещание по физической программе
для эксперимента на коллайдере ВЭПП-6
21 мая 2026



Для проекта Супер c-tau фабрики ранее было реализовано быстрое моделирование [Comput. Softw. Big Sci. **8**, 1 (2024)]. Однако, у него есть ограничения, которые не позволяют полноценно использовать его для разработки проекта детектора и физической программы ВЭПП-6:

- Не учитываются распады частиц на лету и их взаимодействие с веществом, что влияет на вероятность некорректной идентификации,
- Из-за отсутствия подгонки треков невозможно восстановление вершин долгоживущих частиц (K_S^0 , Λ^0) и импакт-параметров отдельных треков.

Для оценки того, какие нужны вершинный детектор и система идентификации, а также для моделирования процессов с долгоживущими частицами необходимо улучшенное моделирование.



- В процессе разработки улучшенного моделирования хотелось бы сохранять возможность проводить моделирование в каждый момент времени.
- Поэтому постепенно реализуется упрощённая геометрия детектора, а элементы быстрого моделирования постепенно заменяются на полное моделирование в Geant4. Такой переходный вариант назван гибридным моделированием.
- В настоящее время реализованы дрейфовая камера и ASHIPH, проводится подгонка трека по сигналу из дрейфовой камеры, его экстраполяция в ASHIPH и вычисление функций правдоподобия для различных массовых гипотез в ASHIPH. Идентификация в дрейфовой камере и сигнал калориметра берутся из быстрого моделирования.



Геометрия как условие

- В проекте детектора есть различные варианты отдельных систем. Необходимо предусмотреть возможность замены, например, дрейфовой камеры на трубки или калориметра КЕДР на кристаллический.
- Может потребоваться моделирование с различными размерами детектора для их оптимизации.
- Необходимо иметь возможность менять геометрию детектора со временем. Например, вершинный детектор может быть установлен после начала работы или заменён калориметр.

Всё это с необходимостью требует, чтобы геометрия детектора загружалась из базы данных.



База данных

База данных основана на идее Belle II:

- Условия содержатся в файлах ROOT. У них есть интервал, в котором они действительны: (experiment 1, run 1; experiment 2, run 2).
- Объект базы данных определяется так:

```
/** ASHIPH geometry. */  
DatabaseObject<AshiphGeometry> m_ashiphGeometry;  
  
/** ASHIPH geometry. */  
const AshiphGeometry* m_geometry;
```

- Объект базы данных загружает класс условия по идентификатору события:

```
m_geometry = m_ashiphGeometry.getObject(eventContext);
```

и далее из класса условия можно получить нужные данные:

```
const double innerRadius = Aurora::AuroraToCLHEP::getLength(  
    m_geometry->getInnerRadiusEndcap());  
const double outerRadius = Aurora::AuroraToCLHEP::getLength(  
    m_geometry->getInnerRadiusBarrel());
```

Геометрия всего детектора создаётся единым скриптом на python. Позже к нему можно будет добавить опции, которые создают различные необходимые варианты. Вначале создаются классы с данными для отдельных поддетекторов:

```
# Drift chamber.
drift_chamber = Aurora.DriftChamberGeometry()
drift_chamber.setLength(1600 * units.mm)
drift_chamber.setInnerRadius(0 * units.mm)
drift_chamber.setOuterRadius(620 * units.mm)
drift_chamber.setNSuperLayers(14)
drift_chamber.setSuperLayerInnerRadius(0, 20.000 * units.mm)
drift_chamber.setSuperLayerInnerRadius(1, 52.827 * units.mm)
...

# ASHIPH.
ashiph = Aurora.AshiphGeometry()
ashiph.setLength(1800 * units.mm)
ashiph.setInnerRadiusBarrel(646. * units.mm)
ashiph.setInnerRadiusEndcap(200. * units.mm)
ashiph.setOuterRadius(730. * units.mm)
...
```



Создание файлов базы данных (2)

