EXPERT ROOT - Developing #56

BeamDet симуляция

06/29/2017 10:07 AM - Vitaliy Schetinin

Status:	Закрыта	**Start date:**	06/29/2017
Priority:	Низкий	**Due date:**	
Assignee:	Mikhail Kozlov	**% Done:**	90%
Category:	BeamDet	**Estimated time:**	0.00 hour
Target version:	v-0.4		

Description

BeamDet это собирательное понятие для всех детекторов необходимых для востановления характеристик исходного пучка.Детектор должен давать энергию, идентификацию и координаты пучкового иона на мишени. Пучковые детекторы - 2 пластины сцинтилляторов, разнесенные метров на 15, для измерения времени пролета
и многопроволочные камеры, пока что возьмем как в He-8, а потом подправим параметры под новую разработку.

Мишень - нержавеющая банка со стенкой толщиной микрон примерно 20, внутри которой жидкий водород. Толщину уточню позже.

Пучок Серы-28 энергией 40 МэВ/нуклон налетает на водородную мишень. Тритий летит в кольцевой телескоп, а кремний и два протона в детектор из нескольких квадратных телескопов позади магнита. Чтобы было понятнее, как это все работает, надо размазать по энергии, XY и углу тэта первичный ион (+- 2% по энергии, по углу 10 мрад(чтоб в мишень попадал), в пространстве на 0.5 см. Все сигма) и подмешать, тоже размазывая, несколько других ионов (с A не больше 31 и разностью числа протонов и нейтронов не более 5) с такой же жесткостью, т.е. имеющих такую же тракеторию в магнитном поле. Равная жесткость, значит равны P/Z где P импульс иона, Z его заряд. Это поможет увидеть работу отбора по TOF и DeltaE.

Станция MWPC это плоская конструкция, наполненная активным газом. Пока что возьмите RPCgas из cbm media.geo.Толщина газа 8 мм давление пока что атмосферное (потом будем снижать) температура 20 градусов цельсия. На самом деле, в свойствах материала задается плотность, которая определяется давлением и температурой. Поперечный размер 5 см * 5 см для начала. Стенки 20 микрон майлара. Вся конструкция вставлена в вакуум, в котором летит пучок. Внутри газа с шагом 1.25 мм проволочки диаметром 50 микрон из, скажем, бронзы (можно взять медь пока что) В дальнейшем надо будет учесть диффузию и рекомбинацию электронов в газе, но для начала надо разбить весь объем на параллельные полоски, в центре каждой полоски - проволочка. Сигнал с проволочки пропорционален энерговыделению в полоске. Всего 4 таких камеры, сгруппированные по две. В каждой паре в одной камере проволочки идут вдоль X, в другой -вдоль Y.

Одна пара камер стоит на 80 мм (close) выше по пучку от мишени, а другая на 280мм (far) от мишени. Какая камера в паре (X или Y) расположена ближе к мишени, пока что все равно.

History

#1 - 06/29/2017 10:12 AM - Vitaliy Schetinin

- Category set to BeamDet

#2 - 06/29/2017 10:14 AM - Vitaliy Schetinin

Комментарий к первому коммиту

1. root Файл геометрии мы заливаем в репозиторий. Да, он бинарный, но все таки очень лёгкий, поэтому не страшно.
2. В файле create_BeamDet_geo_v1.C параметры геометрии (такие как Int_t platesCount = 2; Double_t plateOffsetZ = 1500;) стоит вынести наверх макроса. Фактически обычно именно они изменяются от версии к версии.
   В нуле всегда находится мишень. Все части BeamDet должны быть расположены до мишени. У тебя полчучилось, что MWPC и второй TOF после нуля. Надо глобально сдвинуть назад весь детектор.
   Лучше завязать все на 4 переменные: глобальное расположение по Z двух TOF и двух MWPC, а уже служебные параметры: что как сориентировано внутри Assembly считать из них
3. В классе ERBeamDet. Сейчас в репозитории творится путаница с тем как надо оформлять класс симуляции. Тут я виноват. Правильно так как сейчас написано в документации. То есть текущее состояние поинта хранят переменные класса. Статическими переменными мы не пользуемся. У тебя возникли две переменные eloss и fEloss, которые по смыслу - одно и тоже. Сделай это одной переменной и у тебя будут нормально писать поинты.
4. Увидел особенность, связанную с файлом gconfig/g4Config.C. Мы используем многослойный пирог из библиотек, где каждый норовит добавить что-то свое. Тут долго объяснять пока просто прими на веру, что создание класса должно выглядеть так:
   TG4RunConfiguration* runConfiguration
   = new TG4RunConfiguration("geomRoot", "QBBC", "stepLimiter+stackPopper");

#3 - 06/29/2017 11:52 AM - Vitaliy Schetinin

1. Следует также создать не один, а два класса поинтов: ERBeamDetTofPoint и ERBeamDetMwpcPoint. И заполнять соответственно две коллекции. Пример можно посмотреть в ERGadast

