

EXPERT ROOT - Developing #57

Генератор коктейля ионов ERlonMixGenerator

06/29/2017 11:24 AM - Vitaliy Schetinin

Status:	Закрыта	Start date:	06/29/2017
Priority:	Низкий	Due date:	
Assignee:	Mikhail Kozlov	% Done:	0%
Category:	BeamDet	Estimated time:	0.00 hour
Target version:	v-0.4		
Description В мишень попадает коктейль из ионов разного типа. Для идентификации какой именно ион прилетел в мишень используется BeamDet. Для симуляции работы BeamDet необходимо разработать генератор событий который будет создавать события с разными ионами в зависимости от вероятности их появления. Рабочее название для генератора ERShonMixGenerator. Он должен обладать всеми интерфейсами ERlonGenerator (Уметь испускать ион с размазанной координатой и импульсом) и соответственно быть наследован от него. Конструктор ERlonGenerator имеет следующие параметры: ERlonGenerator(TString name, Int_t z, Int_t a, Int_t q, Int_t mult), Будем считать, что конструктор задает главный ион генератора - события с которым нам интересны. В события с этим ионом должны быть подмешаны события с ионами фона. Для этого необходимо завести интерфейс AddBackgroundlon(name,z,a,q,probability). Вероятность главного иона будет рассчитываться как 1 - сумма вероятностей фоновы ионов. Характер размазывания вершины и импульса для фоновых ионов такой же как для главного. Необходимо перегрузить ERlonGenerator::ReadEvent(). При этом желательно не копипастить код из него реализующий размазывание, а переработать его, выделив в отдельную функцию и отнаследовав.			

History

#1 - 06/29/2017 07:05 PM - Vitaliy Schetinin

- Assignee changed from Vitaliy Schetinin to Mikhail Kozlov

#2 - 07/05/2017 10:38 PM - Mikhail Kozlov

1) Не могу решить как лучше хранить вероятность появления разных ионов и как потом это использовать в ReadEvent.

2) Не совсем понимаю, что делать с ReadEvent. Нужно ли менять его код в ERlonGenerator или просто перегрузить в ERlonMixGenerator? И что при этом выносить в отдельную функцию - ту часть которая внутри цикла по количеству ионов в событий?

#3 - 07/06/2017 07:34 AM - Vitaliy Schetinin

1) Хранить как атрибут класса типа std::map(name, probability) fBGlons. Вся информация по иону будет доступна в TPDGDatabase после run->AddNewlon(flon);

2) В AddBackgroundlon() надо добавить проверку, что в сумме по вероятностям мы не вылетели за 1.

3) В отдельную функцию я бы предложил вынести розыгрыш координаты и импульса:

```
phi = gRandom->Uniform(fPhiMin,fPhiMax) * TMath::DegToRad();

if (fPRangelsSet) { pabs = gRandom->Uniform(fPMin,fPMax); }
else if (fPtRangelsSet) { pt = gRandom->Uniform(fPtMin,fPtMax); }
if (fThetaRangelsSet) {
    if (fCosThetalsSet)
        theta = acos(gRandom->Uniform(cos(fThetaMin* TMath::DegToRad()),
            cos(fThetaMax* TMath::DegToRad())));
    else {
        theta = gRandom->Uniform(fThetaMin,fThetaMax) * TMath::DegToRad();
    }
} else if (fEtaRangelsSet) {
    eta = gRandom->Uniform(fEtaMin,fEtaMax);
    theta = 2*TMath::ATan(TMath::Exp(-eta));
```

```

} else if (fYRangelsSet) {
    y = gRandom->Uniform(fYMin,fYMax);
    mt = TMath::Sqrt(fPDGMass*fPDGMass + pt*pt);
    pz = mt * TMath::SinH(y);
}

if (fThetaRangelsSet || fEtaRangelsSet) {
    if (fPRangelsSet) {
        pz = pabs*TMath::Cos(theta);
        pt = pabs*TMath::Sin(theta);
    } else if (fPtRangelsSet) {
        pz = pt/TMath::Tan(theta);
    }
}

px = pt*TMath::Cos(phi);
py = pt*TMath::Sin(phi);

if (fBoxVtxIsSet) {
    fX = gRandom->Uniform(fX1,fX2);
    fY = gRandom->Uniform(fY1,fY2);
}

```

В перегруженном ReadEvent() надо идти циклом по множественности ионов, разыгрывать с помощью функции вершину и импульс, разыгрывать вероятность появления иона, по fBDlons определять какой именно ион, по его имени из базы доставать PDG код и ион таким кодом, вершиной и импульсом добавлять в стек первичных треков.

#4 - 07/10/2017 05:10 PM - Mikhail Kozlov

1) Сейчас AddBackgraondlon() работает так, что он по порядку помещает в тар все ионы, пока суммарная вероятность не превышает 1. Например, если добавить ионы с вероятностями появления:

- 1) 0.5
- 2) 0.6
- 3) 0.1
- 4) 0.2

то будут генерироваться все, кроме второго.

2) Если добавить два вот таких иона:

```

sgenerator->AddBackgroundlon("Bglon1", 27, 9, 9, 0.3);
sgenerator->AddBackgroundlon("Bglon2", 3, 9, 3, 0.4);

```

то симуляция почему-то падает на 4 событии.

#5 - 07/10/2017 06:21 PM - Vitaliy Schetinin

Она не просто падает. Она страдает и кидается сообщениями:

```
G4NucleiProperties::GetMassExccess: Wrong values for A = 9 and Z = 27
G4NucleiProperties::GetMassExccess: Wrong values for A = 9 and Z = 27
G4NucleiProperties::GetNuclearMass: Wrong values for A = 8 and Z = 26
G4NucleiProperties::GetMassExccess: Wrong values for A = 9 and Z = 27
```

Потому что A меньше Z. Чего быть не должно. A это массовое число = кол-во нейтронов + кол-во протонов в ядре. Z = количеству протонов

Q тоже ставь равным количеству протонов.

С такими параметрами у меня все взлетело:

```
sgenerator->AddBackgroundlon("Bglon1", 27, 36, 27, 0.5);
sgenerator->AddBackgroundlon("Bglon2", 3, 12, 3, 0.3);
```

#6 - 07/14/2017 11:22 AM - Mikhail Kozlov

Для Beamdet нужны следующие поправки:

1) Сделать не равномерное распределение координат в пространстве, а Гауссово для SetBoxXYZ() в генераторе. Сейчас размазывание находится в spreadingParameters.

Как лучше поступить? Можно вынести это дело в отдельную функцию и ее перегрузить в mix генераторе, либо прям в ERLonGenerator поменять Uniform на Gauss.

2) Кинетическую энергию тоже размазать по Гауссу. Вопросы те же, что в п.1.

#7 - 07/14/2017 06:59 PM - Vitaliy Schetinin

1) Явно нужно делать два варианта и к ним два интерфейса: SetBoxXYZ() - работает как было. Set SetGausXYZ() - делает гауссово распределение. Какой интерфейс был пользователем вызван последним - ту логику и выбрать. Сначала надо реализовать в ERLonGenerator. Да лучше две функции иметь. Для разыгрывания по Uniform - та что есть и для разыгрывания по Гауссу. Потом их уже перегрузить в ERLonMixGenerator

2) Логика та же что в пункте 1. Только давай уж конкретно займемся. У нас на данный момент интерфейс только к импульсу. Поэтому в макросе пользователь пересчитывает. Это не удобно. Давай делать как для импульса так и для энергии и Гаусс и униформ

#8 - 08/09/2017 08:09 AM - Vitaliy Schetinin

- Status changed from *Открыта* to *Закрыта*