Далее создаётся класс с общими данными о геометрии. В нём хранится список поддетекторов:

```
# General configuration.  
geometry = Aurora.Geometry()  
geometry.addSubdetector('DriftChamber')  
geometry.addSubdetector('Ashiph')
```

а также списки элементов и материалов:

```
# Hydrogen.  
hydrogen = Aurora.Element('H', 1, 1.007975)  
geometry.addElement(hydrogen)  
...  
# Nitrogen.  
nitrogen = Aurora.Material('N2')  
nitrogen.addElement('N', 2)  
nitrogen.setDensity(0.0012506 * units.g / units.cm3)  
geometry.addMaterial(nitrogen)  
...  
# Silicon dioxide.  
silicon_dioxide = Aurora.Material('SiO2')  
silicon_dioxide.addElement('Si', 1)  
silicon_dioxide.addElement('O', 2)  
silicon_dioxide.setDensity(2.65 * units.g / units.cm3)  
geometry.addMaterial(silicon_dioxide)  
...
```



Создание файлов базы данных (3)

В конце файлы записываются в пользовательскую базу данных в локальном каталоге `local_database`:

```
# Import.
database_importer = Aurora.DatabaseImporter()
database_importer.setName('Geometry')
database_importer.setObject(geometry)
iov = Aurora.IntervalOfValidity(-1, -1, -1, -1)
database_importer.setIntervalOfValidity(iov)
database_importer.writePayload()

# Drift chamber.
database_importer.setName('DriftChamberGeometry')
database_importer.setObject(drift_chamber)
database_importer.writePayload()

# ASHIPH.
database_importer.setName('AshiphGeometry')
database_importer.setObject(ashiph)
database_importer.writePayload()
```



Создание детектора

- Геометрия детектора Супер с-tau фабрики создавалась с использованием библиотеки DD4HEP. Данные при этом хранятся в файлах XML.
- Брать данные из базы данных, как было показано выше, тоже можно. Тогда это нужно желать в коде конструкторов подсистем детектора.
- Но при этом получается, что по крайней мере для создания геометрии нет никаких принципиальных преимуществ перед непосредственным использованием Geant4.
- Пока используется создание детектора в Geant4. Но согласия по этому вопросу нет (преобразования координат для отдельных элементов геометрии? визуализация?).



Создание детектора: пример

Создание средней части ячейки дрейфовой камеры (расположение чувствительного объёма):

```
void DetectorConstructor::createCellMiddle(G4LogicalVolume* monoLayer,
    int iMonoLayer, int iCell)
{
    const int iSuperLayer = m_geometry->getSuperLayer(iMonoLayer);
    const double firstCellAngle = m_geometry->getFirstCellAngle(iSuperLayer);
    const int firstMonoLayerIndex = m_geometry->getFirstMonoLayerIndex(
        iSuperLayer);
    const int monoLayerIndex = iMonoLayer - firstMonoLayerIndex;
    const int nCells = m_geometry->getNCellsMonoLayer(iMonoLayer);
    const double phiCell = M_PI / nCells;
    const double deltaPhi = m_geometry->getDeltaPhiBackwardForwardPlane(
        iMonoLayer);
    const double phi = (2 * iCell + (monoLayerIndex % 2)) * phiCell
        + firstCellAngle + deltaPhi / 2;
    std::string name = "cell_" + std::to_string(iCell) + "_middle";
    HepGeom::Transform3D transform = HepGeom::RotateZ3D(phi);
    ChannelNumber channelNumber = getChannelNumber(iMonoLayer, iCell);
    new G4PVPlacement(transform, m_cellMiddleLogicalVolumes[iMonoLayer], name,
        monoLayer, false, channelNumber);
}
```