2. В поинтах MWPC записывать в какую именно станцию и объём проволоки попал поинт. В Tof в какой по счету пластик. Как доставть номера текущих объёмов при транспорте можно смотерть в ERNeuRad

```
Int_t curVolId =  gMC->CurrentVolID(fiberInModuleNb); - номер копии текущего объема
Int_t corOffVolId = gMC->CurrentVolOffID(1, moduleNb); - номер копии материнского объема
```

#4 - 06/29/2017 07:05 PM - Vitaliy Schetinin

- Assignee changed from Vitaliy Schetinin to Mikhail Kozlov

#5 - 06/29/2017 07:16 PM - Vitaliy Schetinin

Добавить мишень в симуляцию

#6 - 07/01/2017 12:45 AM - Mikhail Kozlov

1) Пробовал разные параметры в TG4RunConfiguration* runConfiguration = new TG4RunConfiguration("geomRoot", "QBBC", "stepLimiter+stackPopper"). Не понимаю, правильно ли сейчас работает симуляция?

2) Если мы убрали статические переменные из processHits(), то, может, следует преписать метод AddPoint() без параметров?

3) В документации сейчас в описании реализации закона Биркса статические переменные в параметрах AddPoint остались.

#7 - 07/01/2017 02:09 PM - Vitaliy Schetinin

1) О правильности симуляции можно судить по двум вещам: eventDisplay.C - визуально и по поинтам в sim.root

На eventDisplay у тебя сейчас такая ситуация:

ed_def_cuts.png

И ~40тысяч вторичных частиц.То есть ион отдает свою энергию на ионизацию воздуха. Потому что материал пещеры - Air. Мы забыли упомянуть, что все это должно происходить в вакуумной трубе. Пока для того, чтобы ее не вводить в геометрию давай глобально поменяем материал пещеры (geometry/cave.geo) на vacuum (проверить как правильно пишется в geometry/media.geo). После такого изменения видим следующую картину:

ed_vacuum.png

Тут видно, что он долетел до второго Tof и до MWPC. И родил уже в них вторичные частцы.

ed_mwpc.png

В gconfig/g4config.in нужно установить каты на рождение вторичных частиц:

```
/mcPhysics/rangeCutForGamma  1. mm
/mcPhysics/rangeCutForElectron 1. mm
```

В sim.root можно видеть такое энерговыделение на всех поинта(используй veiew->Editor для лог шкалы):

eloss_beamdet.png

С помощью условия на атрибут fType, который ты ввел, можно разделить на eloss в Tof и MWPC. Нужно правой кнопкой на дереве Start

Tree Viewer. Перетащить eloss мышкой на X, fType на Experssion. Потом Edit experssion и там c++ синтаксис предикатов.

eloss_Tof.png eloss_mwpc.png

Для симуляции нужно доправить: 1) материал пещеры, 2) Разбить на две ветки поинты 3) Теперь действительно мржно убрать аргументы из AddPoint(). 4) Лучше сделать AddTofPoint() и AddMWPCPoint() 5) Добавить водородную мишень

В остальном отлично! В документации сейчас поправлю, спасибо!

#8 - 07/01/2017 09:29 PM - Mikhail Kozlov

Разбил поинты на две разные ветки.

В Gadast и детекторе из документации несколько отличается инициализация в коструктурах. Является ли один из вариантов более приоритетным?

Метод CopyClone() в Gadast прописан только для ERGadastCsIPoint. Сделал также для ERBeamDetTOFPoint.

#9 - 07/02/2017 07:02 PM - Vitaliy Schetinin

Да пусть CopyClones работает пока только для одного типа поинтов. Это не страшно. В другомслучае и просто пришлось бы наследовать от одного предка, который бы мы сдлеали только для этого. Инициализация пусть будет пока так, как в документации.

#10 - 07/02/2017 10:20 PM - Vitaliy Schetinin

- % Done changed from 0 to 50

На последний коммит:

1) Нужно добавить в класс Tof поинт атрибут fTofNb, в который писать номер Tof станции.

2) Нужно добавить в класс MWPC поинт fMWPCNb {0,1} (первый или второй MWPC - сдвоенные плоскости), fPlaneNb {0,1} (первая или вторая плоскость в сдвоенном MWPC), fWire {0, ...} номер проволочки

3) Все классы данных будем хранить в папке BeamDet/data/. Перенеси туда классы поинтов

4) Стоит переименовать макрос и название геометрии мишени на target.h.geo.root. (git mv)

5) Настроить в eventDisplay.C отображение поинтов. Поправить сторoку FairMCPointDraw *muSiPoints = new FairMCPointDraw ("ERmuSiPoint",kOrange, kFullSquare); выбрать разные цвета для поинтов в tof и в MWPC

#11 - 07/03/2017 07:14 PM - Vitaliy Schetinin

- % Done changed from 50 to 90

1) В Tof поинт лишний атрибут fStilbenNr

В остальном все отлично, можно переходить к написанию генератора и симуляции с ним: http://er.jinr.ru/develop/issues/57

#12 - 07/04/2017 05:14 PM - Mikhail Kozlov

А зачем fStilbenNr в MWPC?