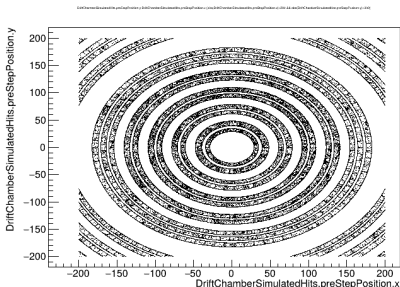
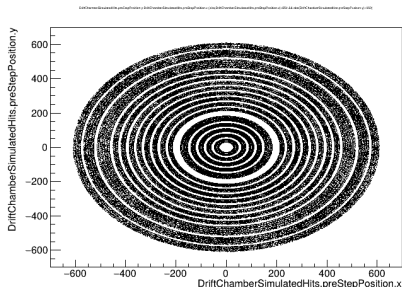
Геометрия дрейфовой камеры

- Параметры дрейфовой камеры взяты из доклада Корнелия Тодышева “Проектные параметры ДК для ВЭПП-6”.
- Во внутренней части ($r < 20$ см) дрейфовую камеру расположить невозможно, это место занято финальным фокусом ускорителя. Поэтому там будет разъёмный вершинный детектор.
- Временно вместо вершинного детектора используются 5 внутренних суперслоёв от проекта дрейфовой камеры для КМД-3. Их длина увеличена до длины дрейфовой камеры ВЭПП-6, а угол аксиальных слоёв уменьшен.
- Далее используются 7 внутренних суперслоёв от проекта дрейфовой камеры для Супер с-tau фабрики.

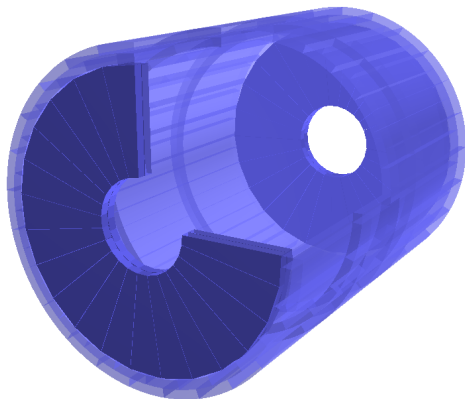


Геометрия дрейфовой камеры: ячейки

Координаты хитов Geant4:



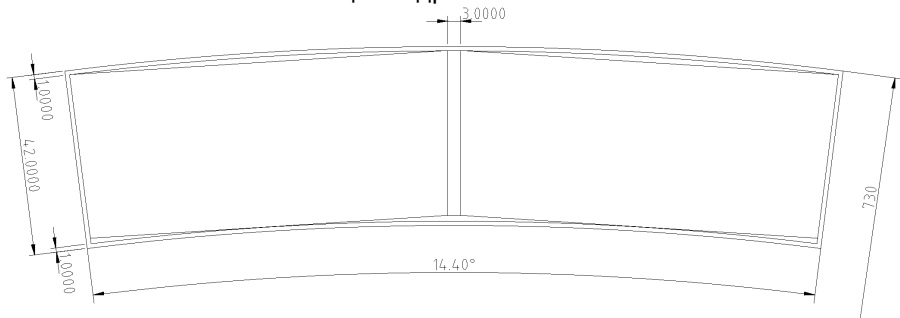
- Реализована только центральная часть гексагональной ячейки между 4 катодными проволоками, которые ближе всего к анодной по радиусу. Эта часть является чувствительным детектором в Geant4. Этого недостаточно для полноценного моделирования, но хватает, чтобы узнать минимальное расстояние от анодной проволоки до трека.
- Проволок, торцевых пластин пока нет, только основной объём поделен на чувствительные объёмы ячеек.



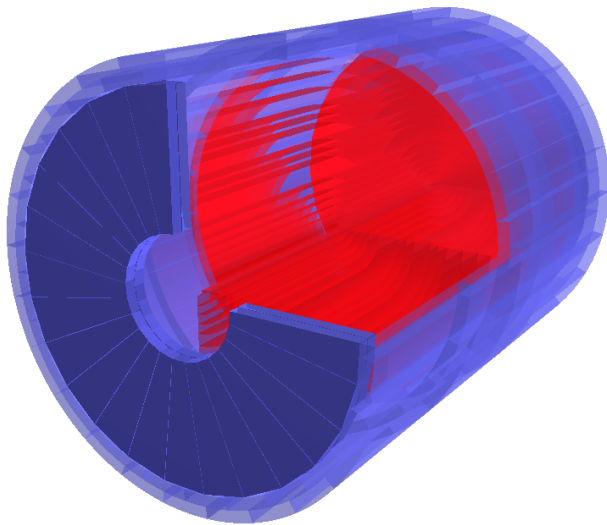
- 2 слоя, 3 секции по 25 счётчиков в центральной части.
- 2 слоя по 25 счётчиков в торцевых частях.



Счётчик во внешнем слое цилиндрической части ASHIPH:

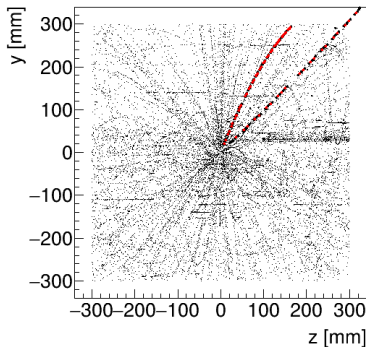
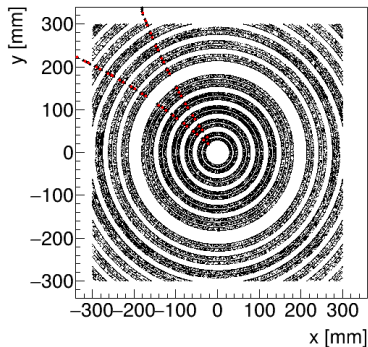


- Счётчик состоит из двух аэрогелевых частей и центральной части. В центральной части должны проходить фиберы, но пока их нет, она тоже заполнена аэрогелем.
- Реализована алюминиевая оболочка толщиной 1 мм.





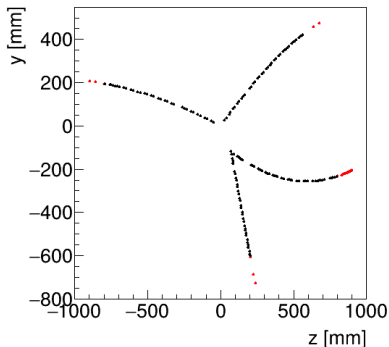
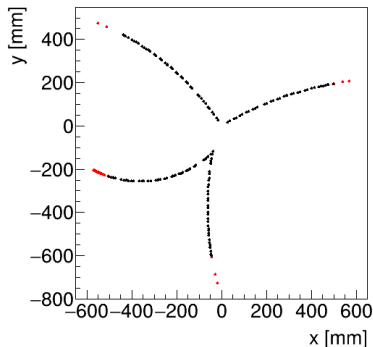
Подгонка треков



- Подгонка треков проводится с помощью библиотеки GenFit. Пока она использует информацию из Монте-Карло, чтобы определить, какие сигналы в дрейфовой камере относятся к какой частице.
- Измерения - расстояние от трека до проволоки в дрейфовой камере. Оно тоже берётся из Монте-Карло, моделирования дрейфа нет.



Экстраполяция треков



Красные точки - хиты экстраполяции, чёрные - в дрейфовой камере.

- Экстраполяция проводится в Geant4.
- Для каждого шага экстраполяции записывается ExtrapolationHit с информацией о координатах, длине участка трека, чувствительном объёме.



Вычисление правдоподобия в ASHIPH

1. По хитам экстраполяции для каждой массовой гипотезы вычисляется ожидаемая длина трека в конкретной счётчике ASHIPH.
2. Ожидаемое число фотоэлектронов вычисляется как

$$N = N_e \frac{(\beta\gamma)^2 - \frac{1}{n^2-1}}{(\beta\gamma)^2} \frac{1}{2} \left[1 + \operatorname{erf} \frac{\beta\gamma - \sqrt{\frac{1}{n^2-1}}}{\sqrt{2}\sigma} \right] + \frac{N_e \xi}{2} \left(1 + \frac{1}{(\beta\gamma)^2} \right) \left(\frac{1}{t} - \frac{1}{(\beta\gamma)^2} \right) \frac{1}{2} \left[1 + \operatorname{erf} \frac{\beta\gamma - \sqrt{t}}{\sqrt{2}\sigma} \right], \quad (1)$$