#13 - 07/04/2017 06:40 PM - Vitaliy Schetinin

Тоже не нужен. Удаляй

#14 - 07/14/2017 11:23 AM - Mikhail Kozlov

Что такое stackPopper TG4RunConfiguration()?

#15 - 07/14/2017 04:19 PM - Mikhail Kozlov

Можно ли задвать разные каты для частиц, в зависимости от материала?

#16 - 07/15/2017 11:16 AM - Vitaliy Schetinin

Mikhail Kozlov писал(а):

> Что такое stackPopper TG4RunConfiguration()?

В симуляциях мы не обращаемся к Geant4 напрямую, а используем некоторую прослойку абстрагирующую нас от конкретной библиотеки транспорта - Virtual Monte Cerlo - https://root.cern.ch/vmc.Идея в том, что в принципе все библиотеки транспорта работают примерно одинаково и можно иметь один и тот же интерфес к ним всем и без проблем переключаться. Раньше это было более актуально, потому что в колаборациях одновременно использовали geant3(https://en.wikipedia.org/wiki/GEANT-3) и geant4(http://geant4.cern.ch/).Сейчас все практически используют geant4. Bvmc также есть поддержка fluka - http://www.fluka.org/fluka.php (в основном применяется для расчетов радиационной защиты) и http://garfield.web.cern.ch/garfield/(в основном для газовых детекторов).

TG4RunConfiguration это класс VMC для настройки Geant4. - http://ivana.home.cern.ch/ivana/g4vmc_html/classTG4RunConfiguration.html

Его параметры: тип используемой геометрии, Physic List - список используемых процессов(https://root.cern.ch/physics-list) и специальные процессы VMC. Под специальными процессами подразумевается то, что VMC сам может вторгаться в логику транспорта. К примеру SpecialCuts означает, что будут применены каты в стиле VMC. Они энергетические и задаются в макросе SetCuts с помощью объекта gMC. Если этого специального процесса нет, то работают каты по умолчанию или используется нативный способ задания катов библиотеки транспорта. Для нас сейчас удобен именно этот вариант. Сейчас стоит задавать каты с помощью g4config.in это макрос настройки geant4. В Geant4 введен свой способ задания катов - range cut(http://geant4.cern.ch/G4UsersDocuments/UsersGuides/ForApplicationDeveloper/html/TrackingAndPhysics/thresholdVScut.html). Это кат на пробег новой частицы в данном материале, То есть если она по расчету пробежит меньше ката, то не надо заниматься ее трекингом. ее энергия уйдет в edep в точке, где разыгрывается процесс.

Строки для задания катов:

```
/mcPhysics/rangeCuts 1 mm
/mcPhysics/rangeCutForGamma  1. mm
/mcPhysics/rangeCutForElectron 1 mm
```

Процесс stackPopper позволяет нам с помощью vmc докидывать свои треки в событие в процессе трекинга. Нам этот процесс очень нужен в связи с реализацией нашего класса распадов.

#17 - 07/16/2017 04:18 PM - Sergey Belogurov

Михаилу актуально задавать разные каты в разных материалах. В ТОФ надо пониже, а в газе MWPC и так ОК.

#18 - 07/20/2017 08:26 AM - Vitaliy Schetinin

Для получения реальной координаты пучка на мишени, ее следует добавить как чувствительный объем в этот детектор (то есть прямо в геометрию). Ввести еще один тип поинта. Мишень как пассивный отдельный элемент убрать из симуляции.

#19 - 07/20/2017 12:16 PM - Mikhail Kozlov

Примитив мишени оставить Tube или как-то по-другому сделать?

#20 - 07/20/2017 12:47 PM - Vitaliy Schetinin

Ровно также. Только она теперь в детекторе. Просто, чтобы не плодить отдельный класс мишени и как то странно будет его несаледовать от ERDetector

#21 - 07/21/2017 07:23 AM - Vitaliy Schetinin

Тест рассылки уведомлений на почту

#22 - 07/31/2017 02:14 PM - Vitaliy Schetinin

Необходимо добавить в симуляцию запись такого объекта как fBeamDetMCProjectile класса ERBeamDetParticle. Это состояние иона на мишени в симуляции. Имя иона мы передаем в класс симуляции через интерфейс. С помощью этого при трекинге через мишень отбираем трек иона. Считываем его параметры при входе в мишень и строим объект fBeamDetMCProjectile.

Также в симуляции необходимо сохранить объект fBeamDetMCTrack класс ERBeamDetTrack. В нем необходимо запомнить теже данные, что и в реконструкции. Только в данном случае мы по честному получаем координаты на мишени и на MWPC.

#23 - 08/09/2017 08:09 AM - Vitaliy Schetinin

- Status changed from Открыта to Закрыта