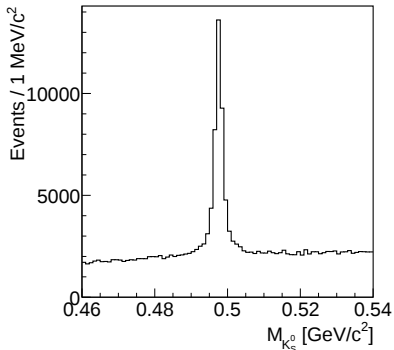
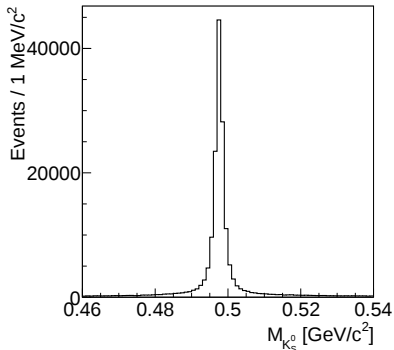
где $N_e = 10.0$ - число фотоэлектронов на 6 см, $n = 1.030$ - показатель преломления, $\xi = 0.14/0.24 \times 0.18$ - вклад δ -электронов, $t = 2.13$ - порог для них, $\sqrt{2}\sigma = 0.0001$.

3. После суммирования по участкам трека добавляется шум SiPM.
4. Функция правдоподобия перемножается для всех счётчиков, которые дают сигнал.



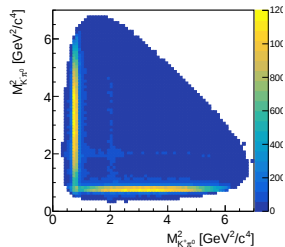
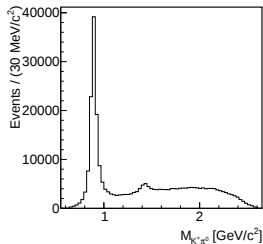
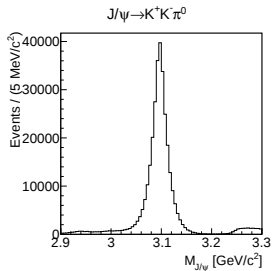
Анализ

- Гибридное моделирование даёт возможность восстанавливать вершину долгоживущих частиц, цилиндрические координаты dr и dz наименьшего удаления от точки взаимодействия для треков.
- Сгенерирован образец Монте-Карло со светимостью 50 пб^{-1} на энергии J/ψ (100 миллионов J/ψ). В качестве примера показано распределение по массе K_S^0 с $r > 5 \text{ мм}$ и $r < 5 \text{ мм}$.



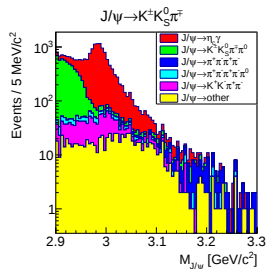
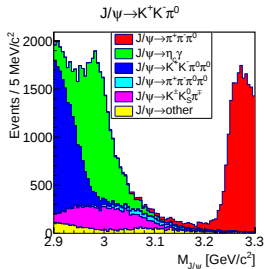
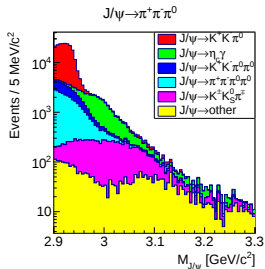


$$J/\psi \rightarrow K^+ K^- \pi^0$$





Фоны в распадах J/ψ





- Реализовано гибридное моделирование эксперимента на коллайдере ВЭПП-6.
- Текущая версия - первая, которая даёт результаты, которые можно использовать для физического моделирования.